SECURITY MEASURES FOR OPEN SOURCE WEBSITE PLATFORMS

Gabriel Eugen GARAIS[1]*

ABSTRACT:

Open Source Website Projects are widely spread among web developers and web users. The ease of installing and handling Open Source Web Site Platforms is known to be a handy solution but also a risky one. The use of such platforms is under heavy discussion because of the transparency that not only a normal user sees but also a hacker.

KEYWORDS: Open Source Websites, Open Source Projects, Security

INTRODUCTION

Open Source Website Projects that are free for download are a common use for unexperienced users but also for professional web developers. These projects are created by web developers and programmers that gather in form of communities. A project has a certain standard structure that is represented through the database design rules and source code rules. The main standard structure of rules is identified as the core. By definition such a project can be obtained for free from the main community website page. On the main community website can be found installation instructions but also important updates that are crucial for maintaining the integrity of the installed website platform. Updates have different justifications depending what concern it is addressed to. There can be updates as patches for solving simple bugs in the program but also security issues that can risk the vulnerability of the entire platform if left unsolved. This article will refer as a security case study to the Open Source CMS Platform called WordPress as this platform is one of the most used free platforms.

SECURITY THREAT SOURCES AND REASONS

The security issue is a very important part of an Open Source Website platform. The identified hackers, for such platforms, are in most of the cases automated remotely scheduled programs that already know the entire structure of the core platform and search for *open doors* to hack the system. Each issue will be addressed in the rest of this article.

Most common identified threat is ***brute forcing the login forms*** so that the hacker gains access to the administration part of the platform. The hacking program will try to identify the administrator username and password by testing a list of commonly used usernames and passwords. The main security risk for this kind of attack comes from the transparent and available for download standard core structure which permits the hacking code to know all vulnerabilities. Unfortunately the commonly most used username for backend administration is admin which also makes the most used username for brute force attacks. The chart presented in figure 1 [1] shows the daily massive brute force attacks during April 2016 on Websites using WordPress. The observations were recorded by the company that developed one of the security plugin for WordPress called SUCURI.

[1]* corresponding author, Lecturer PhD, Romanian-American University, Bucharest, garais.gabriel.eugen@profesor.rau.ro

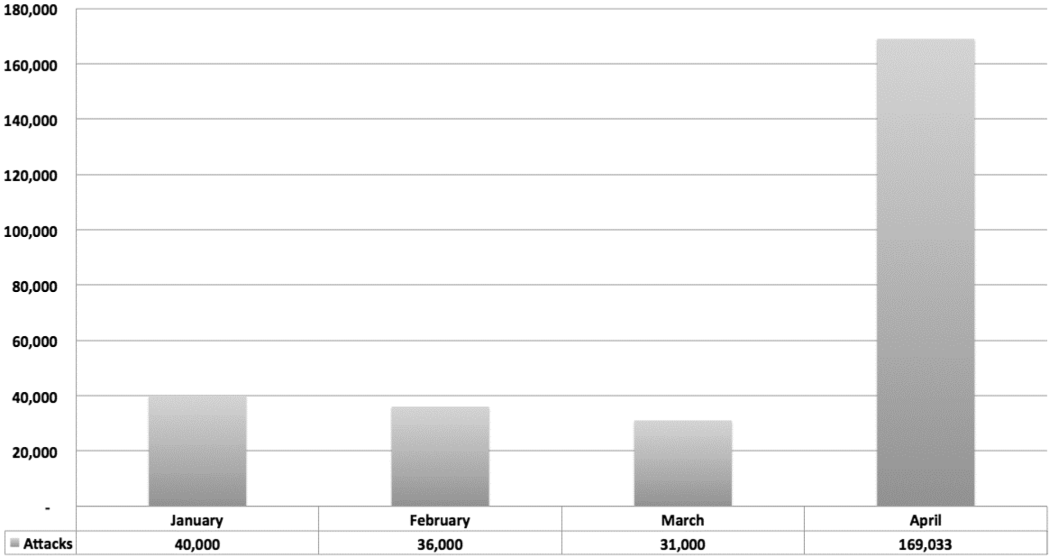


Figure 1 – Daily average number of Brute force attacks on WordPress [1]

The chart in figure 2 [2] presents the number of attacks by usernames during the time span between January and April 2016.

