A FMEA Analysis on Web Applications

Gabriel PETRICĂ[1], Costel CIUCHI[2]
[1] EUROQUALROM, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania
gabriel.petrica@upb.ro
[2] Associate Professor, University POLITEHNICA of Bucharest, Romania
costel.ciuchi@upb.ro

Abstract

Based on the Failure Mode and Effects Analysis (FMEA) method, this paper identifies the potential causes that lead to the failure of a Web application built on the WordPress platform. Both software vulnerabilities identified in the U.S. National Vulnerability Database (NVD) and other platform administration and configuration processes that can be exploited in cyber-attacks against the Web application are considered. Finally, measures to eliminate potential security breaches are proposed in the form of a best practice guide for managing sensitive data and increasing the level of security for this type of application.

Index terms: cybersecurity, FMEA analysis, software vulnerabilities, WordPress

1. Introduction. Content Management Systems (CMS)

Currently and conventionally, the term "*content*" is used to refer to information in the category of text (documents in various formats), audio, video, binary files or any other type of media, transmitted electronically through traditional systems or via the Internet. In the latter case, specifications for media types - Multipurpose Internet Mail Extensions (MIME) - are managed by the Internet Assigned Numbers Authority (IANA), the official authority for standardizing and publishing these specifications. The content can be produced, modified, transmitted, consumed, or traded in parts or in its entirety, being available on demand, possibly under certain conditions, and accessible permanently or during certain periods. In the context of the media industry, a team representing the Society of Motion Picture and Television Engineers (SMPTE) and the European Broadcasting Union (EBU) defined the term "*content*" in 1998 and identified its two components: *the essence* and *metadata* [1] (Figure 1).

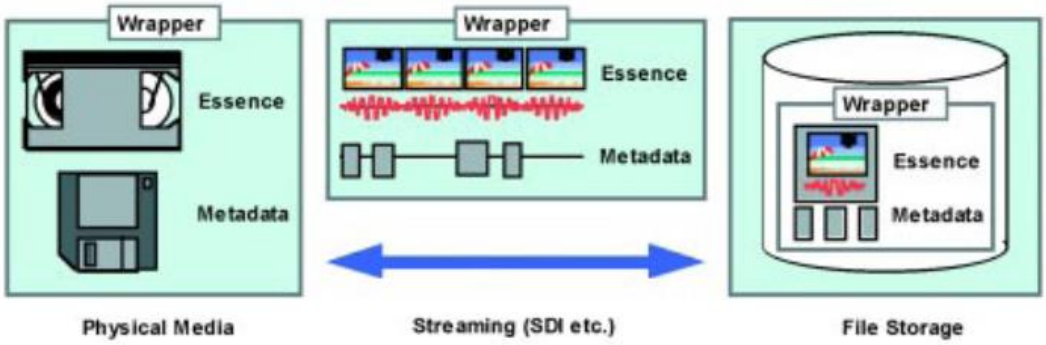

Fig. 1. The "content" elements: essence and metadata [1]

The *essence* refers to the raw material of the program itself - represented by text, images, sounds, video and others. The *metadata* is the part that characterizes the essence and other attributes of the content. Metadata describes the actual content or subject matter, material (available formats, encoding parameters, and specific recording information) or location.

A system that deals with content and metadata management is called a *Content Management System* (CMS). It allows an organization to manage information in real time, provides up-to-date content and respond to changing consumer demands. CMS provides automated control of information collection, management and distribution [2]. The need for a Content Management System within an organization is supported by the following factors:

- large amount of content.
- multiple access.
- finding information on different sites of the same organization, in a format adapted to the type of channel.
- varied content, which changes in a dynamic way.
- personalized content, reflecting the way each organization interacts with its consumers.
- multiple authors, contributors, and publishers.
- systems for recording workflows and managing tasks between teams.
- the need for flexibility in entering and processing data.

There are many CMSs and the most popular of them is WordPress (according to W3Techs statistics from April 2023, 63.3% of monitored websites used the WordPress platform, which represented 43% of all websites [3] - Figure 2).

