IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

Reflected Cross Site Scripting

Arhaam Chandiwal, Vipul Wani, Anurup Mahagaonkar
K.J. Somaiya Institute of Engineering & Information Technology, Mumbai, Maharashtra, India
arhaam.c@somaiya.edu, vipul.wani@somaiya.edu, anurup.m@somaiya.edu

Abstract: *Cross-Site Scripting, also called as XSS, is a type of injection where malicious scripts are injected into trusted websites. When malicious code, usually in the form of browser side script, is injected using a web application to a different end user, an XSS attack is said to have taken place. Flaws which allow success to this attack are remarkably widespread and occur anywhere a web application handles the user input without validating or encoding it. A study carried out by Symantec states that more than 50% of the websites are vulnerable to the XSS attack. Security engineers of Microsoft coined the term "Cross-Site Scripting" in January of the year 2000. But even if it was coined in the year 2000, XSS vulnerabilities have been reported and exploited since the beginning of 1990's, whose prey have been all the (then) tech-giants such as Twitter, Myspace, Orkut, Facebook and YouTube. Hence the name "Cross-Site" Scripting. This attack could be combined with other attacks such as phishing attack to make it more lethal but it usually isn't necessary, since it is already extremely difficult to deal with from a user perspective because in many cases it looks very legitimate as it's leveraging attacks against our banks, our shopping websites and not some fake malicious website.*

Keywords: Cross-Site Scripting

I. INTRODUCTION

As we just read that XSS exploits occur when data goes into a web application via an untrusted source. The data included is residing in the dynamic content which is sent to the web browser without any validation. The malicious content is usually in the form of JavaScript code or it could even be an HTML snippet or flash media code. It is because of this, that the variety of attacks possible because Cross-Site Scripting is nearly endless..
There are broadly two types of XSS attacks:

- Non-persistent XSS attack
- Persistent XSS attack

1.1 Non-Persistent XSS Attack

When the injected script is reflected off the web server, such as in search result, error message or any other response that includes some or all of the input sent to the server as part of the request, it is called a non-persistent XSS attack (or reflective attack). Any website to be vulnerable for Cross-Site scripting attack, it must follow these two conditions:

- The website must allow script injection
- Two users must interact on the website

1.2 Persistent XSS Attack

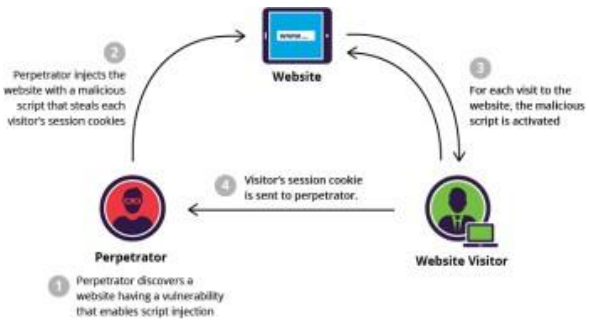

Fig 1.1: General idea behind XSS attacks

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

This attack, also called a stored attack, is those attacks where the injection script is stored permanently on the databases of the server through various means such as comment fields, logs, forums, etc. That injected script is then retrieved whenever the victim requests for the stored information.

Apart from these two attacks, there is a third type of XSS attack known as DOM based attack, which was first identified by Amit Klein in the year 2005. In this type of attack the DOM (Document Object Model) environment is modified to execute the payload.

Suppose a website asks their users to enter their preferred language with a default language also provided in the query string.

```
<scripts>
document.write("<OPTION value=1>" +
documents.locations.href.substrings
(documents.locations.href.index("default=")+8)
+ "</OPTION>");
document.write("<OPTION value=2> English
</OPTION>");
</scripts>
```

And consider that the webpage is invoked with the URL: https://example.site/page?default=Hebrew

A DOM based attack against this could be accomplished with something such as:

https://example.com/page?default=alert(document. co okie)

As we can see, all the types of attacks arise when the data which is input by the user is not verified. Therefore, this was the basis for me to find the websites vulnerable to XSS attacks. Also, while searching for such websites, I had to always keep in my mind that I don't break any of the laws such as the Internet Privacy Law or the IT Act 2000 or any other.

1.3 The Difference

Reflected	Stored
Contains non-persistant data, generally provided by the user through form submission.	Contains persistent data which is stored in the server's database.
Injected code reflects back to the attacker.	Injected code is fetched every time some user requests the page.
Example: Link generated by the attacker containing the injection for user to execute.	Using HTML tags in comments/forums so it (technically) becomes a structure of the main body.

