UNIVERSITY OF PIRAEUS

DEPARTMENT OF DIGITAL SYSTEMS

Postgraduate Programme " Digital Systems Security "

ACADEMIC YEAR 2018-2019

Security overview of Web Content Management Systems:

A survey over the security measures used by the developers

Paschalia Marampouti (MTE1725)

Supervisor
Christoforos Ntantogian

2019

There is no perfect security, but you can always do your best!

Table of Contents

Table of Figures

Table of Graphs

Table of Tables

1 Introduction

Security is one of the most discussed topics of these days. Although, what about the opinion of the developers on this topic? Is hardening of their websites their priory? Do they choose to use Content Management Systems that are more security complex?

In this thesis, we tried to examine these topics in a great extent. We prepared a questionnaire about developers that they work on the security field or not and they use certain Web CMS platforms. Then, they were asked to respond to specific questions - which are mentioned above- concerning their security level and the measures they choose before they go live with their under supervision site.

Specifically, our goal was to examine whether developers modify the default security settings of three certain CMSs or even if they perform some extra security hardenings on them. The platforms about which we wanted to examine more and so, we asked about, was WordPress, Joomla! and Drupal. Each one of them has its own by default security settings and, for sure, there are several blogs, articles or even advices by their community about the best security hardening way for every newly created site. Moreover, we examined which security plugins/extensions/modules, they prefer to use in order to compare if they follow the most popular way proposed or if they prefer to use custom tools that ensure certain security fields that they believe they are more important.

Our mission was to find out who many of the so-called developers believe that the security of a website is important. Moreover, we studied which steps they do apply in order to ensure that their own site is secure enough, alongside with which one of the best plugins, extensions or modules they are using.

Finally, in the last section of the current thesis, we have gathered all the results of the survey and we cite some tables and graphs in order to discuss and combine the results to a certain extent.

2 Web Content Management Systems (CMS)

CMS is an online application that allows the application manager to create and modify its website online and more specifically its digital content, text, videos, pictures, et al. The management and all the changes are made easily and they are appearing instantly in order for the administrator and visitors to see them. (1; 2)

We have examined three certain Content Management Systems (CMS), which were: the WordPress, the Joomla! and the Drupal platform. Moreover, their publishers have already ensured the minimum security for their products. Thus, a great variety of security features is offered by default to each one of these platforms.

In this section, we are going to learn more about these three platforms and their by default security features. Besides that, we are going to find out which security hardening techniques are proposed to be applied more by the developers.

2.1 The WordPress Platform

WordPress (*WordPress.org*) is a free and open-source content management system (CMS) based on PHP & MySQL, which is used by more than 60 million websites (nearly 30% of all websites on all over the world (3)). It is most associated with blogging but supports other types of web content including more traditional mailing lists and forums, media galleries, and online stores. Comparing all the data with its competitors, we can easily say that it is the most popular CMS globally. (4)

WordPress is the least vulnerable platform than its online brothers and sisters. This is due to the WordPress Security Team, who is made up of approximately 25 experts including developers and security researchers. One of the main by default settings of the popular platform is the automated background updates for all minor releases, that being pushed out without the site owner needing to do anything.

In addition, there are several sections on which WordPress has focused in order to strengthen the core software and 3rd party plugins and themes against these potential risks. Several functions are offered by WordPress in order for the developers to be sure that unauthorized code cannot be injected and for administrators to have the right to restrict certain types of file. As the passwords concerns, they are protected with salting and stretching techniques, as they are salted and hashed based on the Portable PHP Password Hashing Framework. A wide range of functions are provided in order for the users to be protected for Cross Site Scripting (XSS) attacks, with the content to be filtered by default in order to remove dangerous entities. Unauthorized requests are prevented by default while all these default settings are continually evaluated by the core WordPress team for achieving the maximum security possible.

Another WordPress concern is platform's protection from Cross Site Request Forgery (CSRF) attacks, so by default it provides an API for the generation of unique and temporary tokens , which are limited to a specific user, action, object and a period of time and they are being invalidated upon logout.

During the period of time, WordPress monitors and decides to replace external components or even to make contributions to several third party plugins in order to make them more secure.

Last but not least, there are several internal access controls and authentication systems which keep the users to safe paths by redirecting them automatically from unwanted destinations. Although, the team of WordPress is always providing additional information to users and hints to in order to increase their security. (5)

2.2 The Joomla! Platform

Joomla! is a free and open-source content management system (CMS) for publishing web content based on PHP & MySQL. Joomla! follows behind WordPress with approximately 2.8% of all websites (2.5 million websites globally) (6; 7). It is built on a model–view–controller web application framework that can be used independently of the CMS. The Joomla! platform is another standout name that comes along with WordPress. It is not considered a friendly platform for a beginner, as the before mentioned platform, although it could be pretty adaptable in some aspects. (8)

Joomla! is the second most popular CMS right now, however it was not easy to find a list with all the default security settings which are applied to the platform. In spite that fact, there is an official portal that highlights the importance of securing your site during the setup and configuration of it. (9)

Although, it is worth mentioning that the Joomla! Project takes security vulnerabilities seriously. With the help of the Joomla! Security Strike Team (JSST) - a team of developers and security experts - and the users who want to improve their security experience inside the platform, the security issues that come up, are evaluated and being patched according their severity level during certain periods of time. (10)

2.3 The Drupal Platform

Drupal is a free and open-source content management framework written in PHP and distributed under the GNU General Public License. Drupal provides a back-end framework for at least 2.3% of all websites worldwide - ranging from personal blogs to corporate, political, and government sites. (11)

Contrary to Joomla!'s lack of a list with the by default security measures, Drupal's community has a detailed "*Security features and settings page*" (12). A Drupal developer knows from the beginning that there are several safety valves, if he chooses not to take any other action when setting up a site (even though they are never enough). Some of them are:

- ✓ the auto-logout feature after a long period of inactivity
- ✓ the allowance of certain number of bad login attempts
- ✓ give you the ability to see the exact lines that have changed when unauthorized changes occur, implements a certain password policy
- ✓ the logging of all the changes on files and especially on permissions given or retreated in any given time in order for the administrator to be able to audit the entire history
- ✓ the prevention of the username enumeration, redirection from unwanted/unauthorized destinations, et al.

2.4 Recommended Security Hardening Techniques

The WordPress platform itself advices the developers to apply certain simple tricks in order to achieve the maximum security on their websites. (13) The security hardening techniques are not very much different among the previously mentioned platforms. For sure, every platform has taken care of its product, but the pillars of securing your site are going to be elaborated below.

The main fields that that should be taken into consideration when a website is constructed are:

1. Limiting access

Developers should make some smart choices concerning the available entry points. By reducing the possible ports to a malicious person automatically the site is more secure.

2. Containment

Even when your site is compromised, you should always keep in mind that you need a backup plan -be ready to confront your damage.

3. Preparation and knowledge

Backups should be your primary concern, in order to recover and get back up online faster in case of a problem.

4. Trusted Sources

You should always trust the sources that you download from. Try to use well-known repositories and not to get plugins/themes from unknown and untrusted contributors.

Respectively, the formal blogs of Joomla! and Drupal (14; 15), provide some instructions that a developer should follow in order to achieve the maximum security for his websites. There are, also, certain advices about security that are pointed out from several article writers and developer forum writers which are recommended to be applied. Some of them are mentioned below: (16; 17; 18; 13; 19; 20):

- ✓ Keep everything up-to-date (CMS core code, themes, plugins).
- ✓ Rename your login URL, as hackers know the direct URL of your login page.
- ✓ Change Database Prefix (eg wp_admin, wp_login), as it is a known one - especially when we are talking about a Wordpress site.

- ✓ Do not forget the simplest steps, such as the strengthen of your password - even this means to change the default password's hashing schema, and of course do not skip to choose more complex usernames and not something like "admin" or "user". In addition, you should use techniques, such as CAPTCHA or a security question to your login screen and/or use two-factor authentication (2FA).
- ✓ It is pretty important, not to forget to hide your CMS version or the names of the plugins/modules/extensions that you are using.
- ✓ Always use the appropriate file permissions for the appropriate users.
- ✓ Do not neglect to harden HTTP security headers, disable XML_RPC and disable users enumeration, if these are some features that are not taken care already by your platform.
- ✓ Do not omit to scan your website for malwares, vulnerabilities and block the spam deliveries, in order to limit the possibilities finding yourself exposed.
- ✓ The use of CSRF token is welcomed, if the platform you are using has not taken care of it.
- ✓ The use of a WAF (Web Application Firewall) should not be omitted, alongside with the monitoring of your site. Thus, you should keep an up-to-date blacklist or whitelist and do not forger to block any malicious action/users.
- ✓ Last but not least, it is proposed your GDPR compliance!

3 Plugins, extensions, or modules used for Security Hardening

This section is about getting to know more about the security plugins, extensions or modules which are used for securing a website.

3.1 WordPress Plugins

There is a pretty wide variety of plugins you can use to protect and harden your WordPress site, even with their free version. In this survey, we asked our participants for the top five of them and we gave them the opportunity within another field to tell us if they are using another one or even none of them. The plugins used are:

- ✓ Wordfence Security
- ✓ Sucuri Security
- ✓ All-In-One WP Security & Firewall
- ✓ BulletProof Security
- ✓ iThemes Security

Wordfence Security: It is one of the most popular WordPress security plugins. It fights spam, malware, and other threats in real time. (21) It includes WAF that identifies and blocks malicious traffic and defends brute force attacks by limiting login attempts, enforcing strong passwords and other login security measures. It also offers two-factor authentication (2FA) and several precautions for your login screen. Besides all the above, Wordfence Security scans for known security vulnerabilities and suspicious content.

Sucuri Security: It is a plugin, which offers auditing and malware scanning and for sure a variety of security hardenings techniques. Just like its competitors, it offers the monitoring and defending a website and of course the WAF, which can prevent a site from hack and DDoS. (22) On top of that, its most valuable asset is the fact that in the event of a hack or attack, Sucuri offers you actionable steps to help you proceed with repairing any damage caused. (21)

All-In-One WP Security & Firewall: It is about a plugin, where the security features are segmented into three categories: basic, intermediate, advanced. (21) Its greater advantage is the fact that it lets you know how well you are protecting your site based in the security features you have activated. (23) Some of the security fields that you can cover up with the use of this certain plugins are: the protection of the user's login page, the integrity of your file system, the possibility of blacklisting IPs and gives to users the possibility to easily backup some important files if needed.

BulletProof Security: It is not as popular as its competitors, although it does not mean that it is, to the least, less important than the others. It offers features like logging the activity of the site, monitoring and scanning for malwares. It also contains anti-spam and anti-hacking tools along the ability to restore your database if case of a breach. (21; 24)

iThemes Security: Its previously known as Better WP Security and it was a pretty popular choice for WordPress users. Its only disadvantage is the fact that it does not offer as many free benefits, as its brothers and sisters. (21) Although, within this basic security offered, a developer can protect a login page from mistreats, can scan for malwares, strengthen password policies and check the logging history and the integrity of all the files which has been changed recently. (25)

3.2 Joomla! Extensions

Joomla! also has an amazing variety of security extensions which can intercept hacking attacks and prevent from unpleasant situations. Into this survey, we aksed our participants to choose between the five most popular of them. These are:

- ✓ Akeeba Admin Tools
- ✓ JHackGuard
- ✓ RSFirewall
- ✓ JomDefender
- ✓ SecurityCheck

Akeeba Admin Tools: It is security extension that helps the Joomla! developers improving the security of their site. Its main task is fixing damaged files and ensures that the site will be safe from small attacks. (26)

JHackGuard: It is considered the best security extension for Joomla! created sites and it is formerly known as JoomlaPack. It can protect Joomla! websites from hacking attacks, such as SQL injections, Remote Code Executions or even XSS attacks. (27) It comes with a set of pre-configurable parameters which can always changes for the administrative area.

RSFirewall: It comes with a set of tools that can be used in securing your site. It scans your site in order to identify its weak points and in the end makes the necessary improvements, as your site's security concerns. (26) Moreover, it offers the ability to block certain IPs or countries, protects your from various known attacks and as all the other extensions offers the maximum possible security as your login page concerns. (28)

JomDefender: It is consider to be one of the best security extensions, in case your site has been hacked. It gives great attention to the security features concerning admin's login screen and front end & back end IP ban/blocking. (29)

SecurityCheck: It is a medium protection suite, which protects its users against SQL, LFI and XSS attacks. It gives great significancy to the permissions given to files and folders. It scans for vulnerabilities, but, to the contrary, with its competitors it cannot help you rebound in case of a breach. (30)

3.3 Drupal Modules

As for the Drupal platform, the most popular modules which are recommended, seems to serve a very specific purpose, such as the CAPTCHA security feature or the change of the password policy. Although, for the purposes of this survey we tried focusing on the four most well-known security modules, which can take a more overall control over the security strategy of one's website:

- ✓ Security Kit
- ✓ Security Review
- ✓ Paranoia
- ✓ Coder

Security Kit: It offers a foolproof to your site which allows you to configure various security options and minimize the changes of any attacks on the site. Moreover, it give you advices in order for you to make your site more resistant to malicious attacks. (31)

Security Review: It is considered to be one of the best security modules of Drupal. It contains the most of the hardening techniques you need to secure your site. It has its own Security check list, check for brute-force attacks and protects the user from XSS and SQL attacks. In conclusion, when it identifies any kind of vulnerability it gives you some of the best recommendations in order to mitigate the risk. (32)

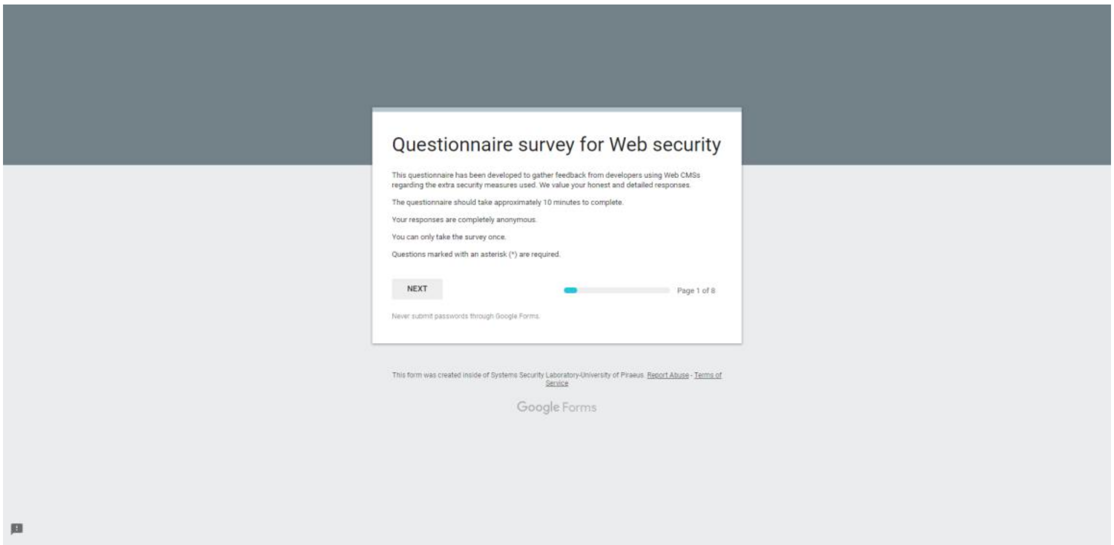
Paranoia: It provides to the developer with the vulnerable areas of his site. More specifically, it identifies them and blocks them reducing the potential an attacked to gain elevated permission. (33)

Coder: It is a module which gives you the ability to perform a deep scan to your site's code alongside with the detection of any possible drifts from the standard coding practices. Its motto is that vulnerabilities cannot lie down only to weak passwords or unpatched core modules. (32)

4 Survey Questions - The Questionnaire

Section 4 includes the questionnaire used in this survey. As we have mentioned before about the motivation behind our survey, we have concluded that measuring security is one of the most challenging tasks. Is there "adequate security"? By definition, once you get breached, your security was not adequate. Although, security is not just a machine problem, but a human too.

So, in this section, you will find out the questions used, in order to examine whether the developers take into serious consideration the importance of a website's security.



Questionnaire survey for Web security

This questionnaire has been developed to gather feedback from developers using Web CMSs regarding the extra security measures used. We value your honest and detailed responses.

The questionnaire should take approximately 10 minutes to complete.

