WatchDog: A Malicious URL Detection System for Social Networks (Twitter)

MADALI SAIPOOJA¹, K.CHANDRA BABU²

¹PG Scholar, Dept of CSE, Annamacharya Institute of Technology and Sciences, Tirupati, AP, India.
²Asst Prof, Dept of CSE, Annamacharya Institute of Technology and Sciences, Tirupati, AP, India.

Abstract: Twitter is vulnerable to malevolent tweets containing URLs for irrelevant, phishing, and malware distribution. Traditional Twitter spam detection schemes utilize account features such as the ratio of tweets containing URLs and the account creation date, or relation features in the Twitter graph. These detection schemes are ineffective against feature creations or use much time and resources. Predictable malicious URL detection schemes utilize several features including lexical features of URLs, URL redirection, HTML content, and active behavior. However, evading techniques such as time-based evasion and crawler evasion exist. In this paper, we propose WatchDog, a malicious URL detection system for Twitter. Our system examines correlations of URL redirect chains extracted from several tweets. Because attackers have limited resources and usually reuse them, their URL redirect chains frequently share the same URLs. We develop methods to discover correlated URL redirect chains using the frequently shared URLs and to determine their maliciousness. We collect numerous tweets from the Twitter public timeline and trained a statistical classifier using them. Evaluation results show that our classifier accurately and efficiently detects malicious URLs. We also present WatchDog as a near real-time system for classifying malicious URLs in the Twitter.

Keywords: Redirect Chains, Correlation of URLs, Feature Extraction, Data Collection.

I. INTRODUCTION

Twitter is a famous social networking and information sharing service that allows users to exchange messages of fewer than 140-character, also known as tweets, with their friends. When Twitter users want to share a URL with friends via tweets, they usually use URL shortening services to reduce the URL length since tweets can contain only a restricted number of characters. Owing to the popularity of Twitter, malicious users often try to find a way to attack it. The most common forms of Web attacks, including spam, scam, phishing, and malware distribution attacks, have also appeared on Twitter. To cope with malicious tweets, several Twitter spam detection schemes have been proposed. These schemes can be classified into account feature-based, relation feature-based, and message feature-based schemes. Account feature-based schemes use the distinguishing features of spam accounts such as the ratio of tweets containing URLs, the account creation date, and the number of followers and friends. The relation feature-based schemes rely on more robust features that malicious users cannot easily fabricate such as the distance and connectivity apparent in the Twitter graph. Extracting these relation features from a Twitter graph, however, requires a significant amount of time and resources as a Twitter graph is tremendous in size.

The message feature-based scheme focused on the lexical features of messages. In this paper, we propose WatchDog, a malicious URL detection system for Twitter. Instead of investigating the landing pages of individual URLs in each tweet, which may not be successfully fetched, we considered correlations of URL redirect chains extracted from a number of tweets. Because attacker’s resources are generally limited and need to be reused, their URL redirect chains usually share the same URLs. Therefore we created a method to detect correlated URL redirect chains using such frequently shared URLs. By analyzing the correlated URL redirect chains and their tweet context information, we discover several features that can be used to classify malicious URLs. We collected a large number of tweets from the Twitter public timeline and trained a statistical classifier using the discovered features. The trained classifier is shown to be accurate and has low false positives and negatives.

II. PROPOSED SYSTEM

Our goal is to develop a malicious URL detection system for Twitter that is robust enough to protect against conditional redirections. Consider a simple example of conditional redirections (Fig.1), in which an attacker creates a long URL redirect chain using a public URL shortening service as well as the attacker’s own private redirection servers used to redirect visitors to a malicious landing page. The attacker then uploads a tweet including the initial URL of the redirect chain to Twitter. Later, when a user or a crawler visits the initial URL, he or she will be redirected to an entry point of the intermediate URLs that are associated with private redirection servers. Some of these redirection servers check whether the current visitor is a normal browser or a crawler. If the current visitor seems to be a normal browser, the servers redirect the visitor to a malicious landing page. If not, they will redirect the visitor to a benign landing
Therefore, the attacker can selectively attack normal users while deceiving investigators.

Therefore the above example shows that, as investigators, we cannot fetch the content of malicious landing URLs, because attackers do not reveal them to us. We also cannot rely on the initial URLs, as attackers can generate a large number of different initial URLs by abusing URL shortening services.

