International Journal of Technology and Engineering Studies
volume 6 issue 1 pp. 1-7 doi: https://dx.doi.org/10.20469/ijtes.6.10001-1

Attacks on Newly Registered Content Management Websites - A Comparison

Marko Niinimaki *
Webster University Thailand,
Bangkok, Thailand

John Lawrence
Webster University Thailand,
Bangkok, Thailand

Kitichai Chanyalikit
Webster University Thailand,
Bangkok, Thailand

Veli Pajula
University Consortium of Seinajoki,
Seinajoki, Finland

Abstract: This paper aims to study the case of hacker/intrusion activities on Content Management System (CMS) websites. CMSs are tools for creating and maintaining commercial-quality websites. Their popularity has increased, but so has their complexity and the number of third-party modules. These, however, increase the risk of vulnerabilities. The current study investigates the amount of incoming traffic that could be potentially malicious and where it originates. Additionally, we study if CMS's based on different CMS software attract different kinds of traffic. Three virtual websites (running on the same computer) have been registered and launched to implement this study. Each site runs its own popular CMS software, but its content is identical (a weblog with a simple template). The sites run for six months on a platform of a commercial web hosting provider. This study is empirical in nature, and the analysis is based on logging every HTTP request that was sent to the sites. This was done using the logging capabilities of the web server software Apache. The sites were compared with each other, with an established website and an empty website. Our analysis shows that more than 90% of all traffic to the websites (both old and new) is potentially malicious. The results highlighted that a large majority of the intrusion attempts are very unsophisticated: they do not try to exploit any specific vulnerabilities of the underlying CMS. Therefore, keeping the CMS up-to-date and following CMS hardening practices is enough to repel these attacks.

Keywords: *Website security, website intrusion, computer security, web site CMS*

Received: 29 December 2020; ***Accepted:*** 21 January 2020; ***Published:*** 28 February 2020

I. INTRODUCTION

According to recent statistics, there are about 1.5 billion websites on the World Wide Web today, though less than 200 million are active [1]. In 2017, Sucuri, a website security company, estimated that over 50 million websites were malicious; These sites were either trying to gain information or install harmful software on the user's computer. However, a website is often built for a completely legitimate purpose and only later gets compromised by an intruder. This can be lucrative; A successful breach can turn a website into a cryptocurrency miner, a distributor of malware (including ransomware), a mass mailer of unwanted advertising ("spam"), or a node in a "botnet" that can be used to send disruptive traffic to other server computers [2, 3, 4, 5, 6, 7, 8].

In the early days of the Web, pages were created and modified by editing HTML files manually. This became untenable with large, interconnected sites; Poorly written HTML code, broken tables, disconnected links, poor quality content, and missing graphics were typical problems [9, 10]. A CMS prevents these problems because the HTML code is generated by the CMS software. The main

*Correspondence concerning this article should be addressed to Marko Niinimaki, Webster University Thailand, Bangkok, Thailand. E-mail: niinimakim@webster.ac.th

functions of a typical CMS (authoring support, templates, combining data sources) were described by Boiko [11]. Based on survey samples, W3Techs, a web technology survey company estimates that more than 50% of active websites utilize a CMS, and the most popular ones are WordPress, Joomla, and Drupal [12].

Software products are, of course, vulnerable and CMS are not an exception. Sucuri Security's survey of compromised websites found that sites utilizing a CMS can be easily infected. In most instances, the compromises which were analyzed had little, if anything, to do with the core of the CMS application itself but more with its improper deployment, configuration, and overall maintenance by the webmasters. In a few cases [13], however, the vulnerability is so-called 0-day, meaning that there is no yet a countermeasure for it.

The focus of this paper is to study how website hackers (users with malicious intent) find out about CMS installations, what kind of tools they use, and where their traffic originates from. Our earlier research concentrated on non-CMS sites and established that almost 90% of all traffic and almost 100% of web server requests were potentially malicious and that the traffic often originated from China, the U.S., Czech Republic, and the Netherlands [14].