Table 1.1: Difference between types of XSS attacks

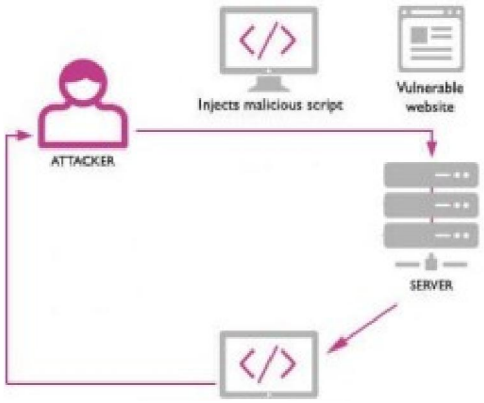

Figure 1.2: Reflected XSS attack

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

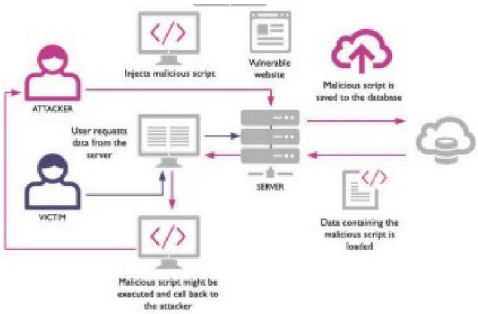

Figure 1.3: Stored XSS Attack

II. REFLECTED XSS VULNERABILITY

So, I found a website named Natas, which is a dictionary website that returns all the words containing the substring you enter in the text field. For example, if we enter the word "camp" in the text field, the following results will pop-up.
Upon looking at the source code I found a piece of code shown below

```
<pre>
<?
$val = "";
if(array_key_exists("needle", $_REQUEST)) {
    $val = $_REQUEST["needle"];
}
if($val != "") {
    passthru("grep -i $val dictionary.txt");
}
?>
</pre>
```

So, from the above source snippet we can see that the input we enter is passed to the grep command and no sort of validation is made into the input with the help of preg_match or regular expressions [12]. Therefore, it becomes an easy target for exploitation by which we can exploit the input data field and that's why this type of attack is called a Reflected attack or Type-II attack or Non-Persistent attack.

2.1 Exploitation

Always being within the limits of the law, we approach the exploitation taking learning to be our intention and not causing any breach of violations in the process. From the code snippet we can see the server on which the website is hosted is Linux, since grep command is used. Therefore, we can treat the input as a simple Linux terminal. Considering so, we can see use semicolon (;) and hash (#) to end the grep command and to comment the dictionary.txt ahead respectively and within these, we can use any other Linux command we want.

A. Directories

We can list all the directories, subdirectories and files using the ls (list) command as shown below.

Figure 2.2: Enumerating directories

B. Environment Variables

First of all, an environmental variable is a dynamic object which contains a modifiable value. They are usually used to store paths to numerous executable files, paths for storing temporary files, paths of the user profile and many other things.
We can list all the environment variables set for the server by using the env command as shown below.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

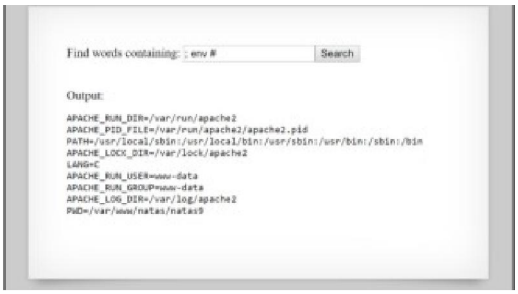

Figure 2.3: Enumerating Environment Variables

C. Open Files

We can even check for the files which are currently open on the server by all the users or the server itself, by which command the process was executed, what is the process ID, what is the file descriptor of the process, whether it's the rtd (root directory), cwd (current working directory), mmap (memory mapped devices), txt (txt file) or rtd (root directory) with the help of lsof command (list open files)

III. TOOLS FOR DETECTING XSS

Some of the tools used for detection of XSS vulnerabilities are discussed below.

S. No	Tool Name	Description
1	N- Stalker	It is a web scanning tool which is used for the detection of XSS vulnerabilities and other known attacks.
2	Acunetix	It scans the website to check whether the website is vulnerable to XSS attack or not.
3	Paros	Paros first crawls the entire website and then executes the canned vulnerability scanner test.
4	Hackbar	This tool helps the security analyzer to analyze the security holes faster and easily.
5	XSS ME	It is a Exploit-Me tool which is used for the detection of reflected type XSS.