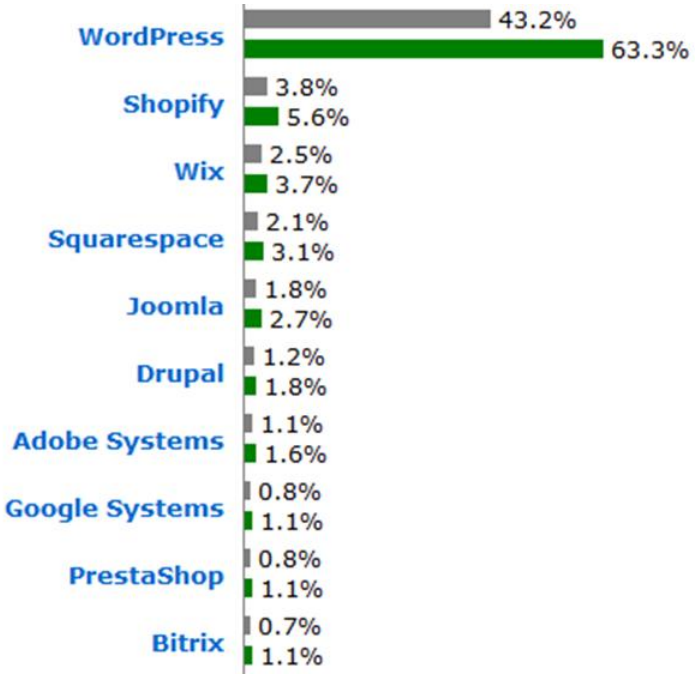

Fig. 2. Content Management System usage statistics - April 2023 [3]

Launched in 2003, WordPress is the most widely used open-source CMS worldwide, with approximately 18 million installations. Started as a blogging system, WordPress has now become a fully functional Content Management System [4]. WordPress has many free and varied templates, easy installation using a wizard, easy-to-search URLs, and management and publishing tools for mobile solutions. As the most popular CMS, WordPress is the most common target of attackers. However, WordPress is built on a very secure code and responds quickly to security vulnerabilities. It also has an auto-update mechanism that allows the system to automatically update when there are new versions.

2. Analysis of failure modes for a Web application based on the WordPress platform

Failure Modes and Effects Analysis (FMEA), an effective method in systems reliability, maintainability, security, and testability studies, involves the exhaustive enumeration of possible failure modes for all system components and highlights the effects of these failures at the component or (sub)system level [5]. FMEA is a tool that helps deliver products or processes that are reliable, acceptable to the customer and, above all, safe to use. Because FMEA helps the designer identify potential critical product/process defects, it is used to:

- develop product or process requirements that minimize the probability of failures.
- evaluate the requirements obtained from the clients or other participants in the design process to ensure that these requirements do not introduce potential defects.
- identify those design features that contribute to the occurrence of critical defects and minimize the resulting effects.
- designs methods and procedures to develop and test the product / process so that there is certainty that defects have been successfully eliminated.
- track and manage potential risks in design.
- ensure that any defects that may occur will not affect or will not have a serious impact on the user of that product / process [5].

In the next chapters we developed a FMEA analysis by identifying and describing the main components of WordPress (Figure 3): *core* (the main code), *theme* (which represents a collection of files that change the appearance of the website, and the way information is presented, keeping its content unchanged), and *plugins* (PHP code sequences that extend the default functionalities of the platform).

Fig. 3. The software components of a WordPress platform

2.1. Exploitation of the software vulnerabilities within the core of WordPress

Considering recent statistics (April 2023) showing that the most used versions of WordPress are currently v.6 and v.5, with over 90% of installations [6] (Figure 4), we can consider vulnerabilities in these versions as the most likely to be exploited by the current potentially cyber-attacks. Released in December 2018, WordPress 5.0 “Bebo” introduced a new editor based on blocks (abstract elements that make up the layout of a page) and a new default theme called “Twenty Nineteen” [7]. The new WordPress version 6, called “Arturo”, was released to the public in May 2022 and enhanced several aspects like performance, accessibility, or design tools and added a better writing experience [8].

