

Notebook 1

CS495 Web and Cloud Security

PDX | Winter 2022

Evan La Fleur

1.1 Setup	5
1.2 Web Programming	5
1.2.3 Sequential Program	6
1.2.4 Timing Decorator	7
1.2.5 Run Sequential Program	7
1.2.6 Multiprocessing Program	8
1.2.8 Matplotlib	8
1.12 Asynchronous Program	9
1.3 Broken Authentication	9
1.3.3 username enumeration via different responses	10
1.3.4 broken-bruteforce-protection-ip-block	11
1.3.4 username enumeration via account lock	11
1.3.5 oauth authentication bypass via oauth implicit flow	12
1.4 HW 1 (2fa-bypass-using-a-brute-force-attack)	15
2fa-bypass-using-a-brute-force-attack	15
1.5 Broken Access Control	18
1.5.1-1.5.2 file-path-traversal	18
simple	18
absolute-path-bypass	19
sequences-stripped-non-recursively	19
Superfluous-url-decode	21
Validate-start-of-path	22
Validate-file-extension-null-byte-bypass	23
1.5.3-1.5.9 access-control	24
Unprotected-admin-functionality	24
Unprotected-admin-functionality-with-unpredictable-url	25
User-role-controlled-by-request-parameter	26
User-role-can-be-modified-in-user-profile	26
Url-based-access-control-can-be-circumvented	28
Method-based-access-control-can-be-circumvented	29
User-id-controlled-by-request-parameter	29

User-id-controlled-by-request-parameter-with-unpredictable-user-ids	31
User-id-controlled-by-request-with-data-leakage-in-redirect	31
User-id-controlled-by-request-parameter-with-password-disclosure	32
Insecure-direct-object-references	32
Multi-step-process-with-no-access-control-on-one-step	33
Referrer-based-access-control	34
1.5.10 information-disclosure	34
Lab-infoleak-in-error-messages	34
Lab-infoleak-on-debug-page	35
Lab-infoleak-via-backup-files	35
1.5.11 WFP1: File Upload	36
Example #1	36
Example #2	36
1.5.12-1.5.13 file-upload	37
Lab-file-upload-remote-code-execution-via-web-shell-upload	37
lab-file-upload-web-shell-upload-via-content-type-restriction-bypass	38
1.6 SSRF	39
1.6.1-1.6.4 ssrf	39
basic-ssrf-against-localhost	39
basic-ssrf-against-backend-system	41
Ssrf-with-blacklist-filter	41
Ssrf-filter-bypass-via-open-redirection	42
1.6.5 ssrf/blind	42
out-of-band-detection	42
1.7 XXE	43
Exploiting-xxe-to-retrieve-files	43
Exploiting-xxe-to-perform-ssrf	44
Xinclude-attack	45
Xxe-via-file-upload	45

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

The screenshot shows a code editor window with a dark theme. The file is named `getUrls.py`. The code implements two functions: `getUrlTitle` and `getSequential`. The `getUrlTitle` function uses the `requests` library to fetch a URL and `BeautifulSoup` to parse the HTML for the title. The `getSequential` function takes a list of URLs and returns a list of titles. The code includes docstrings for both functions.

```
冬 22 > CS495 > w22websec-lafleur-evan > lab1 > 1.2 > getUrls.py
1 import requests
2 from bs4 import BeautifulSoup
3
4 def getUrlTitle(url):
5     """
6         This function returns the <title> of an HTML document given its URL
7         :param url: URL to retrieve
8         :type url: str
9         :return: Title of URL
10        :rtype: str
11    """
12    resp = requests.get(url)
13    soup = BeautifulSoup(resp.text,'html.parser')
14    title = str(soup.find('title'))
15    return(title)
16
17 def getSequential(urls):
18     """
19         Given a list of URLs, retrieve the title for each one using a single synchronous process
20         :param urls: List of URLs to retrieve
21         :type urls: list of str
22         :return: list of titles for each URL
23         :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 import requests, time
    2 from bs4 import BeautifulSoup
    3
    4 def time_decorator(func):
    5     """
    6         Takes a function and returns a version of it that prints out the elapsed time for
    7         :param func: Function to decorate
    8         :return: Function which outputs execution time
    9         :rtype: Function
    10    """
    11    def inner(*args, **kwargs):
    12        s = time.perf_counter()
    13        return_vals = func(*args, **kwargs)
    14        elapsed = time.perf_counter() - s
    15        print(f'Function returned: {return_vals}')
    16        return(elapsed)
    17    return(inner)
    18
    19 def getUrlTitle(url):
    20     """
    21         This function returns the <title> of an HTML document given its URL
    22         :param url: URL to retrieve
    23    """

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

[elafleur]: python3 getUrlTitle.py
Function returned: ['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>']
0.35 secs
[elafleur]: █

