POSTER: Predicting Website Abuse Using Update Histories

Yuta Takata, Mitsuaki Akiyama, Takeshi Yagi,
Kunio Hato
NTT Secure Platform Laboratories
takata.yuta@lab.ntt.co.jp

Shigeki Goto
Waseda University
goto@goto.info.waseda.ac.jp

ABSTRACT

Threats of abusing websites that webmasters have stopped updating have increased. In this poster, we propose a method of predicting potentially abusable websites by retrospectively analyzing updates of software that composes websites. The method captures webmaster behaviors from archived snapshots of a website and analyzes the changes of web servers and web applications used in the past as update histories. A classifier that predicts website abuses is finally built by using update histories from snapshots of known malicious websites before the detections. Evaluation results showed that the classifier could predict various website abuses, such as drive-by downloads, phishes, and defacements, with accuracy: a 76% true positive rate and a 26% false positive rate.

CCS CONCEPTS

• Security and privacy → Web application security;

KEYWORDS

Website Abuse; Software Update; Internet Archive; CMS

ACM Reference Format:

Yuta Takata, Mitsuaki Akiyama, Takeshi Yagi, Kunio Hato and Shigeki Goto. 2018. POSTER: Predicting Website Abuse Using Update Histories. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23-27, 2018, Lyon, France*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3184558.3186903>

1 INTRODUCTION

Threats of hacking other people's websites have been increasing. Websites are hacked by abusing vulnerabilities in web servers and web applications or stealing account credentials by using malware. Especially, abuses of the CMS (Content Management System) pervasively used by many webmasters are increasing year by year.

To prevent such website hacking, it is effective to have webmasters update the version of CMS and apply security patches. However, many webmasters do not update software or check security configurations for a long time, and these abandoned websites are abused by attackers [8]. Security warnings must be sent to the webmasters on the basis of the current website status such as software updates and security configurations before abuse by attackers [6].

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '18 Companion, April 23-27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5640-4/18/04.

<https://doi.org/10.1145/3184558.3186903>

Table 1: Extracted Features During Website Access

Type	Feature Name	Description
Strings	meta[generator]	Value of a generator attribution in a meta tag
	Server header	Value of a Server header
	X-Powered-By header	Value of a X-Powered-By header
Boolean	detect-url	Whether the CMS type can be identified by URL path
	detect-meta	Whether the CMS type can be identified by a meta tag
	detect-header	Whether the CMS type can be identified by HTTP headers

Figure 1: Update Histories Compiled by Comparing Archived Snapshots

In this poster, we focus on update histories of software that composes a website. We propose a method of predicting **potentially abusable websites** by retrospectively analyzing whether the webmasters have appropriately managed their own websites. This method accesses archived snapshots of past websites and analyzes the changes of web servers and web applications used so far as update histories. Finally, it builds a classifier that predicts website abuses by using update histories of known malicious websites before the detections. In our experiment, we used 397,629 snapshots of 10,730 URLs for the classifier. Evaluation results showed that the classifier could predict website abuses with accuracy: a 76% true positive rate (TPR) and a 26% false positive rate (FPR).

2 PROPOSED METHOD

2.1 Collecting Archived Snapshots

Internet Archive provides a service, called Wayback Machine [4], for archiving past websites (snapshots). Wayback Machine archives web content collected by crawling websites that are well linked to from other websites. Therefore, we can identify web servers and web applications used in the past if we can access snapshots archived by Wayback Machine.

Figure 2: Experimental environment for dataset collection

2.2 Compiling Update Histories

We trace an update history by comparing features listed in Table 1 in a time series. If the type of extracted features is Strings, the update history has four categories: 1. “Up” means the value began to be observed; 2. “Stable” means the observed value did not change; 3. “Updated” means the observed value changed; and 4. “Down” means the value was no longer observed. If the type of extracted features is Boolean, the update history has three categories: 1. “Up” means the value changed from False to True; 2. “Stable” means the value did not change; and 3. “Down” means the value changed from True to False. Figure 1 shows two example update histories created from three archived snapshots. More precisely, Update History 0 is created by comparing Archived Snapshot 0 with Archived Snapshot 1, and Update History 1 is created by comparing Archived Snapshot 1 with Archived Snapshot 2. Then, a time interval Δ between archived snapshots was also included in these update histories by comparing the number of months in archived Timestamp.