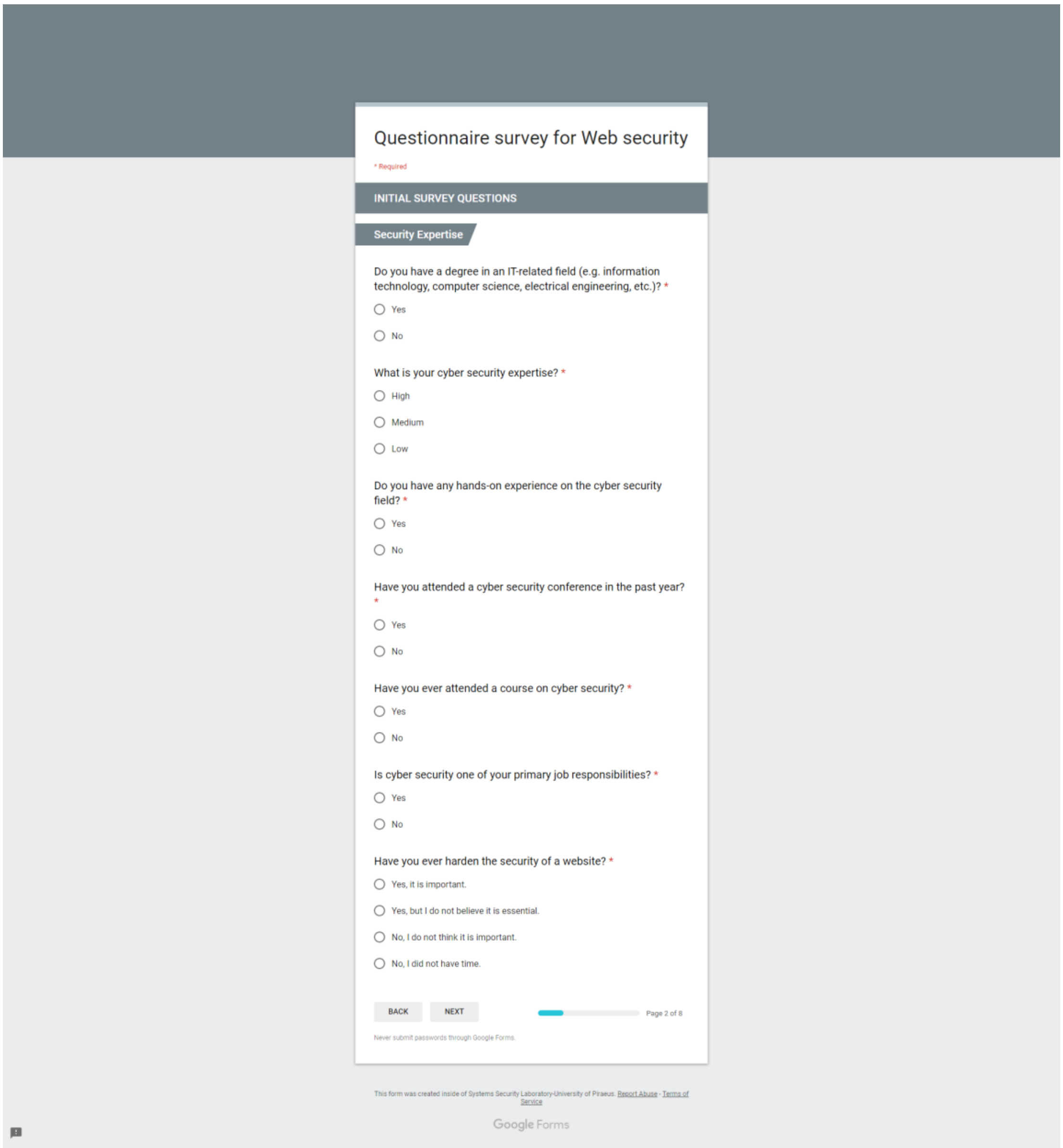
Your responses are completely anonymous.

You can only take the survey once.

Questions marked with an asterisk (*) are required.

NEXT
Page 1 of 8
Never submit passwords through Google Forms.
This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service
Google Forms

Figure 1 - Introduction for the participants



Questionnaire survey for Web security

* Required

INITIAL SURVEY QUESTIONS

Security Expertise

Do you have a degree in an IT-related field (e.g. information technology, computer science, electrical engineering, etc.)? *

○ Yes

○ No

What is your cyber security expertise? *

○ High

○ Medium

○ Low

Do you have any hands-on experience on the cyber security field? *

○ Yes

○ No

Have you attended a cyber security conference in the past year? *

○ Yes

○ No

Have you ever attended a course on cyber security? *

○ Yes

○ No

Is cyber security one of your primary job responsibilities? *

○ Yes

○ No

Have you ever harden the security of a website? *

○ Yes, it is important.

○ Yes, but I do not believe it is essential.

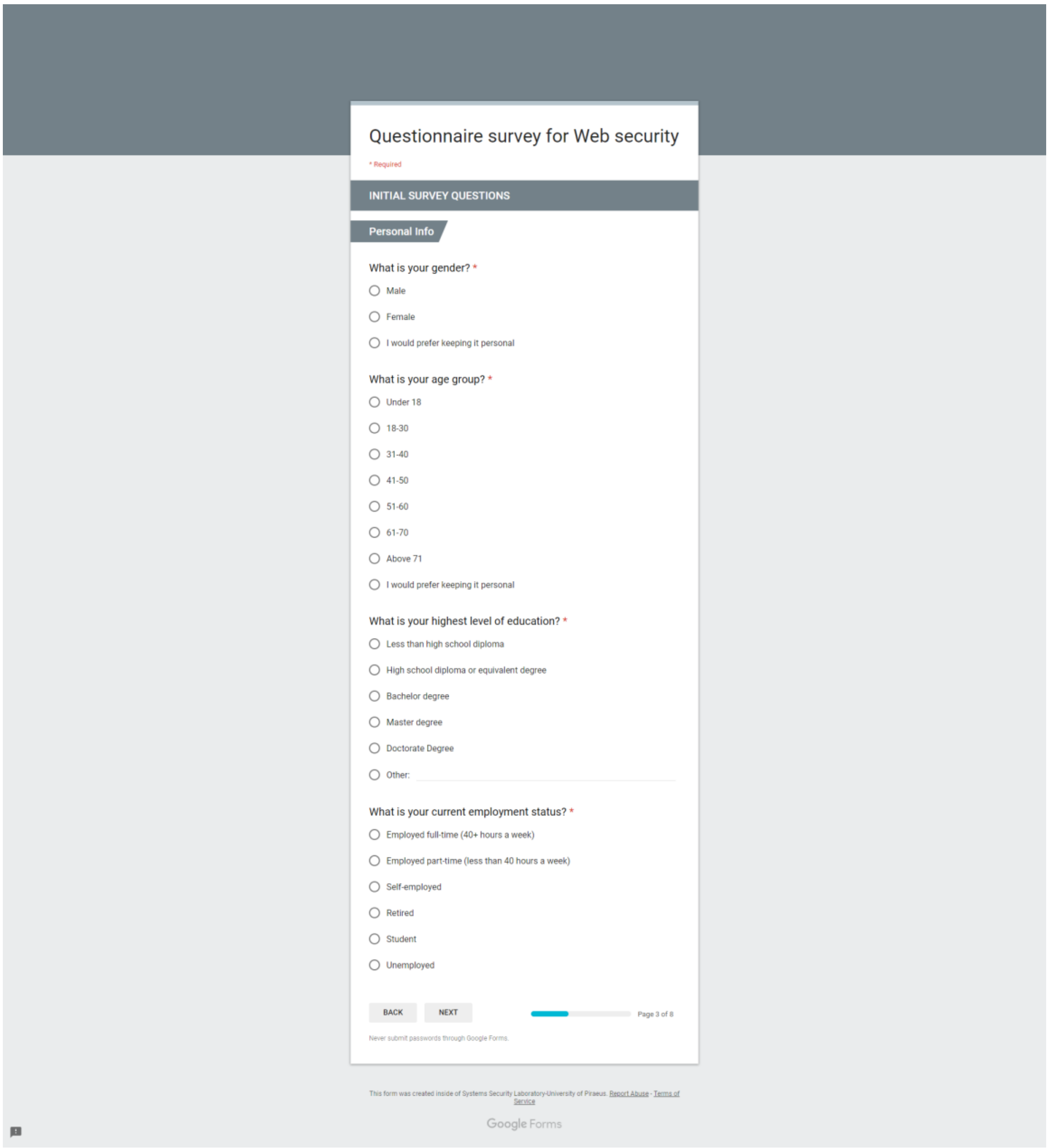
○ No, I do not think it is important.

○ No, I did not have time.

Never submit passwords through Google Forms.

This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service

Figure 2 - Initial Survey Questions (Security Expertise)



Questionnaire survey for Web security

* Required

INITIAL SURVEY QUESTIONS

Personal Info

What is your gender? *

○ Male

○ Female

○ I would prefer keeping it personal

What is your age group? *

○ Under 18

○ 18-30

○ 31-40

○ 41-50

○ 51-60

○ 61-70

○ Above 71

○ I would prefer keeping it personal

What is your highest level of education? *

○ Less than high school diploma

○ High school diploma or equivalent degree

○ Bachelor degree

○ Master degree

○ Doctorate Degree

○ Other: ____________

What is your current employment status? *

○ Employed full-time (40+ hours a week)

○ Employed part-time (less than 40 hours a week)

○ Self-employed

○ Retired

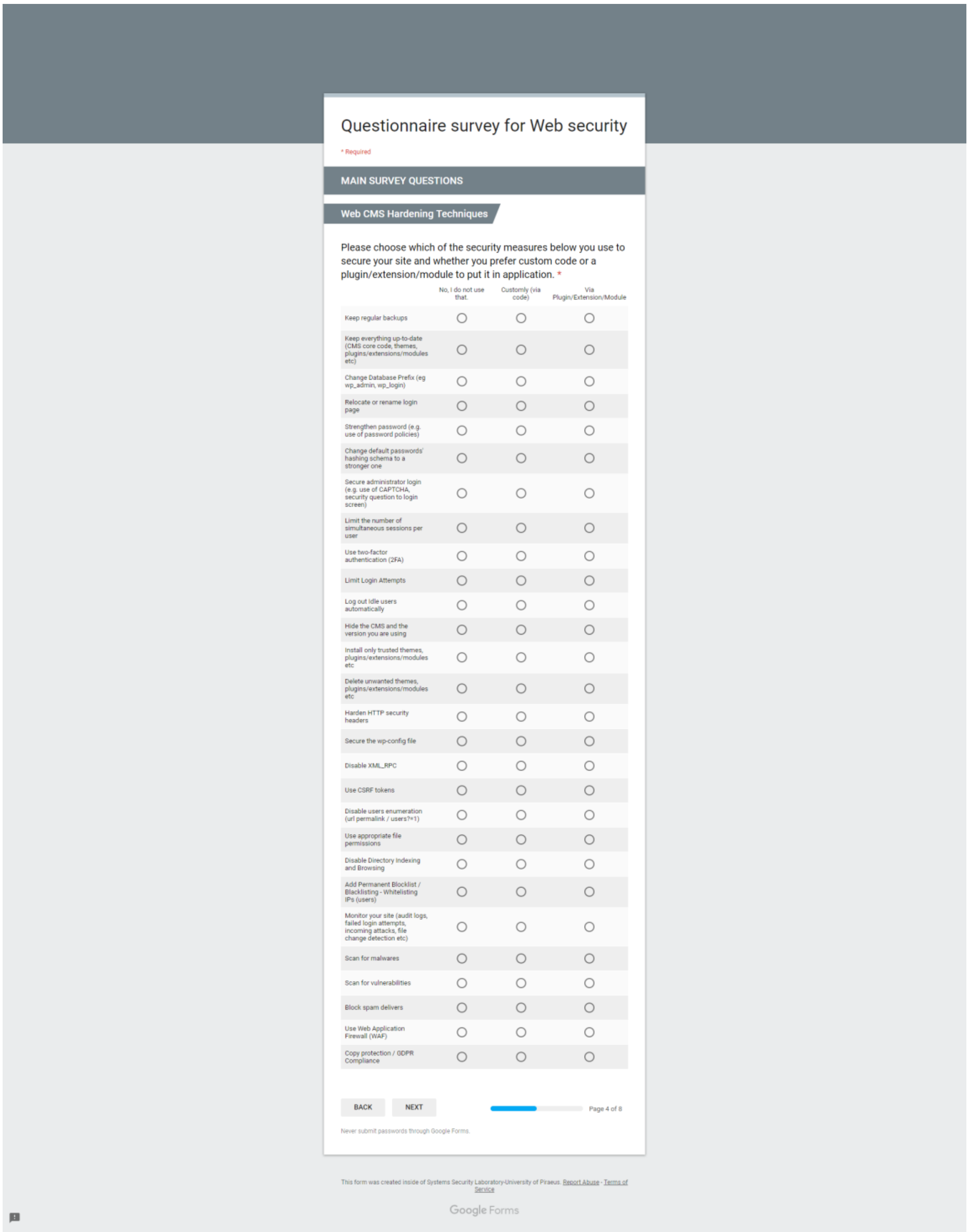
○ Student

○ Unemployed

Never submit passwords through Google Forms.

This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service

Figure 3 - Initial Survey Questions (Personal Info)



Questionnaire survey for Web security

* Required

MAIN SURVEY QUESTIONS

Web CMS Hardening Techniques

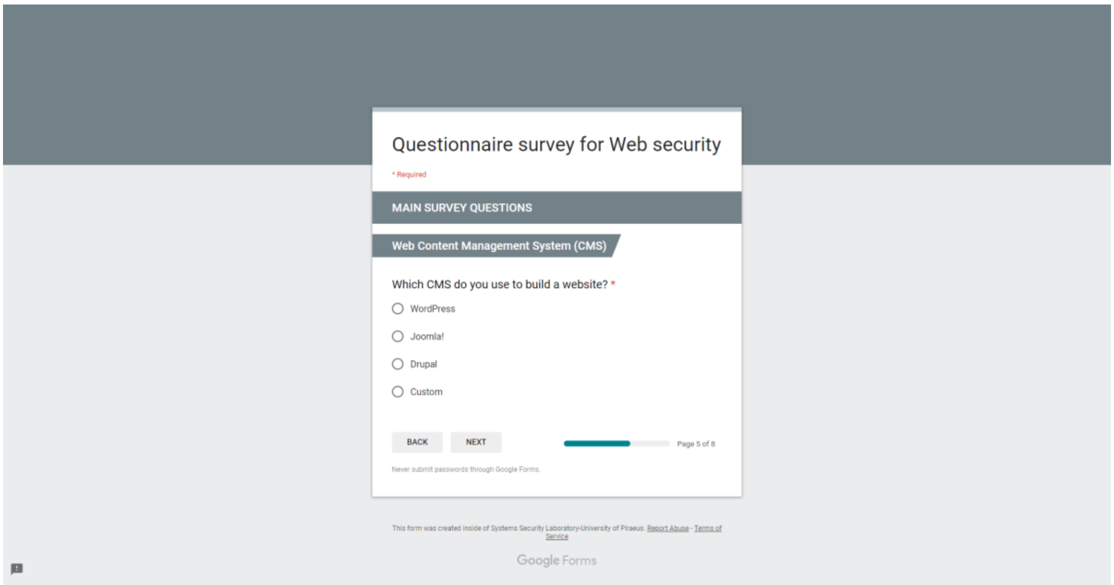
Please choose which of the security measures below you use to secure your site and whether you prefer custom code or a plugin/extension/module to put it in application. *

	No, I do not use that.	Customly (via code)	Via Plugin/Extension/Module
Keep regular backups	○	○	○
Keep everything up-to-date (CMS core code, themes, plugins/extensions/modules etc)	○	○	○
Change Database Prefix (eg wp_admin, wp_login)	○	○	○
Relocate or rename login page	○	○	○
Strengthen password (e.g. use of password policies)	○	○	○
Change default passwords' hashing schema to a stronger one	○	○	○
Secure administrator login (e.g. use of CAPTCHA, security question to login screen)	○	○	○
Limit the number of simultaneous sessions per user	○	○	○
Use two-factor authentication (2FA)	○	○	○
Limit Login Attempts	○	○	○
Log out Idle users automatically	○	○	○
Hide the CMS and the version you are using	○	○	○
Install only trusted themes, plugins/extensions/modules etc	○	○	○
Delete unwanted themes, plugins/extensions/modules etc	○	○	○
Harden HTTP security headers	○	○	○
Secure the wp-config file	○	○	○
Disable XML_RPC	○	○	○
Use CSRF tokens	○	○	○
Disable users enumeration (url permalink / users?=1)	○	○	○
Use appropriate file permissions	○	○	○
Disable Directory Indexing and Browsing	○	○	○
Add Permanent Blocklist / Blacklisting - Whitelisting IPs (users)	○	○	○
Monitor your site (audit logs, failed login attempts, incoming attacks, file change detection etc)	○	○	○
Scan for malwares	○	○	○
Scan for vulnerabilities	○	○	○
Block spam delivers	○	○	○
Use Web Application Firewall (WAF)	○	○	○
Copy protection / GDPR Compliance	○	○	○

Never submit passwords through Google Forms.

This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service

Figure 4 - Web CMS Hardening Techniques


Questionnaire survey for Web security

* Required

MAIN SURVEY QUESTIONS

Web Content Management System (CMS)

Which CMS do you use to build a website? *

○ WordPress

○ Joomla!

○ Drupal

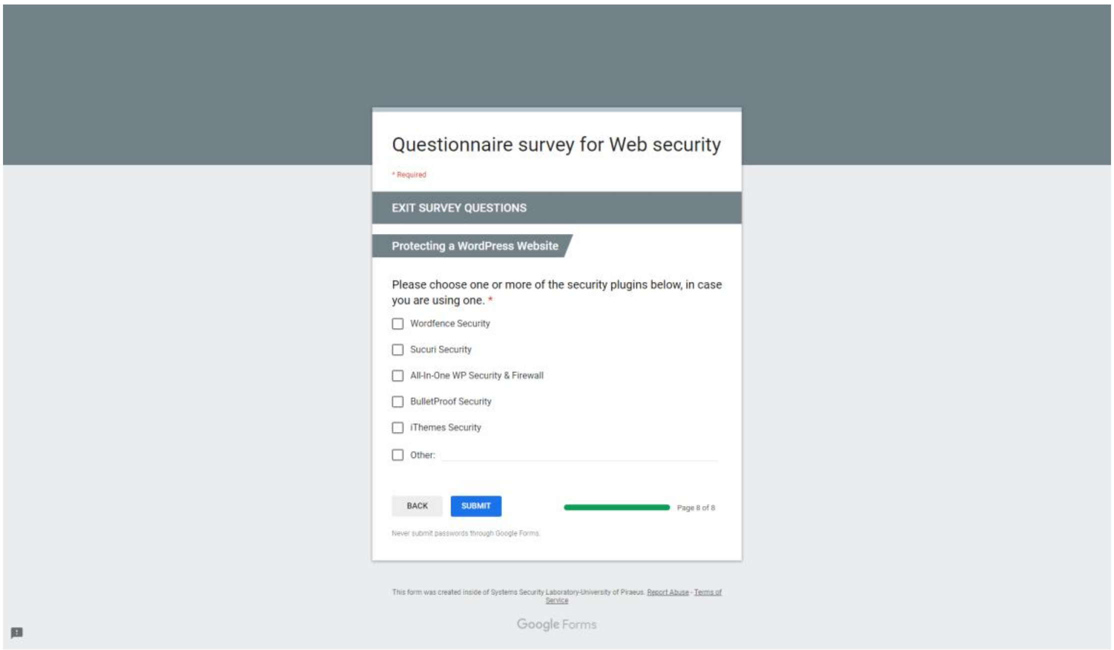
○ Custom

BACK
NEXT
Page 5 of 8
Never submit passwords through Google Forms.
This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service
Google Forms

Figure 5 - Web CMS selection

According to the choice of the participant, he sees the corresponding screen.

