SECURITY OF WEB APPLICATIONS IN PHP

Tomas Slunsky
Doctoral Degree Programme (2), FEEC BUT
E-mail: xsluns01@stud.feec.vutbr.cz

Supervised by: Jiri Prinosil
E-mail: prinosil@feec.vutbr.cz

Abstract: This article deals with the security of web applications, focussing on vulnerabilities in web applications written in PHP language. This work reveals existing security issues, demonstrates the impact of them and propose solution with more approaches. The solution focuses mainly on the level of network filtering with Intrusion Detection System (IDS) or Intrusion Prevention Systems (IPS). There are more issue solution approaches and it will therefore be possible to propose the best one and describe it more.

Keywords: web, web application, vulnerabilities, security issues, HTTP, PHP, OWASP, exploit, IDS, IPS, server-side programming language

1 INTRODUCTION

Nowadays the importance of the Web and web applications are growing continuously. The main reason for that is the important fact - informations from the Web are easily accessible from anywhere in the world and it is practically the most used software for this reason at all. Other reasons for the growth were the simplicity how web applications can be deployed today. Other important fact is the entire communication process provided by the HTTP (Hypertext Transfer Protocol) protocol and others related to it. Due to these circumstances many e-commerce companies appeared and build entire business on this principle. Changes are also happening in the case of innovation, where companies are integrating web technologies, from industry to banking, etc.

With the increasing demands on web presentations, it was necessary to introduce new technologies, thanks to which it was possible to better interact with the user and offer more advanced services and functions. Since the beginning of the Internet, web presentations have moved forward enormously. But the requirements have increased too. With the introduction of new technologies, there are also associated security issues and vulnerabilities. Security emphasis is needed much more than ever.

Web application security has become an increasingly important topic. From this reason the Open Web Application Security Project (OWASP) was established to improve security. OWASP maintains the top 10 web application vulnerabilities.

The goal of this article is to describe the vulnerability of web applications in PHP and describe advanced techniques for dealing with these vulnerabilities. Intrusion Detection System (IDS) / Intrusion Prevention Systems (IPS) technology, its possibilities and a proposal for improvement will be discussed in more detail.

1.1 VULNERABILITY

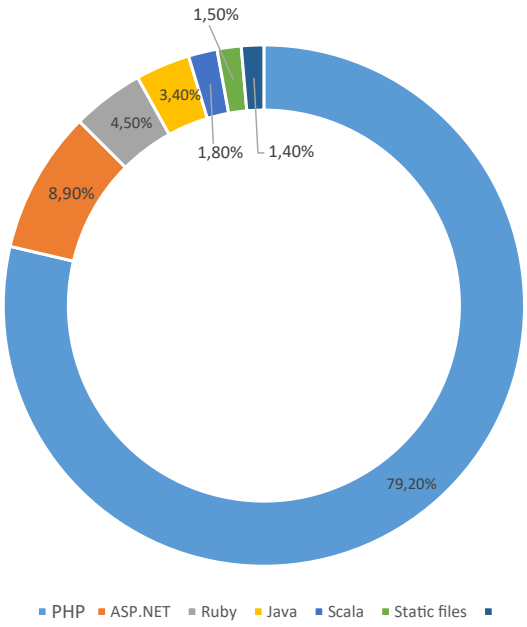
A website vulnerability in this article meaning is a misconfiguration or weakness in a website or web application code that allows an attacker to gain some level of control of the website, and possibly the whole server in the worst case.

1.2 PHP

PHP (Hypertext Preprocessor) is a language used to create dynamic web applications. There are many web frameworks written in PHP for web development. A web framework provides a structure and starting point for creating web application. There are many things solved yet. They offer ready-made tools for authentication, security and have built-in libraries and tools. The time required for development is less. Examples of PHP frameworks are Nette, Laravel, CodeIgniter and others.

2 ANALYSIS OF CURRENT WEB VULNERABILITIES

Nowadays PHP is powering more than 75% of the top ten million websites and is the most widely used server side programming language in Web applications [3]. On PHP is running huge projects as Wikipedia or Facebook for instance. The reason why PHP is a so popular general-purpose scripting language for web is because of his suitability for web development and is free, fast, flexible and also pragmatic. Server-side programming language usage [6] is shown in Fig 1.