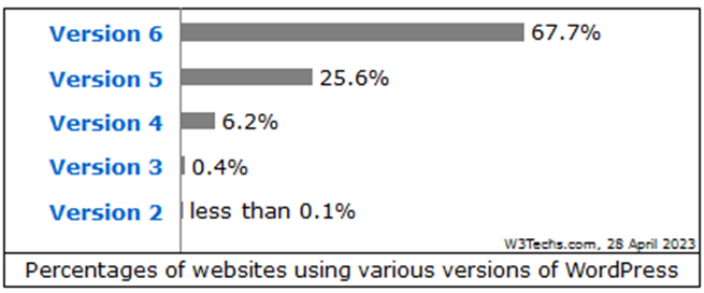

Fig. 4. WordPress versions usage statistics [6]

The existence of different ways of identifying and scoring vulnerabilities and the lack of interoperability between databases and tools that referred to the same vulnerabilities led to the emergence of the Common Vulnerabilities and Exposures (CVE) project in 1999. Designed and coordinated by the MITRE Corporation, CVE is a tool for monitoring and standardizing the most known vulnerabilities, which ensures trust between parties when used to discuss or share information about a unique vulnerability of an application, operating system, service, or firmware [9]. Each known vulnerability is uniquely identified in the National Vulnerability Database (NVD) and is also available in the CVE list, with detailed descriptions for each vulnerability, as well as a system - the Common Vulnerability Scoring System (CVSS) - that quantifies (on a scale from 0 to 10 - maximum severity) the impact of that vulnerability [10]. Software vulnerabilities reported for the WordPress platform from 2019 to 2022 are summarized in Table 1.

Table 1. WordPress vulnerabilities reported in 2019 - 2022 [11]

Year	# of Vulnerabilities	DoS	Code Execution	SQL Injection	XSS	Directory Traversal	Bypass something	Gain Information	Gain Privileges	CSRF
2019	23		4		12	1	2	2		2
2020	21	1	2		7				2	1
2021	8		1		2		2	2		
2022	9			2	3		1			

The 61 vulnerabilities of WordPress version 5.x and 6.x (from 2019 to 2022) have the CVSS score distribution shown in Table 2, having an average score of 5.3 [12].

Table 2. CVSS vulnerability score for WordPress v5.x and 6.x (2019 - 2022) [12]

CVSS score	Number of vulnerabilities	Percent
0-1	4	6.60%
1-2		0
2-3		0
3-4	13	21.30%
4-5	18	29.50%
5-6	11	18%
6-7	6	9.80%
7-8	9	14.80%
8-9		0
9-10		0

2.2. Exploitation of the vulnerabilities in themes and plugins

The two components of a WordPress platform, *themes* and *plugins* implemented by third parties, bring different ways of displaying information, respectively implement new functionality starting from the core code. However, these components are also the most vulnerable, being most of the time exploitable resources by cyber attackers. They typically use automated scripts to scan the Internet for websites that contain known software vulnerabilities. When a target is identified, malware code is executed or can be injected to gain unauthorized access to the compromised environment. The attacker then deploys software tools, depending on available resources, to launch new attacks on other targets.

According to SUCURI statistics from 2022, the ten most frequent vulnerable software components are indicated in Table 3. It is noted that “36% of all compromised websites had at least

one vulnerable component present in the environment at the point of remediation" [13]. However, the data does not necessarily indicate that these plugins were attack vectors, but instead contributed to an overall insecure environment.

Table 3. Top software with vulnerabilities in 2022 [13]

Software Component	Percent
Contact-Form-7	27.44%
Fremius Library	20.85%
WooCommerce	14.51%
UpdraftPlus Free	5.35%
Gutenberg Temp. Library & Redux Framework	3.83%
Advanced Custom Fields	3.23%
WP Fastest Cache	3.21%
Essential Addons for Elementor	3.04%
PageBuilder by SiteOrigin	2.22%
File Manager	1.89%

The tools used by SUCURI in responding to various security incidents detected signatures of the malware application categories shown in Figure 5.

