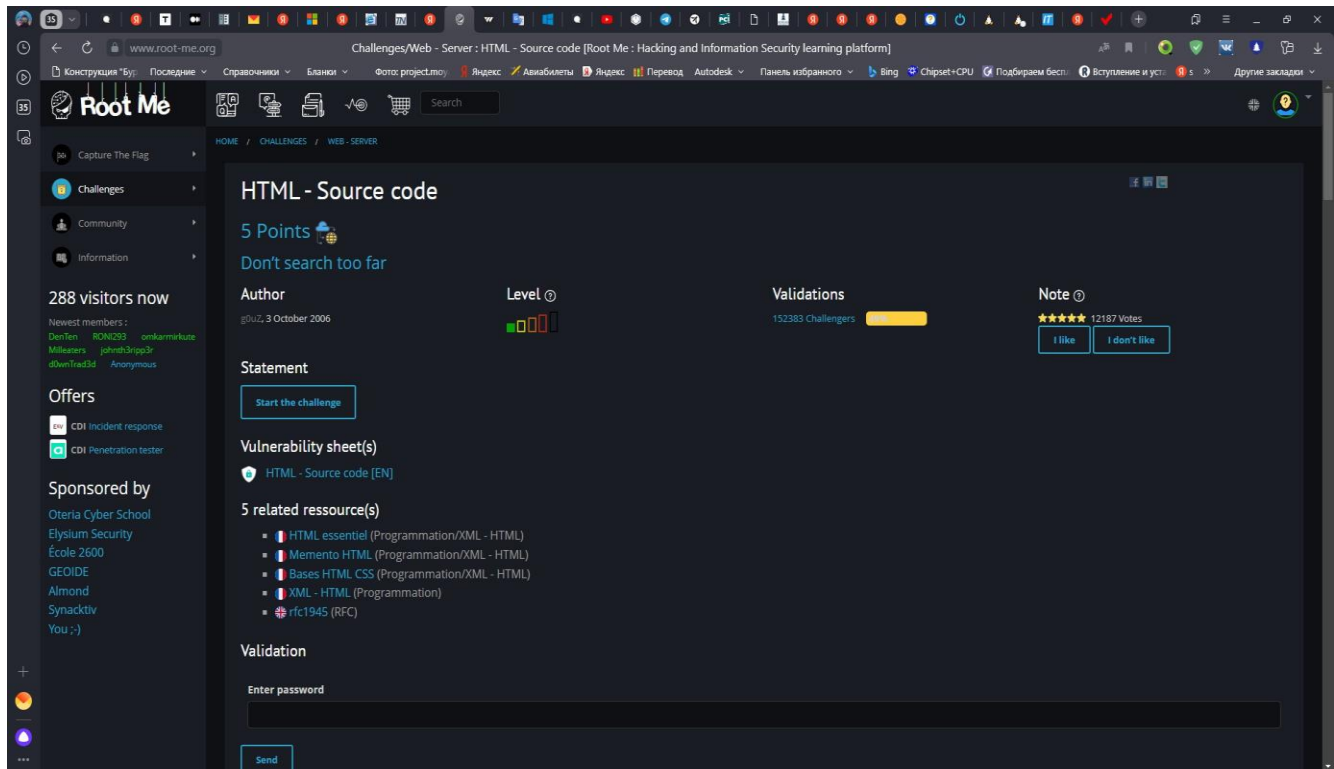


Web Application Security Testing -> Root Me (HTML_SourceCode, HTML_UserAgent, WeakPassword, HTTP-Headers, HTTP_POST, HTTP_VerbTampering)

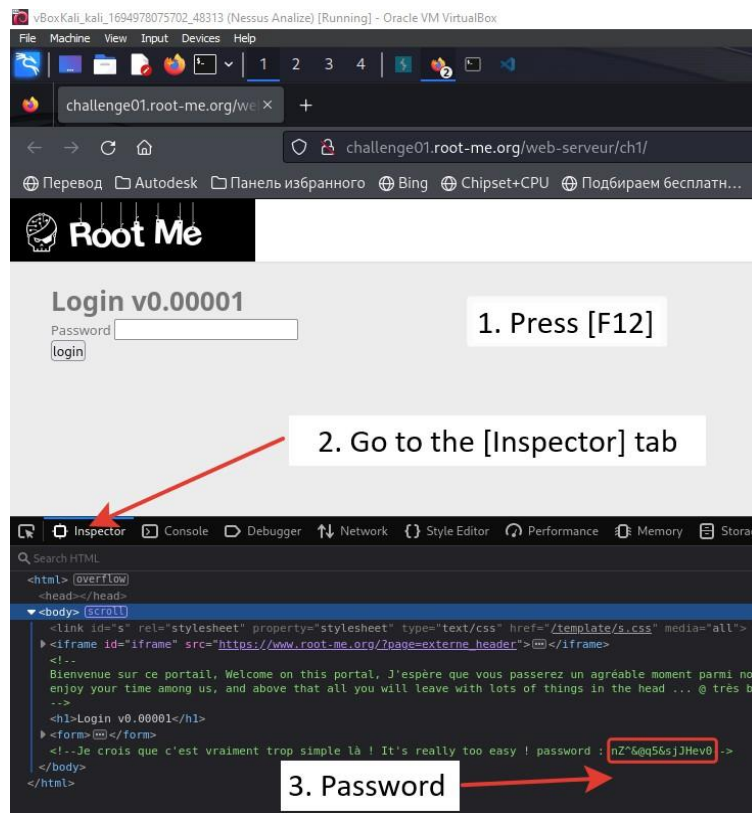
- Web Application Security Testing -> Root Me (HTML_SourceCode, HTML_UserAgent, WeakPassword, HTTP-Headers, HTTP_POST, HTTP_VerbTampering)
 - Root Me (HTML - Source code)
 - Root Me (HTML - User-agent)
 - Root Me (Weak password)
 - Root Me (HTTP - Headers)
 - Root Me (HTTP - POST)
 - Root Me (HTTP - Verb tampering)