Obrázek 1: Usage statistics of server-side programming languages for websites

2.1 OWASP TOP 10

According to OWASP [5] the most fundamental vulnerabilities in recent years are attacks [4] such as:

A1 - Injection An injection is a very widespread attack and can occur when untrusted data is sent to an interpreter as some part of a query or command. [4]. The attack can affect various systems such as SQL (Structured Query Language), NoSQL (non-SQL), LDAP (Lightweight Directory Access Protocol) and can cause an interpreter to execute unwanted commands or access data without proper permission.

Example of vulnerability is shown bellow.

```
mysql_query("SELECT * FROM user WHERE id_user = " . $_GET['id']);
```

At this moment, attack can be done with GET request to endpoint.

```
http://example.com/ourscript.php?id=25~ORDER~BY~2
```

A2 - Broken Authentication Web application features related to user authentication may be implemented incorrectly. This can lead to an attacker being able to steal a user's identity. By exploiting this vulnerability, attacker is able to perform all of user's operations. The user's identity can be stolen temporarily or even permanently.

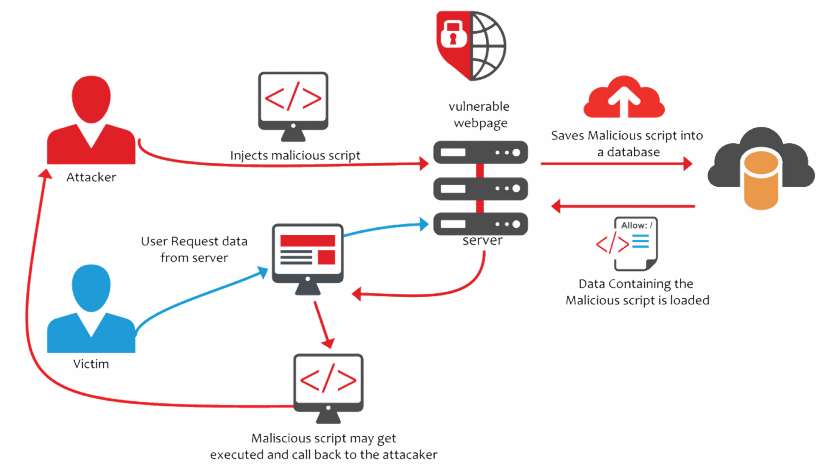
A3 - Sensitive Data Exposure This is one of the most critical security threats. It occurs in case a web application does not sufficiently protect sensitive information from users who aren't allowed to access to data. These are implementation errors that attackers use to gain access to sensitive data stored by applications.

A4 - XML External Entities (XXE) This vulnerability can allow an attacker to view files on the application server filesystem, remote code execution or internal port scanning via XML External Entities. Vulnerability is caused by poorly configured XML (Extensible Markup Language) processors. The problem is little known yet a many of today's security tests do not take it into account. At the same time, the impact of its misuse can be very serious.

A5 - Broken Access Control Authenticated users are allowed to do everything without any restrictions because of they are often not properly enforced and attackers can exploit these flaws to access unauthorized functionality and/or data.

A6 - Security Misconfiguration It's the most commonly seen issue caused by insecure default, ad hoc or incomplete configurations. Another important thing is also to fix the error using the patches and keep the system up to date.

A7 - Cross-Site Scripting (XSS) A security vulnerability occurs when a web application inserts untrusted data into a client browser's output without proper validation or escaping. For instance, it can be done by saving untrusted data into web application database via some of the input forms. By exploiting this weakness, the attacker is able to steal the user's session, modify it page content, redirect users and use the user's browser to run malicious code. XSS attacks are a serious danger to web applications and can have major consequences in the case of a successful attack. XSS attack is demonstrated in Fig 2.



Obrázek 2: Demonstration of XSS attack. [7]

A8 - Insecure Deserialization User-controllable untrusted data is deserialized by a website. This situation potentially enables an attacker to modify serialized objects in order to pass malicious data into the code of application.

