Threat Analysis and Alert Management – 04-Jan-2021

Introduction

Client: A 30 employees law firm based in the capital of the city, that provided consulting and outsourced legal analysis for different companies.

After identifying the malware and recovering the contents of the web-page, it is required to monitor and secure the Wordpress environment to avoid future attacks and to determine the possible vector that was used to acquire access in the first place.

As such, the client stated the following **objective** to be met:

1. Provide all around security for Wordpress

The ideal tool for this is WordFence, due to the integrated nature with WordPress Core and being well documented, free of use[1] and fast to be deployed as a plugin. It also has auto-update features, blacklisting of IPs, usernames and a user interface that is clear and friendly.

Another option that we can implement if the previous is not successful the Sucuri WordPress plugin[2], as it has some excellent features regarding scanning, identifying scripts running and blacklisting status for IPs taken from reputable sources. It as very complete and practical suit.

1 The “premium” features have to be paid, but the basic ones have very interesting applications that are enough for the purposes of the client’s request.

2 The free API Key allows use of the Firewall too.

Threat Analysis and Alert Management

To actually start to protect our asset, we had to state a policy on how the permissions are managed for the client and the cybersecurity analyst.

User roles within the WordPress environment are managed according to certain privileges, which allows them to perform certain actions, similarly as how UNIX systems can work around having a lot of users but only a few of them with root access into the critical components of the system.

Following the philosophy of least privileges, it was accorded with the client that only 2 administrator users would be active at ANY moment, due to the permissions available to that role.[3] One would be a work account for the site-manager, that did maintenance on themes, plugins and other issues related to WordPress, and the other would be the account that the analyst would use to monitor the site and implement the security features with WordFence.
Other restricted users would be editors, there would only be one account for the moderator, who curated the contents of the posts, commentaries and all the related tasks.
Authors would be managed more freely by the client, as it depended on interns who were constantly rotating. As such, it was advised that an individualized account should be created for each user, and deactivated after they left the company. This was recommended for easier logging of each user's activities within WordPress, and if credentials were leaked, they could be more readily tracked.

This information was recommended to be kept only by the client-side, inside an encrypted volume. Veracrypt[4] was selected as the tool for the job, mainly for ease of implementation and previous use by the client.

Coming back to WordPress itself, the first element to be protected was the login form.

Figure 1 WordPress Login

This form can be enumerated in several ways, trying to find the "root", how many users are there, possible vulnerabilities, etc.[5]

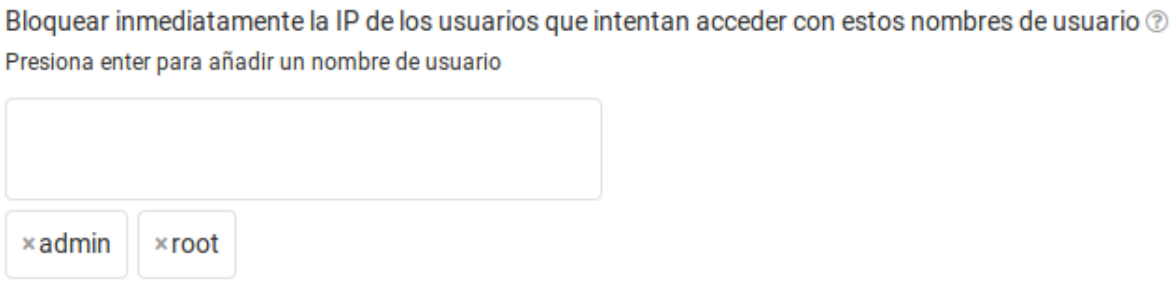
The vanilla version of WordPress also doesn't have any protection whatsoever to dictionary or brute force attacks, as such, we had to configure several rules in the previously mentioned Wordfence.

There is also another issue, that the "login" form most always has the same path, being www.example.com/wp-admin.php or www.example.com/wordpress/wp-admin.php as such, automated programs that take that path can easily start testing for usernames.

3 For more information regardingeach user's role and available permission, check the documentation: https://wordpress.org/support/article/roles-and-capabilities/
4 See: https://www.veracrypt.fr/en/Documentation.html
5 See: https://hackertarget.com/wordpress-user-enumeration/
Note: WPScan tool is ideal for this scenario, but due to legal limitations with the server, it was not possible to be deployed.

Due to this, the login domain was changed to something more specific, so it actually has to be enumerated before a dictionary attack has to be performed.
To actually protect from enumeration, the requests were throttled and the user that performed them, to be blocked.