Root Me (HTML - Source code)

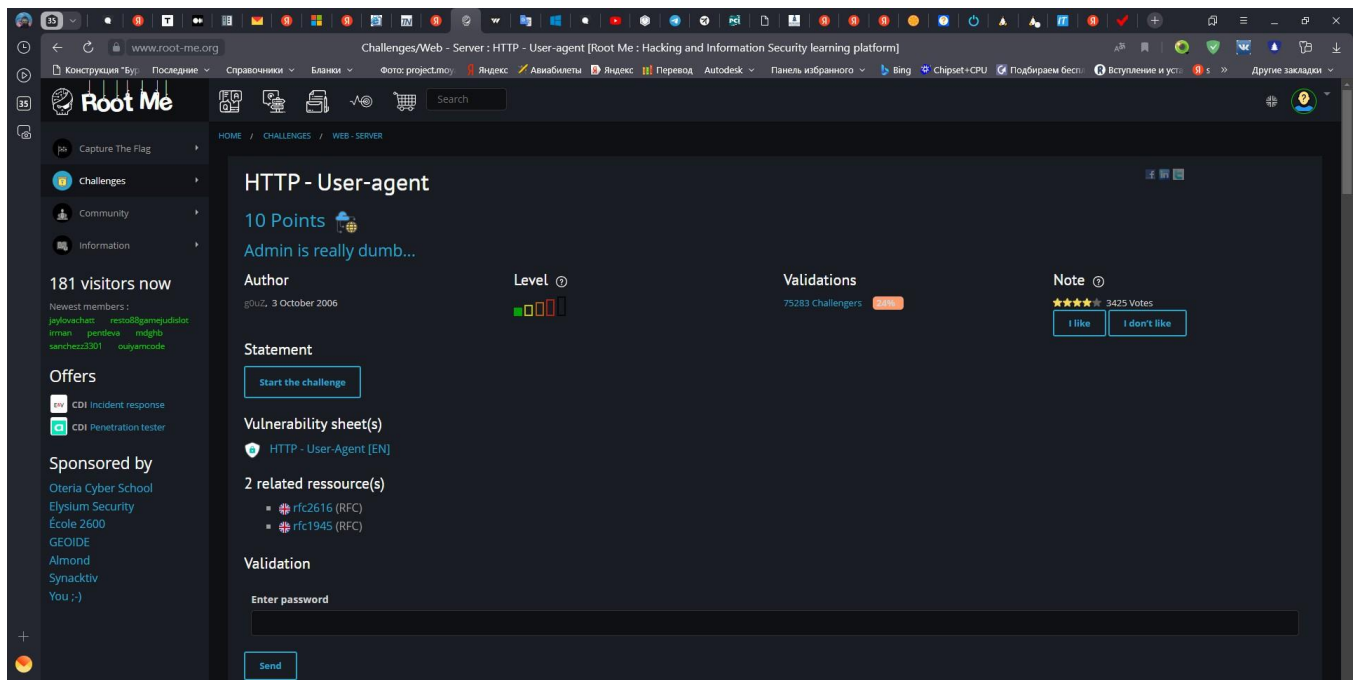


Solution:

1. Press the [F12] key to open the developer tools.
2. Go to the [Inspector] tab to research the source code.
3. The html page contains a commented out password.

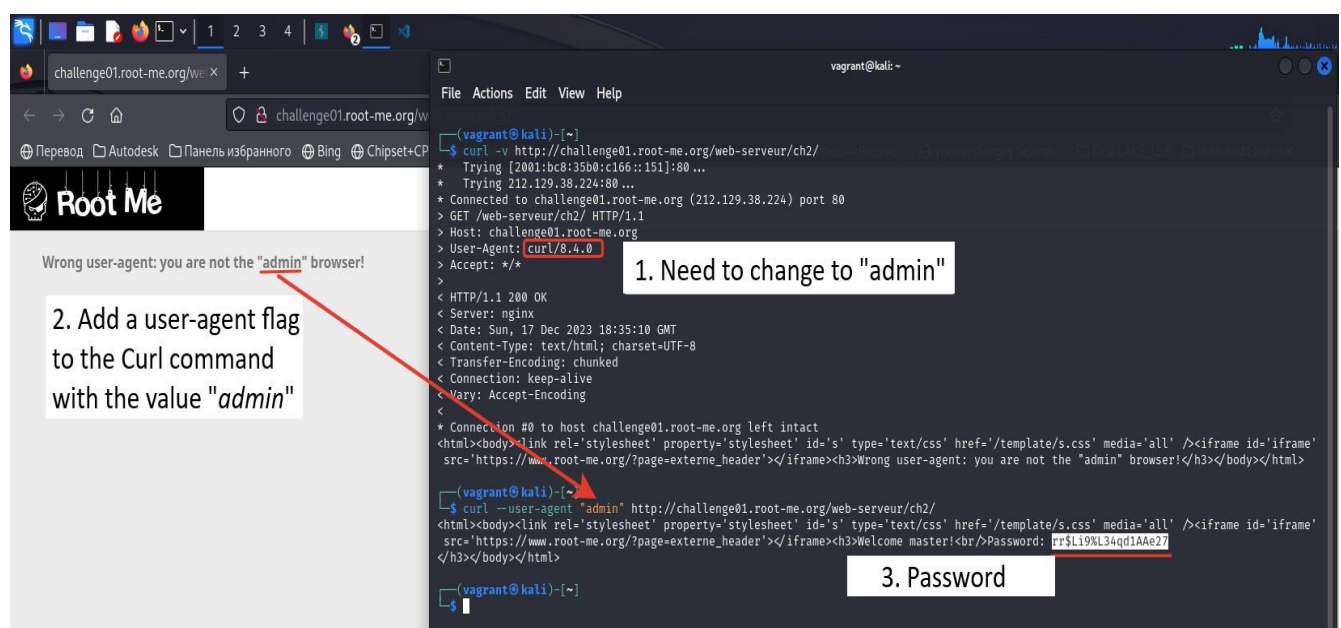


Root Me (HTML - User-agent)