```

1.2.5 Run Requestional Program

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

```

[elafleur]: python3 getUrls.py
Function returned: ['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>']
0.33 secs
[elafleur]: █

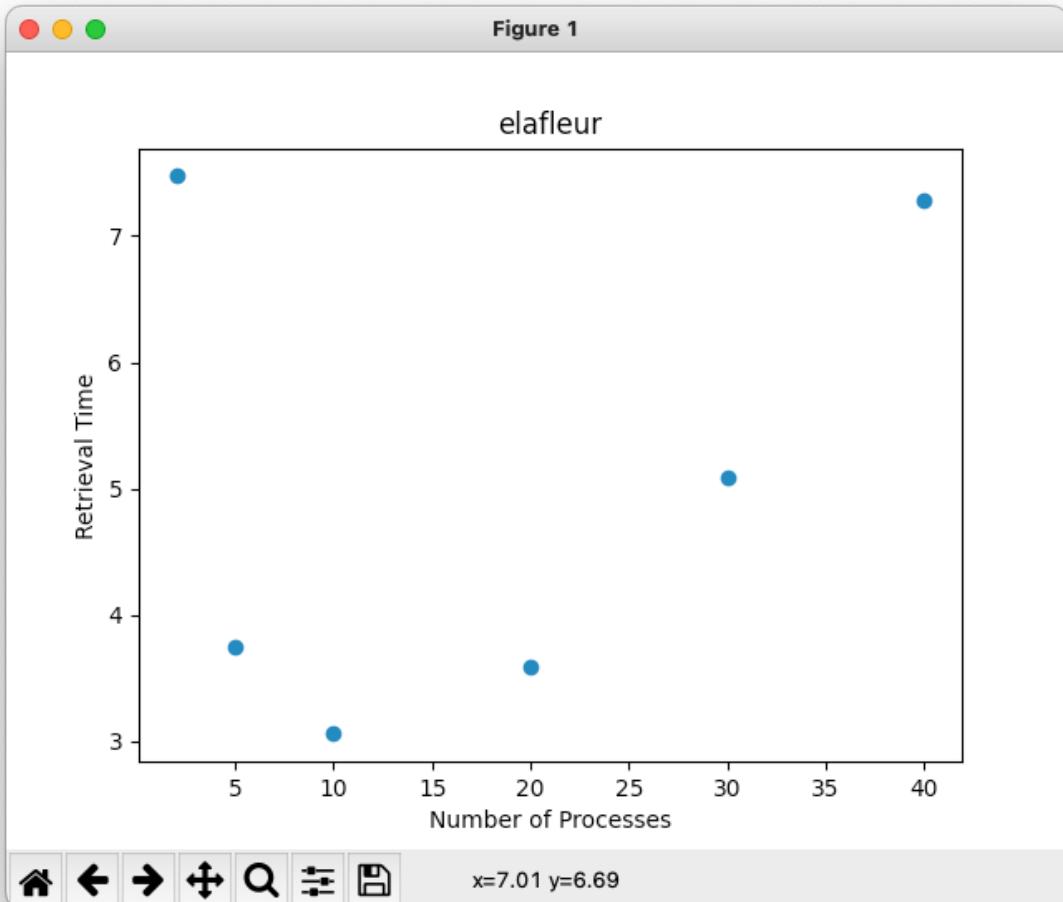
```

1.2.6 Multiprocessing Program

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

[elafleur]: python3 getUrlsWithMultiprocessing.py
Function returned: ['<title>Portland State University - PSU | Portland OR</title>', '<title>Oregon CTF</title>', '<title>YouTube</title>', '<title>GitHub: Where the world builds software · 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 software · 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 software · 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>', '<title>Google Advertising and Analytics</title>', '<title>Google</title>', '<title>Google</title>', '<title>Google , '<title>Google</title>', '<title>\n    Google Account\n    </title>', '<title>Google Chrome - Download 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.

