WordPress Security.

Context problem

Solution

Results

Conclusion

Set up instructions

Evidence or references to back up claims.

Graphics and diagrams to explain every complex concept

Comparison tables for reporting results

Quotations

---- 3 points ----

Why Concurrency for this problem?

Cons and pros with concurrency, pararellism, functional etc.

Evidence of why this is better

Show real cases and why im helping

Social implications

--- 3 points ----

Make use of several paradigms that complement each other or use a single paradigm to tackle a significant technical challenge.

--- 3 points ---

Implementation running and tested with several config and scenarios ( above 30 users) complete documentation to back up.

WordPress History

WordPress is a semantic personal publishing platform with a focus on aesthetics, web standards, and usability. Based on PHP and MySQL, released on May 27, 2003 by its founders Matt Mullenweg and Mike Little.

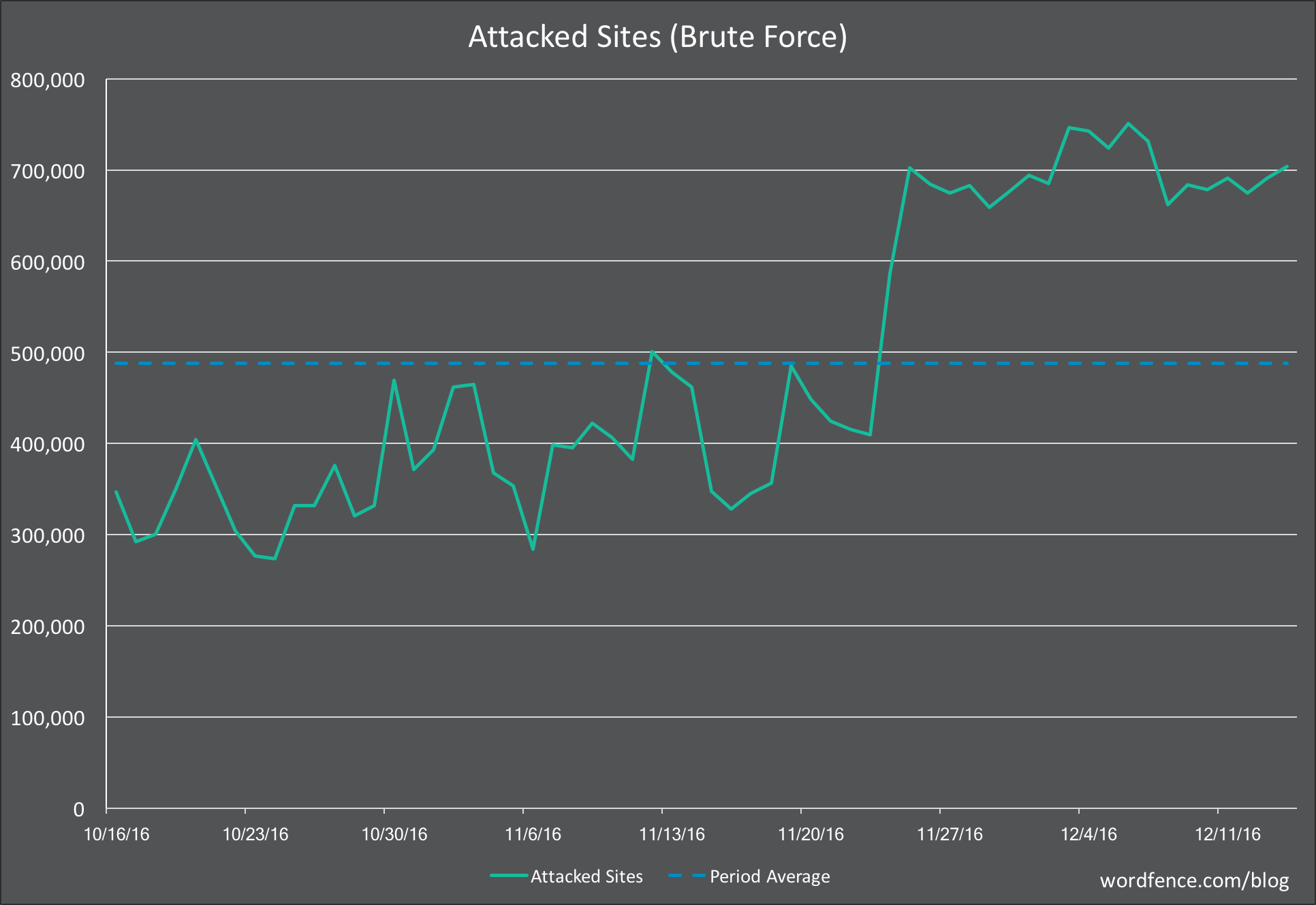
“*Since March 2016, WordPress is powering 26.4% of the Web, and it’s on the rise…Also interestingly on a daily basis there are over 500 sites being created on WordPress*” (Tomovic, 2016).