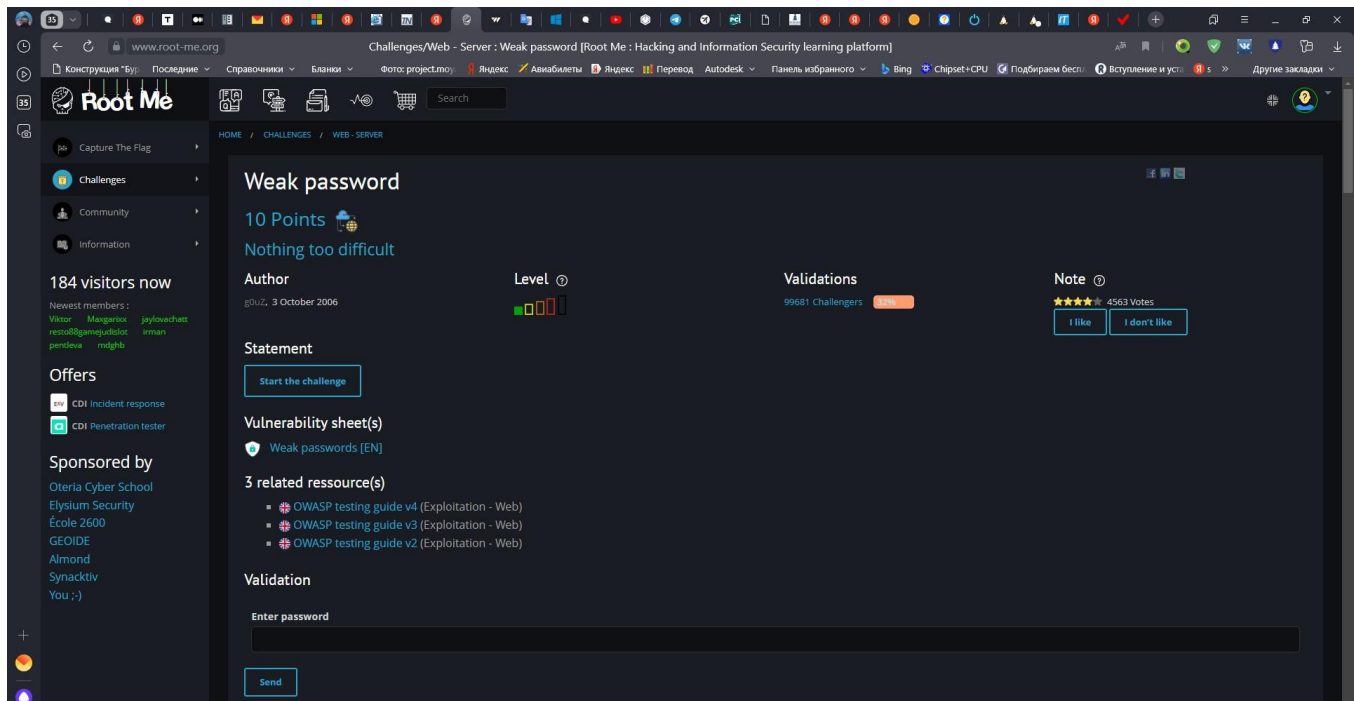


Solution

1. On the challenge page there is a hint that we are using the wrong browser.
2. If we look at the server response, we can verify that the User-agent contains `curl/8.4.0`, but we must use the browser "`admin`". Of course, to solve this problem, we should change the User-Agent request header.
3. Add the `--user-agent` flag with the modified header "`admin`".
4. Collect the flag



Root Me (Weak password)

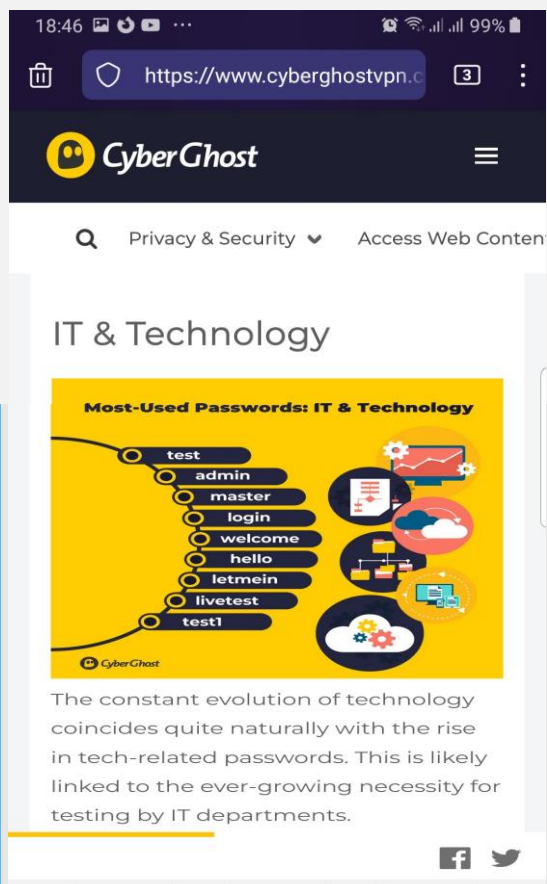


This challenge can be solved in 2 ways:

1. Brute force method
2. Use the lists of *"Most-used Passwords"*

NOTE:

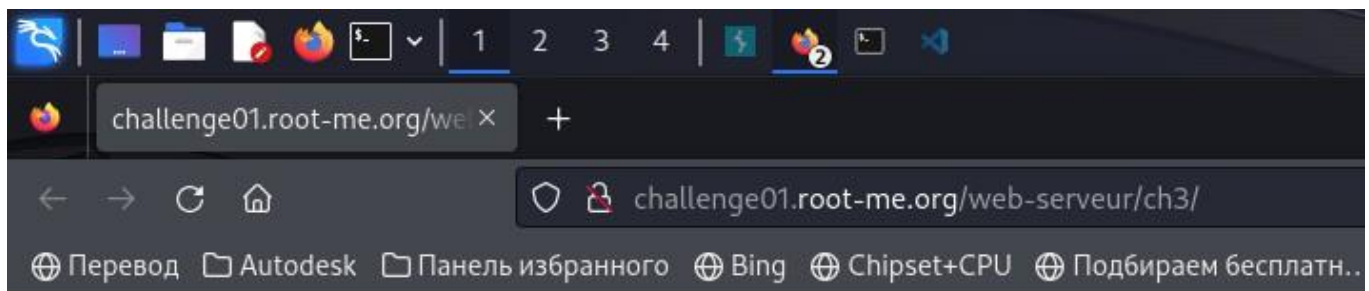
For example, I used 10 lists from CyberGhost



Honestly, every time the first thing I do is try the following pairs:

- `admin/''`,
- `admin/admin`, <- valid
- `admin/password`,
- `admin/password1`,
- `admin/password123`,
- `admin/passw0rd`,
- `admin/passwd`,
- `root/''`,
- `root/root`,
- `root/toor`,
- `root/password`,
- `root/passw0rd`,
- etc.

After manually entering the username - `admin` and password - `admin` I logged in.

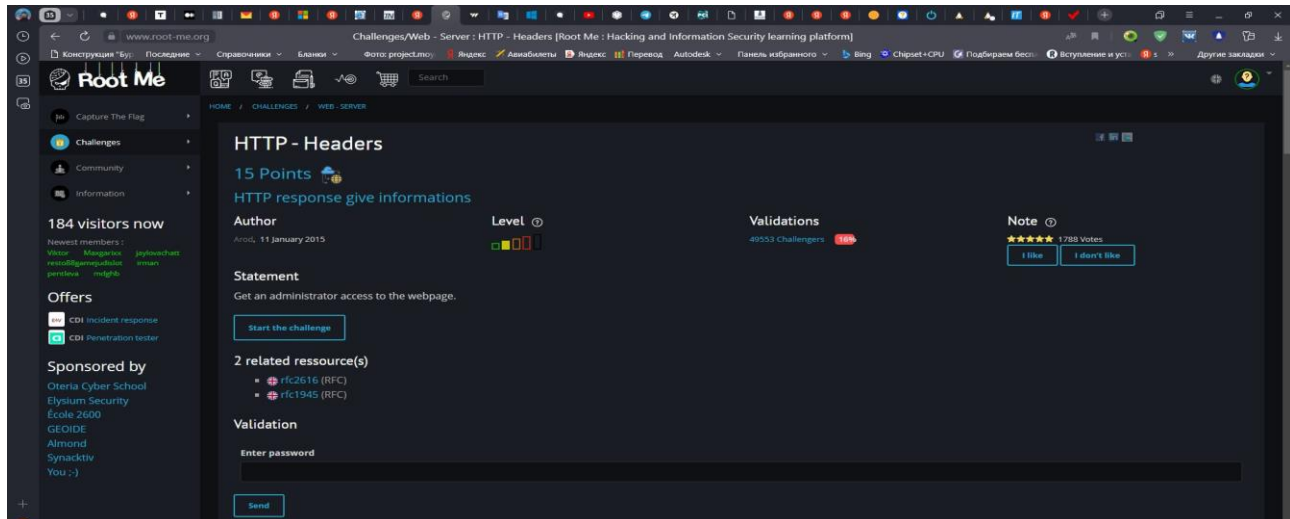


Bien joué, vous pouvez utiliser ce mot de passe pour valider le challenge

User: *admin* password: *admin*

Well done, you can use this password to validate the challenge

Root Me (HTTP - Headers)