If his answer is WordPress:


Questionnaire survey for Web security

* Required

EXIT SURVEY QUESTIONS

Protecting a WordPress Website

Please choose one or more of the security plugins below, in case you are using one. *

□ Wordfence Security

□ Sucuri Security

□ All-In-One WP Security & Firewall

□ BulletProof Security

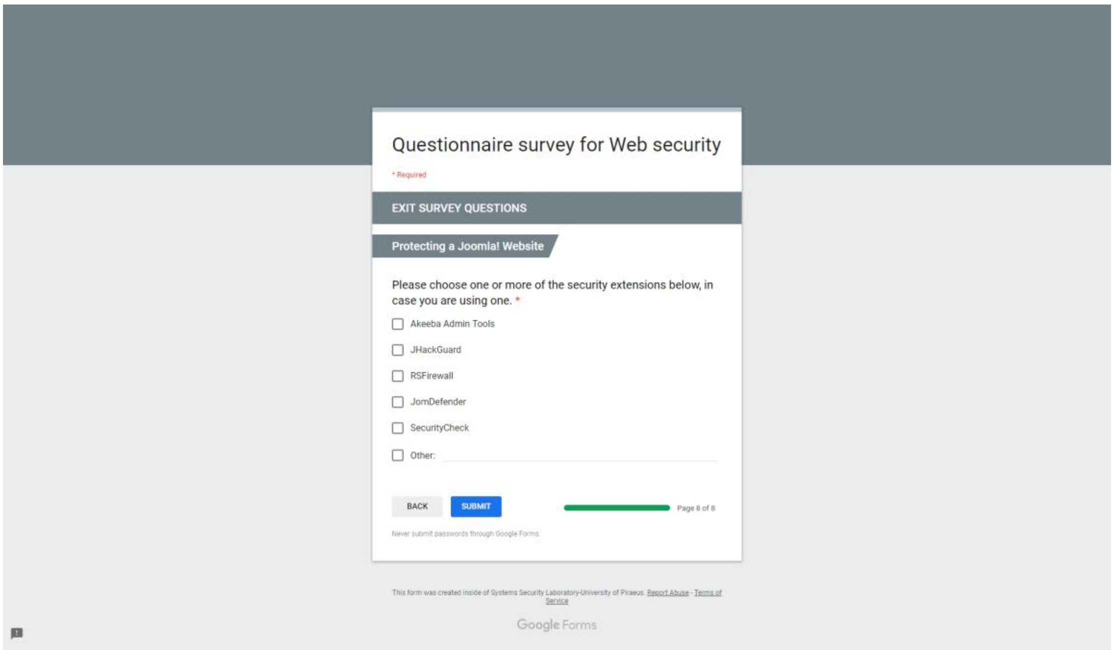
□ iThemes Security

□ Other:

BACK
SUBMIT
Page 8 of 8
Never submit passwords through Google Forms.
This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service
Google Forms

Figure 6 - Protecting a WordPress Website

If his answer is Joomla!:



Questionnaire survey for Web security

* Required

EXIT SURVEY QUESTIONS

Protecting a Joomla! Website

Please choose one or more of the security extensions below, in case you are using one. *

□ Akeeba Admin Tools

□ JHackGuard

□ RSFirewall

□ JomDefender

□ SecurityCheck

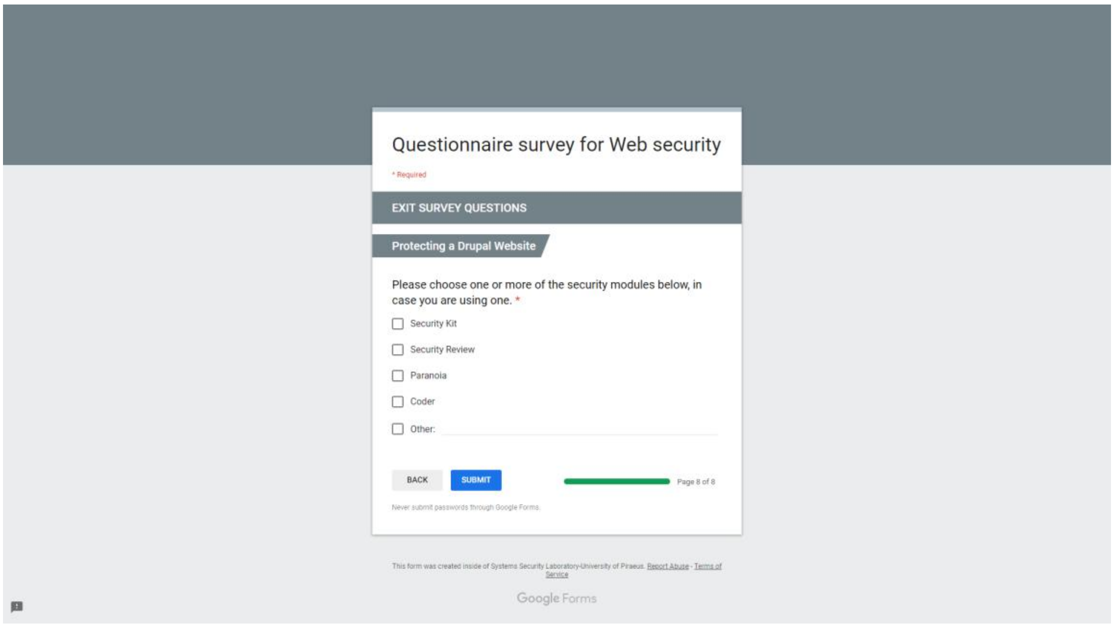
□ Other:

Never submit passwords through Google Forms.

This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service

Figure 7 - Protecting a Joomla Website

If his answer is Drupal:



Questionnaire survey for Web security

* Required

EXIT SURVEY QUESTIONS

Protecting a Drupal Website

Please choose one or more of the security modules below, in case you are using one. *

□ Security Kit

□ Security Review

□ Paranoia

□ Coder

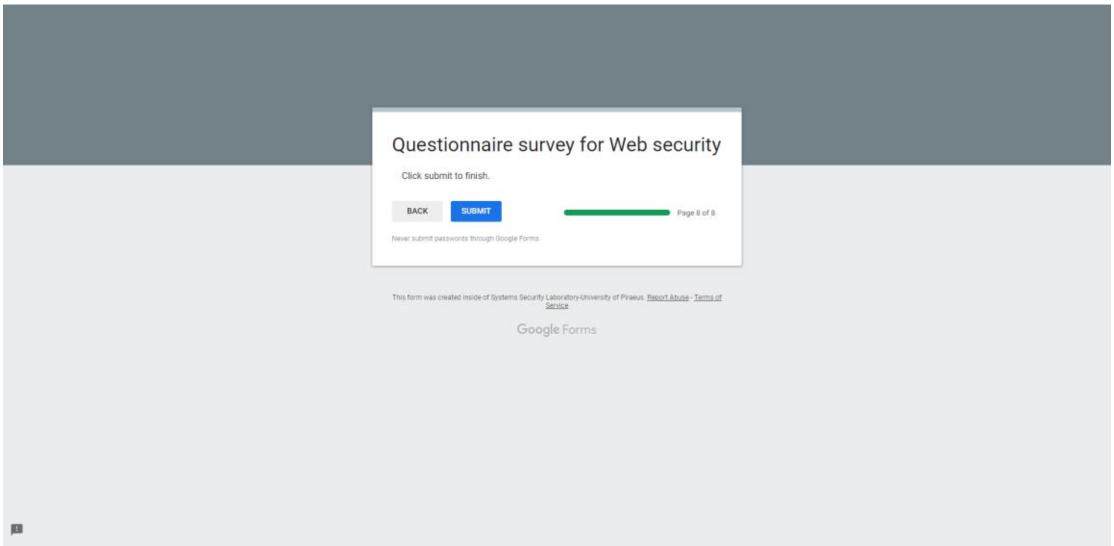
□ Other:

Never submit passwords through Google Forms.

This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service

Figure 8 - Protecting a Drupal Website

After completing the survey, he sees the next screen:



Questionnaire survey for Web security

Click submit to finish.

Never submit passwords through Google Forms.

This form was created inside of Systems Security Laboratory-University of Piraeus. Report Abuse - Terms of Service

Figure 9 - Submission screen

5 Survey Questions - The Results

In this section, we are going to analyze our *collected* data. The questionnaire was created through Google Forms and the survey took place online with a specific link sent either to tech forums (such as el.wordpress.org, joomla.gr, mydrupal.gr, insomnia.gr, adslgr.com, vcdc.gr), companies that occupy web developers (such as Grant Thornton, Develop Greece, WebOlution, Wizdom, Web Builders et al.) or to certain developers working as on the informatics or security field. The responses that we were gathered was 89, so our results are about these people.

First of all, we start with the participants' security expertise level and we are going to continue with their demographic statistics. Then, we are going to present a brief overview about the hardening techniques which are used by the questioned developers. Afterwards, we are going to examine the feedback given by the platforms that are used the most, alongside with which of the plugins, extensions or modules the majority of the users uses.

5.1 Security Expertise

Degree in an IT-related field	
Yes	82% (73)
No	18% (16)
Cyber Security Expertise Level	
High	28.1% (25)
Medium	42.7% (38)
Low	29.2% (26)
Hands-On Security Experience	
Yes	60.7% (54)
No	39.3% (35)
Attending to a cyber security conference (past year)	
Yes	36% (32)
No	64% (57)

Attending to a course on cyber security	
Yes	70.8% (63)
No	29.2% (26)
Security as one of the primary job responsibilities	
Yes	43.8% (39)
No	56.2% (50)
Hardening the security of a website	
Yes, it is important.	75.3% (67)
Yes, but I do not believe it is essential.	1.1% (1)
No, I do not think it is important.	2.2% (2)
No, I did not have time.	21.3% (19)

Table 1 - Security Expertise

Table 1 provides an overview of the security expertise of our participants. It is obvious that the most of them do have a security background with hands-on experience on the field and/or they have attended a cyber security class. In general, they believe that their security expertise level is a medium one, as most of them do not work on security field as their primary job and/or they have not attended to cyber security conference the previous year. It is interesting to observe that a great majority of them believe that the security hardening of a website is essential, even though there is a percentage that it did not had the time to devote on the security of its website. Although, we should not omit to comment the fact that there is still a percentage that it does not believe to the importance of security during the construction of a site.

5.2 Personal Information

Gender	
Male	82.4% (56)
Female	14.7% (10)
I would prefer keeping it personal	2.9% (2)
Age Group	
Under 18	0
18-30	52.9% (36)

31-40	27.9% (19)
41-50	16.2% (11)
51-60	1.5% (1)
61-70	0
Above 71	0
I would prefer keeping it personal	1.5% (1)
Level of Education	
Less than high school diploma	0
High school diploma or equivalent degree	14.7% (10)
Bachelor degree	42.6% (29)
Master degree	32.4% (22)
Doctorate Degree	5.9% (4)
BsC, MsC, OSCP, CEH *(given by the participants)*	1.5% (1)
2 x Master Degrees *(given by the participants)*	1.5% (1)
High. *(given by the participants)*	1.5% (1)
Current employment status	
Employed full-time (40+ hours a week)	69.1% (47)
Employed part-time (less than 40 hours a week)	2.9% (2)
Self-employed	17.6% (12)
Retired	0
Student	10.3% (7)
Unemployed	0

Table 2 - Personal Info

Table 2 summarizes the demographic statistics of the participants that remained to our survey. To be more specific, from this point of our questionnaire, we study the answer of the 68 remaining participants.

The excluded 21 users are those who answered in the previous section that "*they had hardened their sites but they do not believe that the security of it is essential*" or that "*they had not hardened their websites and they do not think that the security of it is important*" who are not belong to our survey's spectrum.

As for the demographics of our participants, we can see that their number is skewed towards male participants. Moreover, it seems that the majority of our sample is pretty young, as it identifies its age between the range of 18 to 30 years.

Lastly, it is worth mentioning that most of the people who participated to this survey are full-time employed and there are no unemployed users, with the majority of them to own a Bachelor degree.

5.3 Web CMS Hardening Techniques

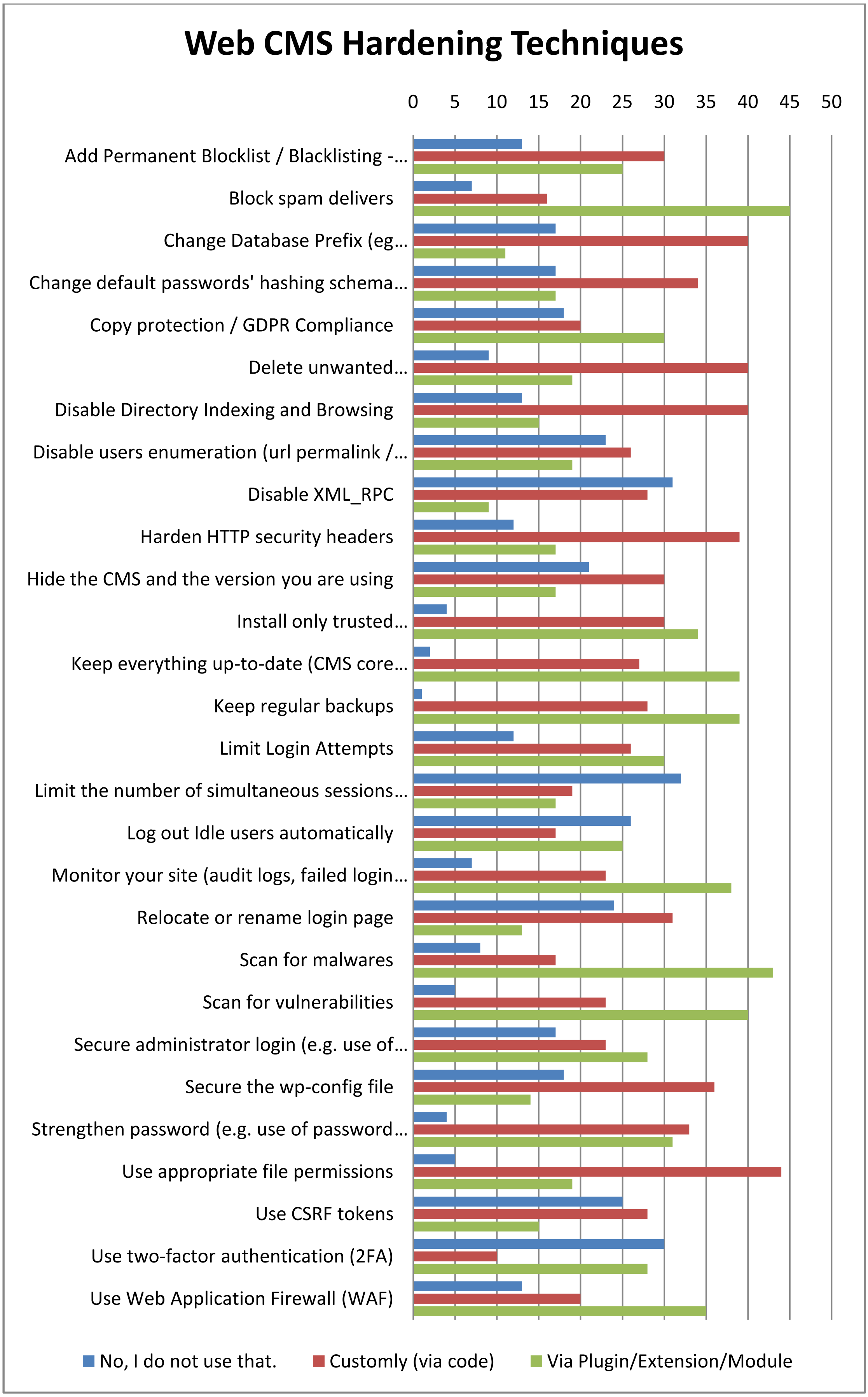
In this section, we can study the results as they emerged from our survey and they concern both participants that use custom CMS platforms and users of the three popular CMSs, such as mentioned before.