Headers	Payload	Preview	Response	Initiator	Timing	Cookies
General Request URL: https://ac131f9c1e142c96c0352e12009d000b.web-security-academy.net/login						
Request Method: POST						
Status Code: 200 OK						
Remote Address: 18.200.141.238:443						
Referrer Policy: strict-origin-when-cross-origin						
Response Headers View source Connection: close						
Content-Encoding: gzip						
Content-Length: 946						
Content-Type: text/html; charset=utf-8						
Form Data View source View URL-encoded username: elafleur						
password: elafleur						

1.3.3 username-enumeration-via-different-responses

The screenshot shows a web browser window for the 'Web Security Academy' challenge titled 'Username enumeration via different responses'. The challenge has been solved, indicated by a green 'Solved' button with a trophy icon. The main content area displays two identical success messages: 'Congratulations, you solved the lab!' followed by a 'Share your skills!' button and a 'Continue learning >>' link. At the bottom of the page, there are links for 'Home', 'My account', and 'Log out'. The 'My Account' section shows the user's details: username 'adm' and email 'adm@adm.net'. The email field is currently set to 'elafleur@pdx.edu', and there is a 'Update email' button. A terminal-like footer shows the command '1.3 @ elafleur [evan] \$'.

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

1.3.4 broken-bruteforce-protection-ip-block

The screenshot shows a completed lab session on the Web Security Academy platform. At the top, the title 'Broken brute-force protection, IP block' is displayed next to a 'Solved' badge with a checkmark icon. Below the title, there's a button to 'Back to lab description'. A prominent orange banner at the top of the main content area says 'Congratulations, you solved the lab!'. To the right of this banner are buttons for 'Share your skills!' (with a Twitter icon) and 'Continue learning >'. At the bottom of the page, there's a 'My Account' section where the user's username 'carlos' is listed. Below this, there's a form for updating the email address, with the current value 'elafleur' in the input field and a green 'Update email' button. At the very bottom of the page, a dark footer bar displays the command-line text: 'Testing case: taylor password is taylor 1.3 @ elafleur [evan] \$'.

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.

The screenshot shows a solved lab page from the Web Security Academy. At the top, it says "Username enumeration via account lock" and has a "Solved" button. Below that, two orange boxes say "Congratulations, you solved the lab!". At the bottom, there are links for "Share your skills!" and "Continue learning".

WE LIKE TO BLOG

Invited user name or password
1.3 @ elafleur [evan] \$

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?

```

▼ Request call stack
  (anonymous)      @ /oauth-callback:13
  Promise.then (async)
  (anonymous)      @ /oauth-callback:12

▼ Request initiator chain
  ▼ https://oauth-ac551f841ecb46b1c0a445f702c60052.web-security-academy.net/interaction/_Hq1NL2CPSL5btF2_zgq8/confirm
    ▼ https://oauth-ac551f841ecb46b1c0a445f702c60052.web-security-academy.net/auth/_Hq1NL2CPSL5btF2_zgq8
      ▼ https://ac561fe01ef34688c0ef455e003200e7.web-security-academy.net/oauth-callback
        ▼ https://ac561fe01ef34688c0ef455e003200e7.web-security-academy.net/authenticate
          https://ac561fe01ef34688c0ef455e003200e7.web-security-academy.net/

```

- 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 to the pieces of a URL. It will allow better manipulation of the query 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-wSMCDnQwk0xXX0aybunhe0xhpG2UKJlHey8CDG607&expires_in=3600&token_type=Bearer&scope=openid%20profile%20email

Web Security Academy 

Authentication bypass via OAuth implicit flow

Back to lab description »

LAB Solved 

Congratulations, you solved the lab!

 Share your skills!  Continue learning »

Home | My account

WE LIKE TO 



The history of swigging port

The 'discovery' of port dates back to the late Seventeenth Century when British sailors stumbled upon the drink in Portugal and then stumbled even more slowly home with several more bottles. It

NE 1.3 @ elafleur [evan] \$]