Here, we use a methodology similar to our previous research, but our data collection spans a longer period. Moreover, in order to focus on the CMS software, we use virtual hosting. This means that all our CMS websites (wordpress.wirlab.net, joomla.wirlab.net, drupal.wirlab.net) are hosted in just one IP address. The web server software directs the users to the right CMS website based on the name (WordPress, Joomla or Drupal) that is used in the HTTP request. If the website is accessed by its IP address, it only shows the Apache startup page. The pages are shown in Fig. 1.

CMS are popular, and there is a number of studies about their security. Patel [15] and Meike [16] give a general account of how popular CMSs assess potential vulnerabilities like cross-site scripting, attempts to upload malicious files, and SQL injection. Vainathyan and Mautone define an eight-dimensional security framework for CMS applications [17]. Trundle and Weippl analyze WordPress plugins, their vulnerabilities, attack types, and countermeasures [18]. Cecina and Popescu [19] have developed a honeypot plugin for WordPress to collect data about attacks. Computer security companies like Sucuri and Symantec publish reports about current CMS vulnerabilities and methods used to compromise CMS websites [20]. Telecommunication and internet companies like Verizon publish yearly reports of data breaches in general [21]. Compared to these studies, our contribution is modest: we want to find out how quickly and effectively a new CMS website is attacked, what are the typical methods used by the attackers, and if they differ according to the type of the CMS.

