A Large-Scale Security Analysis of Web Vulnerability: Findings, Challenges and Remedies

Primož Cigoj[1,2(✉)], Živa Stepančič[2], and Borka Jerman Blažič[2]

[1] Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2] Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
{primoz,ziva,borka}@e5.ijs.si

Abstract. This paper presents an analysis of the results obtained with an efficient and powerful tool developed to recognize the exploitable vulnerabilities of websites on the internet implemented with a web-content management system (WCMS). The key feature of this ethical tool is a dynamic, automated, and fast vulnerability scan of WCMS sites and the attached plug-ins for the live internet by obeying ethical requests. The collected scan results are impressive, and the presented analysis of these results provides an insight into the internet web system's health and the factors that influence the vulnerability levels of websites.

Keywords: Cyberspace · Databases · Software · Security · Search engines · Search methods · Metasearch · Web search · Websites · WCMS

1 Introduction

Web-content management systems (WCMSs) like WordPress are very popular because they are free of charge and simplify the creation of a website and the upgrading of its functionality with plug-ins. Currently, there are more than 54,000 free WordPress plug-ins available in the official directory of plug-ins and the total number of downloads to various websites is estimated to exceed 1 billion. WordPress runs Sweden's official website, Microsoft's news.microsoft.com website, as well as online stores such as that of Bata, a shoe company. Systems like WordPress manage and store valuable information, from banking to electronic prescriptions on medical websites, as well as an enormous amount of personal data, all of which are of interest to cybercriminals and cyber-attackers. Even websites that do not hold valuable data or sensitive information are often attacked by hackers or competitors to cause a reduction in sales or to prevent websites from spreading their messages. For this reason, knowing and being aware of web vulnerability has an important value and is a critical task.

Supported by organization J. Stefan Institute.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12253, pp. 763–771, 2020.
https://doi.org/10.1007/978-3-030-58814-4_64

The need to identify web vulnerability is vital, as it influences the whole of internet security. The most common threats to the internet data space were identified in a 2017 security survey carried out by the UK Government as part of the National Cyber Security Programme. The findings are as follows: fraudulent e-mails (72%), viruses, spyware and malware (33%), attacks based on online impersonation (27%), and ransomware [1].

2 Overview of the Area and the Problem

New web technologies are followed by W3Techs. According to their survey, about 41.1% of websites do not use any WCMS. WordPress is used by 37.5% of all websites, which is a 63.5% market share of the content-management systems [2]. Another survey conducted in 2020 by BuiltWith found that 36.51% of the Alexa database and the top ranked in terms of their market presence contain 1 million websites that use WordPress [3].

Due to a lack of security, WordPress was targeted by an exceptionally large Distributed Denial of Service (DDoS) attack in 2011 that in many cases affected website connectivity [5]. Besides a DDoS attack, the most common attacks on the WordPress platform were identified as SQL Injection, Remote File Inclusion, Directory Traversal, Cross-Site Scripting, Comment Spamming, Remote Command Execution and File Upload. Given that WordPress and its plug-ins are public, anyone can look at the source code and can look for the possibility of vulnerabilities. This situation makes web-page development easier and faster, but at the same time allows hackers to exploit security holes and vulnerabilities in the code and thus misuse websites.

3 Previous Work

The WordPress WCMS is very popular among internet users, but web servers and the associated applications were found [6] to be more vulnerable than those that do not use any form of CMS. There are two major types of web-vulnerability scanners. The first type consists of automatic scanning tools that inspect the security features of the ports within the IPv4 site and share the results publicly (Shodan and Censys). The best-known tools from this group are Mmap, Zmap and Masscan [10–12]. The port information collected with such tools does not provide details of the web-page content and the vulnerability of its plug-ins, as they do not inspect the content of the web pages. The second type is composed of personal, interaction-based scanning tools operated by the user, and their results are returned directly and only to the user. The main examples of such solutions are Nessus, skipfish, Acunetix, IVRE, Vulners, and Vega. The findings are not shared publicly and to obtain the results the user operator must be contacted for a presentation of the results. Several similar search engines [6–9] were created in the past by different researchers, who intended to provide an analysis of vulnerable WCMS websites. The major drawback of these applications is the time needed to inspect the large number of sites. The average scanner of this

type needs a few days to explore a small data set, e.g., a group of 50,000 websites. So, success in identifying the vulnerability of a large internet web space depends on the speed of the crawling. Another drawback is the fact that they do not provide a complete image of a particular website, as some search features are not provided. An additional problem with some of the scanners is the approach used for inspecting the web servers, which is based on a simulated type of attack. In the case of success, the vulnerability is discovered. However, this is not a legal action. The use of this technique, known as the exploit technique, is forbidden in the case that the owner of the site does not allow the site to be inspected in this way. Another drawback of the existing scanners is that they access the websites using the IPv4 address and inspect only one website, thus neglecting the presence of many websites that are hosted on the same web server. The tool presented below overcomes these drawbacks and offers a more accurate vulnerability identification as well as a score for the insecurity level of the WCMS [4].