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

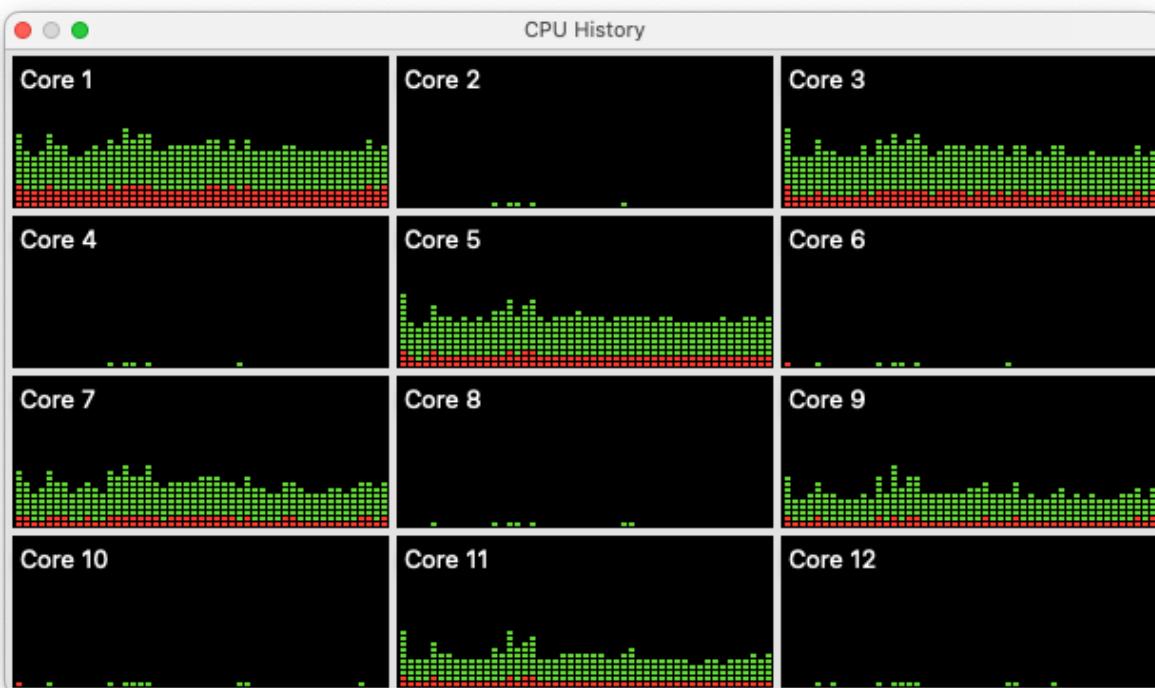
attempting new codes which made the correct code get lost in the mix. To overcome this I found a webpage discussing [Event Handling](#), and with this i was able to create an event that the program would only continue if it did not detect this event stopped.

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] $
```

The screenshot shows a completed lab on the Web Security Academy platform. The top navigation bar includes the 'Web Security Academy' logo, a '2FA bypass using a brute-force attack' title, and a 'LAB Solved' button with a trophy icon. Below the title, there's a link to 'Back to lab description'. The main content area has two orange notifications: 'Congratulations, you solved the lab!' and another identical message. At the bottom, a terminal window displays the same command-line output as the previous image, indicating the successful completion of the challenge.

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.

The screenshot shows a completed lab from the Web Security Academy. At the top, it says "File path traversal, simple case" and "Solved". Below that, a banner says "Congratulations, you solved the lab!". There are buttons for "Share your skills!" and "Continue learning >". A "Home" link is at the bottom right. The main content area features a logo with the text "WE LIKE TO SHOP" and a hanger icon. Below the logo is a terminal window showing the command "cat /etc/passwd" and its output, which includes the line "root:x:0:0:root:/root:/bin/bash". Navigation links "OUTLINE" and "TIMELINE" are visible on the left side of the terminal window.

absolute-path-bypass

The screenshot shows a completed lab page from the Web Security Academy. At the top, it says "File path traversal, traversal sequences blocked with absolute path bypass". A green "Solved" button with a trophy icon is visible. Below that, a link to "Back to lab description >". The main message is "Congratulations, you solved the lab!" with a "Share your skills!" button and a "Continue learning >" link. At the bottom, there's a banner with the text "WE LIKE TO SHOP" and a hanger icon. A terminal window at the bottom shows the command-line session used to solve the lab.