	No, I do not use that.	**Customly (via code)**	**Via Plugin/Extension/Module**
Add Permanent Blocklist / Blacklisting - Whitelisting IPs (users)	19.1% (13)	44.1% (30)	36.8% (25)
Block spam delivers	10.3% (7)	23.5% (16)	66.2% (45)
Change Database Prefix (eg wp_admin, wp_login)	25% (17)	58.8% (40)	16.2% (11)
Change default passwords' hashing schema to a stronger one	25% (17)	50% (34)	25% (17)
Copy protection / GDPR Compliance	26.5% (18)	29.4% (20)	44.1% (30)
Delete unwanted themes, plugins/extensions/modules etc	13.2% (9)	58.8% (40)	27.9% (19)
Disable Directory Indexing and Browsing	19.1% (13)	58.8% (40)	22.1% (15)
Disable users enumeration (url permalink / users?=1)	33.8% (23)	38.3% (26)	27.9% (19)
Disable XML_RPC	45.6% (31)	41.2% (28)	13.2% (9)

Harden HTTP security headers	17.6% (12)	57.4% (39)	25% (17)
Hide the CMS and the version you are using	30.9% (21)	44.1% (30)	25% (17)
Install only trusted themes, plugins/extensions/modules etc	5.9% (4)	44.1% (30)	50% (34)
Keep everything up-to-date (CMS core code, themes, plugins/extensions/modules etc)	2.9% (2)	39.7% (27)	57.4% (39)
Keep regular backups	1.5% (1)	41,2% (28)	57.3% (39)
Limit Login Attempts	17.6% (12)	38.2% (26)	44.1% (30)
Limit the number of simultaneous sessions per user	47.1% (32)	27.9% (19)	25% (17)
Log out Idle users automatically	38.2% (26)	25% (17)	36.8% (25)
Monitor your site (audit logs, failed login attempts, incoming attacks, file change detection etc)	10.3% (7)	33.8% (23)	55.9% (38)
Relocate or rename login page	35.3% (24)	45.6% (31)	19.1% (13)
Scan for malwares	11.8% (8)	25% (17)	63.2% (43)
Scan for vulnerabilities	7.4% (5)	33.8% (23)	58.8% (40)
Secure administrator login (e.g. use of CAPTCHA, security question to login screen)	25% (17)	33.8% (23)	41.2% (28)
Secure the wp-config file	26.5% (18)	52.9% (36)	20.6% (14)
Strengthen password (e.g. use of password policies)	5.9% (4)	48.5% (33)	45.6% (31)
Use appropriate file permissions	7.4% (5)	64.7% (44)	27.9% (19)

Use CSRF tokens	36.8% (25)	41.2% (28)	22% (15)
Use two-factor authentication (2FA)	44.1% (30)	14.7% (10)	41.2% (28)
Use Web Application Firewall (WAF)	19.1% (13)	29.4% (20)	51.5% (35)

Table 3 - CMS Hardening Techniques



Graph 1 - Web CMS Hardening Techniques

Table 3 and Graph 1 provide a summary with all the hardening techniques used or not either by the use of a plugin/extension/module or via code. From the 68 remaining participants, we can observe that most of the developers who use plugin, extensions or modules prefer to secure their websites over spam delivers, to scan their sites for malwares and vulnerabilities and only in the fourth and fifth position, we can see that they tend to keep backups and be certain that they have everything up-to-date. Interestingly, the installation from trusted sources and the strengthen of passwords are in a pretty high position on our graph. However, it should be pointed out that the percentage of the developers who consider changing the database prefix and relocating or renaming the login page though plugin, extension or module is really low.

Though, there are developers who prefer to harden their sites without the use of a plugin, extension or module, but via code. These participants, according to our results, as their priority has to ensure that they are using the appropriate file permissions. Moreover, they prefer to disable the directory indexing and browsing, harden the HTTP headers, secure the wp-config file and finally yet importantly to change the database prefix and to relocate the login page. Last on their list, there are techniques such as the blocking of spam delivers and scanning for malwares.

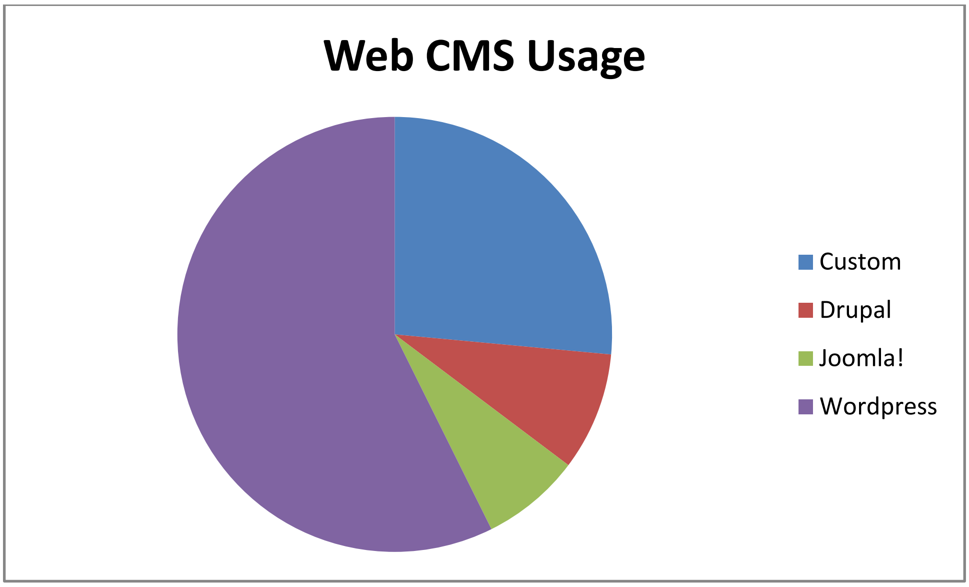
After all the above mentioned, we can observe that the developers prefer to use tools such as security plugins/extensions/modules for very specific procedures, such as scanning their websites and updating everything, although they prefer the plain coding in order to configure the files relevant fields. This could lead us to two results: a) they do not trust automated tools to manage the core files of their site or b) they do not believe that the tools which are provided by each platform are appropriate to manage a security crisis!

In the end, it is important to mention that approximately 44.1% of the developers taken this questionnaire do not use the 2FA (two-factor authentication), as a precautionary security measure.

5.4 Web Content Management System (CMS)

CMS to built a website	
WordPress	57.4% (39)
Joomla	7.4% (5)
Drupal	8.8% (6)
Custom	26.5% (18)

Table 4 - CMSs



Graph 2 - Web CMS Usage

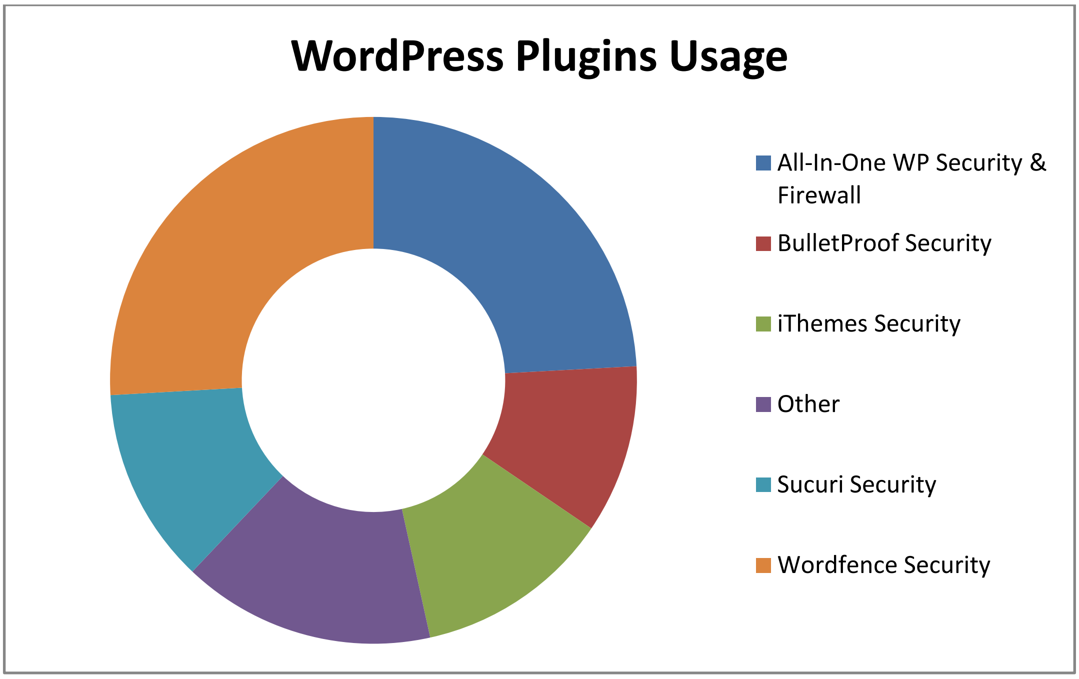
As we can see on the Table 4 and alongside it appears on the Graph 2, the most popular CMS is WordPress among the developers -and not- community. Our participants had to choose among 3 known platforms or a custom one. It is worth mentioning that custom CMS is on the second place followed by Drupal and lastly from Joomla!.

5.5 Protecting a WordPress Website

Security Plugins used to protect a site	
Wordfence Security	38.5% (15)

Sucuri Security	17.9% (7)
All-In-One WP Security & Firewall	35.9% (14)
BulletProof Security	15.4% (6)
iThemes Security	17.9% (7)
Other	23.08% (9)

Table 5 - Plugins protecting a WordPress Website



Graph 3 - WordPress Plugins Usage

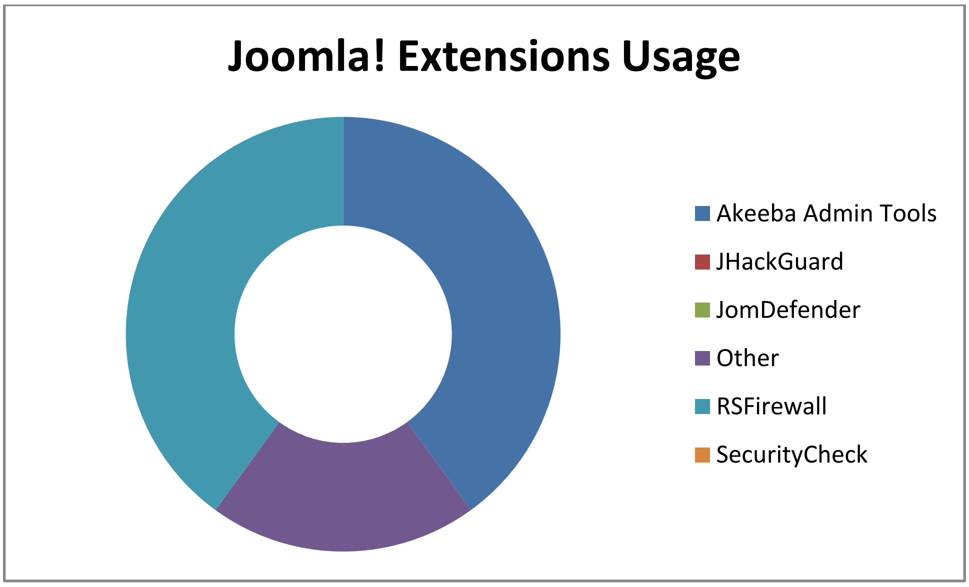
In this section, we received 39 responses (developers who are using WordPress). As you can see in the Graph 3, our prior research has been confirmed in a great extent. To be more specific, we can see that the Wordfence Security is indeed the most popular plugin among the 5 options given. Besides that, BulletProof Security is for sure not as popular as its peers, and iThemes Security is not as high as it should be and this may be due to the change of its name. Also, it should noted that in the third position we received answers like *"Not saying", "None", "Security should be set on the server side instead of plugins" and "Something about preventing cryptojacking- very easy to add mining code to wordpress sites"*.

Finally, pretty high on the list is the All-In-One WP Security & Firewall a fact attributable to its ability to categorize the risk, recommend to the user the best security practice and let him know how well he has protected his site.

5.6 Protecting a Joomla! Website

Security Extensions used to protect a site	
Akeeba Admin Tools	40% (2)
JHackGuard	0
RSFirewall	40% (2)
JomDefender	0
SecurityCheck	0
Other	20% (1)

Table 6 - Extensions protecting a Joomla! Website



Graph 4 - Joomla! Extensions Usage

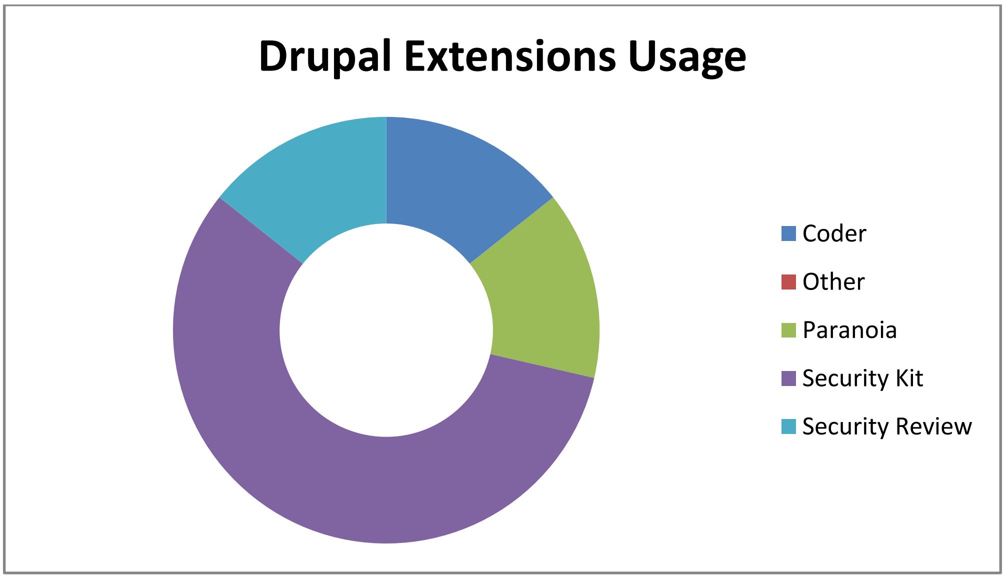
In this section, we received only 5 responses, out of which as shown on the Graph 4 Akeeba Admin Tools and RSFirewall extensions are the most popular ones. They have been chosen equal times (by 2 users each - 40%), leaving behind JHackGuard which was considered to be the most popular among the security extensions in

Joomla! To the Other field, we have received only one answer, by the value *"Other Custom Module"*.

5.7 Protecting a Drupal Website

Security Modules used to protect a site	
Security Kit	66.7% (4)
Security Review	16.7% (1)
Paranoia	16.7% (1)
Coder	16.7% (1)
Other	0

Table 7 - Modules protecting a Drupal Website



Graph 5 - Drupal Extensions Usage

Even though, it was on the second place of the CMS platforms , in the Drupal module section we have received only 6 responses. However, with a great difference for the second place (with half of the participants to use it), Security Kit is considered to be the best module in the Drupal community. Security Review, Paranoia and Coder was equally voted from one user each. Lastly, it is worth noting that one of the users chose both Security Kit and Security Review for his site protection.

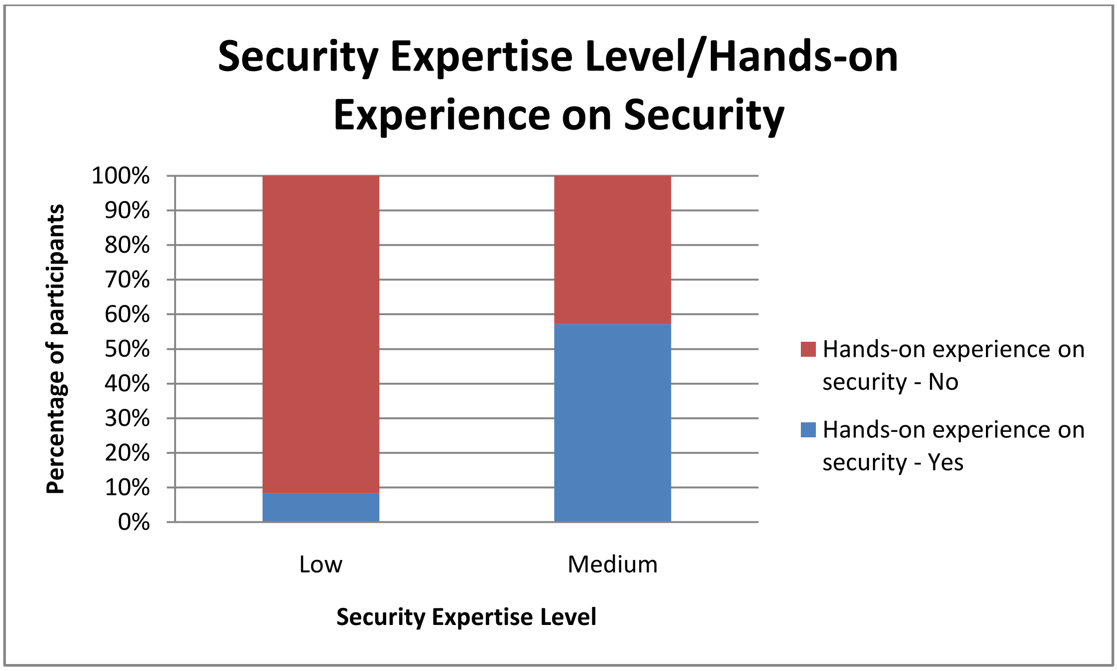
6 Survey Questions - The Discussion

6.1 Importance of security hardening of a website

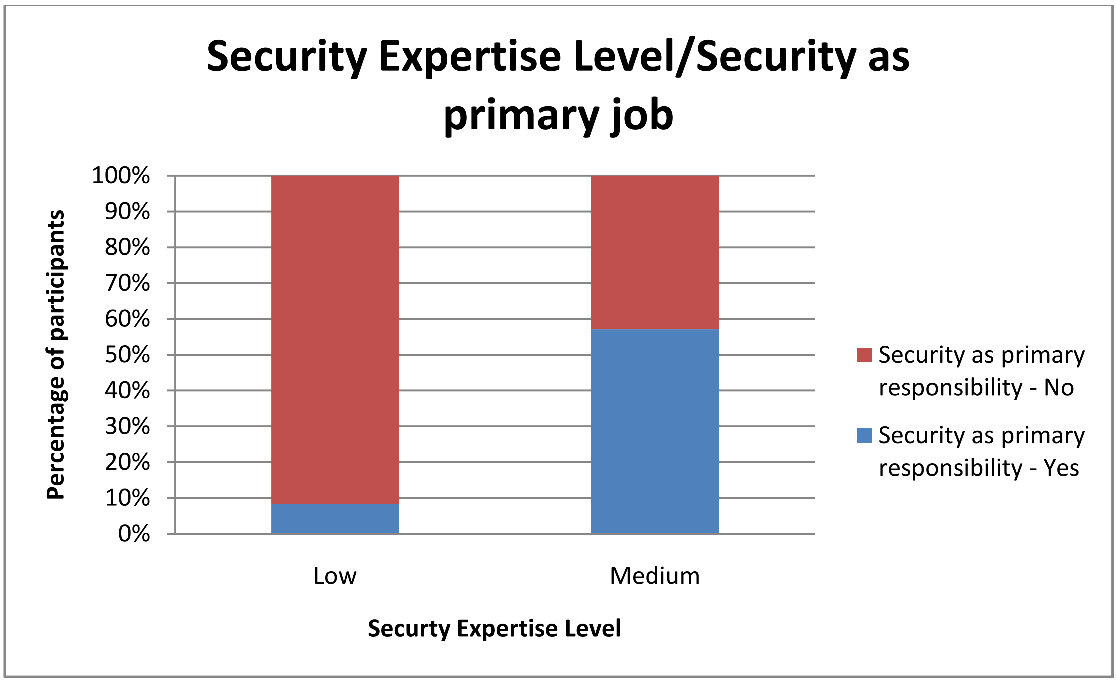
In this section, we are going to comment the 21 answers of our survey that did not continue with the questionnaire until the end, because their answers was not inside our spectrum. Although, it is interesting to take a closer look to their security level, as was declared.