4 The VulNet Tool

The main objective that led to this tool's development was the intention to offer a solution that will improve web security by reducing the number of vulnerable websites with a fast, efficient and powerful method that can detect vulnerability and alert website owners immediately and worn them to improve their website's online hygiene. Another objective was to obtain a clearer picture of the web-security status within the internet space of the European registered domains. By identifying the status and the main characteristics of the vulnerable websites, the basic enablers that influence the level of insecurity in a particular European country can be identified and actions to improve the overall web security can be introduced. As there is no publicly available list of WordPress sites, it was necessary to review the entire web and to find out how many websites are powered by WordPress from the whole population of websites.

The designed methodology for data collection and scanning consists of five steps: 1) searching and collecting the URL queue of websites, where the initial list of websites is collected from the existing DNS domain zone files. A URL from the queue is visited and the links found there are added to the URL queue. The URLs that were not significant for our research were removed from the search (shortened URL providers, and large social sites), 2) accessing the website and asking for responses to the sent queries, 3) obtaining responses (metadata), 4) posting search queries for the application patterns, e.g., plug-ins, and collecting information from the CVE about the detected plug-ins, and 5) ad-hoc validating and storing of the results with the tool's in-built scoring mechanism. The scoring mechanism starts by identifying the vulnerability of the WCMS core in the predefined database identification of the vulnerabilities within the core WCMS pages and the plug-ins. A more detailed description of the search and scoring methodology, including a comparison of our search methodology with that of other researchers and tools, is presented in our published article [4].

5 Results and the Analysis

The scanning results provided by the VulNet tool are displayed with a web-based original interface developed with PHP. The interface displays the process of a live scanning and the obtained data. It also maps the path that the crawler has followed. VulNet provided information about the web space of 194 countries by establishing successfully 126,086,633 links. Among these websites there were 16,274,980 valid WordPress installations with 14,887,047 plug-in installations. In this web set, more than 5,018,262 were found to be vulnerable, representing 31% of all the WordPress installations on the internet. A total of 2,475,337 had a higher vulnerability, with a score of 5 or over. A total of 4,356,067 vulnerabilities were detected and among them 2,795,855 had an insecurity score that was higher than 5. The analysis of the results revealed that there are very vulnerable core versions of the WordPress website, especially the versions from series 4.0 (4.9.8, 4.9.3, 5.0.3, 4.9.6, 4.9.7, 4.9.4, 5.1, 4.9.5, 4.6.1).

Each plug-in or WP core version was assessed to be in one of three categories: secure, unknown, or insecure (with a score). Scores higher than 5 are considered critical and are reported separately (as a variable score). A website is considered secure if the core version and all (if any) plug-in versions are secure. As soon as one of the versions of the core and plug-in is unknown, the vulnerability status of the site is labelled as unknown. Otherwise, the site is considered as insecure. The main data set is composed (by domain) of the total WordPress sites (total WP), the number of insecure, secure, unknown sites, and insecure sites with a score of higher than 5 (variable score). The total number of plug-in versions, the total number of core versions, and their combinations were recorded as well.

A subset of the collected data was selected based on the fact that the parameters that can influence the increased appearance of insecure sites were available for these countries only. The following parameters were considered in the analysis: internet use (IU), frequency of internet use (FoIU), households with access to the internet (HH), and the digital skills (DS) index. The DS index was taken from an Information Society report from ITU [13] and the Eurostat portal [14]. Population-related data, such as the number of citizens (population), the gross net income (GNI), and the fixed cost of broadband access per GNI were analysed as well. These values were downloaded from the Eurostat portal [14].

The aim of the analysis was to explore the relationship between these indicators and the percentage of insecure websites found among the WP websites. Since we found that the internet-based indicators presented above correlate very highly among themselves, we decided to use only the DS index data, as it is considered to be the most indicative parameter in terms of web security for a general user. We hypothesized that the indicator of the DS index can show a statistically significant negative relationship with the degree of insecure sites, since better digital skills among the population implies better knowledge and an awareness regarding the use of cyber-security technology.