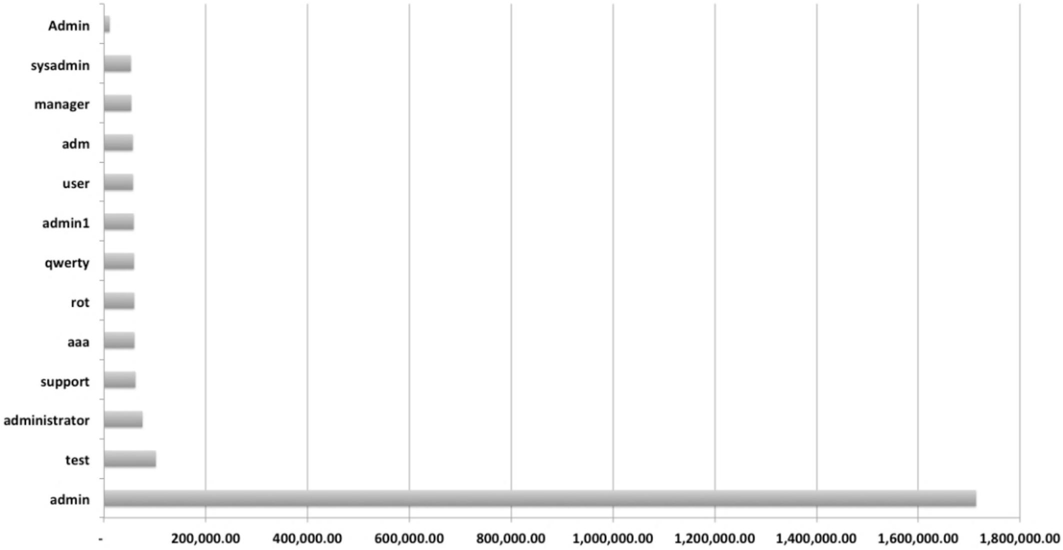


Figure 2 – Username attack distribution [2]

Each open source community project comes with the possibility of adding additional features by installing plugins, modules and themes. Each of these additional features comes with additional source program codding but also with database design adjustments. By adding modifications to the standard core community platform the risk of security vulnerability increases. The risk can get much higher when adding new features from websites that are not directly owned by the community programmers and designers. Each new feature must be tested by the community programmers for security vulnerabilities and approved so that the security issue is held at a certain level of predictability. Unprotected features can lead to ***SQL Injections or virus code uploads*** in the core files structure of the platform.

An interesting analysis [3] conducted by the programmers of the *Wordfence* plugin, which is another security plugin for WordPress, shows the actual reasons of attacking a website suggesting that it is not important if the website is hosting important information because if a website is hacked the compromised platform will conduct other further automatized hacks to other targeted websites and so on.

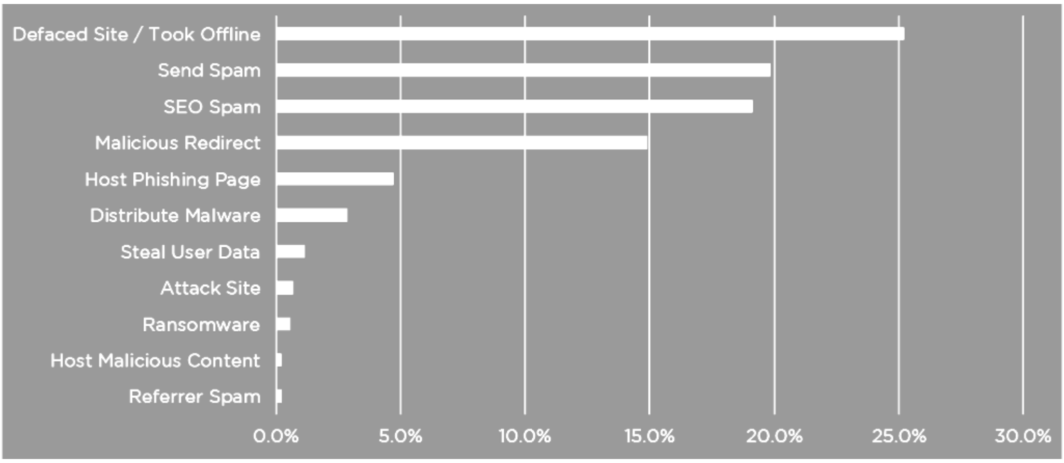


Figure 3 – Purpose for attacking a website by hackers [3]

Each of the presented reasons in figure 3 have different targeted actions but all use almost the same hacking tactics. For hackers the main concern is at first to gain access to the administration backend of the website with full managing privileges then comes the decision of what to use the hacked website for. In most of the cases the hacker uses a complete network of hacked websites to act as soldiers together as an army to hack other websites. This hacking technique has a higher rate of success in hacking systems faster. Using different hacked websites having also different IP's is more successful in hacking because the targeted system has in many cases a firewall which will react to a brute force attack but having many different servers that help in the hacking process has a much more effective success.

