2015

Detecting malicious short URLs on Twitter

Raj Kumar Nepali

Yong Wang

Yazan Alshboul

Detecting malicious short URLs on Twitter

Emergent Research Forum papers

Raj Kumar Nepali
Dakota State University
rknepali@pluto.dsu.edu

Yong Wang
Dakota State University
Yong.wang.dsu.edu

Yazan Alshboul
Dakota State University
yaalshboul@pluto.dsu.edu

Abstract

Short URLs (Uniform Resource Locators) have gained immense popularity especially in Online Social Networks (OSNs), blogs, and messages. Short URLs are used to avoid sharing overly long URLs and save limited text space in messages or tweets. Significant numbers of URLs shared in the Online Social Networks are shortened URLs. Despite of its potential benefits from genuine usage, attackers use shortened URLs to hide the malicious URLs, which direct users to malicious pages. Although, OSN service providers and URL shortening services utilize certain detection mechanisms to prevent malicious URLs from being shortened, research has found that they fail to do so effectively. These malicious URLs are found to propagate through OSNs. In this paper, we propose a mechanism to develop a machine learning classifier to detect malicious short URLs with visible content features, tweet context, and social features from one popular Online Social Network Twitter.

Keywords

Online Social Networks, twitter, short URLs, malicious URLs, machine Learning, classification

Introduction

In the age of web 2.0, information is contained in the form of webpages and shared via Online Social Networks (OSNs), emails, messages, and texts. OSN is a platform to connect and communicate with each other and share knowledge, information, news, announcements, etc. While doing so, people share URLs (Uniform Resource Locator) of the webpages that possess the information. However, URL sharing can be problematic because of its length; some URLs are overly long and complicated. One particular OSN that is very famous as information sharing social network is Twitter. Twitter is a microblogging social network that allows users to post messages up to 140 characters known as "tweets". It is difficult to share long URLs because of this restriction. This led to the popularity of URL shortening services on Twitter. URL shortening services take, usually a long URL from users, and create a short URL "alias" for that long URL. The short URL can then be shared among friends easily. When other users visit the short URL, they will be directed to the shortening services upon which users are redirected to the original long URL "Landing page". These services have gained popularity ever since they appeared in 2001. Now, there are more than 200 URL shortening services. Bitly and tinyurl are the most widely used URL shortening services along with Twitter. Google and Facebook also have their own services.

Unfortunately, malicious users leverage this knowledge to their advantage to hide their identities, and exploit limited text space to spread malicious URLs (Chhabra et al. 2011; Maggi et al. 2013). Attackers are leveraging these services in phishing scams, spamming, and malware campaigns. Especially, social media has been found very vulnerable to these attacks (Castillo et al. 2011). The factors that attract malicious users to social media include, but are not limited to, wide range of easy targets, easily exploiting social

relationships, and very high success rate (Jagatic et al. 2007). Despite of its popularity and users expectation of strong security against malicious URLs, shortening services and OSNs fail to prevent malicious URLs from being shortened and propagated (Maggi et al. 2013). URL Shortening Services and OSNs often use blacklists to block malicious URLs, however, blacklists are generally not comprehensive and up-to-date (Sinha et al. 2008). In many cases, backlists can be bypassed too (Maggi et al. 2013).

In this research, we propose an analytical approach on malicious short URLs by investigating their tweet contents, tweet contexts, and social features, which are directly visible on tweet and profile information without actually visiting the URLs. First, we aim to determine different visible content-based features that are present on tweet contents. Second, context-based features will be explored in detail, for example, relevance of the tweet, mentions, time, etc., and finally, social features of malicious tweets will be explored to develop an efficient mechanism to detect malicious short URLs. The contributions of the paper include, but are not limited to:

- We propose a novel approach to detect malicious short URLs with the help of visible features;
- We demonstrate the feasibility of effectively detecting malicious short URLs with visible features.

Our research questions are:

a. Can content-related features on tweets help in detecting malicious short URLs?

b. Can contexts of tweets help in detecting malicious short URLs?

c. How can visible profile and social features help in detecting malicious short URLs?