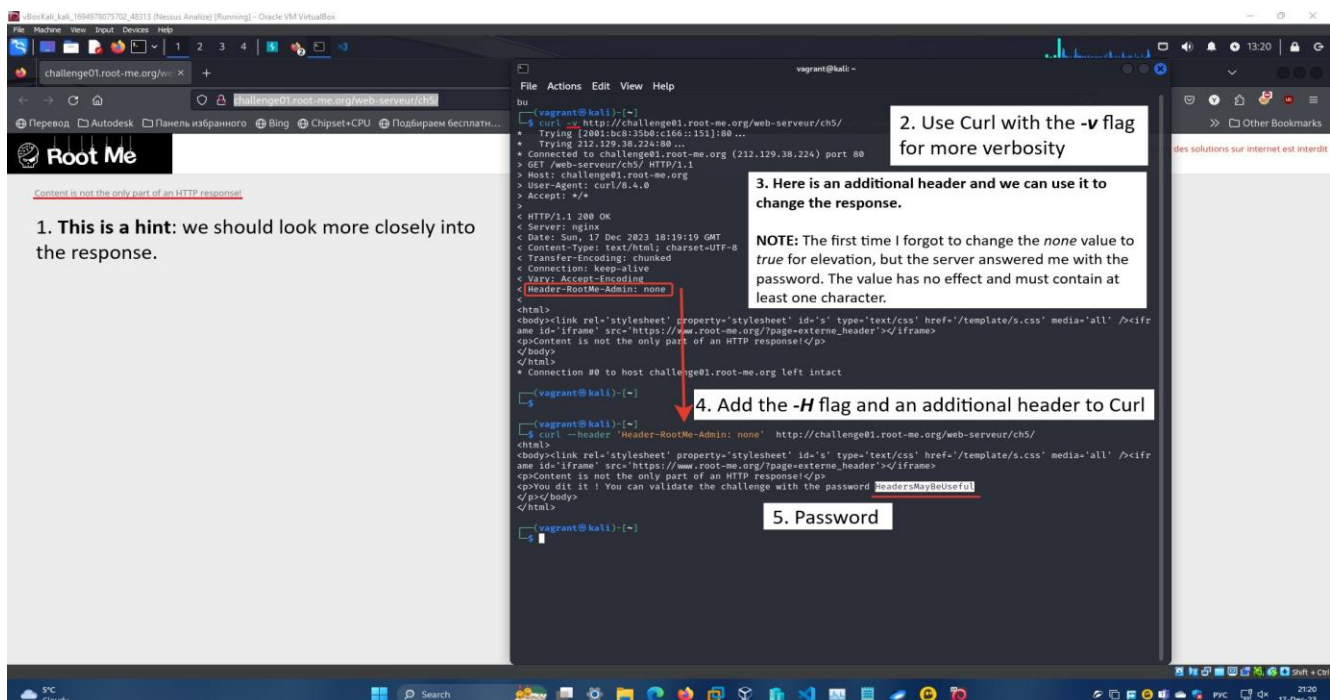
Solution

1. On the challenge page there is a hint that the content isn't only part of an HTTP response.
2. Use Curl with the **-v** flag for more verbosity.
3. Here is an additional header and we can use it to change the response.
4. Add the **-H** or **--header** flag to modify the additional header with the value of **Header-RootMe-Admin: true** to Curl.

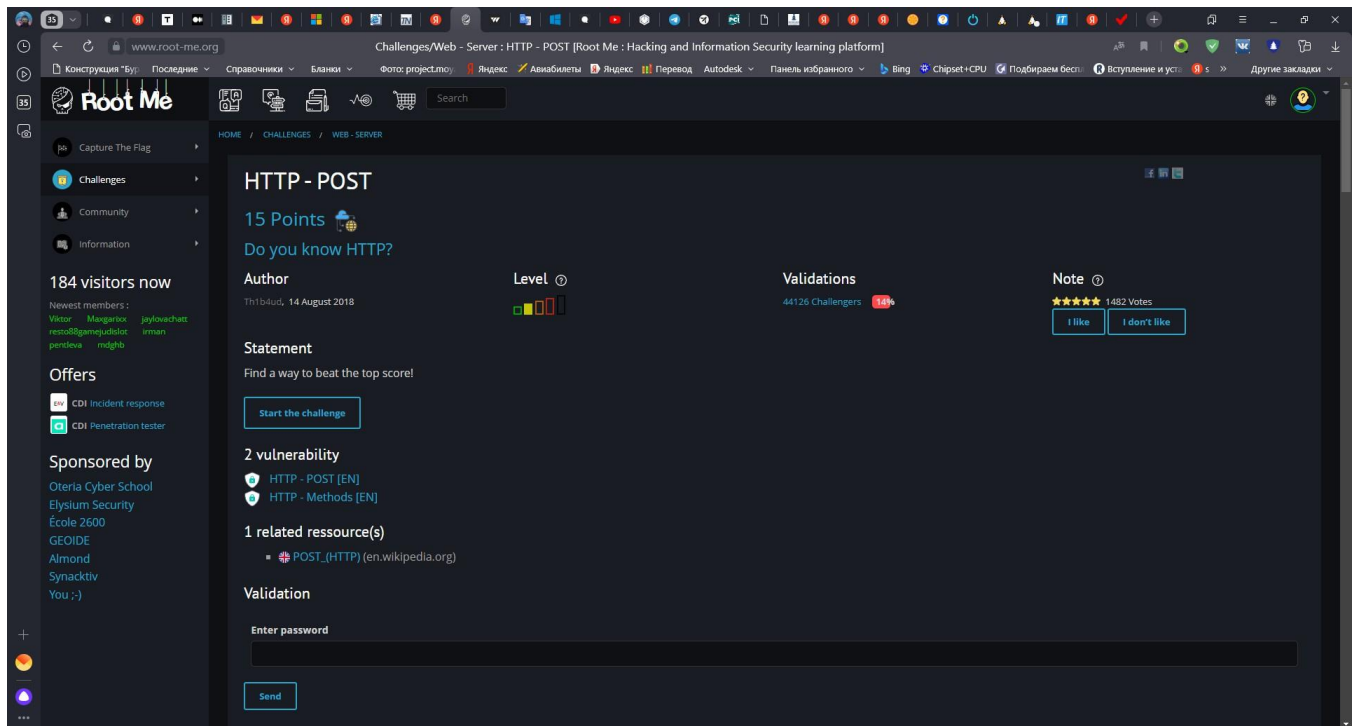
NOTE:

The first time I was in a hurry and copied a line from the answer, I forgot to change the **none** value to **true** (for elevation) and press **[Enter]**. This is amazing, the server responded to me with a password page. I sent a few additional requests and found that the value had no effect and must contain at least one character.

5. The response contains the password.



Root Me (HTTP - POST)