To secure a system we first have to understand the patterns and reasons of hacking so that proper security measures can be undertaken.

An Open Source Website platform has a higher probability of being targeted for hacking but also must be taken into consideration that the programmers of the community will be the first to know about new vulnerability risk that can occur and immediately develop security patches that will harden the user platform. Being part of a community that has many affiliated programmers should give a high level of confidence that when a core security issue occurs it will be solved in a very short amount of time.

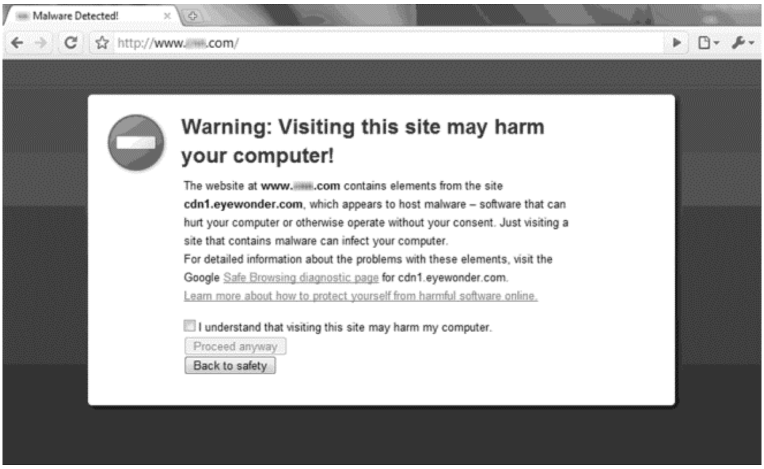
According to [3], a *defaced site* is the one that after it has been hacked the content is replaced with new content serving different purposes. In this kind of hacking it is almost impossible not to identify that the website has been hacked because it distributes a totally different content than the original.

The websites that have been compromised and are *sending spam* emails can be identified after the mail queue system blocks with a long tail of emails that can no longer be sent because of the deferrs transmitted back by the targeted email servers. The immediate result of the spam sent is blacklisting the hacked IP and domain name of the website at services like *Spamhaus*. Most of the small companies use a website only to present minimum information about its activities but rely on the email addresses a great deal to communicate with the customers and partners. If the domain name is blacklisted the real emails will no longer be transmitted and the returning error has a 421 answer code. The final issue is getting the domain name reputation destroyed.

Spam can have also a different form called *SEO Spam*. If the site has been hacked having a SEO purpose then the platform will be used to generate sitemaps and pages with backlinks to other targeted websites. Backlinks are an important factor in ranking websites and fake backlinks are still being used to create a fake good reputation to a certain website. Natural and organic backlinks are achieved with hard work and strategy but for spammers it is easier to fake the results by spamming search engines with generated backlinks. Usually if the hacker decides that the hacked website will be used for SEO spamming it will use the next procedure. After the hacker gains access to the entire filesystem, it uploads scripts that will automatically generate pages and sitemaps targeting links as backlinks to other webpages. The implementation doesn't stop here. The hacker will also authenticate itself as owner of the website for example in the *webmastertools* of the Google aps for developers and upload the links of the generated sitemaps. And the procedure continues with submitting the sitemaps to other networks that manage sitemaps of web links. After such an attack the reputation of the domain name that has been hacked will suffer sever damages and must enter in a process of clearing.

Malicious redirects and *hosting malware scripts* are another way of targeting the web pages of a hacked website. The traffic can be entirely redirected to the hackers websites or partially so that the hacking will be discovered much slower. Malware scripts will be detected after a while by search engines and the website will be tagged as dangerous and a message will be displayed like in figure 4.