The remainder of the paper is organized as follow: we discuss the related work in the domain in next section followed by methodology and the dataset used for our work. Then, we discuss architectural design of the artifact followed by Evaluation and Implementation plans. Finally, we conclude the paper with conclusion and future work.

Related Work

Malicious URLs are pointer to malicious contents on the web. Malicious URLs may be part of scams, phishing, or more sinister malware campaigns. Because of the popularity of OSNs, attackers are leveraging OSNs as medium to propagate malicious URLs. Machine learning is well-accepted technique to classify malicious URLs. This section provides an overview of studies that discuss different approaches to detect malicious URLs using machine learning.

Malicious URL detection:

Malicious URL detection on the web in general is a very important topic. Many researchers have developed techniques to detect malicious URLs. The approaches can be classified into two categories: active detection and passive detection. Active detection is the one that actually visits the webpage to download the contents of the page and performs the analysis on the contents. Passive detection does not visit the suspicious page. It gathers other information to classify the URL as malicious or benign.

Passive detection of malicious URLs typically leverages lexical features and host-based features. Lexical features attempt to find the available properties for malicious URLs that look different to genuine URLs. For example: a malicious URL www.ebay.com.payment.net looks different from a typical URL www.ebay.com. Host based features include WHOIS info, geographic information, etc. Ma et al. (2009a) used host and lexical based features to develop a machine learning classifier with 95-99% accuracy. Ma et al. (2009b) further performed a large scale online learning of suspicious URLs and used lexical and host-based features to classify malicious URLs. It achieved accuracy of 99% on a balanced dataset. Similarly, many works have been performed in other areas like phishing URL detection. Garera et al. (2007) used four types of features: page based, domain based, type based, and word based, to identify the URL as malicious and benign and achieved accuracy of up to 97% with logistic regression algorithm. Their approach is also URL classification without actually visiting the webpage. McGrath and Gupta (2008)

used IP addresses, WHOIS records, geographic information, and lexical features to compare phishing and non-phishing URLs.

Active detection can also be found in the literature. However, it is out of scope of this paper and will not be discussed here.

Malicious short URL detection:

Recently, malicious short URL detection in social networks has gained attention of the researchers. The first large scale exploratory research on short URLs was done by Antonaides et al. (2011). They found that short URLs are mostly famous with OSNs and have word of mouth propagation. Mostly they point to news and informative contents, thus confirming Kwak et al.'s (Kwak et al. 2010) assertion that Twitter is more like an information relaying network than a social network. Shortening services lead to up to 91% reduction in length and 54% overhead.

First study of security implications of short URLs was done by Maggi et al. (2013) over the span of two years. They found that existing countermeasures against the prevention of shortening of malicious URLs fail horribly. Almost every shortening service security can be evaded. Wang et al. (2013) studied the misuse of short URLs and developed a mechanism to detect spam URLs. Authors used click-traffic features to achieve accuracy up to 90.81% with random forest algorithm. Lee and Kim (2013) developed a mechanism to detect suspicious URLs in real time from Twitter stream. They leveraged attacker's URL redirection chain and tweet context to develop a classifier with 91.87% accuracy. Gupta, Aggarwal, & Kumaraguru (2014) did exploratory study on bitly, one of the most widely used URL shortening services and it's spam URL/account detection mechanism. They used short URL based features with two domain specific features to classify bitly short URLs and achieved accuracy of 86.41%.

Among the few researches in malicious short URL detection, as mentioned above, most of the researches fail to look deeper into the OSNs itself. Although, some research include some features from OSNs and with some data from third parties, it is not always possible to obtain data from third parties, provided there are many URL shortening services: proprietary and open. Our assertion is that since malicious URL propagation is embedded in social phenomenon, it can be solely detected from the information obtained from social networks. Particularly, in this work we are interested in leveraging visible features to detect malicious short URLs.