```
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
1.5 @ elafleur [evan] $
```

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.



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](#)

WE LIKE TO
SHOP

```
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
1.5 @ elafleur [evan] $[]
```

Superfluous-url-decode



File path traversal, traversal sequences stripped with superfluous URL-decode

LAB Solved

[Back to lab description >>](#)

Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >>](#)

[Home](#)

WE LIKE TO
SHOP



Photobomb Backdrops
★★★★★ \$19.62



Hologram Stand In
★ ★ ★ ★ ★ \$34.13



Conversation Controlling Lemon
★ ★ ★ ★ ★ \$85.78



What Do You Meme?
★★★★★ \$19.63

[View details](#)

[View details](#)

[View details](#)

[View details](#)



```
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
1.5 @ elafleur [evan] %
```

Validate-start-of-path

Web Security Academy File path traversal, validation of start of path LAB Solved

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

[Home](#)

WE LIKE TO **SHOP** 

			
Pet Experience Days ★ ★ ★ ★ ★ \$34.31	Sprout More Brain Power ★ ★ ★ ★ ★ \$39.96	Sarcastic 9 Ball ★ ★ ★ ★ ★ \$95.37	Real Life Photoshopping ★ ★ ★ ★ ★ \$72.45
View details	View details	View details	View details

			
---	---	---	--

```
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
1.5 @ elafleur [evan] $
```

Validate-file-extension-null-byte-bypass

Web Security Academy File path traversal, validation of file extension with null byte bypass LAB Solved

[Back to lab description »](#)

Congratulations, you solved the lab! [Share your skills!](#) [Continue learning »](#)

[Home](#)

WE LIKE TO **SHOP** 



Com-Tool
★★★★★ \$97.21

[View details](#)



Sprout More Brain Power
★★★★☆ \$58.11

[View details](#)



Pest Control Umbrella
★★★★★ \$6.61

[View details](#)



Paintball Gun - Thunder Striker
★★★★☆ \$25.62

[View details](#)



High End Gift Wrapping



The Giant Enter Key

```
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
1.5 @ elafleur [evan] $
```



BBQ Suitcase



More Than Just Birdsong

1.5.3-1.5.9 access-control

Unprotected-admin-functionality

Web Security Academy | Unprotected admin functionality LAB Solved

[Back to lab description >>](#)

Congratulations, you solved the lab! [Share your skills!](#) [Continue learning >>](#)

[Home](#) | [My account](#)

WE LIKE TO **SHOP** 

			
Adult Space Hopper ★★★★★ \$13.66	Six Pack Beer Belt ★★★★★ \$91.28	The Splash ★★★★☆ \$41.71	Com-Tool ★★★★☆ \$96.95
View details	View details	View details	View details

			
1.5 @ elafleur [evan] \$python3 unp*	1.5 @ elafleur [evan] \$		

Unprotected-admin-functionality-with-unpredictable-url

Web Security Academy Unprotected admin functionality with unpredictable URL LAB Solved 

Back to lab description »

Congratulations, you solved the lab!

Share your skills! Continue learning »

Congratulations, you solved the lab!

Share your skills! Continue learning »

[Home](#) | [My account](#)

WE LIKE TO  **SHOP**



The Lazy Dog ★ ★ ★ ★ ★ \$9.07 View details	Cheshire Cat Grin ★ ★ ★ ★ ★ \$79.72 View details	Conversation Controlling Lemon ★ ★ ★ ★ ★ \$48.85 View details	Eye Projectors ★ ★ ★ ★ ★ \$34.72 View details
--	--	---	---



```
1.5 @ elafleur [evan] $python3 unprotected--url.py
1.5 @ elafleur [evan] $
```

User-role-controlled-by-request-parameter

Web Security Academy User role controlled by request parameter LAB Solved

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

User deleted successfully!

Users

wiener - [Delete](#)

Name	Value	Domain	P..	Expir...	Size	Http...	Secure	Sam...	Sam...	Pri...
Admin	true	ac8c1f461e3f2273c0ed1adc006800...	/	Sessi...	9	✓	✓			Medi...

Console What's New Issues

Highlights from the Chrome 97 update

- New preview feature: Recorder panel
- Enhanced "Edit as HTML" with code completion
- Refresh device list in Device Mode
- Improved code debugging experience

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



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

Web Security Academy URL-based access control can be circumvented LAB Solved

Back to lab description »

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

[Home](#) | [Admin panel](#) | [My account](#)

WE LIKE TO **SHOP** 



Inflatable Holiday Home
★ ★ ★ ★ ★ \$40.15

[View details](#)



Sprout More Brain Power
★ ★ ★ ★ ★ \$19.03

[View details](#)



3D Voice Assistants
★ ★ ★ ★ ★ \$76.57

[View details](#)



Six Pack Beer Belt
★ ★ ★ ★ ★ \$37.04

[View details](#)