In attempting to mislead the visitors to give their own personal information a compromised website could host pages that will not only redirect the solicited information to an ambiguous datacenter but will use the *phishing* strategy to identify itself being the site owner and so to convince to visitor cu apply. Through this strategy the hackers will attempt to receive information starting with email addresses and passwords till complete credit card information.



Warning: Visiting this site may harm your computer!

The website at **www.[redacted].com** contains elements from the site **cdn1.eyewonder.com**, which appears to host malware – software that can hurt your computer or otherwise operate without your consent. Just visiting a site that contains malware can infect your computer.

For detailed information about the problems with these elements, visit the Google Safe Browsing diagnostic page for cdn1.eyewonder.com.

Learn more about how to protect yourself from harmful software online.

□ I understand that visiting this site may harm my computer.

Figure 4 – Malware warning

In the same category as phishing but more aggressively we can also count the *ransomware* scripts that will encrypt and block the access to the website in an attempt to receive money from the real owner of the website. These are actually automated viruses and are very dangerous because removing the virus goes also with losing most of the original website content.

SECURITY MEASURES FOR DEFENDING OPEN SOURCE WEB SITE PLATFORMS

The enumeration of different types of attacks sources and reasons is a good way of understanding how and why a website can and could be hacked. Open Source Web Site Platforms are always a good starting point for hackers because they know the entire structure of the website and now where the vulnerabilities exist. This doesn't mean that such platforms are not reliable but the contrary because of the high number of programmers that defend the community that is gathered around the developed platform. But the final site owner that installs the platform on its own webserver and web domain must implement some security measures and rules to harden the system as instructed by the community. In the rest of the article will be presented measures that must be taken to protect the system.

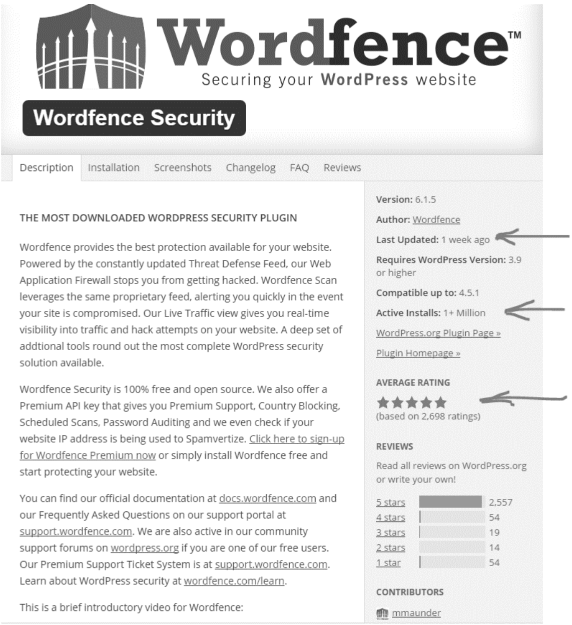
In the WordPress [4] terminology for implementing security measures it is referred as *hardening* the system. First thing that must be taken care of is the actual usernames and passwords not to be identical or similar to the top list of attempted names used by hacking servers to achieve its goals. Usernames as admin and easy passwords must be completely avoided. Instead ***use longer and complex usernames and passwords*** that contain a combination of small, capitalized letters, numbers and special characters. Such passwords are hard to remember but it has a higher level of protection. This kind of passwords can be also saved in its own browser and manage with a software that act as a vault of passwords. It is highly recommended to use a ***two-step authentication*** that can be easily achieved be installing third party plugin in the platform.

The core of Open Source Website Platforms have a certain level of secure but the level decreases and the level of risk and vulnerability rises when third party features are added to the system. Every online platform has features added as plugin, modules and themes but this actions must be done in a controlled environment taking care of security suggestions and rules.

Each additional feature including core files structure must be kept ***up-to-date*.** One important benefit of being part of an open source community is getting important updates through the main website or even directly into the platform. In most of the cases the updates are sent as messages that explain the measures that must be taken and not actually happening. Even if it could generate some incompatibilities with the updates in most of the cases it is advisable to activate the automated update procedure so that patches will be applied immediately as it was announced.

Adding new features to the platform are in some cases for testing purposes. Unfortunately some of the new features remain unused but still active on the platform. It is highly advisable to ***completely remove any unused functionalities*** on the website to prevent having more vulnerable sections in the website and limit the risk of online hacking through unsafe java scripts or unprotected web forms using post or get sending methods.

