LAB 4 – Cross-site-scripting e SQL injection

Sicurezza e Privatezza

The learning objective of this lab is for students to get the basics of web security with cross-site scripting and SQL injection. After finishing the lab, students should be able to craft malicious payload to exploit vulnerable web pages and understanding how to protect a system against these types of attack.

Students are invited to complete the following exercises, writing a txt file that include payloads used for exploitation with a little description of what the payload do and submit it to SPLab@di.unimi.it, with subject: 2020Lab4NameSurnameUniversityid (ex. 2020Lab4MatteoZoia876271). Message not conforming to these rules will be automatically rejected.

The first five students who will respond correctly to the entire lab will receive a bonus of 0.5pt on the final grade, 0.25pt on XSS attacks part and 0.25pt on the SQL injection part.

XSS attacks

Exercise #1: Ma Spaghet! (warmup)

We will start in a simple way, as suggested in the lesson you need to analyze the given code MaSpaghet!.html and find a way to pop-up an alert with an xss attack.
The file xss1.html contains only the necessary lines of code to do the exercise correctly. To make it works well inside a browser you can wrap the lines in a basic html page.

Here is the vulnerable code:

```
<!-- Challenge -->
<h2 id="spaghet"></h2>
<script>
    spaghet.innerHTML = (new URL(location).searchParams.get('somebody') || "Somebody") + " Toucha Ma Spaghet!"
</script>
```

The student has to find the vulnerability (XSS attack) and write in a txt file the payload to pop-up an alert in the website page.

Exercise #2: Ah That's Hawt

Similarly to the previous exercise you need to analyze the given code in AhThatsHawt.html and find a way to pop-up an alert with an xss attack.
The file .html contains only the necessary lines of code to do the exercise correctly. To make it works well inside a browser you can wrap the lines in a basic html page.

Here is the vulnerable code:

```
<!-- Challenge -->
<h2 id="will"></h2>
<script>
    smith = (new URL(location).searchParams.get('markassbrownlee') || "Ah That's Hawt")
    smith = smith.replace(/[\(\`\)\\]/g, '')
    will.innerHTML = smith
</script>
```

The student has to find the vulnerability (XSS attack) and write in a txt file the payload to pop-up an alert in the website page.

Exercise #3: Area 51

In this task you will learn about XSS by doing you first real challenge. The quest is the same as the previous exercise: you need to analyze the given code in Area51.html and find a way to pop-up an alert with an xss attack.
The file .html contains only the necessary lines of code to do the exercise correctly. To make it works well inside a browser you can wrap the lines in a basic html page.

Here's a brief of the vulnerable code:

```
<!-- Challenge -->
<div id="pwnme"></div>


<script>
    var input = (new URL(location).searchParams.get('debug') || '').replace(/[\!\-\/\#\&\;\%]/g, '_');
    var template = document.createElement('template');
    template.innerHTML = input;
    pwnme.innerHTML = "<!-- <p> DEBUG: " + template.outerHTML + " </p> -->";
</script>
```

The student has to find the vulnerability (XSS attack) and write in a txt file the payload to pop-up an alert in the website page.

SQL injection

Exercise #1: User info

In this lab we will use the metasploitable as victim machine and the Kali Linux as attacker machine. The metasploitable comes with several vulnerable services already included, in this lab the student need to attack (with SQL injection) a web site called Mutillidae placed in the metasploitable.
First you need to place the victim (metasploitable) and the attacker (Kali Linux) in the same virtual network as seen in the lesson.
After settings up the virtual network between the hosts you simply need to start the metasploitable, when it boots up, the vulnerable website starts automatically. For visiting the Mutillidae from the Kali Linux you need to type in the URL bar of the Kali browser the IP address of the metasploitable (*ifconfig* on you metasploitable).

The student has to bypass the security with an SQL injection on the Mutillidae website, in the section: OSWAP Top 10 > A1 - Injection > SQLi - Extract Data > User Info and extract some information about users. Write the payload in a file called sql1.txt

The student also need to suggest a possible way to mitigate the SQL injection contained in the exercise.

Exercise #2: Information leak (security medium)

In this exercise we use another vulnerable website on the metasploitable: DVWA. For visiting the DVWA from the Kali Linux you need to type in the URL bar of the Kali browser the IP address of the metasploitable (*ifconfig* on you metasploitable), section SQL injection.

SQL injection is basically a technique through which attackers can execute their own malicious SQL statements generally referred as malicious payload. Through the malicious SQL statements, attackers can steal information from the victim database. **The objective of this task is to get information about database tables and hosting server, you need to write in a txt file sql2.txt the payload with the SQL injection.**

The student also need to suggest a possible way to mitigate the SQL injection contained in the exercise.

Exercise #3: Magic hash php (extra, not necessary for the score)

By following the hint given in the last lesson, can you find a way to bypass the login credential of the given website?
You cannot modify the website code but you can read it.

The student has to find the vulnerability and write in a txt file the payload to login in the website without knowing the right password.
Hint: check out some interesting stuff on the web about “php type juggling”.