```
1.5 @ elafleur [evan] $python3 *circumvented.py
"Access denied"
1.5 @ elafleur [evan] $
```

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



Method-based access control can be circumvented

[Back to lab description >](#)

LAB Solved



Congratulations, you solved the lab!

[Share your skills!](#)

[Continue learning >](#)

[Home](#) | [My account](#)



Sprout More Brain Power

\$56.11

[View details](#)



Folding Gadgets

\$92.15

[View details](#)



Picture Box

\$12.65

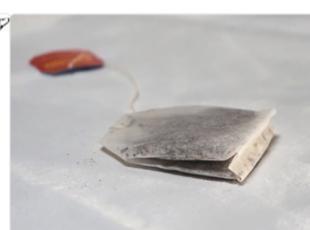
[View details](#)



The Trolley-ON

\$13.26

[View details](#)



1.5 @ elafleur [evan] \$

User-id-controlled-by-request-parameter



User ID controlled by request parameter

LAB Solved



[Back to lab description >>](#)

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

Web Security Academy  User ID controlled by request parameter, with unpredictable user IDs LAB Solved 

[Back to lab description >>](#)

Congratulations, you solved the lab!  Continue learning >>

Home | My account

```
1.5 @ elafleur [evan] $python3 user-id*ids.py
d68dd212-00bf-4afb-811a-6b7ca006a5c1
1.5 @ elafleur [evan] $
```

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

Web Security Academy  User ID controlled by request parameter with data leakage in redirect LAB Solved 

[Back to lab description >>](#)

Congratulations, you solved the lab!  Continue learning >>

Home | My account

```
WE LIKE TO 
1.5 @ elafleur [evan] $
```

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

Web Security Academy  User ID controlled by request parameter with password disclosure LAB Solved 

[Back to lab description >>](#)

Congratulations, you solved the lab!  Share your skills! [Continue learning >>](#)

User deleted successfully!

Users

wiener - [Delete](#)

```
1.5 @ elafleur [evan] $
```

Insecure-direct-object-references

Web Security Academy  Insecure direct object references LAB Solved 

[Back to lab description >>](#)

Congratulations, you solved the lab!  Share your skills! [Continue learning >>](#)

[Home](#) | [My account](#) | [Live chat](#) | [Log out](#)

My Account

Your username is: carlos

Email

```
Hal Pline: Takes one to know one
You: Ok so my password is am30fm7c9d8jwzbd803a. Is that right?
Hal Pline: Yes it is!
You: Ok thanks, bye!
Hal Pline: Do one!
```

```
1.5 @ elafleur [evan] $
```

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

Web Security Academy  Multi-step process with no access control on one step

[Back to lab description >>](#)

Congratulations, you solved the lab!

 [Share your skills!](#) [Continue learning >>](#)

Home | Admin panel | My account

1.5 @ elafleur [evan] \$

Referrer-based-access-control

Web Security Academy Referer-based access control

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

Home | My account

```
1.5 @ elafleur [evan] $
```

1.5.10 information-disclosure

Lab-infoleak-in-error-messages

Web Security Academy Information disclosure in error messages

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills! Continue learning »

The Giant Enter Key

5 stars

\$0.96

Apache Struts 2 2.3.31