When the decision is taken to install new features, ***the plugins or modules must be first investigated*** even before testing. The investigation procedure must include searching about the vendor and programmer of the released new feature, how many downloads it has, read the users comments after testing the product, search for high rankings and references but most of all the search for the last date of the uploaded update like in the presented example on figure 5.



THE MOST DOWNLOADED WORDPRESS SECURITY PLUGIN

Wordfence provides the best protection available for your website. Powered by the constantly updated Threat Defense Feed, our Web Application Firewall stops you from getting hacked. Wordfence Scan leverages the same proprietary feed, alerting you quickly in the event your site is compromised. Our Live Traffic view gives you real-time visibility into traffic and hack attempts on your website. A deep set of addtional tools round out the most complete WordPress security solution available.

Wordfence Security is 100% free and open source. We also offer a Premium API key that gives you Premium Support, Country Blocking, Scheduled Scans, Password Auditing and we even check if your website IP address is being used to Spamvertize. Click here to sign-up for Wordfence Premium now or simply install Wordfence free and start protecting your website.

You can find our official documentation at docs.wordfence.com and our Frequently Asked Questions on our support portal at support.wordfence.com. We are also active in our community support forums on wordpress.org if you are one of our free users. Our Premium Support Ticket System is at support.wordfence.com. Learn about WordPress security at wordfence.com/learn.

This is a brief introductory video for Wordfence:

Version: 6.1.5

Author: Wordfence

Last Updated: 1 week ago

Requires WordPress Version: 3.9 or higher

Compatible up to: 4.5.1

Active Installs: 1+ Million

WordPress.org Plugin Page »

Plugin Homepage »

AVERAGE RATING

★★★★★
(based on 2,698 ratings)

REVIEWS

Read all reviews on WordPress.org or write your own!

5 stars	2,557
4 stars	54
3 stars	19
2 stars	14
1 star	54

CONTRIBUTORS

mmaunder

Figure 5 – Example of high rated new feature on WordPress internal plugin browser

The new plugins can be downloaded for free but they also have premium features, like the example in figure 5, that have additional more complex functionalities. It ***is important not to search for free distribution on ambiguous websites for plugins that are available for sale*** at the original vendor because in most of the cases the codding has been changed so that hackers could easily take over the platform. Because of the high number of community members there is like in any other high membered market place a great competition so the prices for something very complex is very low. So it is highly encouraged to buy the additional features not only because of the low price but most of all the high security level that comes with the legal engagement.

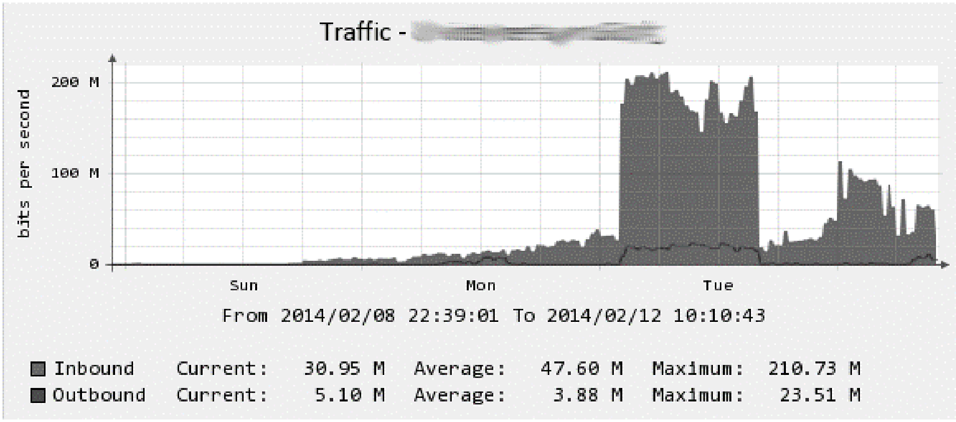
The file permissions should be changed after installing or changing the source code structure of the filesystem. The practical suggestions about file and directory permissions is not to have the permission like 777 set to directories or files but instead ***use permissions configurations of 755 or 750 at folders and 600, 640 or 644 for files***. Limiting such permissions to the filesystem can also mean reducing considerably the risk of a virus uploading to the core structure. It is advisable for directories that are used as a repository for uploaded media files to store a .htaccess file that limits the upload only t certain file types like in to codding example below.