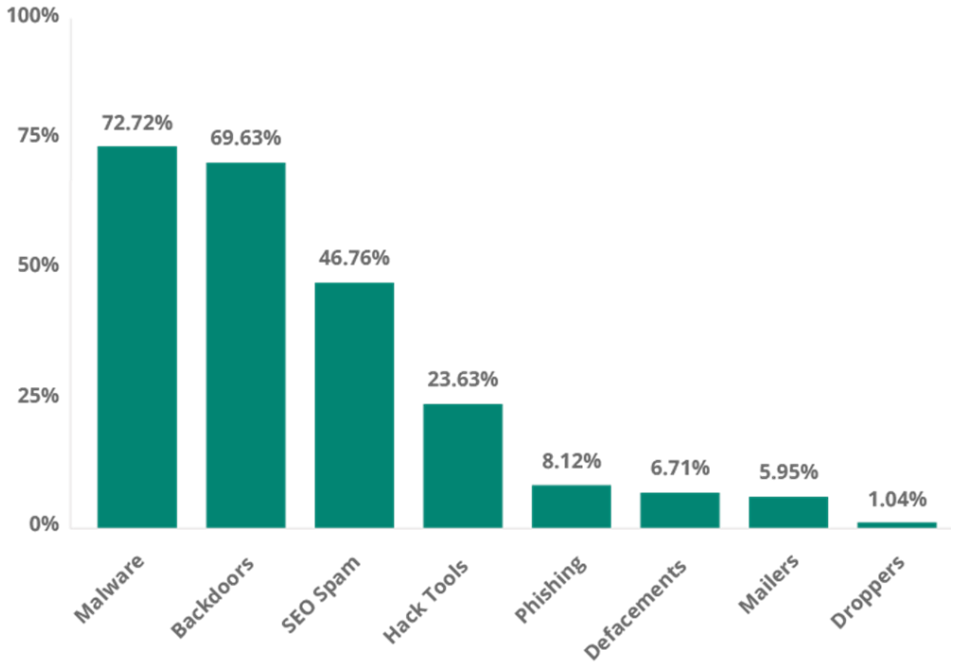

Fig. 5. Distribution of malware applications in 2022 (according to SUCURI) [13]

2.3. Exploitation of PHP or DBMS vulnerabilities

Another sensitive subject is represented by the versions of the PHP language installed on Web servers. Although SUCURI statistics indicate that over 65% of the analyzed websites used in 2021 PHP v7.x or a higher version (Figure 6), there remains a considerable number of those still using outdated versions of the 5.x series [14]. Among the motivations behind this situation, we can mention:

- the added code, themes and plugins used are incompatible with new PHP versions.
- some codes require partial / total rewriting, involving additional time and funds.
- many websites depend on the hosting company and the owners may have limited or no control over the PHP version.
- some owners simply neglect or do not want to update.

PHP versions that have reached EOL (End-of-Life) no longer receive regular security updates, their use being a vulnerability often exploited by attackers.

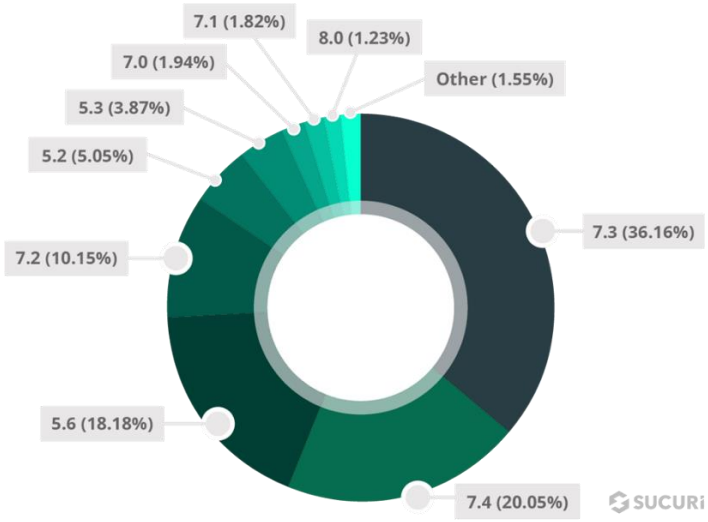

Fig. 6. PHP versions used in 2021 (according to SUCURI) [14]

Regarding Database Management Systems (DBMS), according to Statista, the most popular systems worldwide in February 2023 were Oracle, MySQL and Microsoft SQL Server (Figure 7) [15]. However, it has been found that DBMS systems are generally not delivered as a security-safe package, leaving administrators the task of configuring them optimally from this point of view. Typical vulnerabilities may refer to:

- vulnerable credentials (empty passwords, weak username / password combinations).
- activating some functions that are not necessary.
- insecure configurations, enabled for the convenience of administrators or developers.
- sensitive data stored or transferred in clear text format (not encrypted).