The final subset contained 31 different European countries: Germany (DE), Netherlands (NL), France (FR), Great Britain (GB), Italy (IT), Denmark (DK), Poland (PL), Spain (ES), Sweden (SE), Switzerland (CH), Czechia (CZ),

Ireland (IE), Finland (FI), Austria (AT), Romania (RO), Belgium (BE), Hungary (HU), Bulgaria (BG), Norway (NO), Slovakia (SK), Estonia (EE), Slovenia (SI), Portugal (PT), Croatia (HR), Lithuania (LV), Luxembourg (LU), Greece (GR), Iceland (IS), Latvia (LT), Cyprus (CY), and Malta (MT).

Summary statistics of the detected vulnerabilities are shown in Table 1. On average, we could not determine the vulnerability status for a third of the sites in a domain, due to unknown core or plug-in versions. The percentages of secure sites are between 21% and 40%, with an average of 28%. The percentage of insecure sites is in the range of 30% and 47%, with a mean of 38%, a median of 40%, and a standard deviation of 4.4%. There are no univariate outliers in either variable. In five cases, see Fig. 1, there is a higher percentage of secure sites than insecure: Germany, Denmark, the Netherlands, Switzerland and Norway. The first two are middle ranked in terms of digital skills, while the latter three show a very high DS index.

Table 1. Summary statistics of all the WP sites found, the vulnerability percentages, and the digital-skills (DS) index.

	Total WP	Unknown [%]	Secure [%]	Insecure [%]	Score [%]	DS
Count	31	31	31	31	31	31
Mean	120712.19	33.56	28	38.43	31.27	59.06
Std	182007.02	4.3	5.08	4.44	4.76	14.89
Min	10	20	21.22	30.31	22.13	29
25%	14715	31	24.28	35.5	27.27	49
50%	49345	33.57	26.12	39.65	32.24	57
75%	128049	36.86	31.4	41.59	34.9	71
Max	805361	40.15	40	46.96	40.83	85

The sites with very high insecurity scores are a subset of the all sites classified as insecure sites; therefore, the correlation between the variables score and insecure is very high ($r = .93$). There is also a high negative correlation ($r = -.6$) between the percentages of secure and insecure sites present in the whole set. For the validation of the hypothesis we used the percentage of insecure sites. The categorization of the DS index value was made in three groups, where we classified values below 50 as low skill, above 75 as high skill, and middle skill for the index values between 50 and 75. The descriptive statistics of the DS index are available in the last column of Table 1.

The dependencies among the variables are illustrated in Fig. 1. The countries with high digital skills, a high percentage of secure sites and a low percentage of insecure sites are grouped in the bottom-right corner of Fig. 1. In this group are the Scandinavian countries, Germany, the Netherlands, and Switzerland. In the group of countries with low digital skills and a high percentage of insecure sites are Lithuania, Poland, Bulgaria, Greece, Latvia, Ireland, Czechia, and Cyprus.

In the middle we can find Luxembourg, Belgium, Iceland, Great Britain, France, Austria, Estonia, and Italy. Malta is isolated, although not as a statistical outlier, but due to the small number of WP sites found.

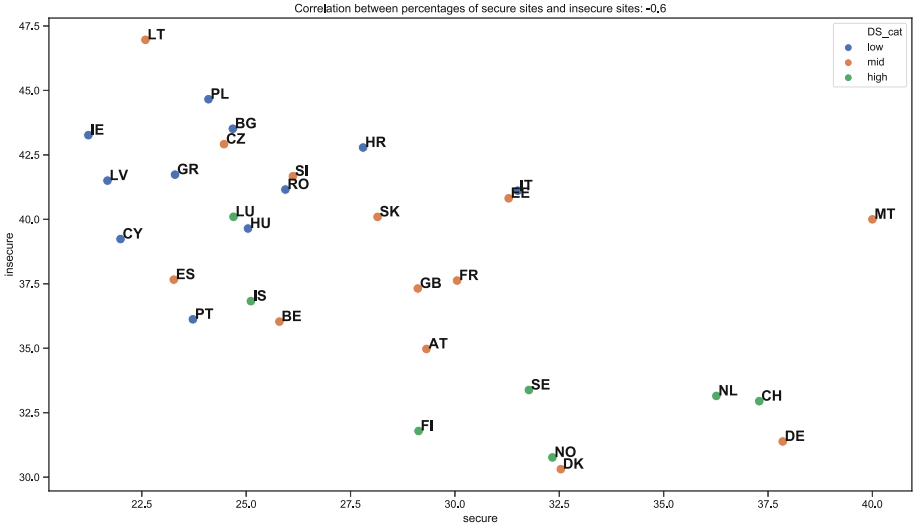

Fig. 1. Scatter plot of secure vs. insecure percentages per country with the digital-skill (DS) index categories.