6.1.1 No, I did not have time.

Out of these 21 responses that did not continued throughout the survey, 19 of them was about participants who did not had the time to secure their sites. They answered that their Security Expertise Level is *Medium* or *Low* and below we can see two charts: one correlating the Security Expertise Level of the participants (*according to their answers*) with their hands-on experience on cyber security and the second one with their having security as their primary job responsibility.



Graph 6 - Security Expertise Level/Hands-on Experience on Security



Graph 7 - Security Expertise Level/Security as primary job

6.1.2 No, I do not think it is important.

Furthermore, among these 21 answers, there are two participants who believe that *securing a website is not important*. From these two, we can stand out that their security level (*as declared*) is *Low* and they do not have *any* security background, according to their results, but only a degree in an IT-related field. More specifically, they answered that they do not have either hands-on experience on the cyber security field nor security is their primary job responsibility. So, as we can observe the participants that they do not believe that hardening a website is essential are not really security aware!

6.1.3 Yes, but I do not believe it is essential.

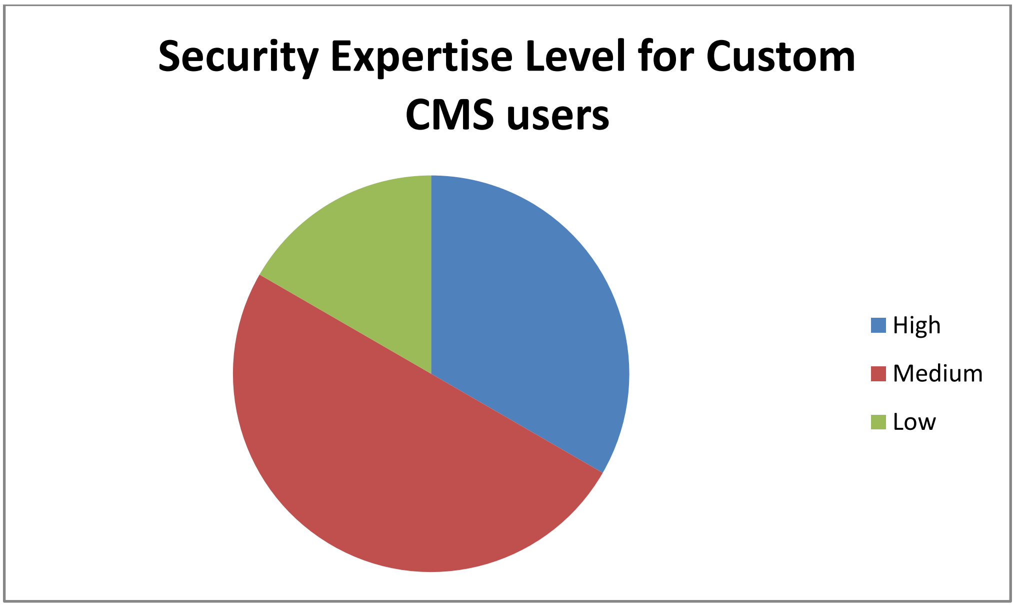
Last but least, it is worth-mentioning the fact that there was an answer from the 68 participants supporting that he have hardened the security of a website, although he does not thing it is essential. As we can observe though our results, he does not have an IT-related field degree and security is not as his primary job. He, also, believes that his security expertise level is high, as he has hands-on experience on security. Furthermore, is between 18-30 years old, he is self-employed, customly harden his websites by using the WordPress platform. He does not choose any of the recommended plugins and he did not want to declare if or which one he is using.

6.2 Web CMS Hardening Techniques Results for Custom CMSs

In this section, we are going to cite the results of the 18 participants, (Chapter 5.4), who answered that they do not use one of the popular platforms (WordPress, Joomla!, Drupal) but they use custom CMSs. We are going to see their security expertise (*according to their answers*) (Graph 8) and analyze how their results about the hardening techniques of a website formed between the three categories we have been defined: *"No, I do not use that."*, *"Customly (via code)"* and *" Via Plugin/Extension/Module"* (Table 8).

Then, we are going to introduce a graph presenting approximately how many of the responders have answered *"Customly (via code)"* at least one time to each one of the pre-defined *Hardening Techniques* and compare these answers with the Security Expertise Level that they believe they have.

Finally but yet importantly, we are going to compare the Security Expertise Level that they have declared with their Hands-on experience on security and with the fact that security is their Primary job responsibility.



Graph 8 - Security Expertise Level for Custom CMS users

	No, I do not use that.	Customly (via code)	Via Plugin/Extension/Module
Add Permanent Blocklist / Blacklisting - Whitelisting IPs (users)	4	3	11
Block spam delivers	1	7	10
Change Database Prefix (eg wp_admin, wp_login)	3	9	6
Change default passwords' hashing schema to a stronger one	5	8	5
Copy protection / GDPR Compliance	1	10	7
Delete unwanted themes, plugins/extensions/modules etc	3	12	3
Disable Directory Indexing and Browsing	4	8	6
Disable users enumeration (url permalink / users?=1)	5	5	8
Disable XML_RPC	4	5	9
Harden HTTP security headers	2	9	7
Hide the CMS and the version you are using	9	4	5
Install only trusted themes, plugins/extensions/modules etc	2	11	5
Keep everything up-to-date (CMS core code, themes, plugins/extensions/modules etc)	2	10	6
Keep regular backups	2	13	3
Limit Login Attempts	1	14	3
Limit the number of simultaneous sessions per user	5	8	5

Log out Idle users automatically	8	9	1
Monitor your site (audit logs, failed login attempts, incoming attacks, file change detection etc)	4	8	6
Relocate or rename login page	6	7	5
Scan for malwares	3	12	3
Scan for vulnerabilities	2	12	4
Secure administrator login (e.g. use of CAPTCHA, security question to login screen)	4	10	4
Secure the wp-config file	2	6	10
Strengthen password (e.g. use of password policies)	4	3	11
Use appropriate file permissions	1	7	10
Use CSRF tokens	4	5	9
Use two-factor authentication (2FA)	4	5	9
Use Web Application Firewall (WAF)	7	5	6

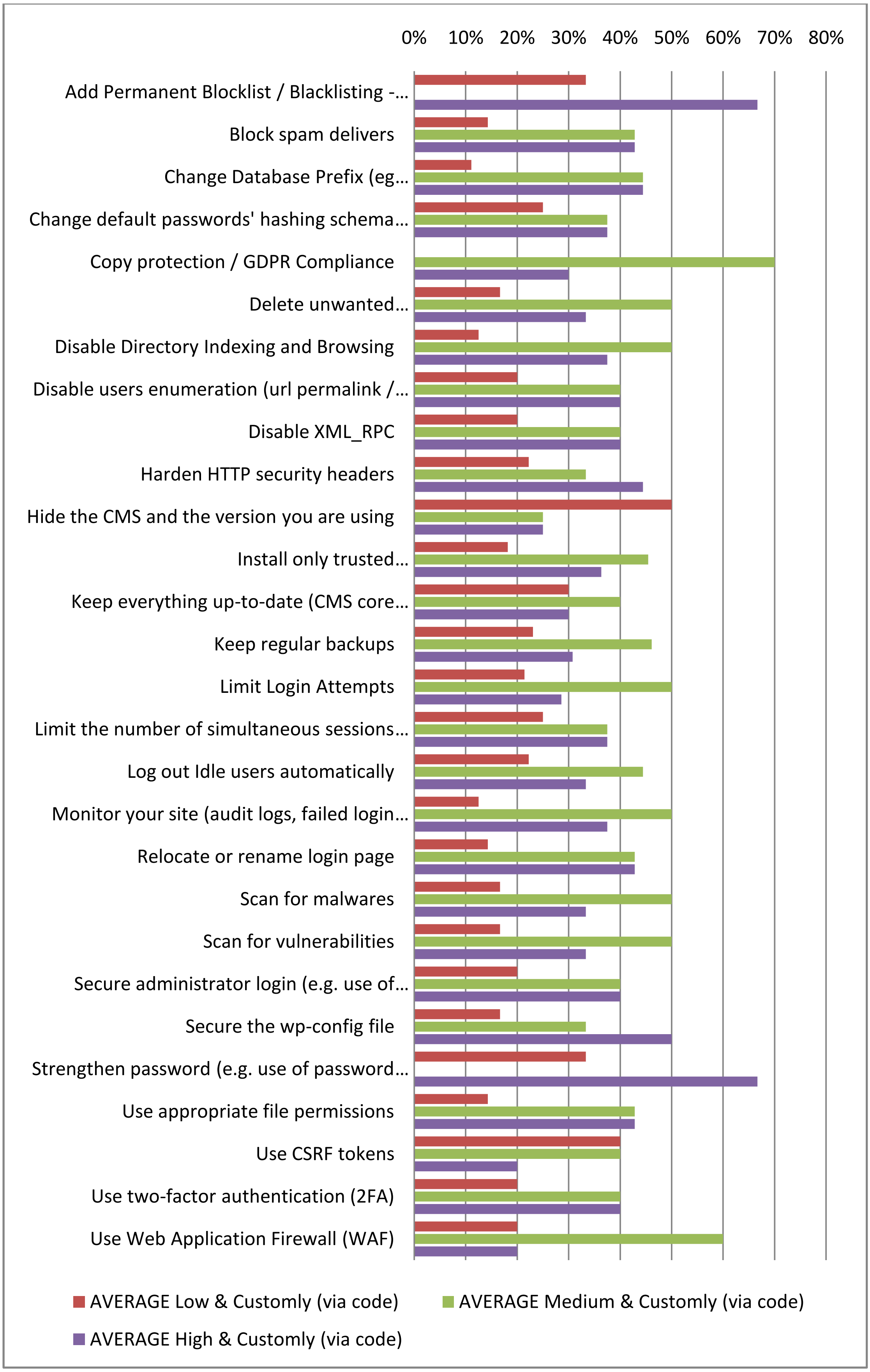
Table 8 - Web CMS Hardening Techniques for Custom CMS users

Graph 9 is a chart presenting an average percentage of the participants who answered they used Custom CMSs and *"Customly (via code)"* at least one time to each one of the pre-defined Hardening Techniques compared with the Security Expertise Level that they believe they have.

As we can observe, people with Low Security Expertise Level they choose customly to Hide CMS and the version of the CMS they use, Use CSFR tokens, Strengthen their passwords and Add permanent blocklists as their primary responsibility. On the contrary, they are least occupied with customly take care of the GDPR Compliance, changing the Database Prefix or even monitor their site.

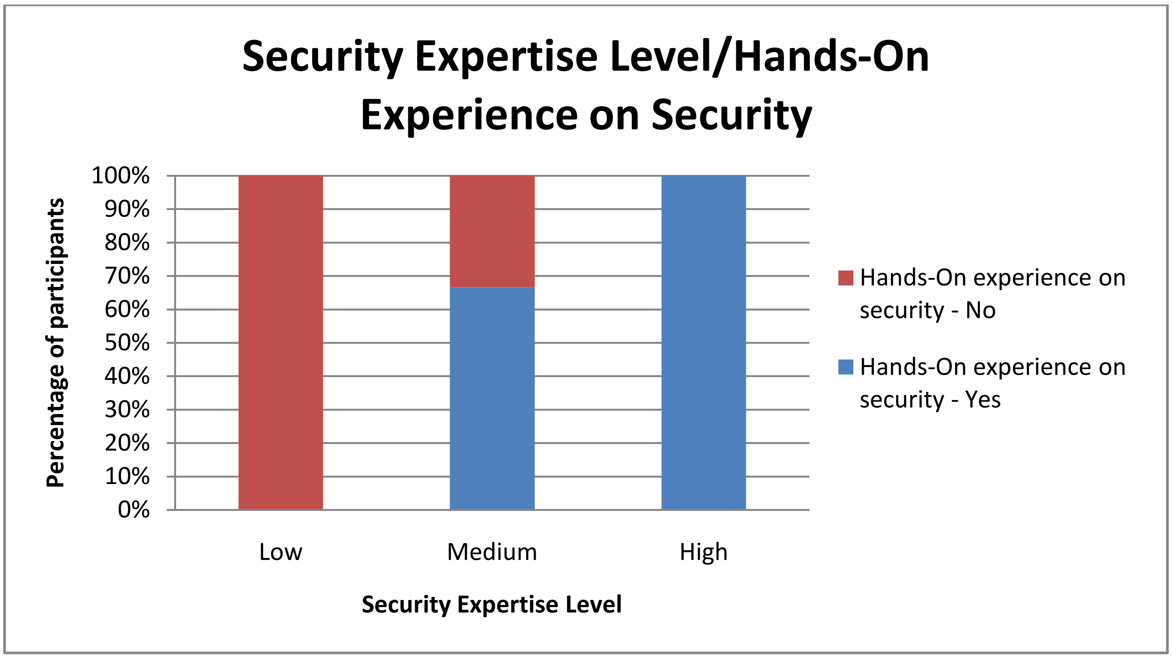
As for the Medium Security Expertise users, they have prioritized to customly take care of the GDPR compliance of their sites, to Use Web Application Firewall (WAF) and to Limit the login attempts in their custom CMS. Although, none of them (as we can of the results) is Strengthening the passwords or Adding permanent blocklists customly.

Lastly, for the High Security Level of Expertise participants we can observe that they mind to customly Strengthen the passwords that the users of their CMSs are using alongside with Adding permanent blocklists (on the contrary with the users with the responders with the medium security expertise level). It is interesting to mention that these users taking care least the Usage of CSRF tokens customly or the usage of WAF.

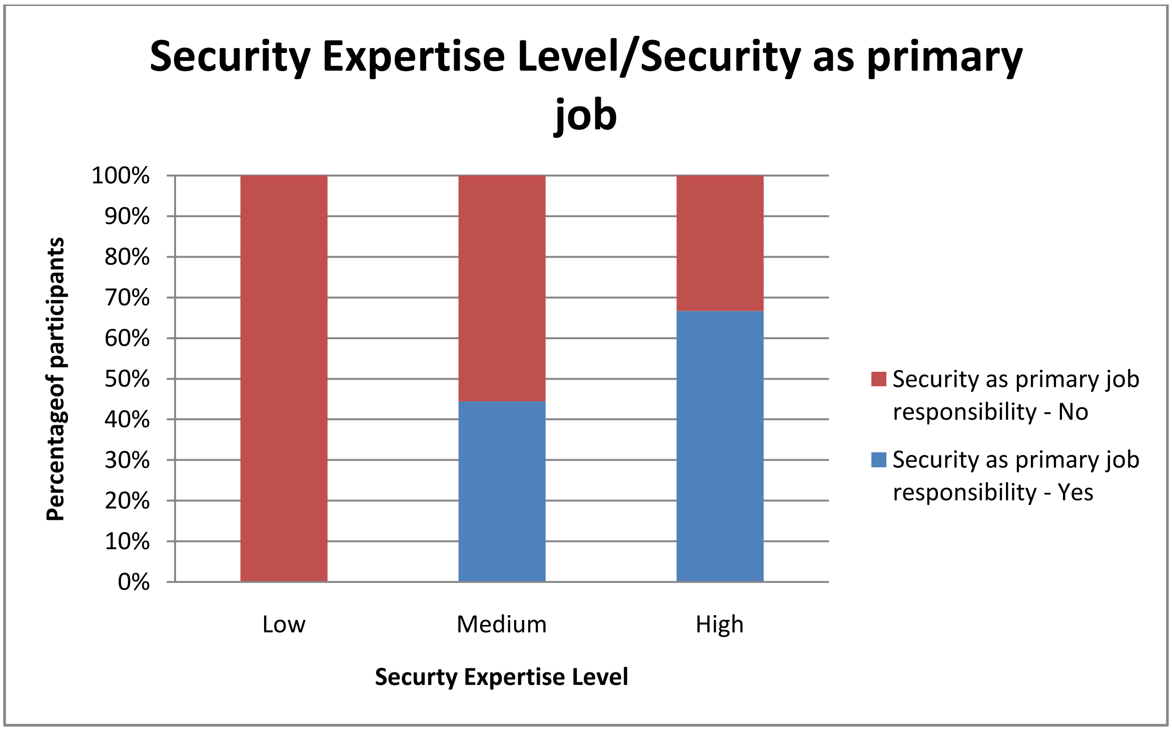


Graph 9 - Customly Hardening Techniques based on Security Expertise for Custom CMS users

After that we can observe the correlation between the Security Expertise Level that our participants answered with their actual hands-on experience on the security field and the fact that if the security is their primary job responsibility.