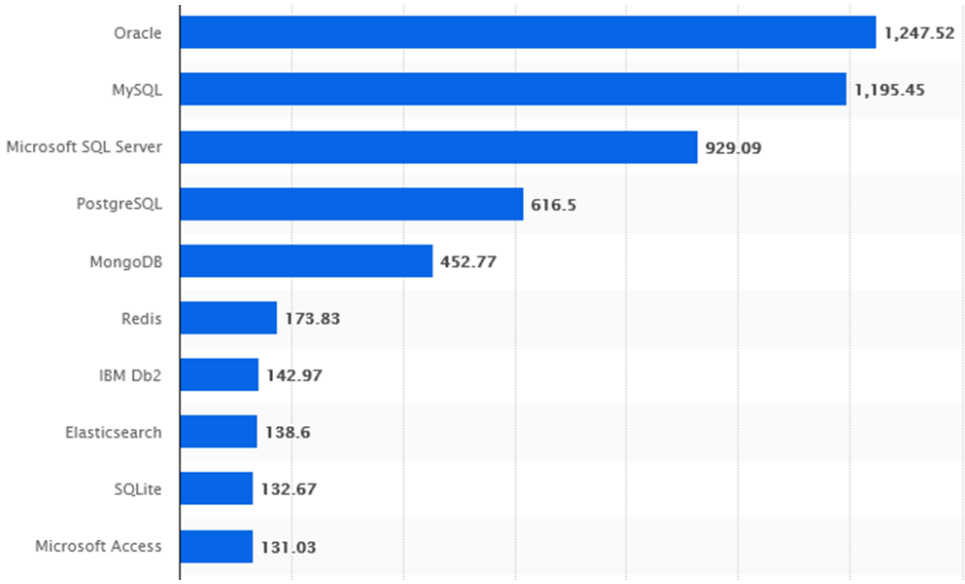

Fig. 7. Top 10 DBMS in February 2023, according to Statista [15]

2.4. Performing the FMEA analysis

Starting from the typical stages of a FMEA analysis and based on the aspects presented in previous chapters, we developed a FMEA analysis that addresses several types of processes (classes of actions) associated with a WordPress platform. In Table 4 the potential causes for each type of process are identified and the modes of manifestation / failure are described. The proposal of solutions that mitigate the impact and the attack surface will be made in chapter 3 as a good practice guide.

Table 4. FMEA analysis for a WordPress platform

Process	Potential causes	Effect description / Failure mode
Exploitation of the core components		
Exploitation of software vulnerabilities specific to the WordPress platform	Vulnerabilities in code (core)	- 61 vulnerabilities identified in the most used current versions (WordPress v.5.x and v.6.x) - the exploitation of these vulnerabilities (Table 1) allows the launch of some types of cyber-attacks with various unwanted effects on the platform
Exploitation of third-party software vulnerabilities (plugins / themes)	Use of vulnerable plugins/ themes	- Stealth code - BASE64 encoding function calls - Functions for displaying some information - e.g. phpinfo() - Running system functions (fopen, chown, chmod, exec...)
Exploiting specific vulnerabilities of the PHP language version	Using an outdated version of the PHP language (e.g. v.5.x)	- Execution of malicious PHP code - Exploitation of known PHP vulnerabilities
Exploitation of specific vulnerabilities in the database management systems	Using an outdated / insecure version of DBMS	- Corruption or loss of data - Loss of right of access - Extraction of sensitive data - passwords, Personally Identifiable Information (PII) - Disclosure of data to unauthorized parties - Interruption of the provision of some services
Other exploits		
Brute-force attacks	There are no mechanisms to limit brute-force attacks	- Valid username / password pairs are obtained (guessed) - Decreases the processing power of the server running the application
Access to sensitive files	Unsecured special file locations	Access to configuration files, installation scripts, or documentation files is allowed
Admin account exploit	The admin account (“admin”) is the default account	Obtaining the admin account password by brute-force attacks on a supposedly existing account (“admin”)
Table prefix	Default Table prefix (“wp_”)	The “wp_” default table prefix is an advantage for an attacker
Password exploitation	Weak passwords - insufficient password length	Illegal application access
	Weak passwords - insufficient password complexity	- Illegitimate access to user / application data - Leaks of personal data and confidential information
Password storage	- Unencrypted password in source code - Unencrypted password in properties / configuration files	Illegally obtaining admin password
	Unencrypted passwords in database	Illegally obtaining user passwords
Password management	- Remember password option is checked - Save password in the browser for subsequent logins	Illegal access to the application if run on a public computer
	Display error messages on login	Disclosure of incorrect elements (username or password) on an incorrect login

3. Proposed corrective actions. Good practice guide

Next, for each of the categories of processes identified in Table 5, corrective actions are proposed, which will lead to the reduction of the effect or the elimination of potential causes. Thus, Table 5 presents recommended actions for increasing the level of cyber security of a Web application based on the WordPress platform.