According to World Wide Web Technology Surveys since the March 1st, 2018 The most popular Content Management System (CMS) is WordPress with 30.5% of webpages in the web. (There are more than 73 million WordPress sites).

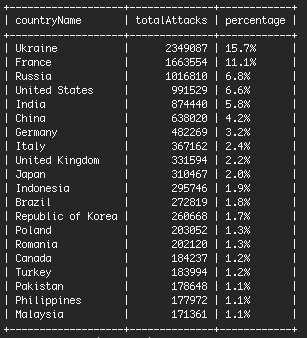
Various famous sites are powered by WordPress, some examples:

* The New York Times.
* Wall Street Journal’s Spe akeasy.
* People Magazine.
* National Geographic.

Because its popularity WordPress has been a natural target for security exploits. Since November 24th, 2016, WordPress Scan (WPScan) noticed a rise in brute force attacks to WordPress sites, as reminder a brute force attack is one that tries to guess your username and password to sign into your WordPress website. The following data will show how the attacks have increased during the next 3 weeks and where are they coming from.

The dotted blue line indicates the average number of attacked sites for the 60 days shown. 

The following table shows the top 20 countries sorted by number of attacks only in a time lapse of 24 hours.



The question after seen this numbers, Is WordPress secure? The answer is “It depends”. WordPress itself is very secure as long as WordPress security best practices are followed. WordPress runs on open source code and according to WordPress.com - Employees has a team of 449 persons specifically devoted to find, identify and fix WordPress security issues. That’s why keeping WordPress updated to the latest version is essential to maintain the website secure.

It’s important to note that WordPress security vulnerabilities extend beyond WordPress core into the themes or plugins installed on a website. According to WPScan, of the 3,972 known WordPress security vulnerabilities:

* 52% are from WordPress plugins. (A plugin adds and extend functionality to a WordPress page, to this day there are 54,908 plugins).
* 37% are from core WordPress. (source code).
* 11% are from WordPress themes. (visual templates).

According to Kristen Wright the 5 must common vulnerabilities found in WordPress sites are:

* Brute Force Attacks.
* File Inclusion Exploits.
* SQL Injections.
  + Is very common with PHP
* Cross-Site Scripting (XSS).
  + Most common vulnerability found in WordPress Plugins.
* Malware.
  + S

Solution

A solution that will help the WordPress community users, owners and developers is a WordPress Security Scan. The best scenario would be a scan that detect all 3972 vulnerabilities, but, because of academic reasons the Security Scan will only cover brute force attacks and SQL injections.

How will the program work?

The program will receive an URL from a WordPress site and will return if the site is vulnerable to the previously mention security issue.

Using concurrent programming in Java every thread will execute an attack.

Why concurrent programming in Java?

* You can handle very well http requests.
* For the http requests, threads can sleep until the server respond.

Why not parallel programming in CUDA?

* You can’t do http requests in CUDA.
* When executing code in parallel it’s not a good practice to use sleep, the ideal code is to run all at the same time without interruptions.

Sites for testing:

* https://chicablogger.com
  + Protected to BruteForce
  + Protected against SQL inyect
* <https://www.hellofashionblog.com>
  + Protected to BruteForce
  + Protected against SQL inyect
* <http://masquewordpress.com>
  + Vulnerable to BruteForce
  + Protected against SQL inyect.
* <http://wpspeak.com>
  + Vulnerable to BruteForce
  + Protected against SQL inyect

SOURCES

WordPressHistory: <https://wordpress.org/about/history/chapter3.pdf>

WordPressFacts: <https://expandedramblings.com/index.php/wordpress-statistics/>

WordPressAttacksCharts: <https://www.wordfence.com/blog/2016/12/how-to-protect-against-brute-force-attacks/>

MostCommonAttacksWordPress: <https://ithemes.com/2017/01/16/wordpress-security-issues/>

WordPressPlugins: <https://es-mx.wordpress.org/plugins/>

WordPressThemes: <https://themeisle.com/wordpress-themes/>

WordPressSecurityManagement: <https://wpscan.org/>

WordPressStatics: <https://managewp.com/blog/statistics-about-wordpress-usage>

WordWideWebTechnologySurveys: <https://w3techs.com/>

Sites WordPress: <https://colorlib.com/wp/blogs-using-wordpress/>

OutPutStreamClass: <https://docs.oracle.com/javase/7/docs/api/java/io/OutputStream.html>

AutomateLogInJava: <http://www.mkyong.com/java/how-to-automate-login-a-website-java-example/>

HTTPCONNJAVA: <https://www.journaldev.com/7148/java-httpurlconnection-example-java-http-request-get-post>

HTTPERRORS: <https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP>

SQLINJECTION: <https://www.owasp.org/index.php/SQL_Injection>

EXECUTORS: <http://www.baeldung.com/java-executor-wait-for-threads>

SQLINJECTION: <https://www.w3schools.com/sql/sql_injection.asp>