Are we sure that WordPress is secure?

WordPress Meetup 13/06/2017

Nicola Laserra

Who am I?

- PHP Developer
- Quite a beginner on WordPress Meetup
- Working for MotorK till September

Where to start? What am I going to talk about?

One question from a previous meetup inspired me.

“What are the risks related to choosing WordPress as the platform of our IT solutions? Is WordPress a good choice in terms of security?”

OK, let’s try to dive deeper...

So what?

- Let’s talk about how WordPress tackles security issues
- Shed a light on the darkness of bad news about WordPress platform weaknesses

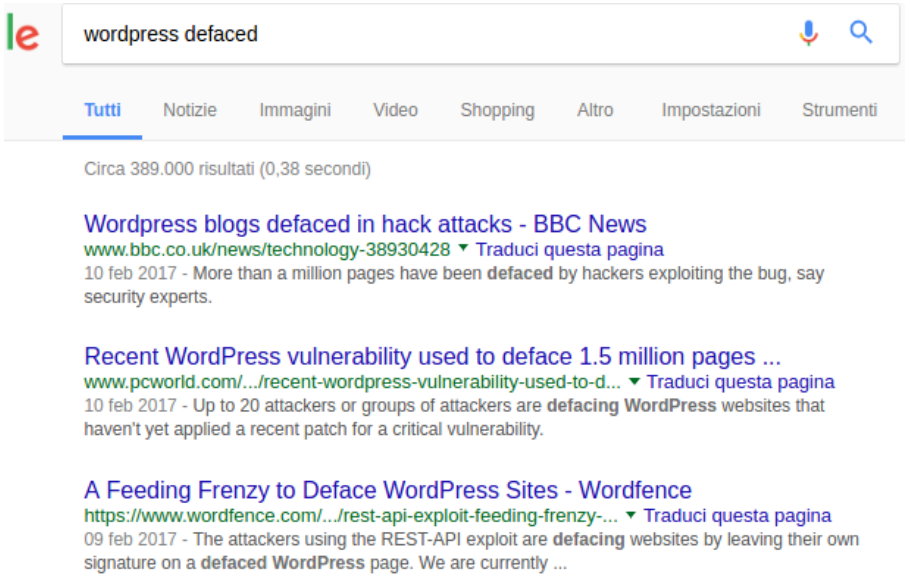
What I’m not going to talk about?

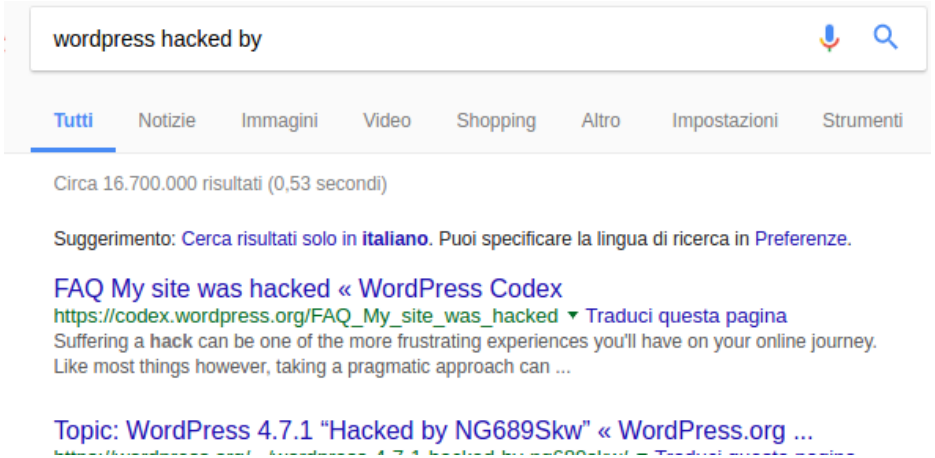
- I don’t want to talk about guidelines to follow for secure WordPress installations
- A bunch of information regarding all the secrets to know in order to have a pretty safe WordPress site
- Fist of all this one

  https://codex.wordpress.org/Hardening_WordPress

A little bit of googling about WP weaknesses

- A lot of bad news regarding WordPress just googling around!