Methodology

Twitter provides streaming API, which can be used to collect public tweets. We will particularly use bitly short URLs, since it is the most widely used URL shortening service provider. Tweets will be collected and necessary information will be extracted using Twitter's streaming APIs. URLs, tweet contents, social information, and profile information will be collected from tweets and APIs. Tweet metrics will also be gathered from OSNs (clicks, favorite, retweets). The relevance of the context will be determined with the help of famous tweet trends. Twitter provides the top ten trends that are famous at a given time. We collect twitter trend specific to US location. This is because of our emphasis on English tweets only. It is believed that attackers leverage relevant contextual information to spread malicious URLs.

Short URLs will be expanded to a long URL and submitted to popular web URL scanning service: PhishTank, and Google's safebrowsing API to obtain the labeled dataset. PhishTank is a phishing blacklisting service operated by OpenDNS and provides developers and researchers with open API to download blacklists and check a particular URL. The results are provided in JSON or XML format (PhishTank n.d.). Google's safebrowsing lookup API, funded by Google, is blacklisting service of phishing and malware hosting sites and supports GET and POST requests for single URL lookup or groups of URLs. These services will be used to check the URLs for malignity or benign. Labeled dataset will be developed based on the result. The labeled dataset will be split into training dataset (TR) with 70% of it and testing dataset (TS) with 30% of it.

Then, based on the features extracted, a classifier will be developed using the labeled training dataset (TR) using the widely used machine-learning tool "Weka" (more info on evaluation section). Weka (WEKA n.d.), data mining software with SVM (Support Vector Machine) and random forest algorithm will be used. SVM and random forest algorithm are used because of their ability to handle large number of features, mixture of binary, numeric, and categorical features, and unbalanced dataset. The results will be compared and the best classification algorithm will be selected. Finally, test dataset (TS) will be used against the classifier developed to test the accuracy of our developed model.

Design

The figure 1 below depicts the design diagram of the artifact. Basically, the artifact will have five major components: Stream Data Collector, Data Cleaning, Feature Extractor, Blacklisting, and Classifier.

Streaming Data Collector: Twitter's streaming API will be used to gather data for planned period of time. This dataset will contain all tweets that have short URLs (particularly bitly in our case). Twitter provides streaming data in JSON file format. Data Collector will run 24/7 for a period of one month. Based on the preliminary results from our testing, our assumption is that one month of data collection will provide us enough labeled dataset, and beside that we have limitations related to API lookups.

Data Cleaning: Data collected through streaming API contains more information than required (profile color, profile image). Hence, only the relevant information will be collected and saved into database. Data will be cleaned and used for further processing. Particularly, it will remove the erroneous data, label the missing values with relevant information, and remove the outliers.

Blacklisting: Blacklisting service will be composed of two services: PhishTank and Google Safebrowsing API. After cleaning the data, full URL of the corresponding short URL will be obtained and checked against the two URL checking services to determine the malignity of the URL. If the URL is malicious, the corresponding features related to the tweet would be labeled as malicious and vice-versa.

Feature Extractor: Feature extractor will extract features from tweets, and OSNs. Features are selected with the help of literature. The concept of content-based features is derived from the concept of lexical features (Ma et al. 2009a) on the existing literatures on malicious URL detection. Context based features are derived from the tweet context. Social features will be derived from the user's profile, and relevance will be obtained from the tweet trends at the time of tweet. Along with some of the features derived from related literature, and using our own concept of visible attributes, we come up with the list of features to look at (but not limited to).

Content-related Features	Context Features	Social Features
Length of tweet	Time of Tweet	Following
Mentions	Relevance (trending topics)	Followers
Hashtags	User mentions	Location
Number of URLs		Tweets
Bag-of-words		Retweets
		Favorite_count

Table 1. List of features

Classifier: Classifier will take the training dataset of short URLs and build a model for classification based on two different machine learning algorithms, i.e., random forest and support vector machines,

based upon features as mentioned in feature extractor. The classification model is then supplied with test dataset to determine if the short URL is malicious or benign.