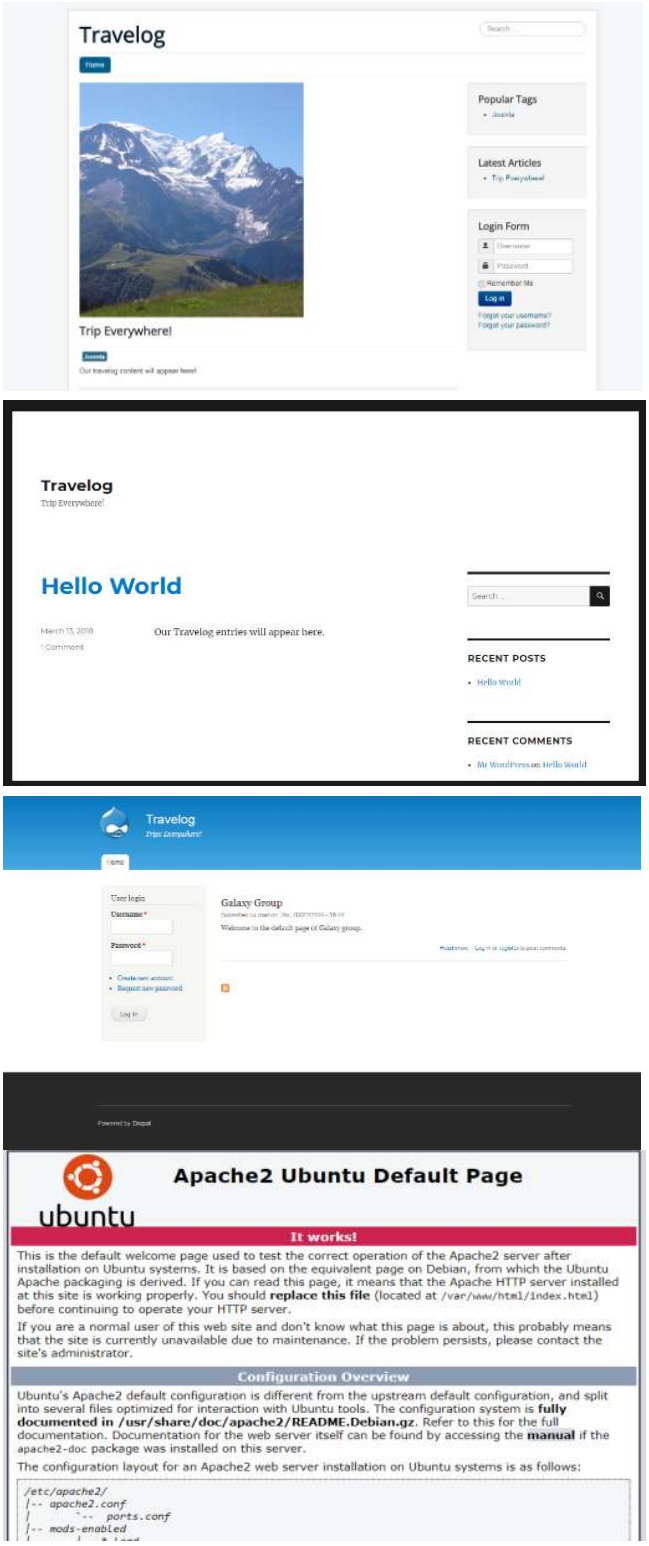

Fig. 1. Joomla.wirlab.net, Worldpress.wirlab.net, drupal.wirlab.net, and the Apache Startup Page

The rest of the paper is organized as follows. In Section 2, we describe the methods used by website hackers to breach a CMS website. In Section 3, we describe our websites and data collection method. In Section 4, we present the analysis of the website traffic and compare it with web traffic in a long-running website. Section 5 contains a summary and directions for future research.

II. TYPICAL CMS WEBSITE INTRUSION METHODS

In this section, we assume that the reader knows the technology behind basic web request methods and authentication. The communication between a client and a server (web site or "domain") uses the HTTP protocol; The client sends a request, typically GET or POST with parameters, and receives a response from the server. The response can be positive (code 200 followed by data) or negative (for example 404 page not found). Most, if not all, CMSs implement authentication using cookie technology. The basis of this technology is well-known: the client's browser (session) contains a cookie with some fields (username, the hash value of password) and values. The server is able to detect the fields and values, and, if they match the server's session data, the user is authenticated [22].

The simplest intrusion method is trying to log in to a CMS by guessing a user's username/password combination. A method like this can utilize a user's personal information or a set of words on which a password can be based (so-called dictionary attack) [23].

Cross-Site Scripting (XSS) gets its name from being able to circumvent the requirement of the same-origin policy: a script (most often written in JavaScript) loaded from a domain cannot access data outside that domain [24]. The same-origin policy can be violated if the malicious content is indeed produced inside the same domain as the script. As an example, let us consider a CMS extension "ext.php that can read content from an external source. A map extension typically does this because the map provider's source map file can be given as a parameter to ext.php. Now the malicious content can be fed to ext.php simply be calling it with a parameter that is a script file instead of a map file. If ext.php does not verify the format of the parameter file, the intruder can embed the harmful script into the content provided by the extension. Since the script has access to the user's session (including the cookies), it can store the authentication cookie information or send it to an intruder who can then adopt the user's identity [24].

The effect of a successful intrusion can range from annoying (the intruder can post insulting CMS contents and the user will be blamed) to devastating (if the user has stored confidential information like credit card numbers in his/her profile). Additionally, an intruder can place malware on the site and use the compromised user's reputation to urge others to download it. Moreover, a breached website can be turned into a Remote Access Tool (RAT) server. A RAT contains two components: a server residing on a victim's endpoint, and a client that is installed on the attacker's machine [25]. The client often uses HTTP requests to send commands to the server, making it carry out tasks for the attacker. This method is often called Command and Control (C2).

III. THE WEBSITES AND DATA COLLECTION

Our data collection started on July 7, 2019 when the websites were configured, started and given their DNS names. The sites ran without interruptions until December 9, 2019. During this time the websites received about 30 000 HTTP requests (excluding the internal cron requests that were made for housekeeping). About a dozen requests came from web content indexing (crawlers like Google and Baidu, identifiable by the user agent string).