```
information-disclosure @ elafleur [evan] $
```

Lab-infoleak-on-debug-page

Web Security Academy Information disclosure on debug page

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

 Share your skills!

Continue learning »

[Home](#)



Lab-infoleak-via-backup-files

Web Security Academy Source code disclosure via backup files

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

 Share your skills!

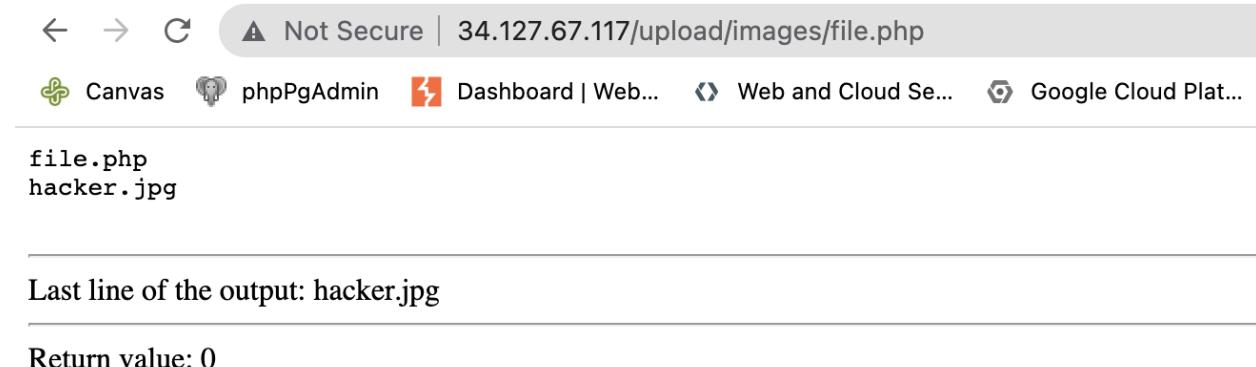
Continue learning »

[Home](#)

information-disclosure @ elafleur [evan] \$

1.5.11 WFP1: File Upload

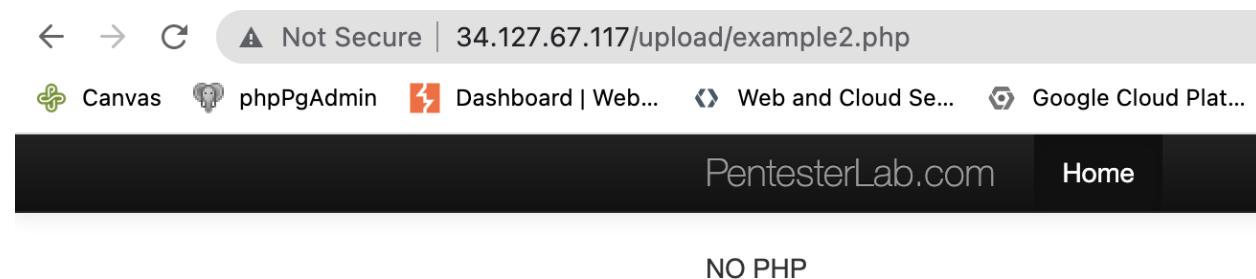
Example #1



A screenshot of a web browser window. The address bar shows 'Not Secure | 34.127.67.117/upload/images/file.php'. Below the address bar is a navigation bar with icons for Canvas, phpPgAdmin, Dashboard, Web and Cloud Services, and Google Cloud Platform. The main content area displays two files: 'file.php' and 'hacker.jpg'. Below this, a horizontal line separates the files from the output. The output section contains the text 'Last line of the output: hacker.jpg' and 'Return value: 0'.

Example #2

Uploading the same file used in the previous exercise:



A screenshot of a web browser window. The address bar shows 'Not Secure | 34.127.67.117/upload/example2.php'. Below the address bar is a navigation bar with icons for Canvas, phpPgAdmin, Dashboard, Web and Cloud Services, and Google Cloud Platform. The main content area has a dark header bar with 'PentesterLab.com' and 'Home' buttons. Below the header, the text 'NO PHP' is displayed.

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

Not Secure | 34.127.67.117/upload/images/file1.pHp

Canvas phpPgAdmin Dashboard | Web... Web and Cloud Se... Google Cloud Plat...

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

A screenshot of a web browser window. The address bar shows the URL: ac441fce1ec0f402c058b3e700f4009a.web-security-academy.net/files/avatars/hello.... The page content displays "Hello World!". Above the browser window, there is a navigation bar with various links like Canvas, phpPgAdmin, Dashboard | Web..., Web and Cloud Se..., Google Cloud Plat..., Truth Tables, React, and a Reading List. On the right side of the browser window, there are several icons for file operations.

A screenshot of the Web Security Academy website. The main heading is "Remote code execution via web shell upload". A green button labeled "LAB" with a checkmark icon and the word "Solved" is visible. Below the heading, there is a link "Back to lab description >>". At the bottom of the page, an orange banner says "Congratulations, you solved the lab!" with a "Share your skills!" button featuring a Twitter icon and a "Continue learning >>" link.

[Home](#) | [My account](#) | [Log out](#)

My Account

Your username is: wiener

A screenshot of a terminal window. The user has attempted to upload files using the command "file-upload" followed by the file path and the command "python3 *upload.py". The output shows three failed attempts, each ending with "\$[]".