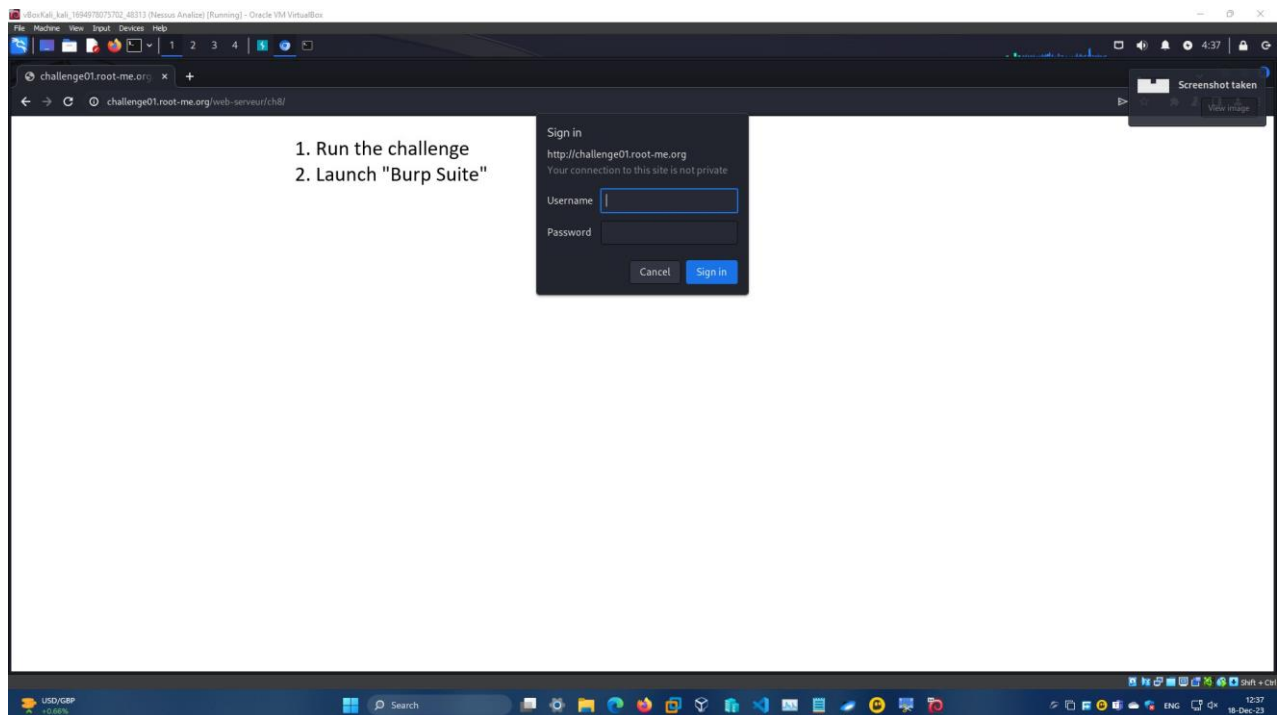


Solution

NOTE:

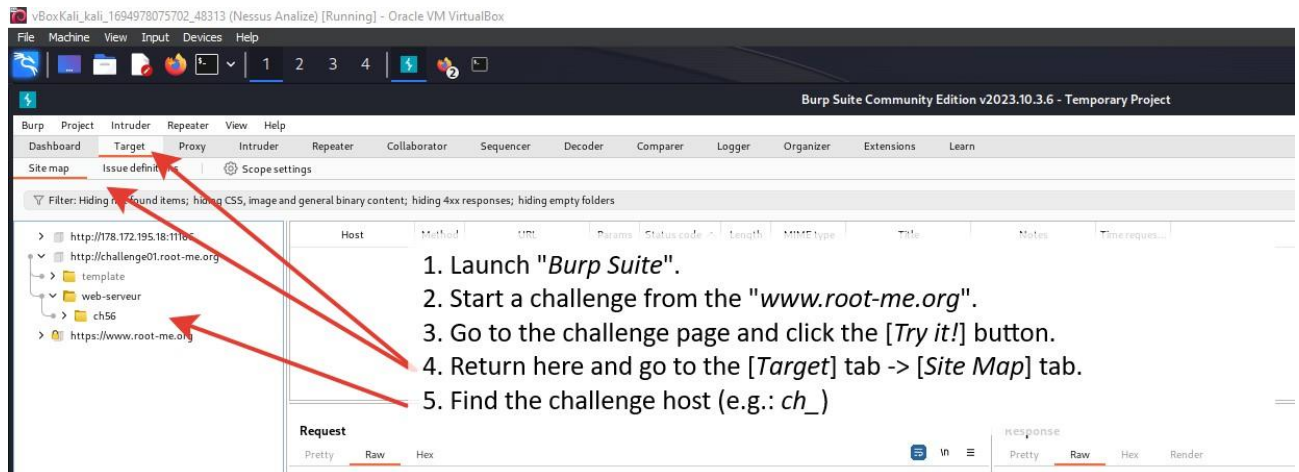
To solve this problem, Curl is not enough, you need to use a proxy (e.g.: *Burp Suite*).