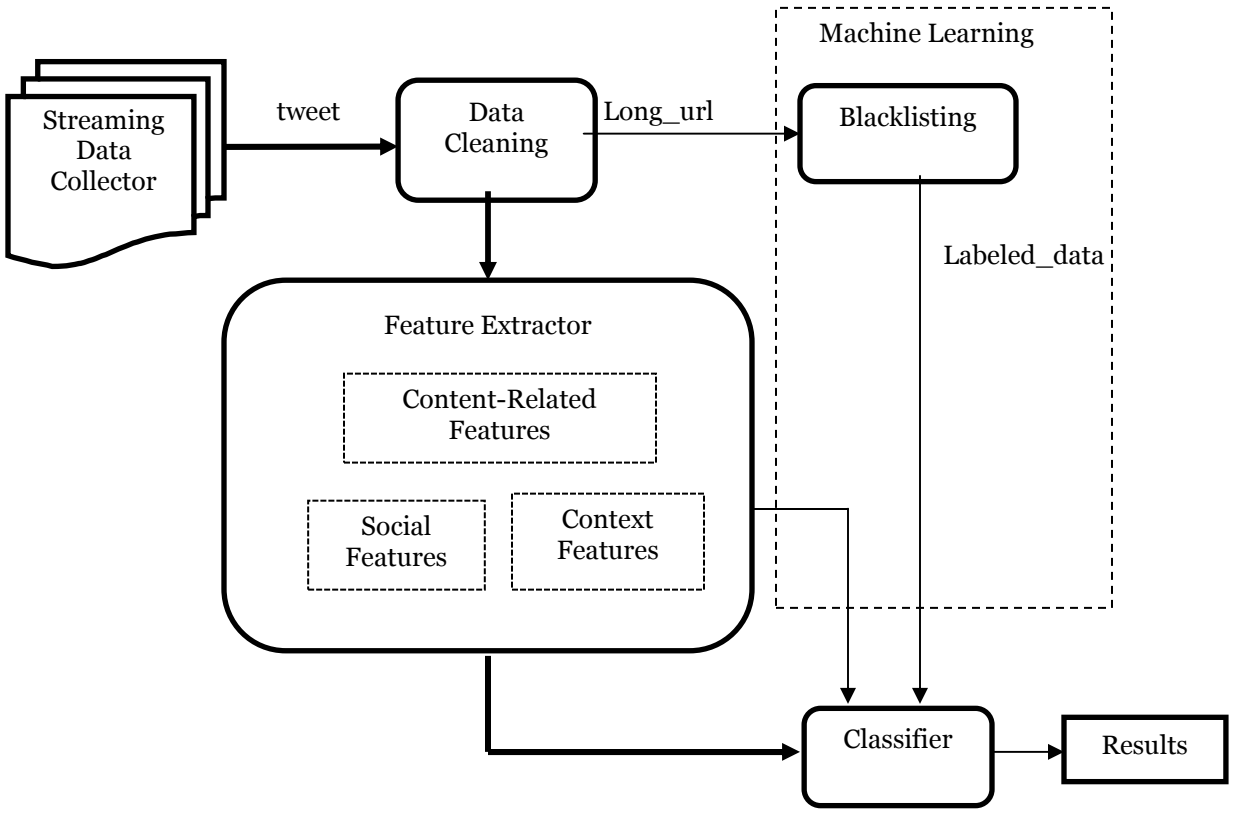

Fig 1: Architectural Diagram

Evaluation

Evaluation will be performed by classifying the short URLs using our methodology. Positive dataset will be checked against the blacklist databases. Evaluation will be based on how fast, compared to the existing blacklists, our approach identifies malicious short URLs and how accurate the result is. Beside this, we will use other evaluation metrics to measure the effectiveness of our approach. The evaluation metrics are:

		Predicted	
		Malicious	Benign
Actual	Malicious	TP	FN
	Benign	FP	TN

Table 2: Output Matrix

True positive (TP): TP is the correct identification of truly malicious URL as malicious.

True Negative (TN): TN is the correct identification of benign URL as benign.

False Negative (FN): FN is the incorrect identification of malicious URL as benign.

False Positive (FP): FP is the incorrect identification of benign URL as malicious.