A9 - Using Components with Known Vulnerabilities Most of today's frameworks are using different third party libraries. If one of these libraries contains a vulnerability and that one is exploited, it can lead to an application crash, etc.

A10 - Insufficient Logging & Monitoring The consequence of this vulnerability is there is no immediate response for bug and potential attackers can exploit this vulnerability for long time until it is typically detected by a third party [2].

3 NEW PHP NETTE VULNERABILITY

3.1 NETTE CVE-2020–15227

A few months ago, at the end of 2020, there was a big vulnerability discovered in the Nette framework. Nette is a full featured component based framework developed by David Grudl and czech PHP community. Nette is powering many websites in the Czech Republic and abroad and runs large websites, e-shops and projects.

The issue that is occurred in the framework was the variant of injection. In certain circumstances it was possible remote code execution via this fatal security issue [1], in other words, there was a code injection belonging to the A1 OWASP category.

The bug affected all versions of nette from version 2.0.19 to version 3.0.6 practically and the attack was carried out using a specially compiled URL (Uniform Resource Locator), which was able to execute PHP code then.

It is not known yet how long this bug could have been exploited by the attackers and whether any damage has occurred so far.

4 SOLUTION WITH IPS/IDS DETECTION

The IPS system can be divided according detection method. Signatures detection method uses its own database of marks, which are strings specific to a given type of attack. An IPS system that uses stateful tag detection monitors network traffic for compliance with these types of attack-specific tags. Once a match is found, the IPS system takes the appropriate action.

Based on the previously described attacks and due to the possibilities of how attacks are performed, it is advisable to choose an IPS system for the protection of the web application, which will use signature detection. For better attacks filtering, the signature database is updated using machine learning. Machine learning deals with algorithms that allow to change internal state. Thanks to machine learning, it is possible to adapt to changes in the surrounding network environment. It can better respond to new attacks and thus increase the security of web applications. The basic task types of machine learning can be summarized in the following three points:

- cluster analysis of data with similar properties
- classification of input data into classes
- estimation of the numerical value of the output according to the input.

5 CONCLUSION

This article outlines current vulnerabilities in web frameworks. Furthermore a specific case of vulnerability in Nette framework was described. Article also discusses the possibilities of defending these vulnerabilities using IPS/IDS and suggests effective ways to mitigate the effects of the security issues.

REFERENCE

[1] CVE-2020–15227: Potential Remote Code Execution Vulnerability. In: Nette – Comfortable and Safe Web Development in PHP [online]. nette.org: nette.org, 2020, 2020 [cit. 2021-02-14]. Available from: https://blog.nette.org/en/cve-2020-15227-potential-remote-code-execution-vulnerability

[2] Insecure deserialization. In: Web Application Security, Testing, & Scanning - PortSwigger [online]. portswigger.net: PortSwigger, 2020, 2020 [cit. 2021-02-14]. Available from: https://portswigger.net/web-security/deserialization

[3] M. Backes, K. Rieck, M. Skoruppa, B. Stock and F. Yamaguchi, "Efficient and Flexible Discovery of PHP Application Vulnerabilities,"2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, France, 2017, pp. 334-349, doi: 10.1109/EuroSP.2017.14.

[4] OWASP Top Ten. In: OWASP Foundation | Open Source Foundation for Application Security [online]. owasp.org: owasp.org, 2020 [cit. 2021-02-14]. Available from: https://owasp.org/www-project-top-ten/

[5] OWASP. Open Source Foundation for Application Security [online]. Owasp.org: OWASP Foundation, 2021 [cit. 2021-03-24]. Available from: https://owasp.org/

[6] Usage statistics of server-side programming languages for websites. In: W3Techs [online]. online: W3Techs, 2021 [cit. 2021-02-13]. Available from: https://w3techs.com/technologies/overview/programming_language

[7] XSS. In: Defender's Notes [online]. notes.defendergb.org: Defender's Notes, 2020, 2020 [cit. 2021-02-14]. Available from: https://notes.defendergb.org/web-sec/vuln/xss