Table 1.2: Tools for Detecting XSS

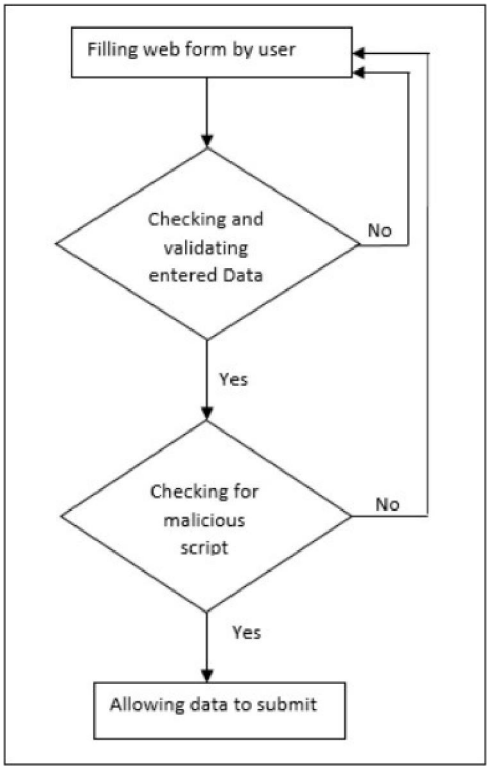

Figure 1.4: Approached system to prevent XSS flowchart

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

IV. APPROACHED SYSTEM TO PREVENT XSS

XSS is to be stated how to happen. it takes place when web forms receive malicious scripting code that has been injected to the victim computer then the web browser will execute. In this approached system, secure code PHP functions are proposed to detect and prevent form XSS attack by using two methods, the first one is to use regular expression to validate data from web forms that has been entered by the user, and the second one is another regular expression to check and protect every input entry that has a possibility to face a malicious script in it so even if the hacker inject XSS script code in the input field , this malicious code will not be allowed to be executed and immediately will be removed. in this work, vulnerable PHP web sites has used to assess the efficiency of the proposed system before and after applying it. For preventing XSS attack such htmlEntities() and htmlspecialchar(), PHP web programming language provides built-in functions that adapt characters to HTML entities, by using regular expressions that can be found it easier, also replace and work with string as shown in algorithm (1), the primary strategy is AllowList regular expression, which is by tolerating just expected and trusted user inputs data it will make a validation while DenyList regular expression includes checking if the information contains unsuitable data and evacuate all conceivable suspicious characters, for example, as HTML starting tags and ending tags <> with any text inside.

V. CONCLUSION

Cross-site scripting vulnerabilities are the most frequently encountered Web-based vulnerabilities today, and have been found on several major websites. These vulnerabilities manifest in Web-based applications whenever best practices, such as input validation, and Web output encoding are not implemented in code. To reduce exposure to these attacks, developers should implement a multi-layer defense strategy that includes coding best practices such as input validation, Web output encoding, and leveraging built-in platform protection. Microsoft has better enabled developers to do so through the guidance, process and tools of the Microsoft SDL.

REFERENCES

[1]. Mohit Dayal, Nanhay Singh, Ram Shringar Raw "A Comprehensive Inspection Of Cross Site Scripting Attack", IEEE, 2018

[2]. Syed Nisar Bukhari , Muneer Ahmad Dar, Ummer Iqbal "Reducing attack surface corresponding to Type 1 cross-site scripting attacks using secure development life cycle practices", IEEE, 2018

[3]. Twana Assad, Murat KARABATAK, "A proposed approach for preventing Cross-Site Scripting", IEEE, 2017

[4]. K.Pranathi, S.Kranthi, Dr.A.Srisaila, P.Madhavi Latha "Attacks on Web Application Caused by Cross Site Scripting ", IEEE, 2018

[5]. Mahmoud Mohammadi, Bill Chu, Heather Richter Lipford, "Automated Repair of Cross-Site Scripting Vulnerabilities through Unit Testing", IEEE, 2019

[6]. Jitendra Kumar, A. Santhana Vijayan, Balaji Rajendran, "Cross Site Scripting Attacks Classification using Convolutional Neural Network ", IEEE, 2022

[7]. Yu Sun , Dake He, "Model Checking for the Defense against Cross-site Scripting Attacks ", IEEE, 2021

[8]. Mehul Singh, Prabhishek Singh, Dr. Pramod Kumar, "An Analytical Study on Cross-Site Scripting ", IEEE, 2020

[9]. Blerton Abazi, Edmond Hajrizi, "Practical analysis on the algorithm of the Cross-Site Scripting Attacks ", IEEE, 2022

[10]. Tianma Wang, Dongmei Zhao, Jianjun Qi "Research on Cross-site Scripting Vulnerability of XSS Based on International Student Website ", IEEE, 2022