From the above results, we can measure precision (P), recall (R), F-measure (FM), and accuracy (A) as shown below:

$$P = \frac{TP}{(TP+FP)} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$R = \frac{TP}{(TP+FN)} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$FM = 2.\frac{P.R}{P+R} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$A = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Implementation

The artifact will be implemented as Firefox's web browser extension. Add-ons SDK extensions provide a simple set of APIs to build Firefox extensions. The system will be programmed with XML User Interface Language (XUL) and Javascript to automate the processes. The program will automatically fetch tweets and extract the features from the data and submit it to the classifier for classification. The result will be displayed back to the user.

Conclusion & Future Work

In this paper, we presented our approach to malicious short URL detection using content-based features, along with tweet context based features, and social features. Our approach is basically based on visible features that are signs of malicious URLs and leveraging those features to classify malicious URLs without actually visiting the malicious URLs. We present the architecture of the system and our implementation plan. Next, we are planning to actually test the model with real-time streaming data from Twitter. Moreover, we are also planning to implement our idea as a Firefox's extension that can be used by OSN users. Our long-term goal is to develop an efficient short URL classification mechanism with high accuracy.

REFERENCES

Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulous, E., Ioannidis, S., P.Markatos, E., and Karagiannis, T. 2011. "we.b: The web of short URLs," in *WWW*, Hyderabad, India, pp. 715–724.

Castillo, C., Mendoza, M., and Poblete, B. 2011. "Information credibility on twitter," in *World Wide Web*, New York, USA: ACM, pp. 675–684.

Chhabra, S., Aggarwal, A., Benevenuto, F., and Kumaraguru, P. 2011. "Phi.sh/$oCiaL: The phishing landscape through short URLs," in *CEAS*, Perth, Australia: ACM, pp. 92–101.

Garera, S., Provos, N., Chew, M., and Rubin, A. D. 2007. "A framework for detection and measurement of phishing attacks," in *Proceedings of the 2007 ACM workshop on Recurring malcode*, pp. 1–8.

Gupta, N., Aggarwal, A., and Kumaraguru, P. 2014. "bit.ly/malicious: Deep dive into short URL based e-crime detection," in *eCrime*, pp. 14–24.
Jagatic, T., Johnson, N., Jacobsson, M., and Menczer, F. 2007. "Social Phishing," *Communications of the ACM* (50:10), pp. 94–100.
Kwak, H., Lee, C., Park, H., and Moon, S. 2010. "What is Twitter, a social network or a news media?," in *WWW*, New York, NY, USA: ACM, pp. 591–600.
Lee, S., and Kim, J. 2013. "WarningBird: A near real-time detection system for suspicious URLs in Twitter stream," *IEEE transactions on dependable and secure computing* (10:3), pp. 183–195.
Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. 2009a. "Beyond blacklist: Learning to detect malicious web sites from suspicious URLs," in *KDD*, Paris, France: ACM.
Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. 2009b. "Identifying suspicious URLs: An application of large-scale online learning," in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada.
Maggi, F., Frossi, A., Stringhini, G., Stone-Gross, B., Kruegel, C., and Vigna, G. 2013. "Two years of short URLs Internet Measurement: Security threats and countermeasures," in *WWW*, Rio de Janerio, Brazil.
McGrath, D. K., and Gupta, M. 2008. "Behind phishing: An examination of phisher modi operandi," in *Proceedings of the USENIX workshop on Large-Scale Exploits and Emergent Threats (LEET)*, San Francisco, California.
PhishTank. (n.d.). "PhishTank," *www.phishtank.com*.
Sinha, S., Bailey, M., and Jahanian, F. 2008. "Shades of Grey: On the effectiveness of reputation based black-lists," in *Proceedings of the International Conference on Malicious and Unwanted Software(Malware)*, Alexandria, Virginia, US.
Wang, D., Navathe, S. B., Liu, L., Irani, D., Tamersoy, A., and Pu, C. 2013. "Click traffic analysis of short URL spam on Twitter," in *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*, IEEE, pp. 250–259.
WEKA. (n.d.). "Weka 3: Data mining Software in Java," *cs.waikato.ac.nz*.