Figure 2 Banned users



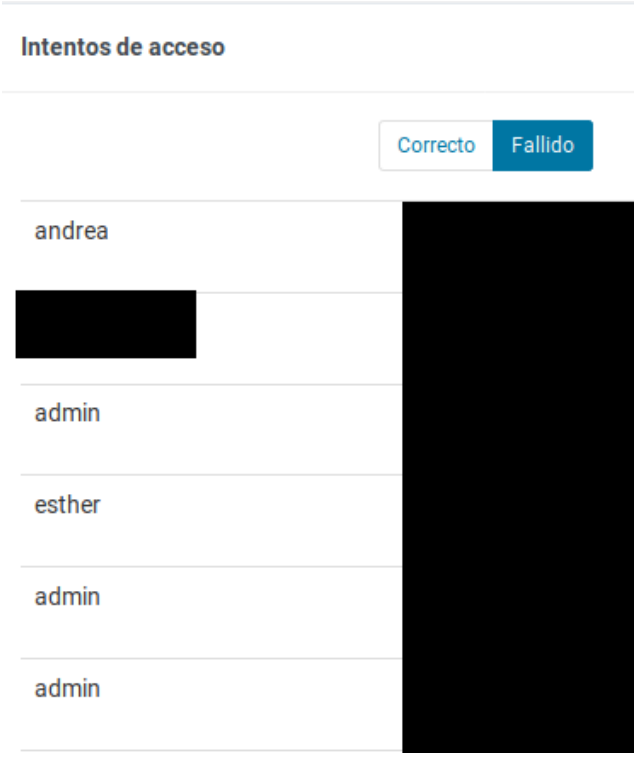
The user “admin” and “root” was banned, and if someone tried to login as such, the user would be instantly banned, as this was the most common used credentials for brute-forcing.

Figure 3 Latest attacks

IP	País	Recuento de bloqueos
[redacted]	Italia	53
	Países Bajos	44
	Estados Unidos	44
	Alemania	16
	Alemania	12
	Estados Unidos	12
	Alemania	11
	Alemania	8
	Estados Unidos	8
	Estados Unidos	7

As we can see in Figure 3, many attempts were frustrated in a month span with a simple plugin with the correct options defined.
In Figure 4, we can see some of users that were tried in failed access attempts, mainly taking names from interns and also the prevalent “admin”.

Figure 4 Users



It was also instructed to WordPress and all plugins, to automatically update themselves, to acquire security and stability fixes, in order to avoid any new exploits that may have appear publicly in the form of CVEs.[6]

The information mentioned is relayed to the cybersecurity analyst and the client in real time, as an E-Mail can be set in options to receive real time alerts.

These E-Mails can be later collated in more readable format (.csv) if required to do statistics or simple keep everyone appraised of the situation.

An interesting issue that had arisen while doing this report, was that a disgruntled employee that was part of the client's company[7] tried his own brute force attack testing, the usernames with the personal E-Mail account of ex-colleagues, supplying passwords that he had acquired, possibly through old breaches or hackerspaces.[8]

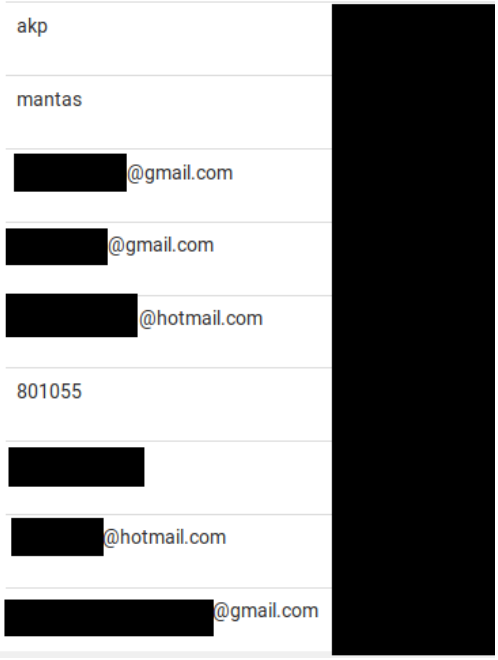
His level of sophistication was low, or perhaps was not expecting any defense at all, that is why took no measures to hide his true identity, sadly falling in the category of "lamer".

6 See: https://cve.mitre.org/

7 This was confirmed by the individual responsible for this after being confronted legally, since he tried to this from his home, with his public IP exposed.

8 The list of personal E-Mails (Hotmail and Gmail domains) were all mentioned in the #1 Collection Breach, confirmed by https://haveibeenpwned.com/ with its free checking service.

Figure 6 Disgruntled employee attack



With this last attack and favorable resolution, it was considered that, **objective 1** *"Provide all around security for Wordpress"*, was achieved.

Conclusion: A zero cost firewall can be deployed and administrated within WordPress, providing protection against enumeration, brute force attacks and ensuring no new hacking attempts would be successful.