A simple linear-regression analysis was made to predict the percentage of insecure WCMSs in the country based on the country's DS index. The data set was shown to be near-normally distributed and homoscedastic. A high correlation between the DS and the percentage of insecure sites (r = −.65) suggests a linear dependence. By modelling the regression on the whole data set, we obtained a regression equation as follows:

$$insecure = 50.41 - 0.20 * DS. \tag{1}$$

The first regression model with $R^2 = 0.462$ and RMSE = 3.202 was found to be significant, as $F(1, 29) = 24.94$ ($p < 0.001$) and both coefficients were significant at $\alpha = 0.001$ (the beta coefficient was −.68). TThe analysis of residuals, however, suggested improvements to the model, see Table 2. We used the Jarque-Bera (JB) test for normality [15], the Heteroscedasticity Test by Breusch-Pagan (LM-statistic) to test the homoscedasticity [16], the Harvey Collier Test for Linearity [17], and the Durbin-Watson (DW) test for first-order autocorrelation [18]. Improvements could be made upon JB and DW statistics. An analysis of the influential observations exposed three residual outliers (LT, DE, DK), three leverage points (IS, RO, BG), and one influential observation).

We removed the biggest outlier (LT), the influential observation (LU), and one leverage point (IS) from the data set to improve the model. These are also

the same observations found to be significant in the bivariate outlier analysis using the Mahalanobis distance. The new model has the following regression equation:

$$insecure = 51.77 - 0.26 * DS, \tag{2}$$

which was again found to be significant with $F(1, 26) = 45.50$ ($p < 0.001$) and both coefficients significant at $p < 0.001$ (the beta coefficient was -0.81). The R^2 has increased to 0.665 and the RMSE dropped to 2.47. The residual analysis supports all four assumptions of linear regression, as shown in Table 2.

Table 2. Regression summary and residual analysis results for both models.

Model	Regression results summary							Residual analysis						
	Regressors	Coefficient	SE	CI 0.025	CI 0.095	AIC	BIC	JB	Prob (JB)	LM-statistic	p-value (LM)	HC-statistic	p-value (HC)	DW
1	const	50.41	2.47	45.355	55.457	164.1	167	1.096	0.578	2.495	0.114	1.649	0.111	1.507
	DS	−0.2027	0.041	−0.286	−0.12									
2	const	52.77	2.094	48.466	57.074	134.1	136.8	0.304	0.859	0.0948	0.758	0.713	0.483	1.917
	DS	−0.2553	0.036	−0.328	−0.182									

The regression analysis confirmed the hypothesis that the level of DS has a positive impact on the provision of security within the websites with WCMS applications.

6 Discussion

We can conclude that the scanning with VulNet gave us a thorough internal security view of popular websites across the internet. However, the dependencies between the number of insecure sites and the other potential influential parameters were investigated only for some European countries, because reliable data is only available for these countries on the Eurostat portal.

The analysis showed that all the parameters per country, like internet use, the cost per fixed broadband access, or the percentage of households with internet access correlate very closely and for that reason only digital skills was selected as an independent variable in the statistical research. The hypothesis that the percentage of insecure sites is lower, when the DS of the population in a particular country are higher, was confirmed to be valid. This finding holds for almost all the studied European countries. The other finding that resulted from the study is related to the source of the vulnerabilities. It was found that the major contributors to the website's high insecurity score are from installed plug-ins. Figure 2 presents the comparison between the percentages of insecure plug-ins and the core versions of Word Press against the percentage of all the insecure sites. It is clear that the plug-ins are those software applications that mostly influence the overall site insecurity of the website.