1. Run the challenge
2. Launch "*Burp Suite*"

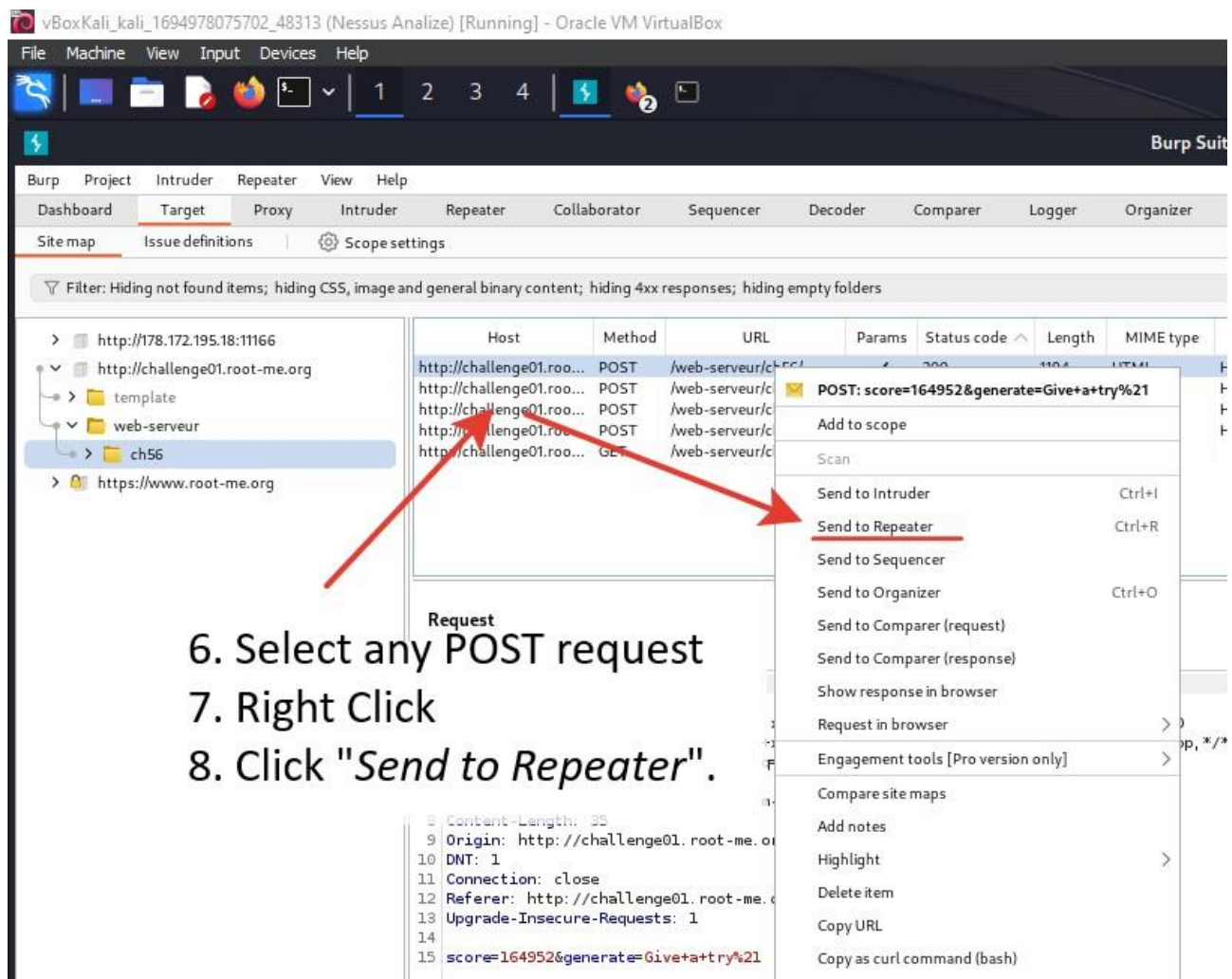


3. Start a challenge from the "<www.root-me.org>".
4. Go to the challenge page and click the [Try it!] button.

5. Return here and go to the [Target] tab -> [Site Map] tab.
6. Find the challenge host (e.g.: ch_)



7. Select any POST request
8. Right click
9. Click "Send to Repeater"



10. Go to the [Repeater] tab
11. Change the value to 1000000

12. Click the [Send] button

13. Collect the flag

9. Go to the [Repeater] tab

11. Click the [Send] button

10. Change the value to 1000000

12. Collect the flag

Request

```
1 POST /web-servreur/ch56/ HTTP/1.1
2 Host: challenge01.root-me.org
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: ru,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 86
9 Origin: http://challenge01.root-me.org
10 DNT: 1
11 Connection: close
12 Referer: http://challenge01.root-me.org/web-servreur/ch56/
13 Upgrade-Insecure-Requests: 1
14
15 score=1000000&generate=Give+a+try%21
```

Response

```
22 </li>
23 <li>
24   Score to beat: <strong>
25     999999
26   </strong>
27 </li>
28 </ul>
29
30 <p>
31   Wow, 1000000! How did you do that? :o
32 </p>
33
34 <p>
35   Flag to validate the challenge: <strong>
36     H7tp_h4s_N0_s3cr37s_F0r_y0U
37   </strong>
38 </p>
39
40 <form action="" method="post" onsubmit="document.getElementById( 'score' ).value = Math.Flo
41 <input type="hidden" name="score" value="-1" />
42 <input type="submit" name="generate" value="Give a try!">
43 </form>
44 </body>
45 </html>
```

Root Me (HTTP - Verb tampering)

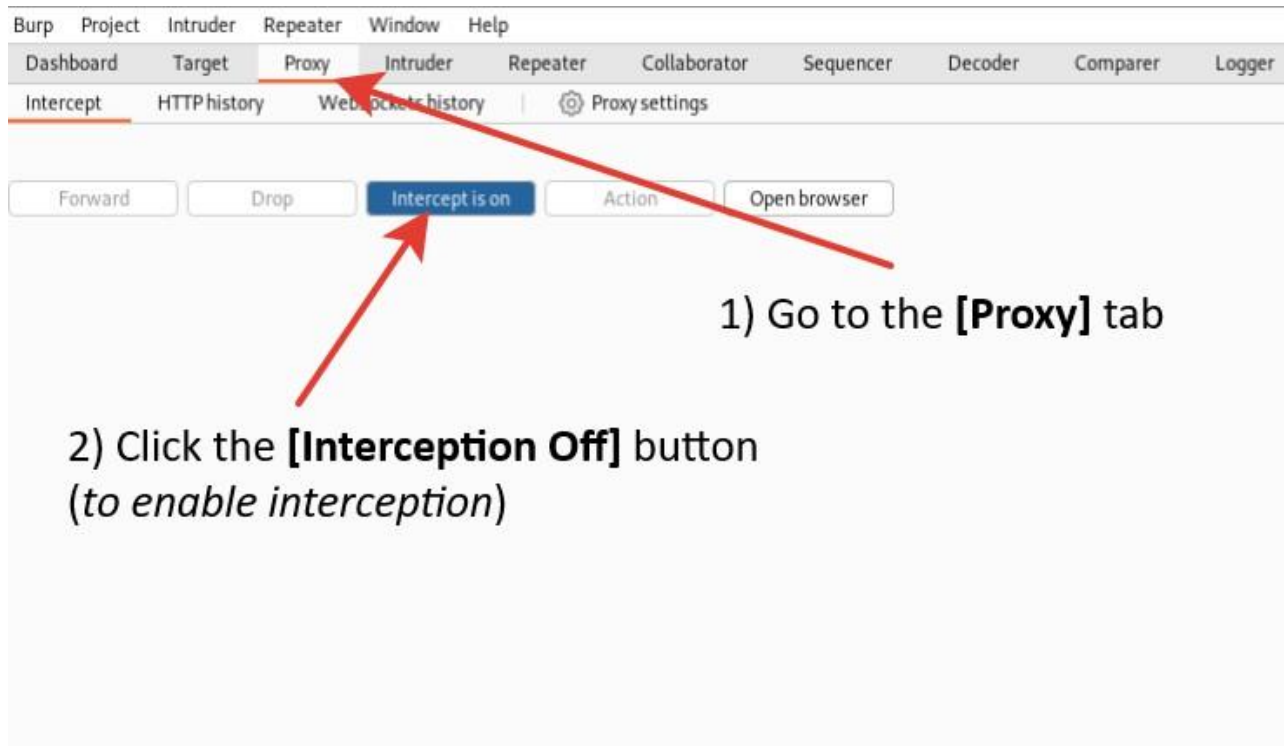
The screenshot shows the Root Me website interface. The top navigation bar includes links for Getting Started, Parrot OS, Hack The Box, OSINT Services, Vuln DB, Privacy and Security, Learning Resources, and Kali Linux. The main header features the Root Me logo and a search bar. The left sidebar contains a menu with 'Capture The Flag', 'Challenges', 'Community', and 'Information', along with a visitor count of 286 and a list of newest members. The main content area displays the challenge details for 'HTTP - Verb tampering', which is worth 15 points and is categorized under 'HTTP authentication'. The author is g0uZ, and the challenge was created on 3 February 2011. The statement reads: 'Bypass the security establishment.' A 'Start the challenge' button is prominently displayed. Below this, there is a 'Vulnerability sheet(s)' section with a link to 'HTTP - Methods [EN]' and a '3 related ressource(s)' section listing RFC 2617, RFC 2069, and a link to 'HTTP basic authentication and digest authentication (Exploitation - Web)'. The page also includes a 'Validation' section.