Other results on Google about WP weaknesses

- “REVOLUTION SLIDER WORDPRESS PLUGIN IS POSSIBLE CAUSE OF PANAMA PAPERS LEAK”

  The Panama Papers leak is the largest in the history. It published a whopping 2.6 terabytes of data scored by 11 million documents. Previous leaks were in gigabytes scale, for example 230 GB (Sony Pictures) and 30 GB (Ashley Madison).

  Reasons?

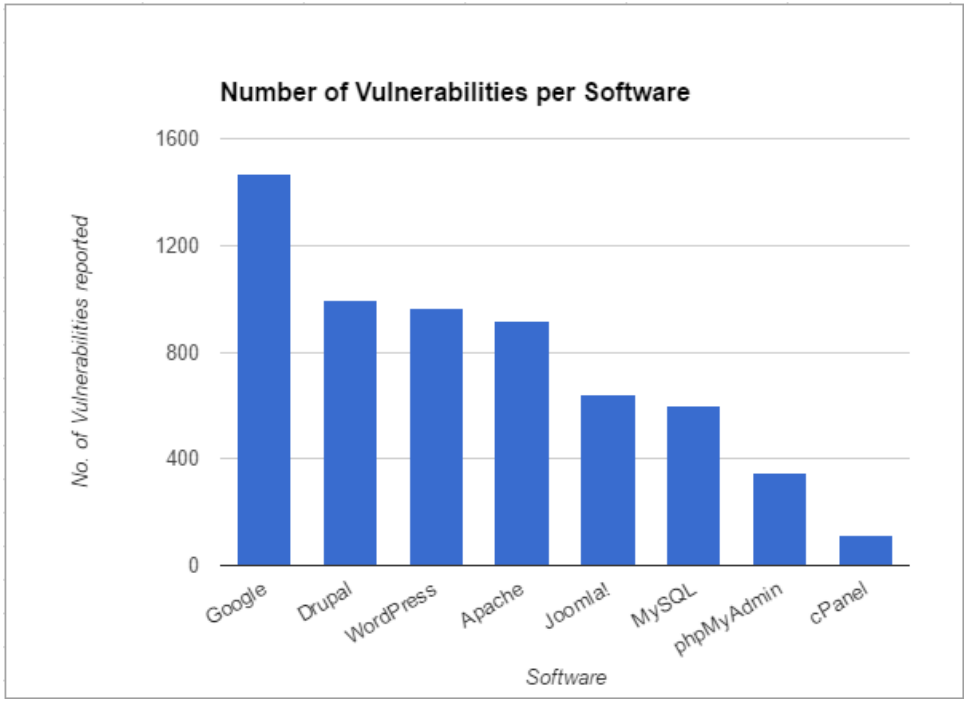
  Outdated CMSes (WordPress and Drupal), backdated plugin (RevSlider 2.1.7) and unencrypted & old email server!

Keep calm

Don’t panic, try to collect as much (detailed) information as you can regarding that problem.

What is there outside?

- There is no software with no vulnerability! Every software comes with its own problems. The more a product is commonly and widely used, the more it will be exposed to attacks.
- WordPress is not the only (potentially) vulnerable platform!
- Have a look at a report (CVE vulnerability data):



WordPress vulnerability trends

Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2004	2						1		1						
2005	10		5			3	2				3				
2006	16	1	2			1	5	1			3				
2007	40	2	13			7	19			3	5		2		3
2008	27	2	4			3	9	4		1	2		2		8
2009	14	3	1				3			1	3	1			4
2010	2		1			1									
2011	11					1	2				4				
2012	24	2	2			2	9			5	3		3		6
2013	19	1	1				8			3	2		1		
2014	29	3	3			1	8	1		6	2		3	1	
2015	11	1	2			1	7			1	1		1		
2016	20	1					9			6	1		1		
2017	29	1	1			1	9	2		4	2		5		
Total	254	17	35			21	91	8	1	30	31	1	18	1	21
% Of All		6.7	13.8	0.0	0.0	8.3	35.8	3.1	0.4	11.8	12.2	0.4	7.1	0.4	

Warning : Vulnerabilities with publish dates before 1999 are not included in this table and chart. (Because there are not many of them and they make the page look bad; and they may not be actually published in those years.)