Graph 10 - Security Expertise Level/Hands-On Experience on Security (Custom CMS)



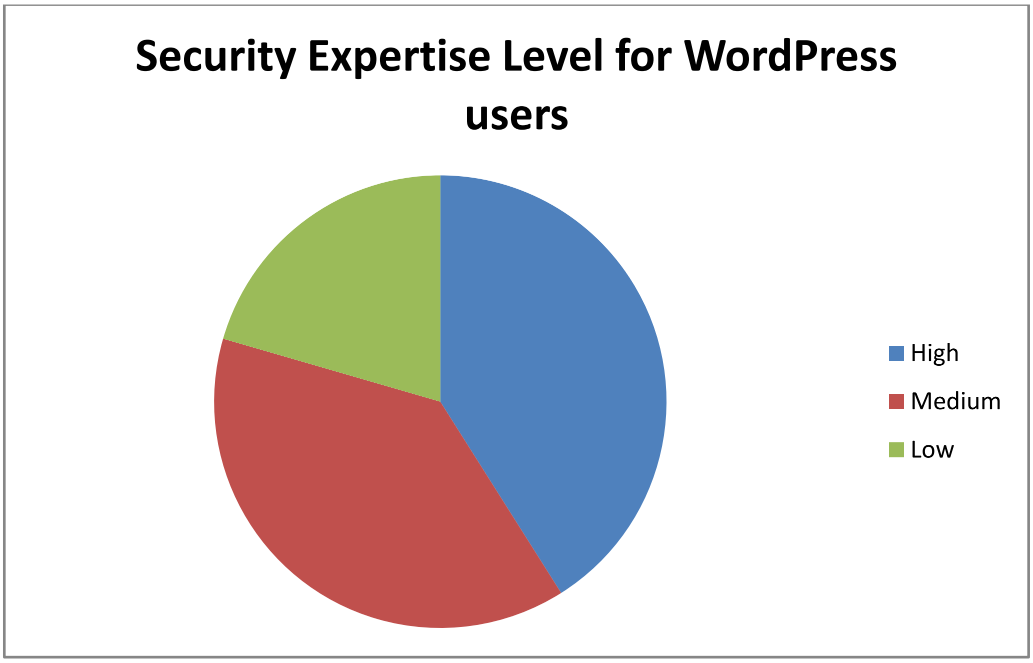
Graph 11 - Security Expertise Level/Security as primary job (Custom CMS)

6.3 Web CMS Hardening Techniques Results for WordPress

In this section, we are going to study the results of the 39 participants, (Chapter 5.4), who answered that they use the WordPress CMS. Moreover, we are going to observe their security expertise (*according to their answers*) (Graph 12) and analyze how their results about the hardening techniques of a website formed between the three categories we have been defined: *"No, I do not use that."*, *"Customly (via code)"* and *" Via Plugin/Extension/Module"* (Table 9).

In addition, in Graph 13, it appears approximately how many of the responders have answered *"Customly (via code)"* at least one time to each one of the pre-defined *Hardening Techniques* and compare these answers with the Security Expertise Level that they believe they have.

At last, we are going to compare, as before, the Security Expertise Level that they have declared with their Hands-on experience on security and with the fact that security is their Primary job responsibility.



Graph 12 - Security Expertise Level for WordPress users

	No, I do not use that.	Customly (via code)	Via Plugin/Extension/Module
Add Permanent Blocklist / Blacklisting - Whitelisting IPs (users)	8	14	16
Block spam delivers	3	8	27
Change Database Prefix (eg wp_admin, wp_login)	11	20	7
Change default passwords' hashing schema to a stronger one	13	17	8
Copy protection / GDPR Compliance	8	12	18
Delete unwanted themes, plugins/extensions/modules etc	5	20	13
Disable Directory Indexing and Browsing	8	22	8
Disable users enumeration (url permalink / users?=1)	16	11	11
Disable XML_RPC	18	14	6
Harden HTTP security headers	9	18	11
Hide the CMS and the version you are using	17	13	8
Install only trusted themes, plugins/extensions/modules etc	2	16	20
Keep everything up-to-date (CMS core code, themes, plugins/extensions/modules etc)	1	14	23
Keep regular backups	0	15	23
Limit Login Attempts	7	12	19
Limit the number of simultaneous sessions per user	22	6	10

Log out Idle users automatically	15	9	14
Monitor your site (audit logs, failed login attempts, incoming attacks, file change detection etc)	5	13	20
Relocate or rename login page	17	14	7
Scan for malwares	3	10	25
Scan for vulnerabilities	3	12	23
Secure administrator login (e.g. use of CAPTCHA, security question to login screen)	13	10	15
Secure the wp-config file	10	21	7
Strengthen password (e.g. use of password policies)	3	17	19
Use appropriate file permissions	2	27	10
Use CSRF tokens	17	15	7
Use two-factor authentication (2FA)	20	4	15
Use Web Application Firewall (WAF)	8	11	20

Table 9 - Web CMS Hardening Techniques for WordPress users

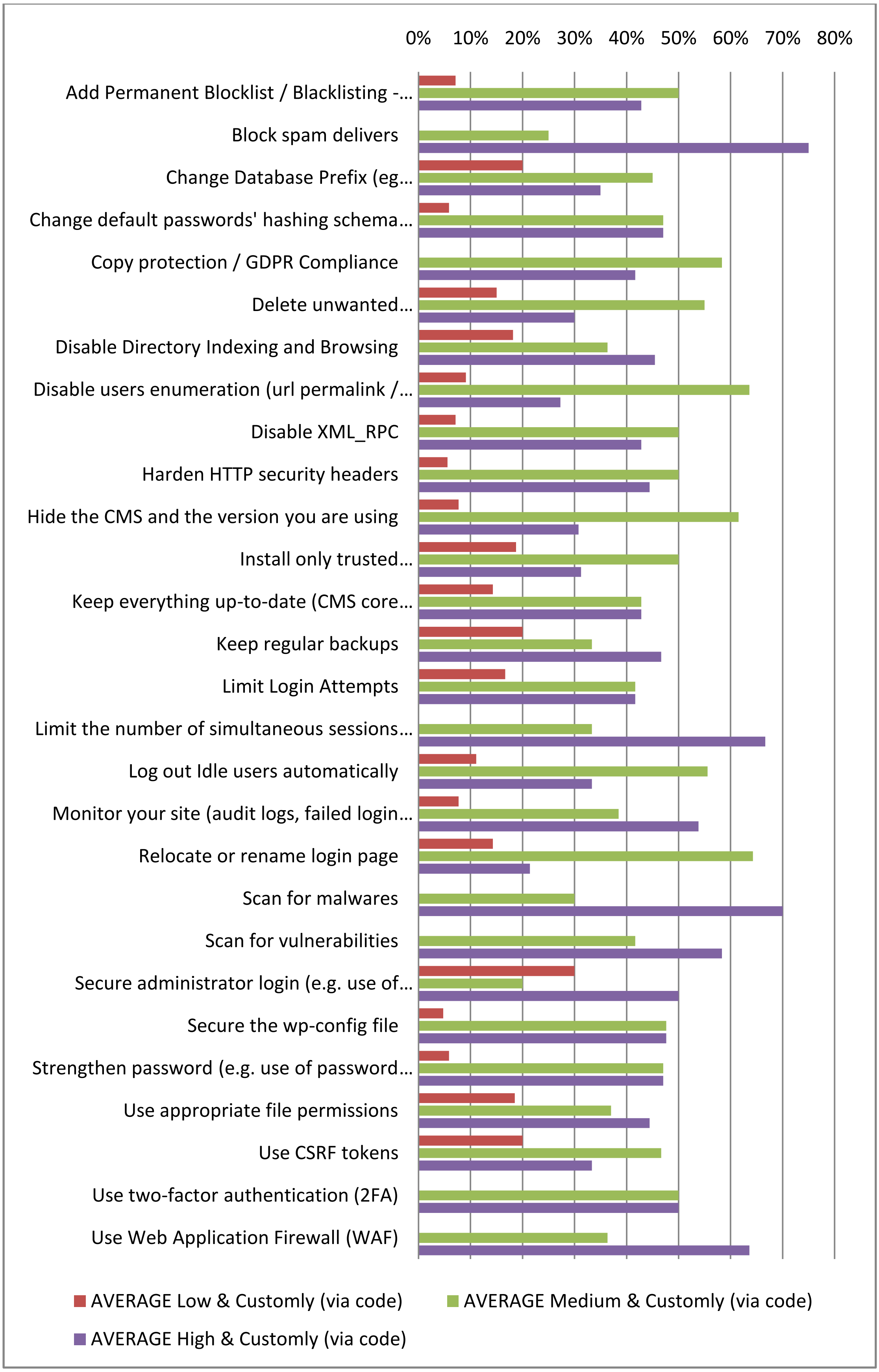
In Graph 13, we are presenting an average percentage of the participants who answered they used the WordPress platform and *"Customly (via code)"* at least one time to each one of the pre-defined Hardening Techniques compared with the Security Expertise Level that they believe they have.

As we can observe, WordPress users with Low Security Expertise Level they choose customly to Monitor their sites (audit logs, failed login attempts, et al.) and Limit login attempts, Block spam delivers and Hide the CMS and the version that they are using. However, there are plenty techniques which they avoid doing customly, such as to Strengthen their passwords, Keep regular backups, Use appropriate file permissions or even be GDPR compliant.

As for the Medium Level security users of WordPress, they prefer to customly Install only trusted themes and plugins, Disable users enumeration and Use two-factor authentication. Nonetheless, they are not font of customly Monitoring their sites and Strengthen passwords, -on the contrary to the Low Security Expertise users.

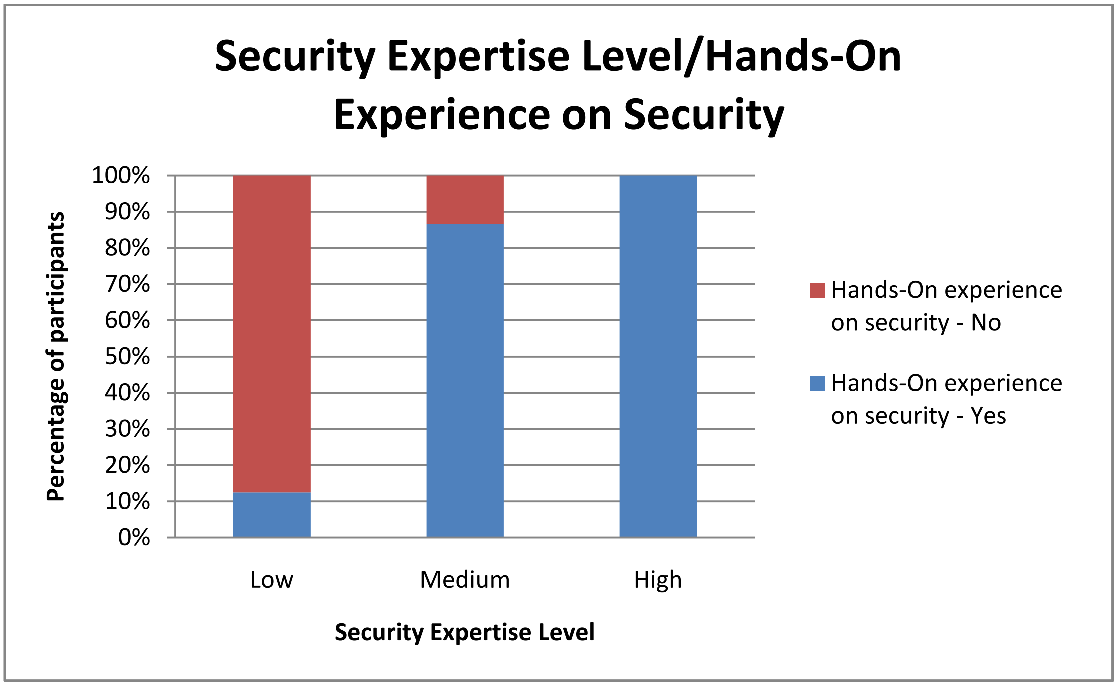
At the end, we have the High Security Expertise level WordPress users who have as their priority to Strengthen passwords customly, Keep regular backups and Use appropriate file permissions. Although, they do not prefer this much the custom way to Install the trusted themes and plugins or disable users enumeration.

Through these results we can understand that the more High leveled you are on the security field the more crucial are the hardening techniques you choose to take care customly.

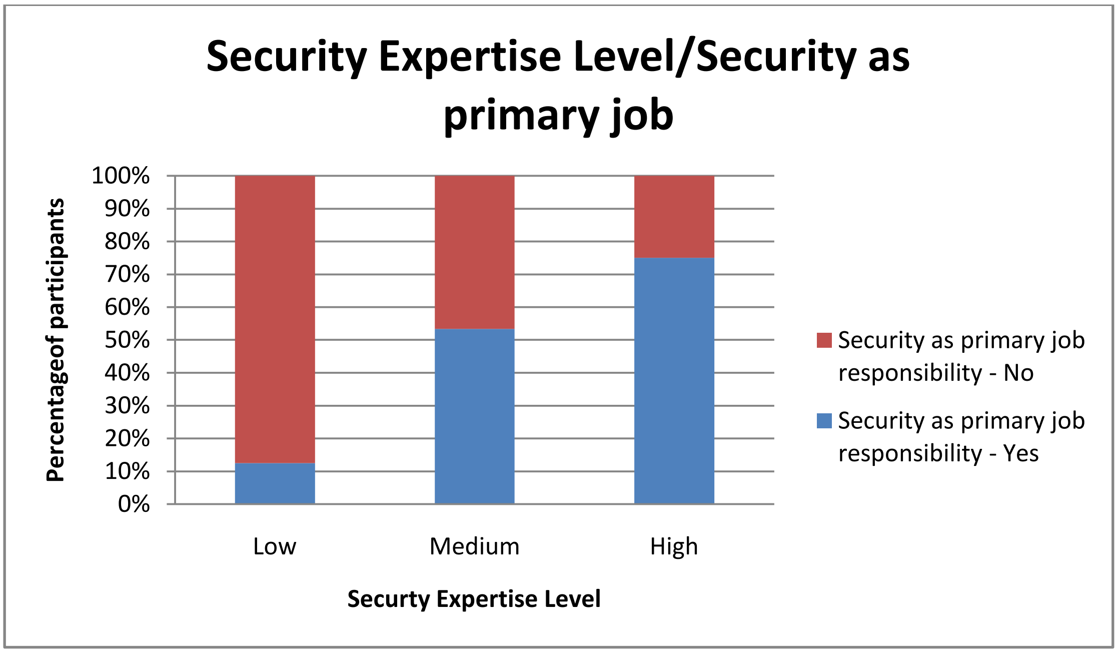


Graph 13 - Customly Hardening Techniques based on Security Expertise for WordPress users

As described above, here we can observe the correlation between the Security Expertise Level that our participants answered with their actual hands-on experience on the security field and the fact that if the security is their primary job responsibility.



Graph 14 - Security Expertise Level/Hands-On Experience on Security (WordPress)



Graph 15 - Security Expertise Level/Security as primary job (WordPress)

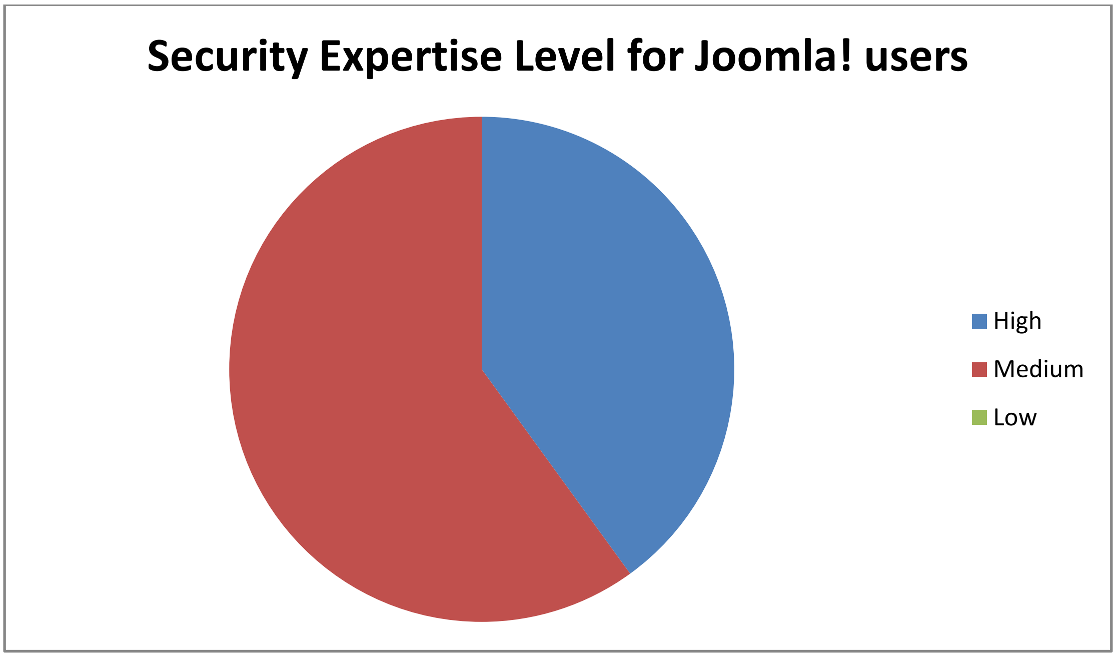
6.4 Web CMS Hardening Techniques Results Joomla!