Solution

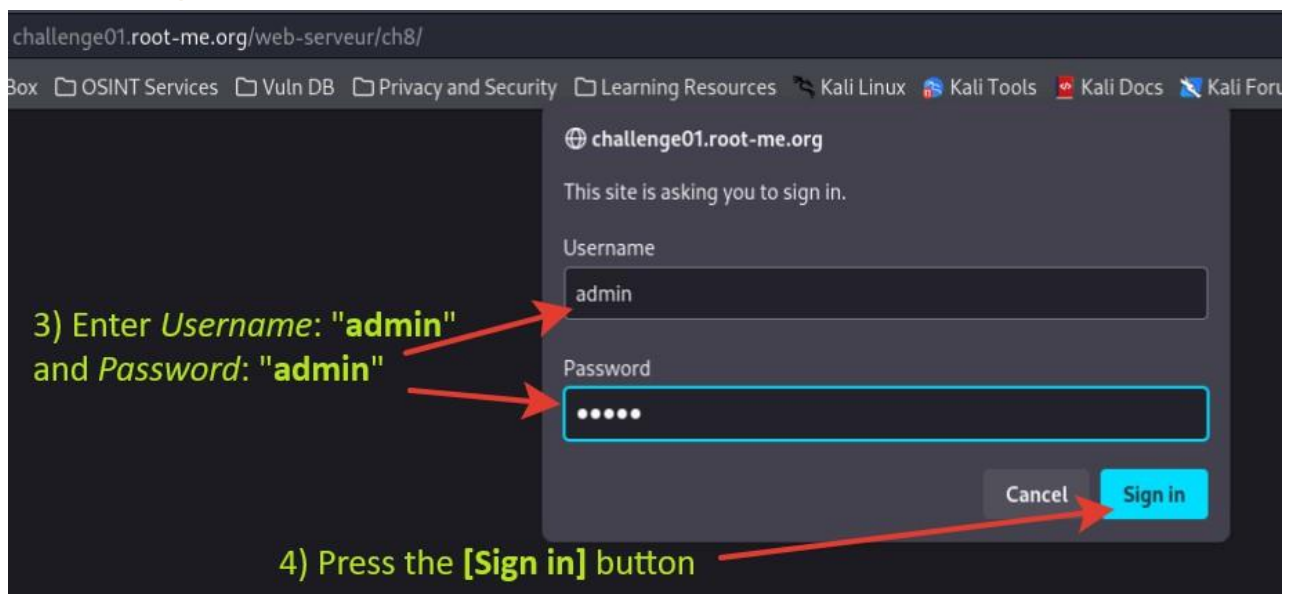
1. Run the challenge

The screenshot shows a login dialog box for the website challenge01.root-me.org. The dialog box has a title bar with the website name and a message stating 'This site is asking you to sign in.' It contains two input fields: 'Username' and 'Password'. At the bottom right, there are two buttons: 'Cancel' and 'Sign in'.

2. Launch "Burp Suite".
3. Go to the [Proxy] tab.
4. Click the [Interception Off] button (to enable interception).



5. Enter Username: `admin` and Password: `admin`.
6. Press the [Sign in] button.



7. After intercepting the request, the Burp Suite window will appear.

The *GET method* is used to retrieve data from the server. This is a **read-only** method. But inside the intercepted request we can find the line `Authorization: Basic YWRtaW46YWRtaW4=` - these are specific changes to the resource. To apply the change, we need to use another method that can change the resource. These methods:

- *POST* - creates a new resource,

- **PUT** - updates an existing resource, but its body must contain the complete structure of the modified resource,
- **PATCH** - updates an existing resource and contains in its body only specific changes for the resource.

In this case, we **must replace** the **GET** method with **PATCH** to apply the authorization state change.

8. Click the **[Forward]** button (to submit the modified request).
9. Click the **[Interception is on]** button (to disable interception).

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A request to `http://challenge01.root-me.org:80 [212.129.38.224]` is intercepted. The 'Intercept is on' button is highlighted. The request details show a `GET /web-serveur/ch8/ HTTP/1.1` with an `Authorization: Basic YWRtaW46YWRTaW4=` header. Red arrows and text annotations are overlaid on the image:

- 5) Replace the "GET" method with "PATCH"
- 6) Click the **[Forward]** button (to submit the modified request)
- 7) Click the **[Interception is on]** button (to disable interception)
- NOTE:** These are specific changes to the resource that need to be applied.

10. Return to the browser.

11. Collect the "Flag"

The screenshot shows a web browser at `challenge01.root-me.org/web-serveur/ch8/`. The password field is highlighted with a red box and contains the text: **Mot de passe / password : a23e\$dme96d3saez\$\$prap**

8) Collect the "Flag"