Other cms vulnerability trends

Vulnerability Trends Over Time

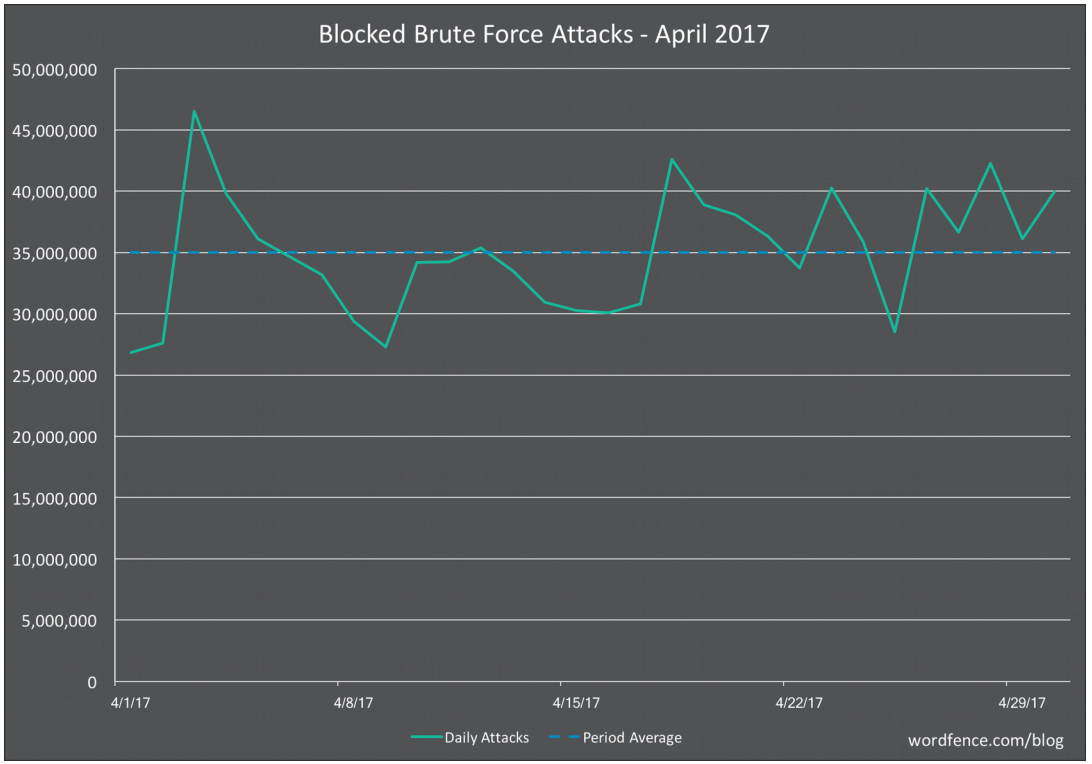
Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2002	1						1								
2005	6		1				2			1		1			
2006	37		7			6	20			1	2	2	1		
2007	36	1	6			2	13	1	1	4	2	1	4		1
2008	75		13			7	32			8	3	4	9		
2009	52		7			7	29			3	1	1	6		
2010	8						3			5					
2011	3						1			1	1				
2012	13	1	2			1	3				3		1		2
2013	14	2	2				2			2	3		2		
2014	35	4				1	22			1	2		1		4
2015	10		1			1	2				2		1		
2016	19	1	1				1		1	4	3	2			
2017	4		1							2			1		
Total	313	9	41			25	131	1	2	32	22	11	26		7
% Of All		2.9	13.1	0.0	0.0	8.0	41.9	0.3	0.6	10.2	7.0	3.5	8.3	0.0	

