Web Security and Malware Analysis

Assignment 3 – 26/11/2020

Goal: Solve each task by providing a brief explanation of the solution that you adopted. Answers should mainly include screenshots and some brief comments. Use the provided template to write your assignment. Remember that solving the Bonus task is not compulsory (it is proposed only for interested people).

What to do: Save the Assignment as a PDF and submit it to the E-Learning platform (**Section: Assignment 3**). The file should be named as follows: **"websec_report_3_name_surname.pdf".** Remember to include your name, surname, and matriculation number in your report.

Starting Notes: Most of this assignment's tasks will concern attacking the Damn Vulnerable Web Application (DVWA).

You can download the Virtual Machine of the course to execute DVWA - https://hub.docker.com/r/vulnerables/web-dvwa (or you can download it directly from Docker).

To run the DVWA:

docker run --rm -it -p 80:80 vulnerables/web-dvwa

Then open the browser and give localhost:80 in the URL bar.

You will be asked to solve tasks by changing their difficulty progressively. To change it, go to DVWA and select the difficulty by clicking on DVWA Security on the menu (you will be asked to solve tasks from low to high).

Task 1 – Dictionary-Based Brute-Force

Attack the Brute-Force section of the DVWA at **easy** and **medium** levels by performing a dictionary attack.

For each level, you are required to retrieve the passwords for the following users:

pablo

admin

1337

gordonb

smithy

You have to complete the two tasks by using **one of the following tools** (you can try a different tool for the two levels or use the same one):

- **BURP Turbo Intruder** (download it from Burp Extender)
- **WFUZZ**
- **Hydra**

For a tutorial on Turbo Intruder, check:

https://portswigger.net/research/turbo-intruder-embracing-the-billion-request-attack

For good password sets, check:

https://github.com/danielmiessler/SecLists/tree/master/Passwords/Common-Credentials

Task 2 – Attacks against Sessions

Complete the "Weak Session IDs" tasks of the DVWA at the **low, medium,** and **high** difficulty levels. The goal of these tasks is to understand how the *server generates the dvwaSession cookie*. You are required to answer the following questions:

- Try to solve each task *without looking at the PHP source code*. What can be inferred by only observing the server response to the generated requests? How is the cookie created? **Hint: Use BURP Repeater.**
- Use **BURP Sequencer** (tutorial: https://portswigger.net/burp/documentation/desktop/tools/sequencer) to launch multiple requests at medium and high difficulty. What can we say in terms of the predictability of the cookie? How are the plots related to the characteristics of the cookie?

Task 3 – More PHP

Given the PHP code in the file **assignment_3_3.php**, answer the following questions:

1. What does the PHP code do?
2. Is this code vulnerable? If yes, describe how you would crack it.
3. How would you enforce the security of the code?

Note: especially for question number 3, there is not only one "right" answer. Try to use your creativity.

Bonus Task – Harder Bruteforce

Solve task 1 at the **high** difficulty level.

Carefully read this tutorial:

https://blog.g0tmi1k.com/dvwa/bruteforce-high/

The best way possible to solve the task is by using the **patator** tool: https://github.com/lanjelot/patator. However, patator would require Linux to work at its best. A Windows version is also available at https://github.com/maaaaz/patator-windows, but it seems to create some problems.

By following the previous tutorial, you can also use **BURP+Intruder (careful: Turbo Intruder will not work)**. However, this solution is significantly slower.

Hence:

A. If you use Linux+Patator, provide the passwords **for all the users**.
B. If you use Burp+Intruder, provide the passwords for the users **admin, smithy, and pablo**.