Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021 (NetSys 2021)

Large Scale Monitoring of Web Application Software Distribution to Measure Threat Response Behavior

Fabian Marquardt and Lennart Buhl

4 pages

Guest Editors: Andreas Blenk, Mathias Fischer, Stefan Fischer, Horst Hellbrueck, Oliver Hohlfeld, Andreas Kassler, Koojana Kuladinithi, Winfried Lamersdorf, Olaf Landsiedel, Andreas Timm-Giel, Alexey Vinel
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Large Scale Monitoring of Web Application Software Distribution to Measure Threat Response Behavior

Fabian Marquardt[1] **and Lennart Buhl**[2]

[1] marquard@cs.uni-bonn.de
[2] buhl@cs.uni-bonn.de
Department of Computer Science 4
University of Bonn, Germany

Abstract: Web application software may be affected by vulnerabilities and a fast deployment of security updates is required to protect users from attacks. With daily scans of over 50 million websites we are able to measure the threat response behavior. Preliminary results indicate significant differences between the different observed web application softwares.

Keywords: Threat Response, WordPress, MediaWiki

1 Introduction

In the modern Internet web application software has become a widespread technical basis for many websites. For example, the WordPress content management system (CMS) is used on about 33% of the top 10 million websites [W3T]. New versions of web application software may be released for a number of reasons. Typically, major releases introduce new functionality to the software, while minor releases fix bugs contained in the previous version of the software. However, the release of the new version does not automatically imply that all installations of the software get updated. In some cases, an automated update mechanism is embedded into the software or invoked by external tools such as package managers. In many other cases, updating the web application software requires manual actions of the administrators. This raises an interesting question: How long does it take until new releases are applied to existing installations? Especially if the new release fixes one or more security issues, minimizing the time-to-patch is a very important step to reduce the exposure of systems to known vulnerabilities.

Our work focuses on measuring the distribution of different web application software versions on a large scale. We detect the used web application software versions from the client-side and inspect millions of websites on a day-by-day basis. With the gained data we can track the deployment of new releases for multiple widely used open source systems. While still in an early stage, our work already shows that the availability of auto-update mechanisms has a great impact on minimizing the time-to-patch especially for security-relevant releases.

2 Related Work

Past measurement studies have often focused on tracking one specific vulnerability to measure threat response. For example, Durumeric et al. have analyzed the population of the *Heartbleed*

vulnerability [DLK+14], and Quach et al. tracked the occurrence of a Linux TCP stack vulnerability on top websites over time [QWQ17]. Our approach goes beyond tracking specific security incidents, but focuses on tracking the web application software version distribution in a more generic way. Apart from threat response analysis, this method can also be used for many other statistical analyses. It is comparable to Censys, which started as a research project but has now been converted to a commercial service [DAM+15].

3 Methodology

To perform daily scans of a high volume of websites we have created a pipeline-based infrastructure in Python. In this Extended Abstract paper we can only briefly outline the different elements of this pipeline: First, the individual scan targets are read from an input file by the *URL Selector*. URLs and the contained domains are matched against common malware/abuse blacklists to avoid triggering false positive alert messages in intrusion detection systems. In addition, a manual blacklist is maintained for some domains explicitly by request of their operators. In the next step, the *HTTP Fetcher* issues an HTTP request for each given URL. To provide a high scanning throughput, many scans are performed in parallel. In the current setup, it can easily perform several thousand requests per second on a normal desktop computer. The exact numbers depend on many factors, e.g. the performance of the used DNS resolvers. Once the HTTP response is received, it is forwarded to the *Version Analyzer*, which applies a pattern-based fingerprinting strategy to identify different web application software and their specific versions. In the current state we focus on several well-known open source systems, including *WordPress* and *MediaWiki*. Finally the *Result Writer* collects all detected version information as well as error messages and stores this information in a JSON-based file structure. This files can then later be processed to analyze the gathered results.

4 Preliminary Analysis

For a first analysis of our methodology we collected a list of over 52 million URLs from the Top List Study project [SHG+18]. Specifically, we merged the contents of all Alexa 1M, Majestic 1M and Umbrella 1M top lists released after January 01, 2018. The first daily scan of all URLs has been conducted on March 24, 2020, leading to a time span of about six months at the time of writing this paper. Obviously, for an in-depth analysis and the identification of long-term trends it is necessary to collect even more data. Hence, the following results only present a preliminary evaluation.