.htaccess codding example to limit file types upload to a directory:

```
<IfModule mod_php5.c>
php_flag engine 0
</IfModule>
AddHandler cgi-script .php .phtml .php3 .pl .py .jsp .asp .htm .shtml .sh .cgi
Options -ExecCGI
```

Regarding also to user accounts security measures it is advisable that the users change periodically their passwords. An important factor to prevent Brute Force Attacks is to limit the number of failed logins and even ***obscure the actual login form***. Through a Brute Force attack the login form is repeatedly called using different combinations of usernames and passwords. To prevent such brute force attacks, which is the most common type of attack, certain security measures can be implemented. To ***limit the number of failed logins*** the system must keep tracking the number of same usernames called by one single IP in a certain amount of time. Through a security algorithm distributed brute force attacks from different IP's can be discovered and blocked. Most of the attacks are driven by automated programs that run by Cron Jobs on a calculated schedule divided through many synchronized different already hacked servers. To block this kind of attacks blacklisted databases of IP's can be queried to prevent the access even before it reaches the login form for a certain IP address and the platform can also collaborate by sending back to the community the initiated attack. ***Login forms can be protected for nonhuman submitting's*** by so called captcha which nowadays are not so used anymore because of smarter algorithms that can detect if the submitter is human or not.

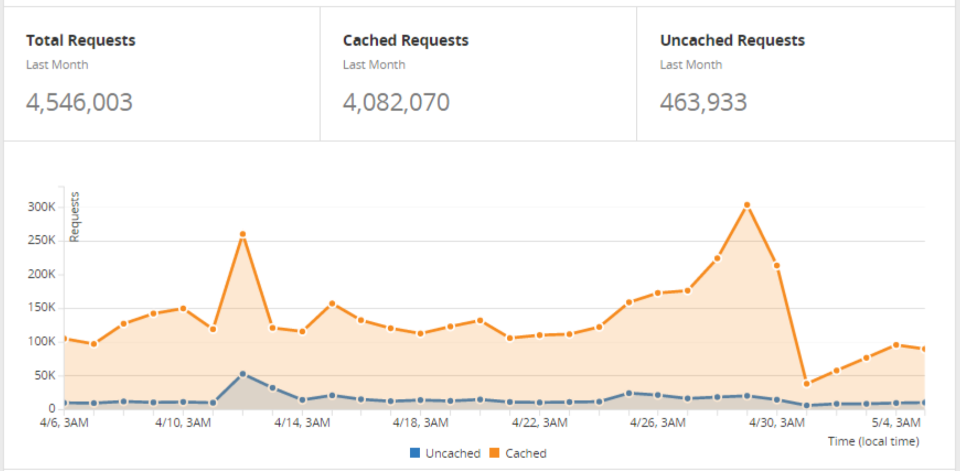
Among modern strategies for limiting access to the website there is used a service which not only can filter the type of user access but it can also deliver the page content in a much faster cached form to the final user. The CloudFlare system intermediates the entire traffic to the Website Platform through its own servers and delivers a cached but dynamically result in the same time to the end user. The filter that intermediates the traffic can also determine if the online Access has a hacking pattern or not. There is a [5] released analysis of a DDoS attack which was intermediated and filtered by the CloudFlare servers which states that the attacker used 4529 NTP servers and each used an average of 87Mbps of traffic to complete an over 400Gbps DDoS attack which is presented in the figure 6 chart.



```
▨ Inbound   Current:   30.95 M   Average:   47.60 M   Maximum:  210.73 M
▨ Outbound  Current:    5.10 M   Average:    3.88 M   Maximum:   23.51 M
```

Figure 6 – 400Mbps DDoS attack detected and filtered through CloudFlare system [5]