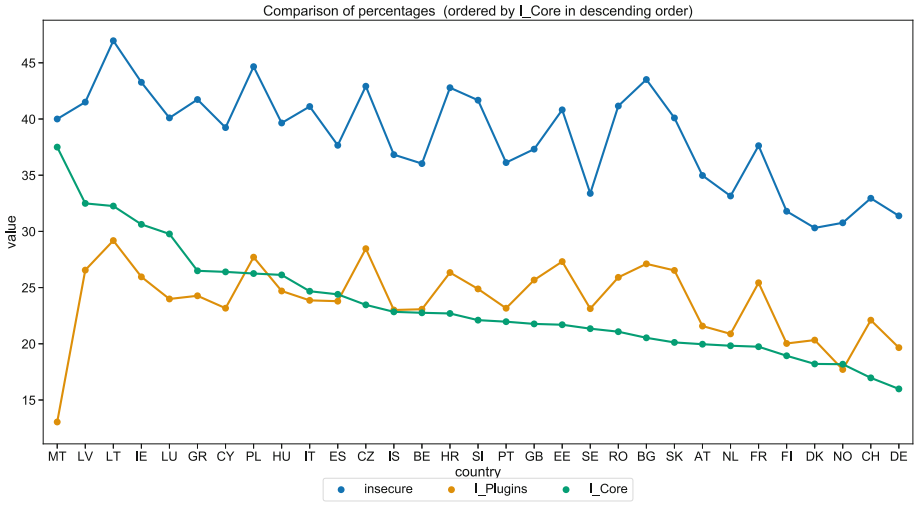

Fig. 2. Comparison of vulnerability percentages of core versions, plug-in versions, and overall insecure percentage.

7 Conclusions

The present paper provides a good insight into the security health of websites established with the Word Press core software for the whole internet. The major contributors to the insecurity of these sites are the used plug-ins. These are freely obtainable on the internet, with there being more than 54,000 available. This finding implies that the used plug-ins should be the major concern of the website owners when they decide to use them for improving the functionality and services offered by the website. However, an information service for providing awareness about the security health of the website for these types of website is still missing on the internet. Our investigation also found that there is a statistically significant relationship between a country's digital-skills index and the percentage of insecure web-content management systems: the higher the digital index of a country's population, the lower is the percentage of insecure sites in that country.

We can conclude that the security of websites is still a topic that needs further investigation and the provision of new remedies as the number of cyber-crime attacks are growing and the number of victims, although not known in detail, is growing as well.

References

1. Klahr, R., et al.: Cyber security breaches survey 2017: main report (2017)
2. W3Techs. https://w3techs.com. Accessed 5 May 2019
3. BuiltWith. https://builtwith.com. Accessed 5 May 2019

4. Cigoj, P., Blazic, B.J.: An intelligent and automated WCMS vulnerability-discovery tool: the current state of the web. IEEE Access **7**, 175466–175473 (2019)
5. Tsotsis, A.: WordPress.com suffers largest DDoS attack in its history (2011)
6. Vasek, M., Wadleigh, J., Moore, T.: Hacking is not random: a case-control study of webserver-compromise risk. IEEE Trans. Dependable Secure Comput. **13**, 206–219 (2015)
7. van Goethem, T., Chen, P., Nikiforakis, N., Desmet, L., Joosen, W.: Large-scale security analysis of the web: challenges and findings. In: Holz, T., Ioannidis, S. (eds.) Trust 2014. LNCS, vol. 8564, pp. 110–126. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08593-7_8
8. Stock, B., Pellegrino, G., Li, F., Backes, M., Rossow, C.: Didn't you hear me? – towards more successful web vulnerability notifications. In: Network and Distributed Systems Security (NDSS) Symposium (2018)
9. Schagen, N., Koning, K., Bos, H., Giuffrida, C.: Towards automated vulnerability scanning of network servers. In: Proceedings of the 11th European Workshop on Systems Security. ACM (2018)
10. Nappa, A., Rafique, Z.M., Caballero, J., Gu, G.: CyberProbe: towards internet-scale active detection of malicious servers. In: The Network and Distributed System Security Symposium (NDSS) (2014)
11. Kim, H., Kim, T., Jang, D.: An intelligent improvement of internet-wide scan engine for fast discovery of vulnerable IoT devices. Symmetry **10**, 151 (2018)
12. Li, F., et al.: You've got vulnerability: exploring effective vulnerability notifications. In: 25th USENIX Security Symposium (USENIX Security 16) (2016)
13. Digital Skills Insights 2019 - ITU. https://academy.itu.int/digital-skills-insights-2019. Accessed 15 Dec 2019
14. European Statistics - Eurostat. https://ec.europa.eu/eurostat. Accessed 15 Dec 2019
15. Jarque, C.M., Bera, A.K.: Efficient tests for normality, homoscedasticity and serial independence of regression residuals. Econ. Lett. **6**(3), 255–259 (1980)
16. Breusch, T., Pegan, A.: A simple test of heteroscedasticity and random coeffient variation. Econometrica **47**, 1287–1294 (1979)
17. Harvey, A., Collier, P.: Testing for functional misspecification in regression analysis. J. Econ. **6**, 103–119 (1977)
18. Durbin, J., Watson, G.S.: Testing for serial correlation in least squares regression III. Biometrika **58**(1), 1–19 (1971)