2.3 Designing Feature Vectors

We design a feature vector used for our classifier by combining update histories in the previous subsection. In other words, the number of a feature vector’s dimensions is determined by the number of update histories. Therefore, to align the number of dimensions, we predefined the number of times to compare archived snapshots N when compiling update histories. Note that we used a missing value (e.g., -1) when we could not compare for the predefined number of times due to the number of archived snapshots.

3 EVALUATION

3.1 Experimental Environment and Datasets

In this experiment, we collected datasets in three steps using the experimental environment shown in Fig. 2. First, our crawler accesses inputted seed URLs and identifies CMS types of the websites on the basis of the features listed in Table 1. In this experiment, we focus on the top three CMSs: WordPress, Drupal, and Joomla!. Next, we collected archive URLs of websites whose CMS types were identified using Time Travel API [3]. Finally, the crawler accesses the archive URLs and collects snapshots used for compiling update histories.

Using this environment, we collected 341,431 snapshots of URLs listed in Alexa Top 250K Sites [1] as a benign dataset. We also collected 56,198 snapshots of three types of malicious URLs as a malicious dataset: URLs from which our high-interaction honeypots detected drive-by downloads, phishing URLs listed in PhishTank [2], and defacement URLs listed in Zone-H [5].

Table 2: AUC by random forest with each parameter

$\delta \setminus n$	1	2	3	4	5	6
3	0.576	0.701	0.788	0.825	0.848	0.857
4	0.589	0.734	0.808	0.850	0.862	0.860
5	0.589	0.763	0.841	0.858	0.864	0.863
6	0.600	0.769	0.852	0.863	0.863	0.859
7	0.612	0.762	0.857	0.870	0.866	0.869
8	0.642	0.801	0.865	0.871	0.871	0.872

3.2 Evaluating Detection Capability

In this evaluation, we measured the accuracy of our classifier built by random forest using learning data and test data separated from the datasets in the previous subsection by 10-fold cross validation. Then, we defined the number of times to compare archived snapshots as n and the minimum time interval (the number of months) between archived snapshots as δ . Table 2 shows the calculated results of AUC (Area Under Curve) values with various parameters. As shown in Table 2, the larger the number of comparisons n , the longer the time interval δ , the higher the AUC value. This means that we can detect more website abuses on the basis of webmaster behaviors by retrospectively analyzing websites rather than analyzing websites at the certain time.

3.3 Evaluating Prediction Capability

In this subsection, we separated malicious datasets into learning data that was detected before a certain time and test data that was detected after the certain time and evaluated the accuracy the same way as in the previous subsection. This method can evaluate the capability for predicting whether websites in the test data are **potentially abusable in the future** at the certain time [7]. In this experiment, we separated datasets from January 2017. From the results of evaluating the accuracy of our classifier with the best performance parameters $n = 6, \delta = 8$, the AUC was 0.803. Compared with the prediction capability of the existing method (AUC=around 0.7) [7], our proposed method achieved better accuracy.

4 CONCLUSIONS

We proposed a method of predicting website abuses by retrospectively analyzing update histories of software that composes a website by using Internet Archive. Evaluation results showed that a classifier built by the method could predict website abuses with accuracy: a 76% TPR and a 26% FPR. Using such accurate prediction results, we can prevent website abuses by preliminarily sending security warnings to webmasters.

REFERENCES

- [1] 2017. Alexa Top Sites. (2017). <https://aws.amazon.com/jp/alexa-top-sites/>.
- [2] 2017. PhishTank. (2017). <https://www.phishtank.com/>.
- [3] 2017. Time Travel API. (2017). <http://timetravel.mementoweb.org/guide/api/>.
- [4] 2017. Wayback machine. (2017). <https://archive.org/web/>.
- [5] 2017. zone-h unrestricted information. (2017). <http://www.zone-h.org/>.
- [6] Frank Li, Grant Ho, Eric Kuan, Yuan Niu, Lucas Ballard, Kurt Thomas, Elie Bursztein, and Vern Paxson. 2016. Remedying Web Hijacking: Notification Effectiveness and Webmaster Comprehension. In *World Wide Web Conference*.
- [7] Kyle Soska and Nicolas Christin. 2014. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *USENIX Security Symposium*.
- [8] Marie Vasek, John Wadleigh, and Tyler Moore. 2015. Hacking is not random : a case-control study of webserver compromise risk. *IEEE Transactions on Dependable and Secure Computing* 13, 2 (2015), 206–219.