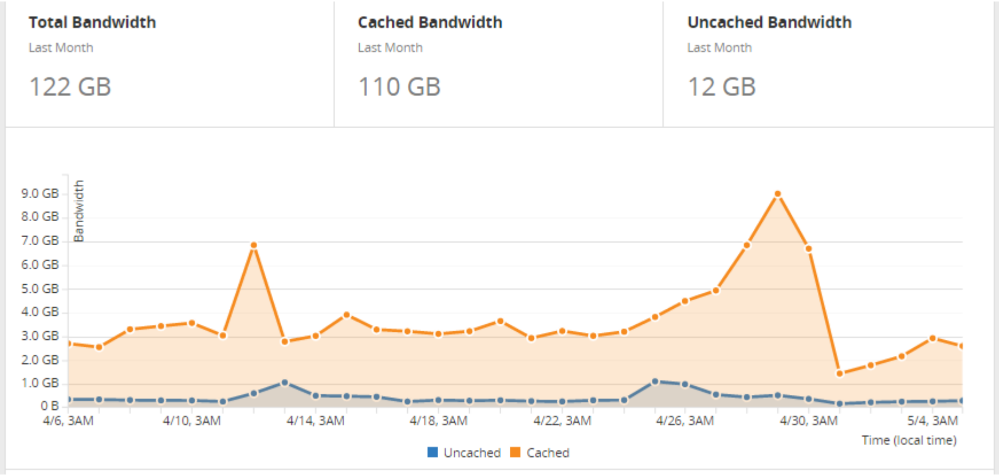
As a case study for this article regarding the usefulness of using the CloudFlare system the website http://www.depozitulderetete.ro was analyzed during the active action of the intermediate system that was used. The analysis presents charts which indicate the actual traffic that was solicited by users compared with the amount of traffic that actually reached the host server for our case study. First chart in figure 7 presents the amount of cached requests that limited the real traffic to the website server.



Total Requests	Cached Requests	Uncached Requests
Last Month	Last Month	Last Month
4,546,003	4,082,070	463,933

Figure 7 – Traffic cached for all the requests received by the website case study

In figure 8 the compared consumed bandwidth of all the traffic including text, images and client side scripts.



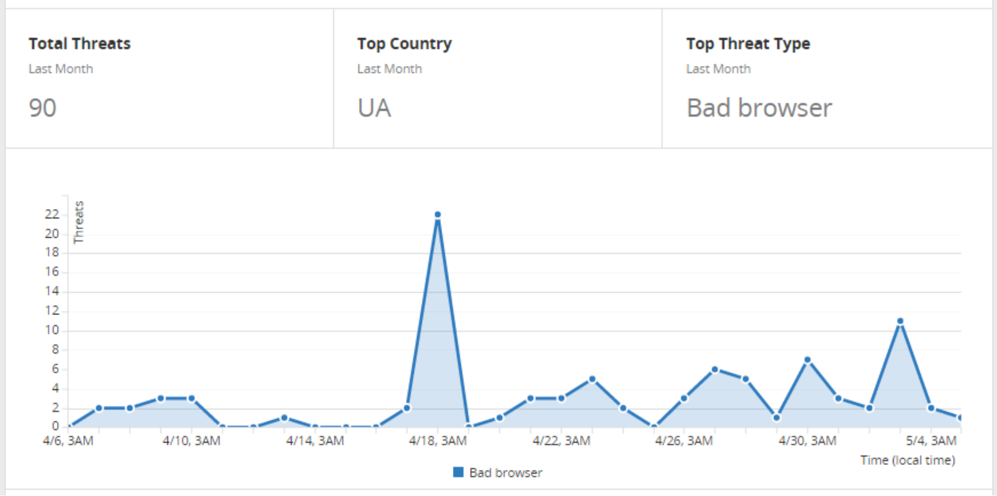
Total Bandwidth
Last Month
122 GB

Cached Bandwidth
Last Month
110 GB

Uncached Bandwidth
Last Month
12 GB

Figure 8 – Total bandwidth saved by using the intermediate system

In figure 9 are presented the total threats that have been intercepted for our web site case study which demonstrates the usefulness no matter the case if it is a small or complex website.



Total Threats
Last Month
90

Top Country
Last Month
UA

Top Threat Type
Last Month
Bad browser

Figure 9 – Threats filtered by the intermediate system for the case study

The final pie diagrams presented in figure 10 show in the most cost effective way the benefits of using such an intermediate system like CloudFlare which among other advantages it is important to mention that it is also a free service.