```
File-upload @ elafleur [evan] $python3 *upload.py
file-upload @ elafleur [evan] $python3 *upload.py
file-upload @ elafleur [evan] $[]
```

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

The screenshot shows a web browser window for the 'Web Security Academy' lab titled 'Basic SSRF against the local server'. On the left, there's a sidebar with a 'View notes as gallery' button. The main content area has a heading 'Basic SSRF against the local server' and a 'Back to lab description >>' link. Below this, there's a large input field with placeholder text 'Enter URL to test...'. A note below the input field states: 'Admin interface only available if logged in as an administrator, or if requested from loopback'. At the bottom left, there's a user profile icon with the name 'Elafleur'.

▼ 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
```

X	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
▼ Form Data	view source	view URL-encoded					
<code>stockApi: http://stock.weliketoshop.net:8080/product/stock/check?productId=1&storeId=1</code>							

Web Security Academy Basic SSRF against the local server LAB Solved 💡

[Back to lab description »](#)

Congratulations, you solved the lab! Share your skills! [Continue learning »](#)

[Home](#) | [My account](#)

Gym Suit

★★★☆☆

\$27.15
1.6 @ elafleur [evan] \$[REDACTED]

basic-ssrf-against-backend-system

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

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-*--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

Web Security Academy  SSRF with filter bypass via open redirection vulnerability

[Back to lab description >>](#)

Congratulations, you solved the lab!

 [Share your skills!](#) [Continue learning >>](#)

[Home](#) | [My account](#)

WE LIKE TO  SHOP

```
1.6 @ elafleur [evan] $python3 *redirection.py
1.6 @ elafleur [evan] $
```

1.6.5 ssrf/blind out-of-band-detection

The screenshot shows the 'Blind SSRF with out-of-band detection' lab from the Web Security Academy. The top navigation bar includes the academy logo, the lab title, and status indicators for 'LAB' and 'Solved'. Below the title is a link to 'Back to lab description'. A large orange banner at the bottom of the page congratulates the user on solving the lab, with options to 'Share your skills!' or 'Continue learning'. The main content area displays a product listing for an 'Eco Boat' with a 5-star rating and a price of \$22.33. It features a silhouette of a boat against a sunset background. At the bottom of the page, there is a snippet of terminal output showing a successful exploit attempt.

1.7 XXE

Exploiting-xxe-to-retrieve-files

WebSecurity Academy | Exploiting XXE using external entities to retrieve files

LAB Solved 

[Back to lab description »](#)

Exploiting-xxe-to-perform-ssrf

Web Security Academy | Exploiting XXE to perform SSRF attacks

Back to lab description »

LAB Solved 

Congratulations, you solved the lab!

 Share your skills!

Continue learning »

[Home](#)



Portable Hat

★★★★★ \$65.75

[View details](#)

Hydrated Crackers

★★★★★ \$43.50

[View details](#)

Potato Theater

★★★★★ \$90.25

[View details](#)

Balance Beams

★★★★★ \$18.32

[View details](#)

```
1.7 @ elafleur [evan] $ python3 *ssrf.py
"Invalid product ID: {
  "Code" : "Success",
  "LastUpdated" : "2022-01-24T22:04:54.533979078Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "mY0nXHxAc0L4aT0ljIn",
  "SecretAccessKey" : "x8SqJIN4mf6DMkyt9ghWgTzjEp3VTL2vmKbcXD6X",
  "Token" : "SPWJmkkkfbjpmKJPipBvPJMNhV5B1NIDIHoUKjpCrnNTLb9hSKyRMYCJ13mD477L6Z8FxFtznI0IDEE7T1otVw9wga5R1fjm0T6bKpyAuavBMt1HfToyItBd2X
AD4p5Frz3uD1LxxCa14CCWTa3xZ7b9C6xicq2Gc6TQp7yL1M3kDjCywEhMUJZcCKxx7ZTICZTY0EoKU8QC07k4ePDLMyT6zoeBNVyqjMvUgYNuZYedwDAUvMVSEJ1jV02n3qX",
  "Expiration" : "2028-01-23T22:04:54.533979078Z"
}"
```

Xinclude-attack

Web Security Academy Exploiting XInclude to retrieve files

Back to lab description »

LAB Solved

Congratulations, you solved the lab!

Share your skills!

Continue learning »

Home



1.7 @ elafleur [evan] \$

Xxe-via-file-upload

A screenshot of a web browser window. The address bar shows the URL: "ac561f7d1e603d91c06e42a4009a0070.web-security-academy.net/post/comment/avatars?filename=1.png". The page content is a black background with a single white rectangular box in the center containing the text "elafleur". The browser interface includes various navigation and extension icons at the top.



Exploiting XXE via image file upload

LAB Solved



[Back to lab description >>](#)

Congratulations, you solved the lab!

Share your skills!

[Continue learning >>](#)

[Home](#)



Exploit successful! Exploit successful!
1.7 @ elafleur [evan] \$