Not all vulnerabilities are created equal

- The number of security holes is not relevant by itself
- A bunch of minor weaknesses can be outweighted by just one critical vulnerability
- Common Vulnerability Scoring System (CVVS) provides a way to classify a given vulnerability by the means of its severity
- 3 factors: existence, access to the flaw, difficult in exploiting the weakness

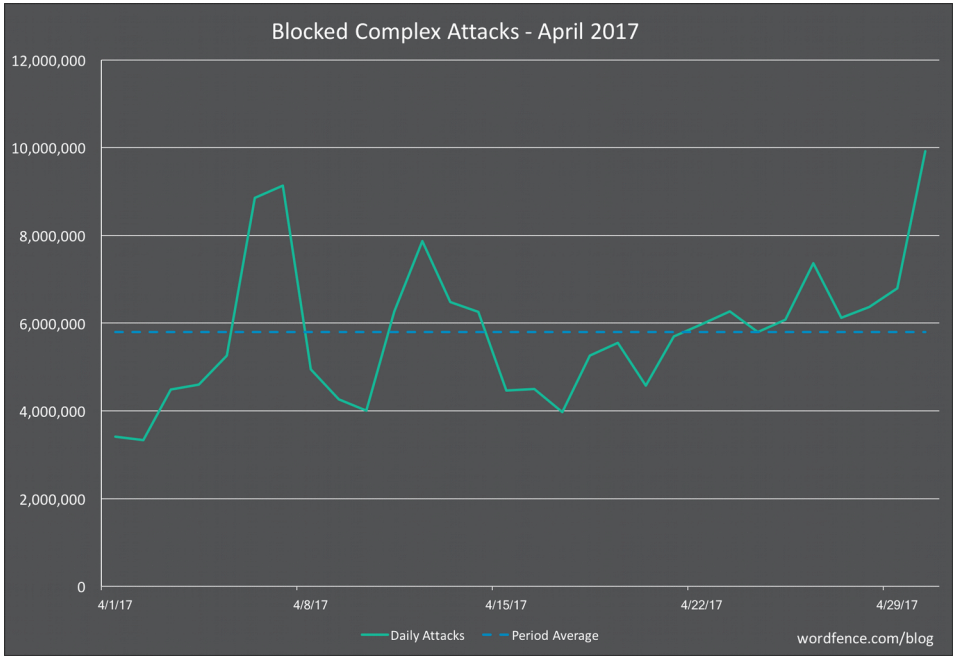
WordPress Attacks report (1)

Brute force attacks per day from March to April 2017 (source Wordfence.com)



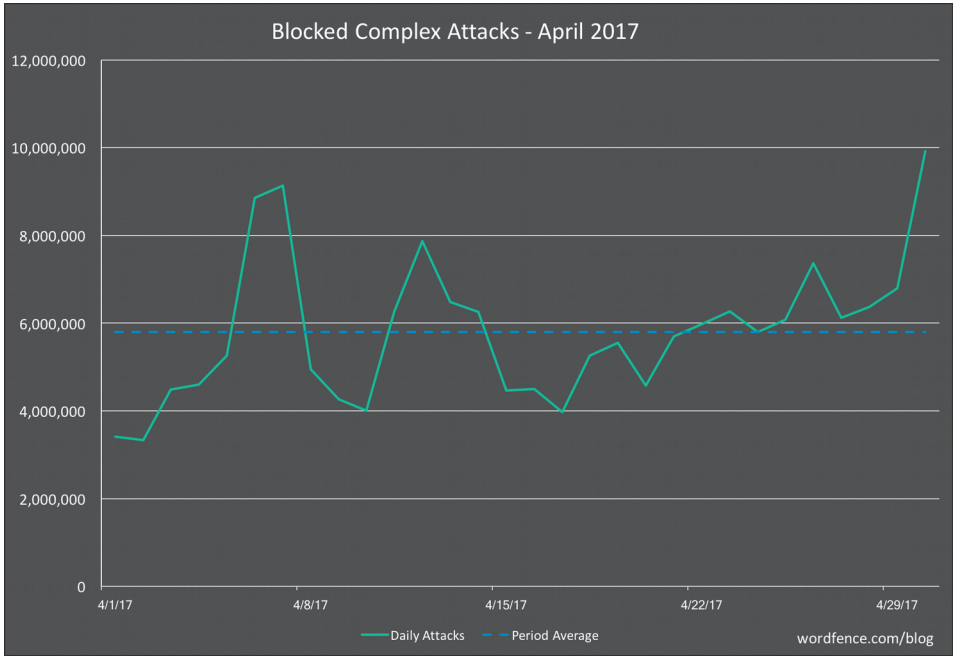
WordPress Attacks report (2)