Table 5. FMEA analysis - proposed solutions

Process	Recommended corrective actions
Exploitation of the core components of the WordPress platform	
Exploitation of WordPress platform-specific software vulnerabilities	Update platform (core)
Exploitation of third-party software vulnerabilities (plugins / themes)	- Update plugins / themes - Deleting unused plugins / themes - Installing trusted plugins from safe sources - View user comments and ratings before installation
Exploiting version-specific vulnerabilities of the PHP language	Using the latest versions of the PHP language
Exploitation of specific vulnerabilities of the database management system	Regular application of patches
Other exploits of the WordPress platform	
Password storage and management	- Controls for a minimum password length - Warnings for weak passwords - Random password generator - Controls for a complex password - Reject typical content of passwords - Password strength indicator - Encryption of stored passwords - Re-authentication request before changing sensitive settings - Transmission of passwords using secure protocols (HTTPS) - Disconnecting inactive (idle) accounts - Mechanism for resetting the password if it has been forgotten - Imposing a periodic change of the password
Brute-force attacks	- Full validation of inputs - Limiting responses to queries (useful in case of a cyber- attack) - Limitation of login attempts for a user / IP address - CAPTCHA mechanisms - Deactivation of the XML-RPC protocol - Use of WAF (Web Application Firewall) - Implementation of Two-Factor Authentication (2FA) - Showing minimal information if login fail (for example: “Incorrect credentials”) - Remove WordPress or PHP version info - phpinfo() - Disabling or removing verbose debugging or error messages
Access to sensitive files	Rules in .htaccess (file used by the Apache Web server)
Admin account exploit	Creating an admin account with an unpredictable name, followed by deleting the default admin account (“admin”)
Prefix for tables	Changing the default prefix (“wp_”), at installation or later (manually or via a plugin)
Admin interface access	Restrict access to the CMS admin interface from approved or internal IP addresses

Other measures consist of “scanning the application to discover any vulnerabilities and fixing them as quickly as possible, but also applying security procedures in the development and maintenance life cycle of Web applications” [16]. For this purpose, dedicated tools can be used to scan for security vulnerabilities at the setup of WordPress (for example, WPScan) or later perform vulnerability assessments of code or plugin modules with third-party applications (such as RIPS, a static code analysis tool for automatically detecting security vulnerabilities in PHP applications).

A starting point in building secure, vulnerability-free web applications is the *OWASP Top 10*, an awareness document that has also been adopted as an industry standard to ensure cybersecurity. The Open Web Application Security Project (OWASP) community makes it easier for organizations

to develop, acquire, and maintain trusted applications and APIs (Application Programming Interfaces) by periodically publishing a Top 10 cybersecurity risks that developers need to be aware of. The latest OWASP statistics on security risks are displayed in Figure 8.

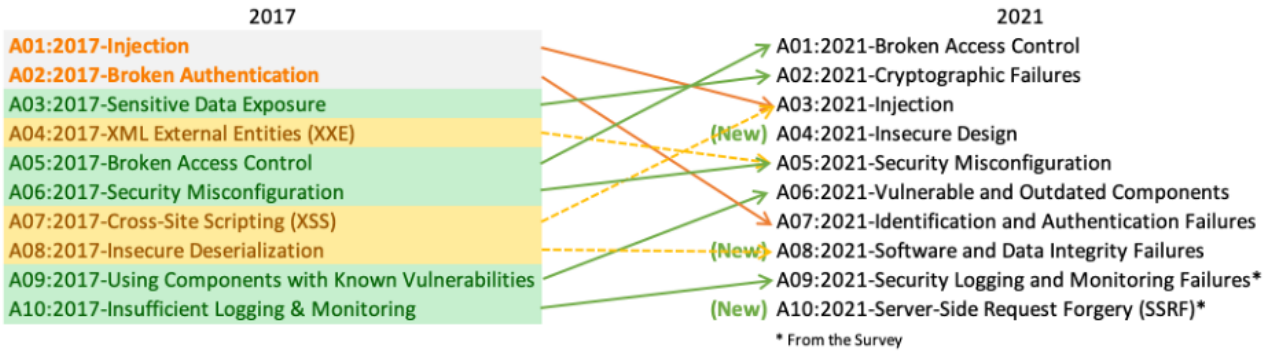

Fig. 8. The latest Top 10 Web Application Security Risks by OWASP [17]