Therefore, if we analyze several correlated redirect chains instead of an individual redirect chain, we can find the entry point of the intermediate URLs in these chains. Consider the three redirect chains shown in the top half of Fig.2 which share some URLs: $A_3=C_3$, $A_4=B_3=C_4$, and $A_6=B_5$. By combining the three redirect chains using these shared URLs, we can generate the correlated redirect chains that share the same entry point URL, $A_4$ (because $A_4$ is the most frequent URL in these chains).

The correlated redirect chains show that the entry point has three different initial URLs and two different landing URLs, and participates in redirect chains that are six to seven URLs long. Even the entry point, $A_4$, does not allow our crawler to visit the latter URLs, we could infer that the chains are malicious because it has many initial URLs for the same landing (entry point in reality) URLs. Therefore, this correlation analysis can help in detecting malicious URLs even when they perform conditional redirections.

III. SYSTEM DESIGN

Our system consists of four components: data collection, feature extraction, training, and classification this can be shown in the below fig.3.

Data Collection: The data collection component has two subcomponents: the collection of tweets with URLs and crawling for URL redirections. To collect tweets with URLs and their context information from the Twitter public timeline, this component uses Twitter Streaming APIs. Whenever this component obtains a tweet with a URL, it executes a crawling thread that follows all redirections of the URL and looks up the corresponding IP addresses. The crawling thread appends these retrieved URL and IP chains to the tweet information and pushes it into a tweet queue. As we have seen, our crawler cannot reach malicious landing URLs when they use conditional redirections to evade crawlers.

Feature Extraction: The feature extraction component has three subcomponents: grouping of identical domains, finding entry point URLs, and extracting feature vectors. This component monitors the tweet queue to determine whether a sufficient number of tweets have been collected. Specifically, our system uses a tweet window instead of individual tweets. When we group domain names or find entry point URLs, we ignore white listed domains to reduce false positive rates. White listed domains are not grouped with other domains and are not selected as entry point URLs.

Training: The training component has two subcomponents: retrieval of account statuses and training of the classifier. Because we use an offline supervised learning algorithm, the feature vectors for training are relatively older than feature vectors for classification. To label the training vectors, we use the Twitter account status; URLs from suspended accounts are considered malicious whereas URLs from active accounts are considered benign. We periodically update our classifier using labeled training vectors.

Classification: The classification component executes our classifier using input feature vectors to classify malicious URLs. When the classifier returns a number of malicious
WatchDog: A Malicious URL Detection System for Social Networks (Twitter)

In this paper, we proposed a new malicious URL detection system for Twitter, called WatchDog. Unlike the conventional systems, WatchDog is robust when protecting against conditional redirection, because it does not rely on the features of malicious landing pages that may not be reachable. Instead, it focuses on the correlations of multiple redirect chains that share the same redirection servers. We introduced new features on the basis of these correlations, implemented a near real-time classification system using these features, and evaluated the system’s accuracy and performance. In the future, we will extend our system to address dynamic and multiple redirections. We will also implement a distributed version of WatchDog to process all tweets from the Twitter public timeline.

IV. CONCLUSION

In this paper, we proposed a new malicious URL detection system for Twitter, called WatchDog. Unlike the conventional systems, WatchDog is robust when protecting against conditional redirection, because it does not rely on the features of malicious landing pages that may not be reachable. Instead, it focuses on the correlations of multiple redirect chains that share the same redirection servers. We introduced new features on the basis of these correlations, implemented a near real-time classification system using these features, and evaluated the system’s accuracy and performance. In the future, we will extend our system to address dynamic and multiple redirections. We will also implement a distributed version of WatchDog to process all tweets from the Twitter public timeline.

V. REFERENCE


Author’s Profile:

Maddali Sai pooja has received B.Tech degree in Computer science and Engineering from Narayana engineering college, Nellore affiliated to JNTU, Anantapur in 2012 and Pursuing M.Tech in Computer Science and Engineering in Annamacharya Institute of Technology and Sciences, Tirupati affiliated to JNTU, Anantapur in 2012-2014.

K Chandra Babu has received B.Tech degree in Computer Science from Vysya College, Salem affiliated to Periyar University in 2003 and M.Tech degree in Computer Science and Engineering from QUBA College of Engineering &Technology, Venkatachalam affiliated to JNTU, Anantapur in 2012. He is dedicated to teaching field from the last 10 years. He has guided 12 P.G and 36 U.G students. His research areas included Non Polynomial Time Algorithms and Computer Graphics. At present he is working as Assistant Professor in Annamacharya Institute of Technology and Sciences, Tirupati, Andhra Pradesh, India.