The technology that drives the websites is summarized in Table 1. It should be noticed that the WordPress CMS running in the webserver was vulnerable to an exploit of arbitrary file deletion [26]. However, it does not look like this vulnerability was exploited.

TABLE 1
SOFTWARE

	Description
Operating System	Linux Ubuntu 16.04, Kernel 2.6.32-042stab134.8
Web server	Apache2 version 2.4.18
PHP	7.0.33
Drupal	7.44
WordPress	4.4.2
Joomla	3.9.8

The item concerning our sites' security by Sucuri security's vulnerability checker sitecheck.sucuri.net are shown in Table 2. The table includes our comparison site, pc123.wirlab.net. The upmost row refers to a request directly to the server's IP address. As a response, the Apache start page is shown.

TABLE 2
VULNERABILITY ASSESSMENT

	Risk Assessment
By IP only	Medium security risk. Outdated software: Apache. No firewall.
Drupal	High-security risk. Outdated software: Drupal, Apache. No firewall.
WordPress	High-security risk. Outdated software: WordPress, Apache. No firewall.
Joomla	High-security risk. Outdated software: Joomla, Apache. No firewall.
Pc123	Medium security risk. Outdated software: Apache. No firewall.

Table 3 shows the breakdown of incoming HTTP requests by CMS. It looks like almost all of the traffic reached the site by its IP address. Only 20 requests were made to joomla.wirlab.net, and they were not obvious intrusion attempts. 344 requests were made to wordpress.wirlab.net. These seem to be naive intrusion attempts by submitting the login form (but the login is unsuccessful). This traffic originated mainly from Turkey. It is possible that the name of our website resembled another one, and the users entered the site by mistake and tried to log in. No requests were made to drupal.wirlab.net.

TABLE 3
INCOMING HTTP REQUESTS

	All	Drupal	Joomla	WordPress
Requests	30077	0	20	344
Valid	24568	0	20	344
Failed	5509	0	0	0
Unique IP's	1832	0	2	107

IV. TRAFFIC ANALYSIS AND COMPARISON

A. *Traffic to CMS*

Our data collection and analysis indicate that web servers are under constant probes and intrusion attempts. However, we can see that the intrusion attempts are not very sophisticated: the intruders favorite methods seem to be brute force or dictionary-based break-in attempts (see e.g., [7]) and scripts that try to exploit web server vulnerabilities that our web server does not have. Table 4 lists the most typical methods that were used by intruders. All of the attacks were ineffective, however. The generic vulnerability scan requests failed (the server response was 404 page not found), most of the dictionary attacks did not even reach the login page (since they used the IP address, not wordpress.wirlab.net) and there was no RAT server to respond to RAT client commands.

TABLE 4
THE IP ADDRESSES WHERE TRAFFIC ORIGINATES

Method	Number of Attempts	Typical Request
Generic vulnerability scanning	Thousands	GET /phpmyadmin/scripts/setup.php GET /appserv.php
Dictionary attack	350	POST /wp-login.php
XSS	103	GET /cacti/plugins/weathermap/editor.php
RAT	9	Gh0st/xad

Table 5 shows the most frequent IP addresses and their countries of origin. It must be emphasized that we are not accusing any individual or organization. Internet addresses can be spoofed and traffic that looks like hacking can be legitimate vulnerability scanning by security companies and researchers. However, the rightmost column of Table 5 gives an estimate by AbuseIP.com, an internet security site, of the IP address being used for hacking.

TABLE 5
THE IP ADDRESSES WHERE TRAFFIC ORIGINATES

IP address	Number of Requests	Country of Origin	Abuse IP Confidence
27.124.x.y	887	Hong Kong	8%
111.23.x.y	886	China	16%
58.56.x.y	884	China	0%
104.221.x.y	873	USA	0%
203.158.x.y	855	Thailand	18%
218.89.x.y	854	China	0%
190.119.x.y	845	Peru	1%
129.211.x.y	845	China	77%
203.195.x.y	843	China	0%
106.13.x.y	843	China	0%

The pattern of traffic originating from the IP addresses is remarkably similar: it consists of a single burst of requests lasting about 3 minutes and mechanically testing generic vulnerabilities. Sequences of this kind of request can be found for example at GitHub (a source code management site).