In this section, we are going to observe the results of the 5 participants, (Chapter 5.4), who answered that they use the Joomla! CMS. It is important to note that our sample is small, so our results may have a deviation from the real scenarios.

In Graph 16, we can observe that the security expertise (*according to our participants' answers*) is limited between the *Low* and the *Medium*. Table 10 summaries the allocation of the responses about the hardening techniques of a website between the three categories we have been defined: *"No, I do not use that."*, *"Customly (via code)"* and *" Via Plugin/Extension/Module"* (Table 10).

More than that, we are going to introduce a graph presenting approximately how many of the responders have answered *"Customly (via code)"* at least one time to each one of the pre-defined *Hardening Techniques* and compare these answers with the Security Expertise Level that they believe they have.

Eventually, we are going to overall compare the Security Expertise Level that they have declared with their Hands-on experience on security and with the fact that security is their Primary job responsibility.



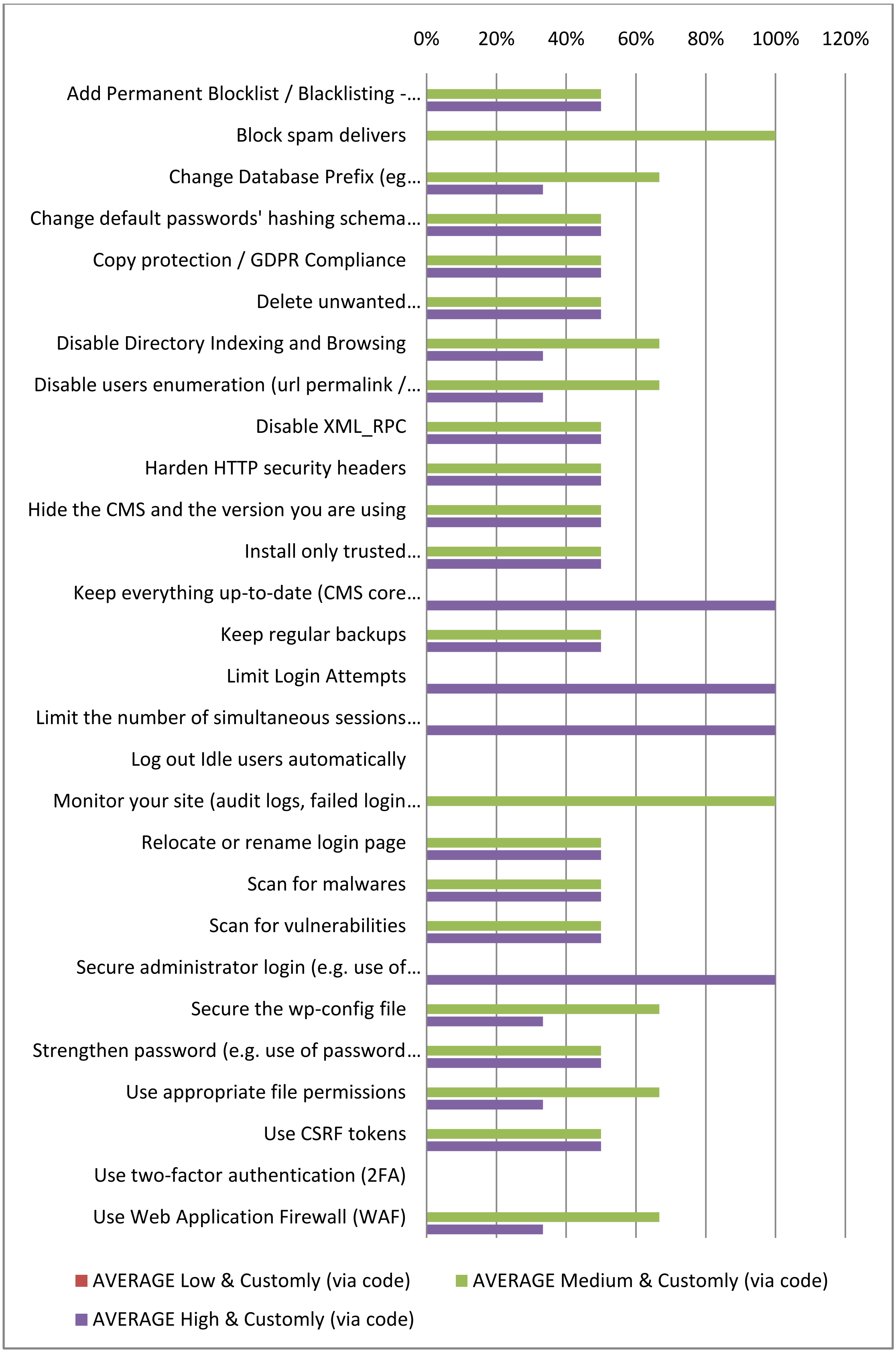
Graph 16 - Security Expertise Level for Joomla! users

	No, I do not use that.	Customly (via code)	Via Plugin/Extension/Module
Add Permanent Blocklist / Blacklisting - Whitelisting IPs (users)	1	2	2
Block spam delivers	0	1	4
Change Database Prefix (eg wp_admin, wp_login)	0	3	2
Change default passwords' hashing schema to a stronger one	0	2	3
Copy protection / GDPR Compliance	1	2	2
Delete unwanted themes, plugins/extensions/modules etc	0	4	1
Disable Directory Indexing and Browsing	1	3	1
Disable users enumeration (url permalink / users?=1)	1	3	1
Disable XML_RPC	2	2	1
Harden HTTP security headers	1	4	0
Hide the CMS and the version you are using	1	2	2
Install only trusted themes, plugins/extensions/modules etc	0	2	3
Keep everything up-to-date (CMS core code, themes, plugins/extensions/modules etc)	0	1	4
Keep regular backups	0	2	3
Limit Login Attempts	2	1	2
Limit the number of simultaneous sessions per user	1	1	3

Log out Idle users automatically	2	0	3
Monitor your site (audit logs, failed login attempts, incoming attacks, file change detection etc)	0	1	4
Relocate or rename login page	2	2	1
Scan for malwares	0	2	3
Scan for vulnerabilities	0	2	3
Secure administrator login (e.g. use of CAPTCHA, security question to login screen)	1	1	3
Secure the wp-config file	1	3	1
Strengthen password (e.g. use of password policies)	0	2	3
Use appropriate file permissions	0	3	2
Use CSRF tokens	2	2	1
Use two-factor authentication (2FA)	2	0	3
Use Web Application Firewall (WAF)	0	3	2

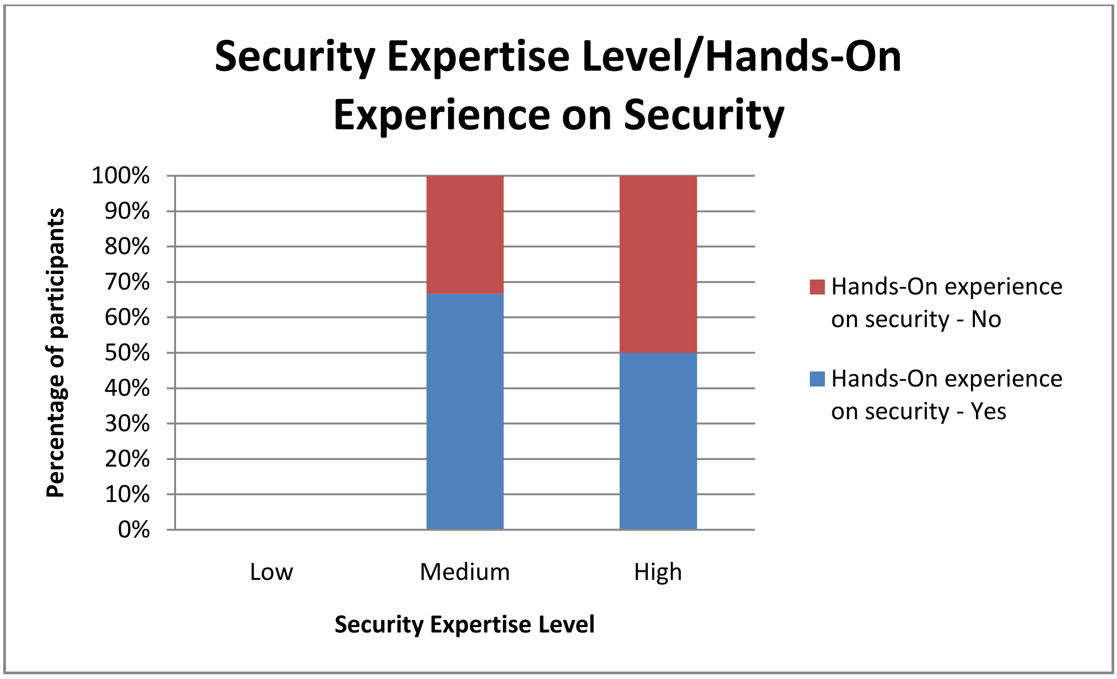
Table 10 - Web CMS Hardening Techniques for Joomla! users

In Graph 17 we can observe the dispersion of the *"Customly (via code)"* response on each of the pre-defined Hardening Techniques which is compared with the Security Expertise Level that our participants believe they have.

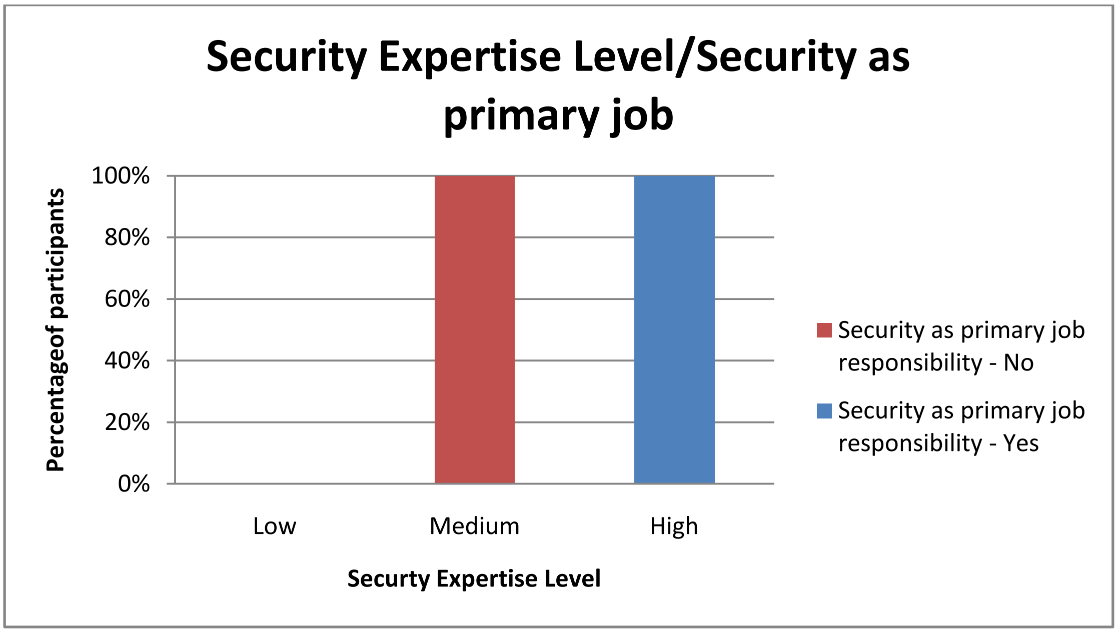


Graph 17 - Customly Hardening Techniques based on Security Expertise for Joomla! users

It follows the correlation between the Security Expertise Level that our respondents answered with their actual hands-on experience on the security field and the fact that if the security is their primary job responsibility.



Graph 18 - Security Expertise Level/Hands-On Experience on Security (Joomla!)

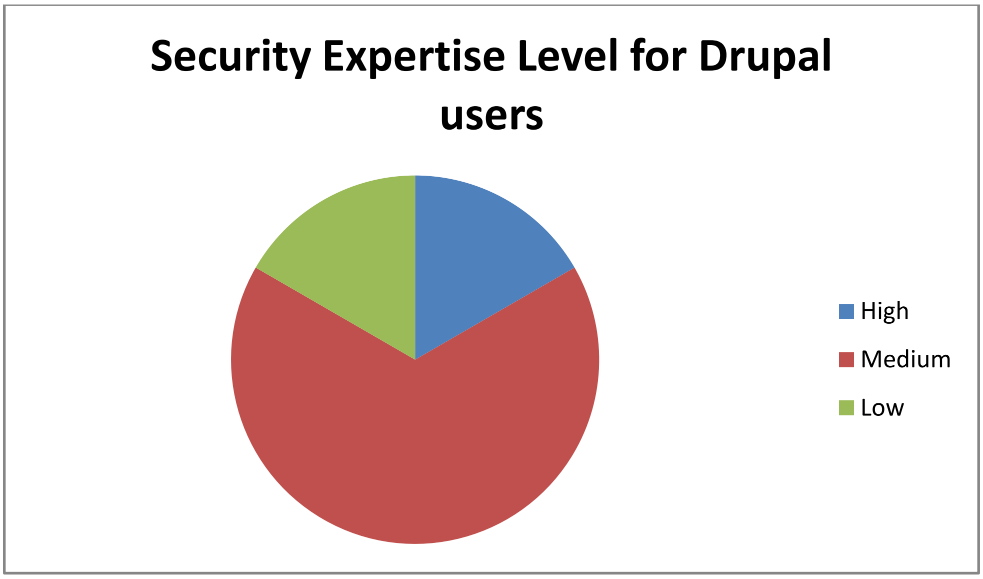


Graph 19 - Security Expertise Level/Security as primary job (Joomla!)

6.5 Web CMS Hardening Techniques Results for Drupal

Similarly to the previous section, our Drupal sample is not that efficient, so our results may have a deviation from the real scenarios, too. We are going to study the results of the 6 participants, (Chapter 5.4), who answered that they use the Drupal CMS and observe their security expertise (*according to their answers*) (Graph 20). Then, we will analyze how their results about the hardening techniques of a website formed between the three categories we have been defined: *"No, I do not use that."*, *"Customly (via code)"* and *" Via Plugin/Extension/Module"* (Table 11).

On top of that, Graph 21 presents the average of responders have answered *"Customly (via code)"* at least one time to at least one of the pre-defined *Hardening Techniques* and compare these answers with the Security Expertise Level that they believe they have.



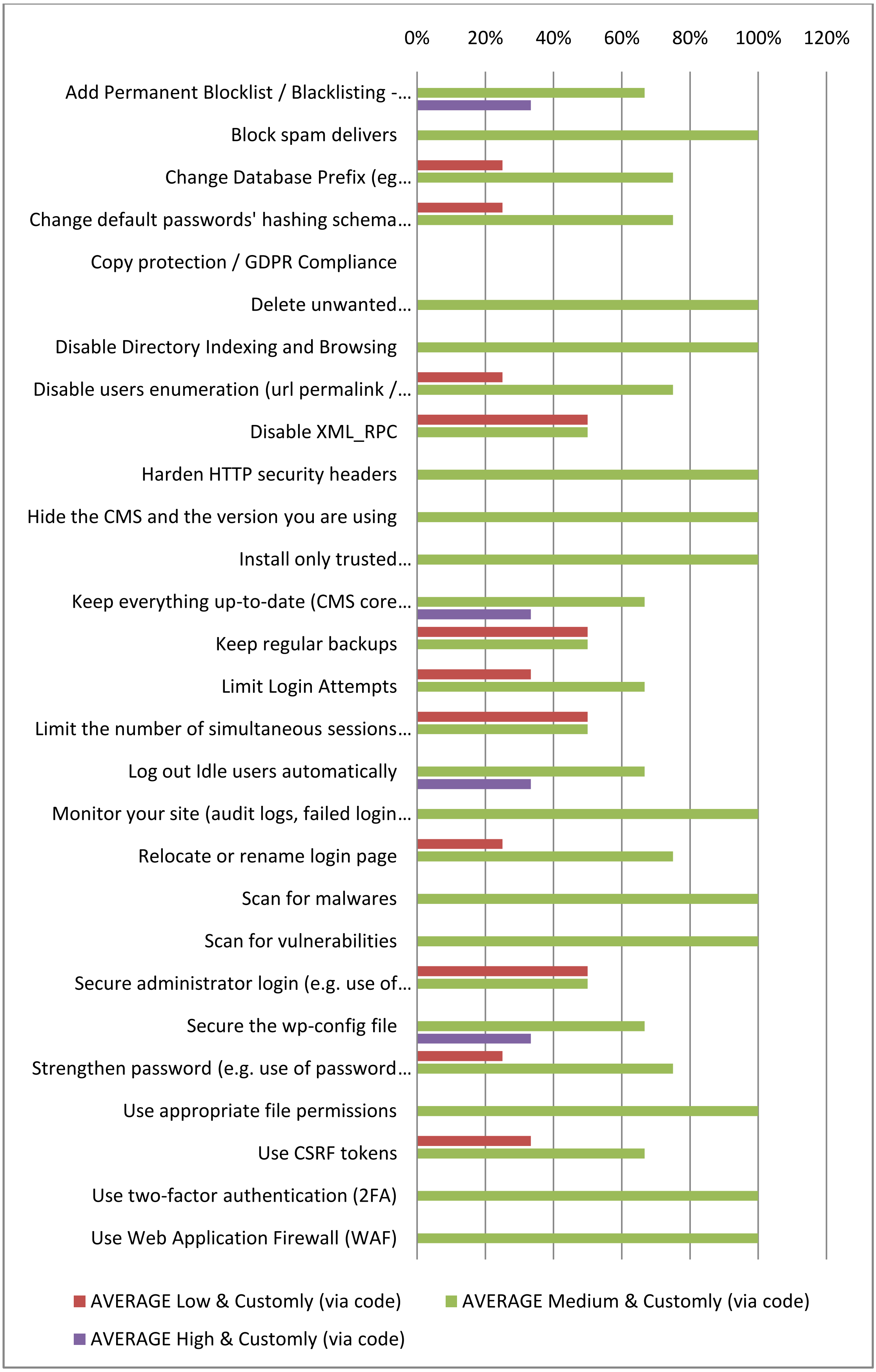
Graph 20 - Security Expertise Level for Drupal users

