Homework #2: Web Hacking Competition

Due: May. 2, 11:59 PM
Responsible TA: Zeewung Shin (zeeshin@unist.ac.kr)

- **Assignment Description.** Capture The Flag (CTF) is a competition that involves capturing a flag as a proof of solving challenges. For each problem, there is a specifically configured server that stores a flag, which is a secret string used by participants as a proof of successfully attacking the corresponding server.
- **Submission guidelines.**
  - You should submit both your report and flags.
    - * Solve the problems and submit the flags to our CTF homework webpage. The URL of this webpage is kept secret. You can obtain it by solving Problem 0. This server can only be accessed from the UNIST internal network. Please use a VPN to access from outside.
    - * You should upload a single PDF file on BlackBored. Your report must describe the answer to each question in this homework. Your report (`Your_ID-hw2.pdf`) can be written in either English or Korean.
  - The name of the PDF file should have the following format: `Your_ID-hw2.pdf`. For example, if your ID is 20231234, then you should submit a file named `20231234-hw2.pdf`.
  - Late submission will be assessed a penalty of 10% per day (We will only accept late submissions of up to 3 days).
- **Capture The Flag (CTF) guidelines.**
  - You can find each problem on our CTF homework webpage. To access the URL of our CTF webpage, you must first solve Problem 0.
  - If you solve each challenge, you will be able to obtain a flag. Submit the found flag to the website. Each flag is in the following format: `flag{[0-9a-f{32}]}`
    (e.g., `flag{1a79a4d60de6718e8e5b326e338ae533}`)
  - Your score in the CTF scoreboard is nothing to do with the actual score for your homework. Your work will be evaluated according to *the quality of the submitted report*. Please make sure to write the report as thoroughly and clearly as possible.
  - Do not attack the CTF environments, including web services!
  - If you think the services are not working properly or have any question, please post your question on the BlackBored.
- **XSS attack guidelines.**
  - For XSS challenges (Problems 8–11), we prepared an imaginary victim and stored the flag in this victim's cookie. Therefore, you need to find a vulnerability in the web service, create a malicious URL, and send the URL to this victim so that you can exfiltrate the cookie.
  - Especially, you should manually send a malicious URL to this victim user via `check.php`. When you send the URL to the victim user, victim's browser will automatically visit the URL. This process might take at most 10 seconds.

- To read the user's cookie, you may need your own server (Recall the“5. Send victim's sensitive data” step in pages 56 of the lecture slide “Client-side Web Security (1)” [1]). We recommend using the following service: `https://webhook.cool/`. Through this service, you will get your unique URL that acts like `attacker.com`, and you will be able to see logs (e.g. query strings, form contents, etc.) of all accesses to that URL.

Problem 0. Warm-up: Find the CTF Webpage (2 points)

The URL of the CTF homepage is hidden somewhere inside our course webpage [2].

(a) (1 point) What is the URL of our CTF homepage?

(b) (1 point) Describe how you discovered this URL.

Problem 1. Login (10 points)

In this challenge, you should sign in to the web service as `admin` without knowing the password. Once you successfully sign in to the service, you will find the flag.

(a) (2 points) Describe the SQL query used in this web service.

(b) (2 points) Describe the vulnerability in this web service.

(c) (5 points) Explain in detail how you can exploit this vulnerability to get the flag.

(d) (1 point) What is the flag?

Problem 2. Password (15 points)

In the same web service used in Problem 1, you now need to find out the password of `admin`, which is the flag of this challenge.

(a) (13 points) Explain in detail how you can exploit this vulnerability to get the flag (i.e., `admin`'s password).

(b) (2 points) What is the flag?

Problem 3. Password++ (25 points)

As in the previous challenge, your goal is to find out the password of `admin`, which is the flag of this challenge.

(a) (3 points) Guess the SQL query used in this web service.

(b) (1 point) What is the HTTP method of the request sent by the browser when you click the “Sign in" button?

(c) (1 point) What is the body content of the HTTP request sent by the browser when you click the “Check!" button?

(d) (5 points) Describe the vulnerability in this web service.

(e) (10 points) Explain how you can exploit this vulnerability to get the flag (i.e., `admin`'s password). Include the script (e.g, python3 script) to exploit it.