B. *Comparison with a Long-Running Web Site*

Our comparison site, pc123.wirlab.net, has been running a web server continuously since 2017. Table 6 shows traffic to the site during the period Jul 7 Dec 9, 2019. We see that the traffic pattern to this site is a bit different than that of Table 3. The volume is larger, and there are more unique IP addresses.

TABLE 6
TPC123 INCOMING HTTP REQUESTS

No. of Requests	Details
Requests	127869
Valid	74469
Failed	53400
Unique IP's	5307

However, the contents of the queries are very similar to that of the CMS sites almost all the queries are potentially malicious. Table 7 lists the IP addresses that are most frequently used to access the contents. Of all the IP addresses used to access the site, 338 can be found in the log files of the CMS sites, too.

TABLE 7
THE IP ADDRESSES WHERE TRAFFIC ORIGINATES, PC123

IP Address	Number of Requests	Country of Origin	Abuse IP Confidence
154.8.x.y	1804	China	58%
45.192.x.y	916	Hong Kong	0%
139.199.x.y	912	China	26%
49.234.x.y	911	China	0%
139.155.x.y	911	China	0%
123.207.x.y	911	China	0%
36.67.x.y	909	Indonesia	27%
106.54.x.y	907	China	0%
122.136.x.y	905	China	55%
203.195.x.y	896	China	72%

V. CONCLUSION AND RECOMMENDATIONS

Three web sites, drupal.wirlab.net, joomla.wirlab.net, and wordpress.wirlab.net were registered, using a low-cost virtual machine provider. In all of the newly registered sites, potential intrusion started almost as soon as the sites were set up. Most of the incoming traffic was intrusion attempts by trying to exploit generic vulnerabilities that would be present in badly configured or outdated CMS modules. We estimate that the proportion of malicious/hacking traffic was more than 90%. However, a huge majority of the intrusion attempts were naive in the sense that they tested the vulnerabilities blindly, even when it was obvious after the first request that the site did not have the CMS that the requests were trying to exploit. The requests that were used to access the CMS sites were very similar to requests received by our longer running server. A study with an intentional run of CMS web sites with vulnerable plug-ins will be a part of future research. Researchers are also encouraged to investigate this domain and highlight the ways through which such attacks can be prevented.

Declaration of Conflicting Interests

The authors declare no conflict of interests.

REFERENCES

[1] Internet Live Statistics. (2019) Total number of web pages. [Online]. Available: https://bit.ly/3pry48f

[2] N. Kiyoshi, "Website evaluation using cluster structures," *Journal of Advances in Technology and Engineering Research*, vol. 5, no. 1, pp. 21–26, 2019. doi: https://doi.org/10.20474/jater-5.1.3

[3] S. Grossenbacher. (2019) If your CMS is attacked, can your security protect you? [Online]. Available: https://bit.ly/3purj5U

[4] The Federal Bureau of Investigation. (2007) Operation: Bot roast - bot-herders charged as part of initiative. [Online]. Available: https://bit.ly/37SfrEx

[5] R. Tahir, M. Huzaifa, A. Das, M. Ahmad, C. Gunter, F. Zaffar, M. Caesar, and N. Borisov, "Mining on someone elses dime: Mitigating covert mining operations in clouds and enterprises," in *Research in Attacks, Intrusions, and Defenses (RAID)*, M. Dacier, M. Bailey, M. Polychronakis, and M. Antonakakis, Eds. Cham, Switzerland: Springer, 2017.

[6] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement,* Rio de Janeriro, Brazil , 2006.