	No, I do not use that.	Customly (via code)	Via Plugin/Extension/Module
Add Permanent Blocklist / Blacklisting - Whitelisting IPs (users)	0	3	3
Block spam delivers	0	1	5

Change Database Prefix (eg wp_admin, wp_login)	1	4	1
Change default passwords' hashing schema to a stronger one	1	4	1
Copy protection / GDPR Compliance	2	0	4
Delete unwanted themes, plugins/extensions/modules etc	2	2	2
Disable Directory Indexing and Browsing	2	2	2
Disable users enumeration (url permalink / users?=1)	0	4	2
Disable XML_RPC	3	2	1
Harden HTTP security headers	1	2	3
Hide the CMS and the version you are using	1	3	2
Install only trusted themes, plugins/extensions/modules etc	0	1	5
Keep everything up-to-date (CMS core code, themes, plugins/extensions/modules etc)	0	3	3
Keep regular backups	0	2	4
Limit Login Attempts	1	3	2
Limit the number of simultaneous sessions per user	3	2	1
Log out Idle users automatically	0	3	3
Monitor your site (audit logs, failed login attempts, incoming attacks, file change detection etc)	0	2	4

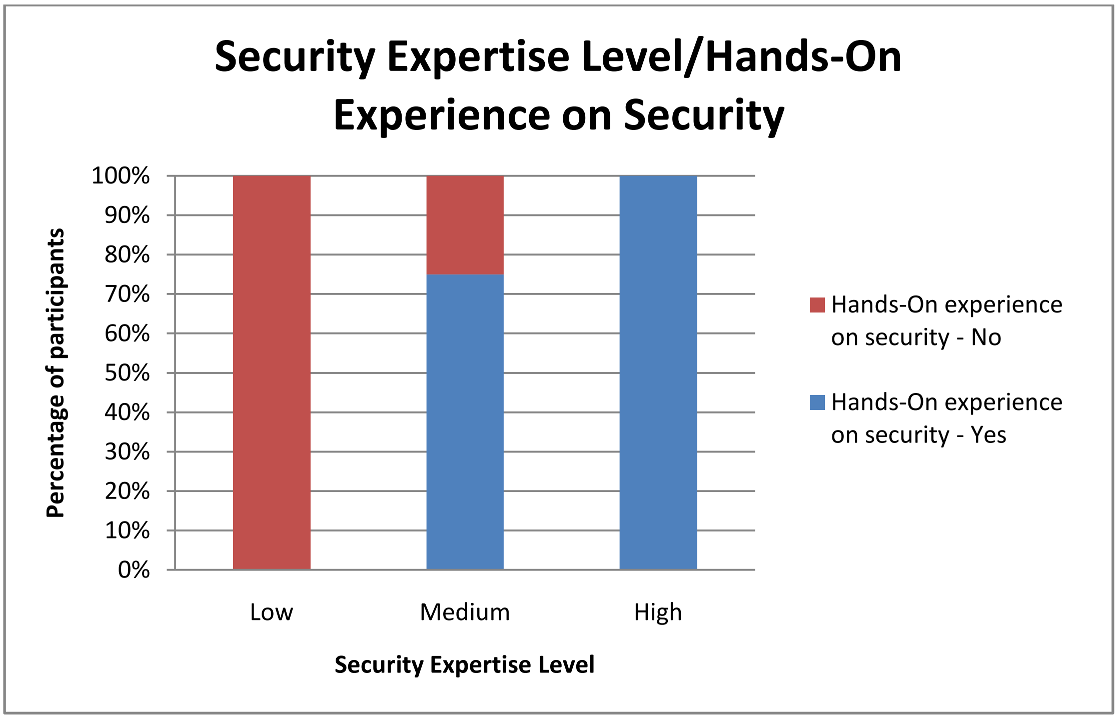
Relocate or rename login page	1	4	1
Scan for malwares	1	1	4
Scan for vulnerabilities	1	1	4
Secure administrator login (e.g. use of CAPTCHA, security question to login screen)	0	2	4
Secure the wp-config file	2	3	1
Strengthen password (e.g. use of password policies)	0	4	2
Use appropriate file permissions	0	2	4
Use CSRF tokens	2	3	1
Use two-factor authentication (2FA)	3	1	2
Use Web Application Firewall (WAF)	1	1	4

Table 11 - Web CMS Hardening Techniques for Drupal users

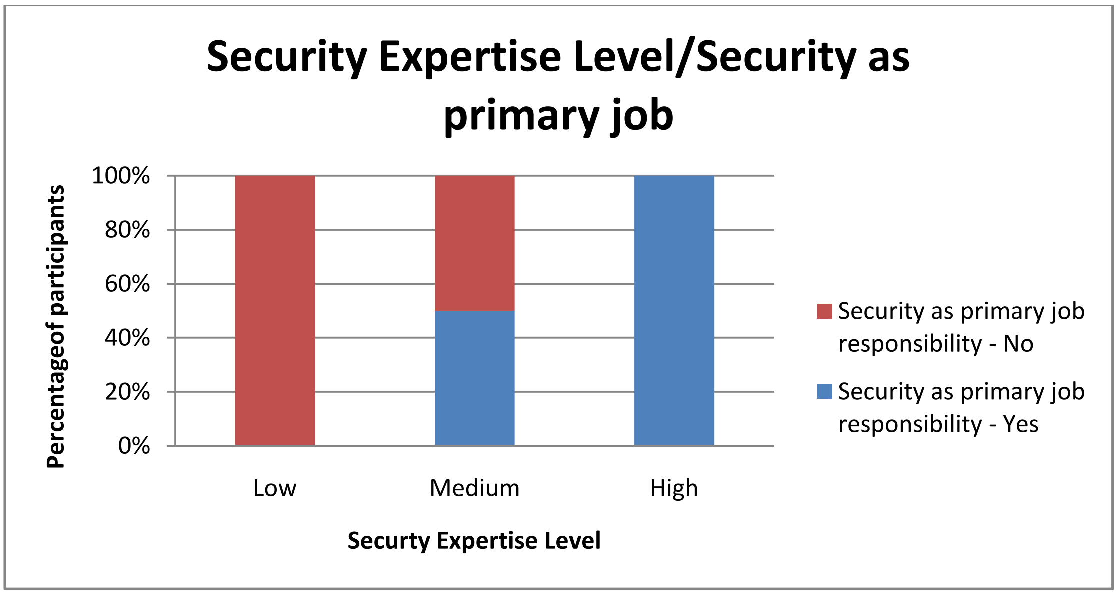


Graph 21 - Customly Hardening Techniques based on Security Expertise for Drupal users

Last yet important, in Graph 22 and Graph 23 you can observe the correlation between the Security Expertise Level that our participants answered with their actual hands-on experience on the security field and the fact that if the security is their primary job responsibility.



Graph 22 - Security Expertise Level/Hands-On Experience on Security (Drupal)

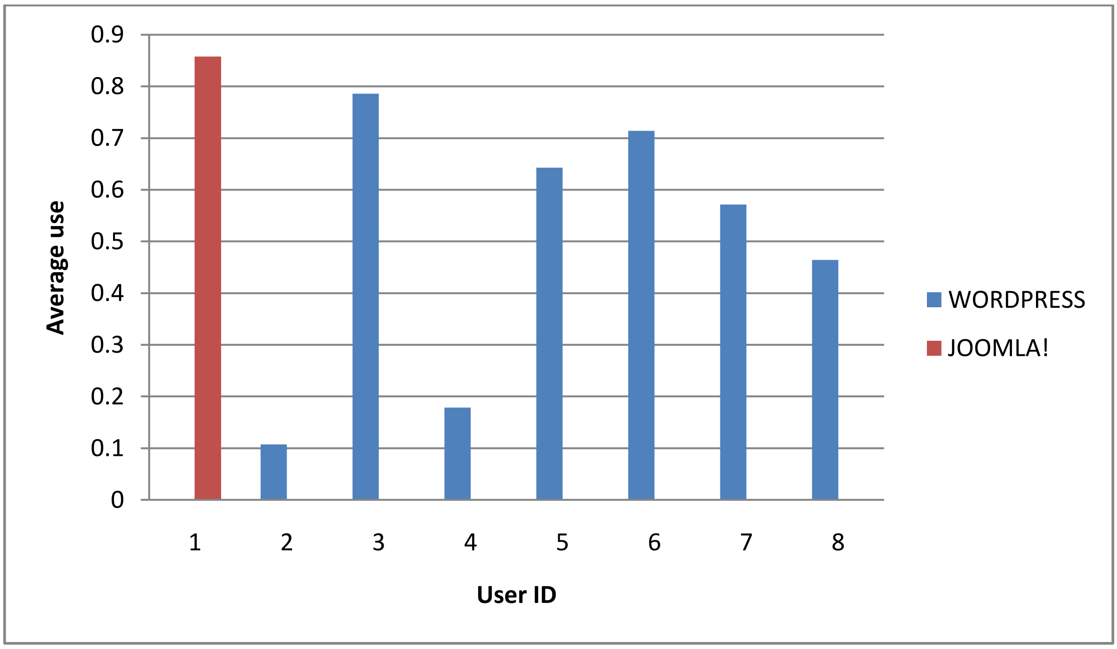


Graph 23 - Security Expertise Level/Security as primary job (Drupal)

6.6 Correlation between Customly Hardening Techniques and Usage of custom plugins/extensions/modules

In this part, we study only the participants that answered something different (*filled in the "Other" field*) and did not choose one of the proposed plugins, extensions or modules, when they asked how they protect their websites. There was a variety of answers and only 5 respondents (4 WordPress users and 1 Joomla! user).

In Graph 24, we can see for each one of these participants an average percentage of *"Customly (via code)"* answer given to Web CMS Hardening Techniques section.



Graph 24 - Average percentage of "Customly (via code)" given

Thus, we can understand that especially the Joomla! user (who according to the results has a High Security Expertise Level and Security is his primary job) answered to almost every hardening technique "Customly"! This leads us to the conclusion, that more users like him prefer to secure their sites their own way, but they prefer to build them under a well-known platform, like WordPress or Joomla!.

7 Conclusion

This paper was about a questionnaire created for developers who are using CMS platforms to built their websites. The main goal was to find people with cyber security expertise or not and examine whether they are aware of the default security settings offered by WordPress, Joomla! and Drupal and if they perform any other security hardenings. The final conclusion is that yes, the majority of them takes the security of their websites seriously.

In general, our participants showed quite similar preferences as concerns the platforms they use. As an overall, we can highlight that developers building sites are mainly men between 18-30 years old and pretty educated (with a Master or Bachelor degree). Their cyber security expertise level is around medium with hands-on experience. Furthermore, even though the fact that the majority of them is using the WordPress CMS they have the tend to use plugins, such as the Wordfence or All-In-One, in order to secure their sites in a very "basic" way, meaning to scan for vulnerabilities & malwares, update themes, change password policies, monitor their sites et al. For the security fields where more attention to detail is needed, they prefer to use their IT-related knowledge and write their own code. Last but important, developers should take more into consideration that the 2-factor authentication is a very serious hardening measure, especially when these sites are going to pass on to simple users!

In conclusion, with this survey we presented the whole idea behind the use of CMSs and their security plugins/extensions/modules. The developers are not ready yet to trust such automated tools for their security and our next step is to research if this hesitation is originated from their non trust to the authors of these tools and plugins themselves or from the fact that they cannot find notable extensions to place their hopes up, so they customly create their own tools based on their unique needs.

8 References

1. ip.gr. *Τι είναι τα CMS (π.χ. Joomla, Wordpress, Drupal)?* [Online] https://www.ip.gr/Web_Development/%CE%A4%CE%B9_%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9_%CF%84%CE%B1_CMS_(%CF%80.%CF%87._Joomla_Wordpress_Drupal)-246.html.

2. Wikipedia. *en.wikipedia.org.* [Online] June 8, 2019. https://en.wikipedia.org/wiki/Content_management_system.

3. Web site Setup. *websitesetup.org.* [Online] 12 16, 2018. https://websitesetup.org/popular-cms/.

4. Wikipedia. *en.m.wikipedia.org.* [Online] https://en.m.wikipedia.org/wiki/WordPress.

5. **Sara Rosso.** Wordpress. *wordpress.org.* [Online] 03 2015. https://wordpress.org/about/security/.

6. W3Techs-Web Technology Surveys. *w3techs.com.* [Online] https://w3techs.com/technologies/details/cm-joomla/all/all.

7. Hosting Tribunal. *hostingtribunal.com.* [Online] https://hostingtribunal.com/blog/joomla-statistics/.

8. Wikipedia. *en.m.wikipedia.org.* [Online] https://en.m.wikipedia.org/wiki/Joomla!.

9. Joomla! Docs. *docs.joomla.org.* [Online] 12 08, 2016. https://docs.joomla.org/Security_Checklist/Joomla!_Setup.

10. Joomla Developers. *developer.joomla.org.* [Online] 01 30, 2017. https://developer.joomla.org/security.html.

11. Wikipedia. *en.m.wikipedia.org.* [Online] https://en.m.wikipedia.org/wiki/Drupal.

12. Drupal. *www.drupal.org.* [Online] 09 02, 2016. https://www.drupal.org/node/2573525.

13. Code in WP. *www.codeinwp.com.* [Online] https://www.codeinwp.com/blog/secure-your-wordpress-website/.

14. Drupal. *www.drupal.com.* [Online] https://www.drupal.com/feature/security.

15. Joompla Shack. *www.joomlashack.com.* [Online] 03 27, 2017. https://www.joomlashack.com/blog/joomla/17-point-checklist-to-harden-your-joomla-site-security/.

16. Trip Wire. *www.tripwire.com.* [Online] 05 07, 2018. https://www.tripwire.com/state-of-security/featured/8-tips-harden-joomla-installation/.

17. Site Ground. *www.siteground.com.* [Online] https://www.siteground.com/tutorials/joomla/security/.

18. Kinsta. *kinsta.com.* [Online] 05 20, 2019. https://kinsta.com/blog/wordpress-security-plugins/.

19. **Jamie Maguire.** DevTeam.Space. *www.devteam.space.* [Online] https://www.devteam.space/blog/drupal-security-how-to-protect-your-website/.

20. **Brian Jackson.** KeyCDN. *www.keycdn.com.* [Online] 01 23, 2018. https://www.keycdn.com/blog/drupal-security.

21. **Lars Lofgren.** Quicksprout. *www.quicksprout.com.* [Online] 02 13, 2019. https://www.quicksprout.com/best-wordpress-security-plugin/.

22. Sucuri. *sucuri.net.* [Online] https://sucuri.net/.

23. Wordpress Plugins. *wordpress.org/plugins.* [Online] https://wordpress.org/plugins/all-in-one-wp-security-and-firewall/.

24. Wordpress Plugins. *wordpress.org/plugins.* [Online] https://wordpress.org/plugins/bulletproof-security/.

25. iThemes. *ithemes.com.* [Online] https://ithemes.com/tutorials/getting-started-ithemes-security/.

26. **Souvik Banerjee.** RS Websols. *www.rswebsols.com.* [Online] 03 15, 2016. https://www.rswebsols.com/extensions/joomla-extensions/best-joomla-security-extensions.

27. Siteground. *www.siteground.com.* [Online] 2019. https://www.siteground.com/joomla-hosting/joomla-extensions/ver1.5/jhack.htm.

28. **RSJoomla!** Joomla Extensions. *extensions.joomla.org.* [Online] 05 21, 2019. https://extensions.joomla.org/extension/rsfirewall/.

29. **'corePHP'.** Joomla Extensions. *extensions.joomla.org.* [Online] 04 10, 2019. https://extensions.joomla.org/extension/jomdefender/.

30. **Texpaok.** Joomla Extensions. *extensions.joomla.org.* [Online] 05 13, 2019. https://extensions.joomla.org/extension/securitycheck/.

31. **Ana.** AgileDrop. *www.agiledrop.com.* [Online] 01 17, 2019. https://www.agiledrop.com/blog/best-drupal-8-security-modules.

32. **Adriana Cacoveanu .** Optasy Blog. *www.optasy.com/blog.* [Online] 06 04, 2018. https://www.optasy.com/blog/these-are-15-best-drupal-security-modules-worth-installing-your-website.

33. Drupal. *www.drupal.or.* [Online] 2019. https://www.drupal.org/project/paranoia.

34. Wikipedia. *en.m.wikipedia.org.* [Online] 06 2019. https://en.m.wikipedia.org/wiki/Joomla!.

35. Wikipedia. *en.m.wikipedia.org.* [Online] 06 2019. https://en.m.wikipedia.org/wiki/Drupal.

36. Wikipedia. *en.m.wikipedia.org.* [Online] 06 2019. https://en.m.wikipedia.org/wiki/WordPress.

37. WordPress. *wordpress.org.* [Online] https://wordpress.org/support/article/hardening-wordpress/.