(f) (5 points) What is the flag?

Problem 4. Check Duplicate (35 points)

In this challenge, your goal is the same as in the previous challenge. However, unlike in the previous web services, this web service manages `admin`'s account information in a separate table. That is, `admin` does not exist in the `user` table (hint: Our server uses SQLite database system).

(a) (4 points) Guess the SQL query used in this web service.

(b) (2 points) Describe the vulnerability in this web service.

(c) (12 points) What is the name of the table used for managing the admin's account? Explain in detail how you found it.

(d) (12 points) Explain in detail how you can exploit this vulnerability to get the flag.

(e) (5 points) What is the flag?

Problem 5. Get Color (10 points)

In this challenge, you need to read the contents of the file `/var/www/flag.txt`.

(a) (3 points) Describe the vulnerability in thie web service.

(b) (6 points) Explain in detail how you can exploit this vulnerability to get the flag.

(c) (1 point) What is the flag?

Problem 6. Uploader (15 points)

In this challenge, you have to upload an arbitrary PHP file that reads the `/var/www/flag.txt` file via a shell command.

(a) (6 points) Describe in pseudocode what content-filtering checks are implemented on the server for uploaded files.

(b) (6 points) Explain how you can exploit this vulnerability to get the flag.

(c) (3 points) What is the flag?

Problem 7. Uploader++ (25 points)

This challenge has more advanced content-filtering checks than the previous challenge. You have to upload an arbitrary PHP file that reads the `/var/www/flag.txt` file via a shell command.

(a) (10 points) Describe in pseudocode what content-filtering checks are implemented on the server for uploaded files.

(b) (10 points) Explain how you can exploit this vulnerability to get the flag.

(c) (5 points) What is the flag?

Problem 8. Search V1 (10 points)

In this challenge, you need to read the victim's cookie!

(a) (2 points) Describe the vulnerability in this web service. You need to specify which type of XSS this vulnerability is.

(b) (7 points) Explain how you can exploit this vulnerability to get the flag.

(c) (1 point) What is the flag?

Problem 9. Search V2 (20 points)

In this challenge, Search V1 has been more safer. You need to read the victim's cookie!

(a) (5 points) Compared to Problem 8, what defense mechanism has been added? Describe the added defense logic in detail.

(b) (8 points) Describe how you can bypass the deplyoed defense mechanism.

(c) (4 points) Explain how you can exploit this vulnerability to get the flag.

(d) (3 points) What is the flag?

Problem 10. Service Center (25 points)

In this challenge, you need to read the victim's cookie!

(a) (10 points) Describe the vulnerability in this web service. You need to specify which type of XSS this vulnerability is.

(b) (11 points) Explain how you can exploit this vulnerability to get the flag.

(c) (4 points) What is the flag?

Problem 11. Search V3 (30 points)

In this challenge, Search V2 has been more safer. You need to read the victim's cookie!

(a) (5 points) Describe the vulnerability in this web service. You need to specify which type of XSS this vulnerability is.

(b) (10 points) Explain the deployed defense logic to prevent vulnerabilities.

(c) (10 points) Explain how you can exploit this vulnerability to get the flag.

(d) (5 points) What is the flag?

Note

- **Be careful about plagiarism!** Do not share the flag value. If your problem-solving process is incorrect but the flag value is correct, it will be considered plagiarism.
- **Grading policy.** We will grade your report based on how self-contained it is and how clearly you describe your problem-solving process. Even if you cannot solve the problem completely, we recommend that you describe your approach in as much detail as possible to receive partial credit.
- **Questions.** If you have any requests or questions (technical difficulties, late submission due to inevitable circumstances, etc.), please ask the TAs on Blackboard. We generally encourage the use of Blackboard for discussions. However, for urgent issues or secret issues, you can send an email to the responsible TA, Zeewung Shin.

References

[1] CSE467: Computer Security. 2025. 10. Client-side Web Security (1). https://websec-lab.github.io/courses/2025s-cse467/slides/lecture10-client-side.pdf.

[2] CSE467: Computer Security. 2025. Course Homepage. https://websec-lab.github.io/courses/2025s-cse467/.