[7] ZDNet. (2018) A botnet of over 20,000 wordpress sites is attacking other wordpress sites. [Online]. Available: https://zd.net/3rAcQqW

[8] N. Kiyoshi, "Website evaluation using cluster structures," *Journal of Advances in Technology and Engineering Research*, vol. 5, no. 1, pp. 25–36, 2019. doi: https://doi.org10.20474/jater-5.1.3

[9] S. McKeever, "Understanding web content management systems: Evolution, lifecycle and market," *Industrial Management & Data Systems*, vol. 103, no. 9, pp. 686–692, 2003.

[10] N. Ugtakhbayar, B. Usukhbayar, S. H. Sodbileg, and J. Nyamjav, "Detecting tcp based attacks using data mining algorithms," *International Journal of Technology and Engineering Studies*, vol. 2, no. 1, pp. 1–4, 2016. doi: https://doi.org/10.20469/ijtes.2.40001-1

[11] B. Boiko, "Understanding content management," *Bulletin of the American Society for Information Science and Technology*, vol. 28, no. 1, pp. 8–13, 2001. doi: https://doi.org/10.1002/bult.221

[12] W3Tech. (2013) Usage statistics and market share of content management systems for websites. [Online]. Available: https://bit.ly/3pk82Uy

[13] ZDNet. (2020) Millions of wordpress sites are being probed and attacked with recent plugin bug. [Online]. Available: https://zd.net/2KHz6hR

[14] M. Niinimaki, V. Pajula, J. Lawrence, and K. Chanyalikit, "Attacks on newly registered websites, a comparison," *Kasem Bundit Engineering Journal*, vol. 8, pp. 183–192, 2018.

[15] S. K. Patel, V. R. Rathod, and J. B. Prajapati, "Comparative analysis of web security in open source content management system," in *International Conference on Intelligent Systems and Signal Processing (ISSP),* Gujarat, India. IEEE, 2013, pp. 344–349.

[16] M. Meike, J. Sametinger, and A. Wiesauer, "Security in open source web content management systems," *IEEE Security & Privacy*, vol. 7, no. 4, pp. 44–51, 2009.

[17] G. Vaidyanathan and S. Mautone, "Security in dynamic web content management systems applications," *Communications of the ACM*, vol. 52, no. 12, pp. 121–125, 2009. doi: https://doi.org/10.1145/1610252.1610284

[18] H. Trunde and E. Weippl, "Wordpress security: An analysis based on publicly available exploits," in *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services,* Brussels, Belgium, 2015.

[19] I. Cernica and N. Popescu, "Wordpress honeypot module," in *16th International Conference*

on Embedded and Ubiquitous Computing (EUC), Bucharest, Romania. IEEE, 2018.

[20] Symantec. (2019) Internet security threat report. [Online]. Available: https://bit.ly/2KWBFMH

[21] Verizon. (2020) Data breach investigations report. [Online]. Available: https://vz.to/3hnGfj3

[22] K. Fu, E. Sit, K. Smith, and N. Feamster, "The dos and don'ts of client authentication on the web," in *USENIX Security Symposium,* Washington, DC, WA, 2001, pp. 251–268.

[23] A. K. Kyaw, F. Sioquim, and J. Joseph, "Dictionary attack on wordpress: Security and forensic analysis," in *Second International Conference on Information Security and Cyber Forensics (InfoSec),* Cape Town, South Africa, 2015.

[24] P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna, "Cross site scripting prevention with dynamic data tainting and static analysis," in *NDSS Symposium*, San Diego, CA, 2007.

[25] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *International Conference on Communications and Multimedia Security*, Madgeburg, Germany, 2014.

[26] C. Valli, P. N. Rabadia, and A. Woodward, "A profile of prolonged, persistent SSH attack on a Kippo based honeynet," in *Proceddings of Annual Confernece on Digital Forencics, Security and Law,* Daytona Beach, FL, 2015.