Complex attacks - that exploits vulnerabilities - on April 2017 (source Wordfence.com)



WordPress Attacks report (2)

Complex attacks (that exploits vulnerabilities) on April 2017



The WordPress formula

Core (Automattic) + Plugins&Themes (from WP Library) + Plugins&Themes (from external Library) + “Magic” ingredient = WP platform

Wanna now what is the Magic ingredient?

Hey, wait…that is the end of the story!

The WordPress core

- A special core development team is the main actor here.
- A few Lead developer + some core developers with permanent commit rights + a variety of core commitiers
- But...anyone can join and start to be a contrinutor!

  https://make.wordpress.org/core

WordPress core – Security & plugins team

- Responsible for tackling security issues on core (both on wordpress.com and self-hosted installations)
- First barriers for security issues on WP plugins on the WP plugin repository
- Plugins and Theme reviewers are responsible for verifyng the quality of plugins and themes on the official archive. The goal is to decine standard and to verify the respect of the guidelines

The responsible disclousure principle

- Responsibly and privately disclose to the vendor the vulnerability
- The goals are
  - ✔ to avoid to publicly report the issue before it has been investigated and verified
  - ✔ to prepare a fix as soon as possible
  - ✔ to minimize the damages (if it is not publicly known, likely it is not used)

Case history

- WordPress security issue on REST API on version 4.7.0 & 4.7.1
- Originally discovered by a Sucuri researcher
- Reported by sucuri to wp and secretly fix on January 26th, witihin the new release 4.7.2
- A bunch of sites put in secure by this new release
- The security fix was kept secret for a week, after the publishing of the new release 4.7.2. The fix for this security issue was hidden within other issues in order to give time for everyone to patch
- Sucuri and wordfence start to see attacking on the old flaw on february 2nd on sites **left unupgraded**. In a few days they see from 67000 - 4 groups of attackers- pages defaced to 1,5 milion- 20 groups of attackers

From insult to injury

- Over the next weekend, Google also warned WordPress website owners registered in the Google Search Console. Google attempted to send security alerts to all WordPress 4.7.0 and 4.7.1 website owners, but some emails reached WordPress 4.7.2 owners, some of which misinterpreted the email and panicked, fearing their site might lose search engine ranking.

A (successful) experiment: WordPress on Hackerone

- Aaron Campbell (Security Team Lead) promoted the use of a Bug Bounty program.
- Launched officially on May 15th , but tested internally for about a year
- Two objectives:
  - ✔ Vulnerability coordination within all the teams involved (reporting, triages)
  - ✔ Create a network of “hackers” that gets paid as they found vulnerabilities (excellent reporting, disclousure agreement)

Remember the magic ingredient?

The community as a whole, is the magic ingredient!

- It is not a “private club”
- Anyone can be part of it, given that he respects some important principles (inclusion, etc)
- It is made of human being and networking between them (Slack channels, Meetups, Wordcamps)

A great opensource project

- Uses standards
- Evolve the product
- Has a solid testing and bugs reporting procedure
- Entrust the community
- Not only contributors

So, WordPress is really safe?

- Core is safe, plugins and themes are potential problems because there is less control.
- Another bottleneck: of 27% of all sites in the world great part are left un updated
- It is not a matter of not having any security issue. It is a matter how (faster) the community respond to that flaw.

Q&A

Thanks!

https://www.linkedin.com/in/nicolalaserra/