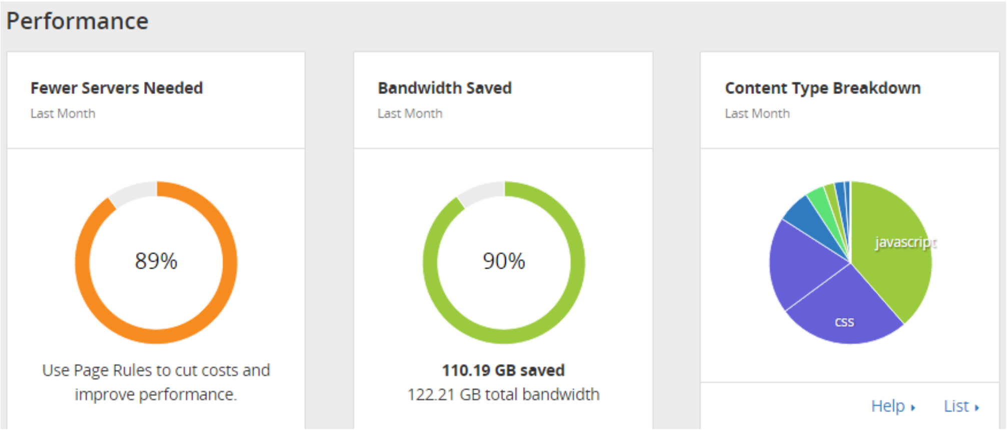


Figure 10 – Overall benefits of using an additional tire of intermediating the traffic

The analysis for the presented case study was conducted for the time period between 6th of April 2016 and 4th of May 2016. If there can be any intermediate system that is different from an antivirus or firewall and it can block malicious traffic then it is a new tire that should be strongly considered to be added to the security system of a website no matter the size. The advantages of such a system starts with lowering the bandwidth traffic that reaches the web site server and end with filtering the type of traffic which significantly can reduce the amount of attacks to the online website platform.

Only securing the system is not enough to have a reliable status of the overall protection. There should be a backup system implemented that preferably runs with a certain frequency. The backups can be stored locally or remotely on a third party server. There are already a number of free plugins that provide backup directly in a Dropbox account, google drive account, SkyDrive account and so on. The backup schedule should include not only the files of the website but also a complete copy of the database. The backups can be used not only when a website was compromised by an attack but also when different combinations of plugins tested committed some irrecoverable damages to the whole platform.

Last in this security article but not least is installing some security plugins into the website platform which also must be updated periodically. The security plugin must be tested before relying on it. One example of highly used security plugin for the WordPress CMS platform is Wordfence. It has over 1 million downloads and high rankings and references. A security plugin or a collection of many security plugins that should be compatible and complete one another through its features must ensure the services like: scanning the entire website filesystem for suspicious codding; limit the login failures, limit internal plugins to upload other file types as configured; throttle suspicious traffic if not completely block it; send messages to the administrator regarding detailed and status of the entire platform; and so on.

CONCLUSION

Using a free open source platform for developing websites which can be small sized or very complex that has also source code available open to hackers can be a subject of discussion when considering the security risks that it involves. But open source platforms compared to proprietary finite-sized software programmed by enterprises have many more programmers to inspect and test the software using wider and broader development base. The potential vulnerabilities and program flaws can be detected much faster and it depends on who detects these flaws faster meaning the community programmers or the hackers. The problem of security will always be an important part of the system no matter if it is an open source or proprietary software.

BIBLIOGRAPHY:

[1] https://blog.sucuri.net/2013/04/the-wordpress-brute-force-attack-timeline.html/screen-shot-2013-04-16-at-11-24-04-am

[2] https://blog.sucuri.net/2013/04/the-wordpress-brute-force-attack-timeline.html/screen-shot-2013-04-16-at-11-15-35-am

[3] https://www.wordfence.com/blog/2016/04/hackers-compromised-wordpress-sites/

[4] http://codex.wordpress.org/Hardening_WordPress

[5] http://thehackernews.com/2014/02/NTP-Distributed-Denial-of-Service-DDoS-attack.html

[6] http://www.infoworld.com/article/2985242/linux/why-is-open-source-software-more-secure.html

[7] http://www.computerweekly.com/feature/Open-source-software-security

[8] https://premium.wpmudev.org/blog/wordpress-security-tips/

[9] http://codex.wordpress.org/Hardening_WordPress

[10] https://hostingfacts.com/how-to-secure-wordpress/

[11] George Căruţaşu *Sustainable innovation in Computer Communication and Management*, Journal of Information Systems & Operations Management, Vol. 7 No.1 / May. pag. 35-50, 2013, ISSN –1843-4711

[12] Coculescu, C., *Possibilities of dynamic systems simulation,* în Journal of Information Systems&Operations Management, Vol. 7, No. 2, December 2013, ISSN: 1843-4711, pag. 319_324