4. Conclusions

Content management systems facilitate task management across teams and provide the context an organization needs to effectively create, update, and publish content. This paper has classified and identified the main causes (*failure modes*) that can cause a WordPress platform to stop working or that can compromise its security. The aspects and conclusions presented in the FMEA analysis are valid for any typical CMS structure, keeping in mind, however, that some elements (for example, vulnerabilities) depend on its type and version.

This analysis is a starting point in securing a Web application built on WordPress platform. Further developments of the study consist of extending the analysis to other software components of a complex Web application. Cyber-attacks and vulnerabilities can be related to the operating system, Web server or other applications components. Other situations that can disrupt the operation of a Web application may refer to the possibility of a physical attack in order to steal or destroy the equipment (access to the premises where the application runs or to the interconnection equipment).

Defending against various security threats is a continuous process that must involve awareness of the latest techniques and tools, correlated with the hardware and software environment in which the application operates, but also with user training, the weakest link in ensuring cyber security in an organization.

References

[1]. "Final Report of the EBU / SMPTE Task Force for Harmonized Standards for the Exchange of Television Programme Material as Bitstreams," 1998. Accessed: Mar. 15, 2023. [Online]. Available: https://tech.ebu.ch/docs/techreview/ebu-smpte-tf-bitstreams.pdf.

[2]. C. Benevolo, *Evaluation of Content Management Systems (CMS): a Supply Analysis*, 2017.

[3]. "Usage statistics of content management systems." W3Techs. https://w3techs.com/technologies/overview/content_management (accessed Apr. 5, 2023).

[4]. "CMS comparison 2022: The most popular content management systems." IONOS Digital Guide. https://www.ionos.com/digitalguide/hosting/cms/cms-comparison-a-review-of-the-best-platforms/ (accessed Apr. 2, 2023).

[5]. V.M. Cătuneanu and I.C. Bacivarov, *Fiabilitatea sistemelor de telecomunicații*, Ed. Militară, București, 1985.
[6]. "Usage statistics and market share of WordPress." W3Techs. https://w3techs.com/technologies/details/cm-wordpress (accessed Apr. 28, 2023).
[7]. "WordPress 5.0 Bebo." WordPress. https://wordpress.org/news/2018/12/bebo/ (accessed Mar. 20, 2023).
[8]. "WordPress 6.0 Arturo." WordPress. https://wordpress.org/news/2022/05/arturo/ (accessed Mar. 20, 2023).
[9]. "CVE - Common Vulnerabilities and Exposures." http://cve.mitre.org/ (accessed Mar. 10, 2023).
[10]. "NIST Common Vulnerability Scoring System Calculator Version 3." https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator (accessed Mar. 10, 2023).
[11]. "WordPress: Vulnerability Statistics." CVE Details. https://www.cvedetails.com/product/4096/Wordpress-Wordpress.html (accessed Mar. 10, 2023).
[12]. "CVSS scores for WordPress between 2019 and 2022." CVE Details. https://www.cvedetails.com/cvss-score-charts.php?fromform=1&vendor_id=&product_id=4096&startdate=2019-01-01&enddate=2022-12-31 (accessed Mar. 10, 2023).
[13]. "2022 Website Threat Research Report." SUCURI. https://sucuri.net/reports/2022-hacked-website-report/ (accessed Apr. 10, 2023).
[14]. "2021 Website Threat Research Report." SUCURI. https://sucuri.net/reports/2021-hacked-website-report/ (accessed Apr. 25, 2023).
[15]. "Ranking of the most popular database management systems worldwide, as of February 2023." Statista. https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/ (accessed Apr. 20, 2023).
[16]. C. Ciuchi, G. Petrică, a.o. *Cybersecurity Guide*. (2021). Accessed Apr. 10, 2023. [Online]. Available: https://dnsc.ro/vezi/document/ghid-securitate-cibernetica-2021.
[17]. "Top 10 Web Application Security Risks." OWASP. https://owasp.org/www-project-top-ten/ (accessed Apr. 1, 2023).