4.1 Verification

To find out whether or not our system can measure the deployment of certain web application software versions correctly, we compare our data set to an external data source: The *WP Central* project collects the download counts for different versions of WordPress over time [Hej]. Even if a download of the WordPress software does not strictly imply that it will be installed on a web

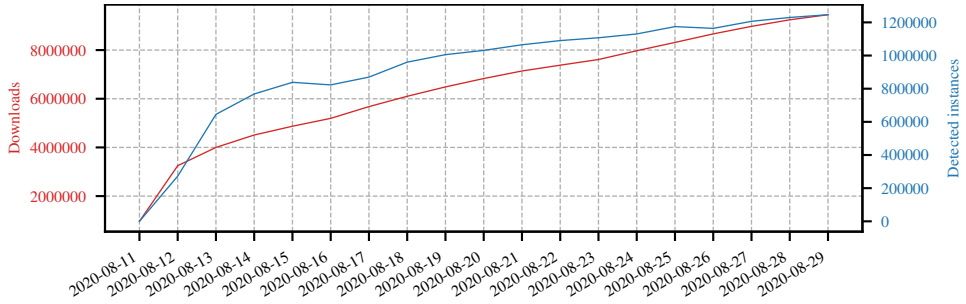


Figure 1: WordPress 5.5 downloads and detected instances (Pearson correlation: $r = 0.96$)

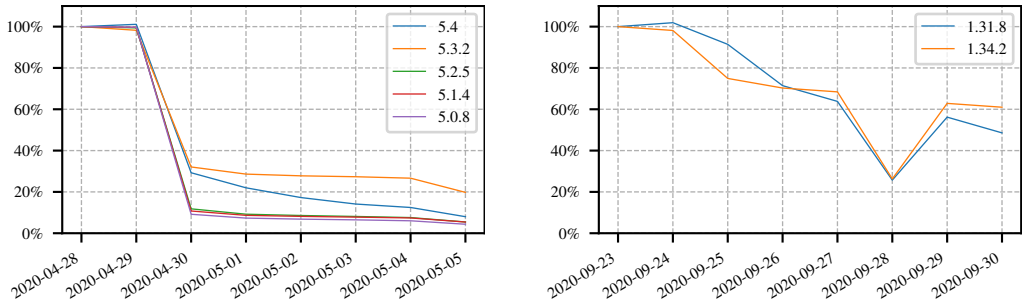
page, we can assume that there is a strong correlation between the number of downloads and the number of installations.

We have retrieved the download counts for the days following the release of the new 5.5 major version on August 11, 2020 and compare it to our data set. As is visible in Figure 1 there is a strong correlation with a Pearson coefficient of $r = 0.96$. Small variations between the two data sets can be expected due to the time interval between download and deployment or the 24 hour time difference between two of our daily scans.

4.2 WordPress and MediaWiki update behavior

One reason for the high popularity of WordPress is the availability of an easy to use update system, which is already built into the software and can deploy updates to existing installations automatically. By contrast, MediaWiki requires the user to manually deploy the source code of the new version and run upgrade procedures. This might have a negative effect on threat response behavior. To identify such effects we analyze and compare two security incidents: Several WordPress vulnerabilities with medium to high *CVSS score* were disclosed on April 30. Patched versions for all supported branches of WordPress were released on April 29. MediaWiki was affected by several vulnerabilities of medium to high severity on September 27. Patched versions for the supported branches were released on September 24.

Using our data set we want to inspect how quickly the fixed versions were applied to existing installations. To do so, we focus on the versions of each software which were current before the security fixes were released and monitor their decline over the next days. In the best case, the occurrence of affected versions drops very quickly after the release. The results are shown in Figure 2. Each plot starts at and is normalized to the day before the new versions got released. For WordPress, some of the older supported major versions have been removed for better visibility, but show similar results in our evaluation. It is clearly visible that the number of affected WordPress installations declines very quickly just one day after the release of the patched versions, which is a good result. The results for the affected MediaWiki installations are significantly worse. The decline of vulnerable versions happens much slower and after one week about 50% of the installations are still not patched. It should be noted that the results for September 28 are skewed due to a power grid failure which affected the network connectivity of the measurement system.



(a) WordPress April 2020 incident (b) MediaWiki September 2020 incident

Figure 2: Version distribution after security incidents

5 Conclusion and Future Work

The first evaluation already shows the potential of our large-scale data collection methodology. For the future it is important to develop additional detection techniques to improve coverage of other web application software. The resulting data sets will be used for a more in-depth analysis to gain a better understanding of web technology updates. It is also planned to publish anonymized versions of our data sets under open access conditions.

Bibliography

[DAM+15] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, J. A. Halderman. A search engine backed by Internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Pp. 542–553. 2015.

[DLK+14] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey et al. The matter of heartbleed. In *Proceedings of the 2014 ACM Internet Measurement Conference*. Pp. 475–488. 2014.

[Hej] M. Hejnen. WP Central: WordPress 5.5 download statistics. Online (accessed on Oct. 15, 2020).
https://wpcentral.io/version/5.5/

[QWQ17] A. Quach, Z. Wang, Z. Qian. Investigation of the 2016 linux tcp stack vulnerability at scale. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, pp. 1–19, 2017.

[SHG+18] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, N. Vallina-Rodriguez. A long way to the top: Significance, structure, and stability of internet top lists. In *Proceedings of the 2018 ACM Internet Measurement Conference*. Pp. 478–493. 2018.

[W3T] W3Techs. Usage statistics of content management systems. Online (accessed on Oct. 15, 2020).
https://w3techs.com/technologies/overview/content_management