IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 1, Ver. II (Jan.-Feb. 2017), PP 06-12
www.iosrjournals.org

State of the Art Analysis Approach for Identification of the Malignant URLs

Amruta Rajeev Nagaonkar , Umesh L. Kulkarni
Shivaji University Department of Computer Science & Engineering Kolhapur, India
Mumbai University Department of Computer Engineering Mumbai, India

Abstract: *Malicious URLs have been universally used to ascend various cyber attacks including spamming, phishing and malware. Malware, short term for malicious software, is software which is developed to penetrate computers in a network without the user's permission or notification. Existing methods typically detect malicious URLs of a single attack type. Hence such detection systems are failed to protect the users from various attacks. Malware spreading widely throughout the area of network as consequence of this it becomes predicament in distributed computer and network systems. Malicious links are the place of origin of all attacks which circulated all over the web. Hence malicious URLs should be detected for the prevention of users from these malware attacks. In this paper we described a novel approach which analyze all types of attacks by identifying malicious URLs and secure the web users from them. This technique prevents the users from malignant URLs before visiting them. Therefore efficiency of web security gets maintained. For such anatomization we developed an analyzer which identifies URLs and examine as malicious or benign. We also developed five processes which crawl for suspicious URLs. This approach will prevent the users from all types of attacks and increase efficiency of web crawling phase.*
Keywords: *Malicious URLs, Analyzer, Malware*

I. Introduction

Malicious web content has become the primary instrument. Malware is a common term for a variety type of malicious software. In general, Malwares include Worm, Botnet, virus, Trojan horse, Backdoor, Rootkit, Logic bomb, Rabbit and Spyware used by miscreants to perform their attacks on the Internet.

To address Web-based attacks, a great effort has been directed towards detection of malicious URLs. A common countermeasure is to use a blacklist of malicious URLs, which can be constructed from various sources, particularly human feedbacks that are highly accurate yet time-consuming. Blacklisting incurs no false positives, yet is effective only for known malicious URLs. It cannot detect unknown malicious URLs. This weakness of blacklisting has been addressed by Anomaly based detection methods designed to detect unknown malicious URLs. This paper presents an analysis system for malicious websites before the website is displayed in the browser. In contrast to previous work, this approach is much more general. It can detect many different forms of malicious attacks. Our system combines generality with usability since it is executed directly in real time on the client running the web browser.

The main advantage of this approach is that it is instantly usable by end-users. The approach is also rather flexible since it does not restrict the ways in which malicious websites are analyzed and detected, i.e., it is able to cover a broad range of malicious behaviour. However, the approach relies on a dedicated infrastructure, i.e., crawlers to analyse websites and a central database to which web browsers can connect. A number of approaches have been proposed to detect malicious web pages. Traditional anti-virus tools use static signatures to match patterns that are commonly found in malicious scripts [1].

Unfortunately, the effectiveness of syntactic signatures is thwarted by the use of sophisticated obfuscation techniques that often hide the exploit code contained in malicious pages. Very common approach which is widely used is a blacklisting. The blacklists are particularly human feedbacks that are highly accurate yet time consuming. Blacklisting [4] is effective only for known malicious URLs. Predictably, many malicious sites are not blacklisted either because they are too new, were never evaluated, or were evaluated incorrectly.

Another approach is based on low-interaction honeyclients, which simulate a regular browser and rely on specifications to match the behaviour, rather than the syntactic features, of malicious scripts [2, 3]. A problem with low-interaction honeyclients is that they are limited by the coverage of their specification database; that is, attacks for which a specification is not available cannot be detected. Finally, the state of- the-art in malicious JavaScript detection is represented by high interaction honeyclients.

However, high-interaction client honeypots face some challenges of their own. First, they require considerable resources, in that a dedicated system – a physical machine or a virtualized environment – is necessary for them to function, which has a direct negative impact on the performance and scalability of these systems.

Second, high-interaction client honeypots have a tendency to fail at identifying malicious web pages, producing false negatives that are rooted in the detection mechanism. The client honeypot uses a vulnerable client that interacts with the web server and then makes an assessment based on the state changes that occur following an attack. However, if the attack does not meet a vulnerable client, no state changes occur and the malicious page is therefore not detected. Before using a particular URL if one could inform users that it was dangerous to visit, much of this problem could be solved. On the basis of this to avoid malware attacks, we designed a framework based on data mining technique where analyzer classifies the URLs into malicious and benign precisely. Hence it has ability to find compromised, legitimate URLs or web pages as well as newly formed malicious executables which are directly set up by attackers.

In this paper, a presented framework detects automatically and accurately, previously and also newly generated URLs by attackers. Our goal is to design and build software which has an effective analyzer that accurately detects malicious executables before they execute o n user's machine.

This system consists of three step processes to identify malicious URLs. Firstly suspicious URLs are collected from five functional processes which are examined in depth using analyzer for classifying URLs into malicious and benign, then last step is submitting these malicious URLs to our database

II. System Architecture

A. Proposed System Architecture:

This approach is going to be used for detecting and analyzing URLs on web efficiently for classifying into malicious or not. In proposed work, analyzer classifies web pages into malicious and benign. This also classifies new malicious content added into web pages. To find out other corresponding pages different types of methods has been added for better classification. These collected pages or URLs will be stored in dataset by proxy server. This will avoid direct contact with search engine. Proxy server will act as firewall between user and malicious contents. The following figure indicates proposed system architecture.

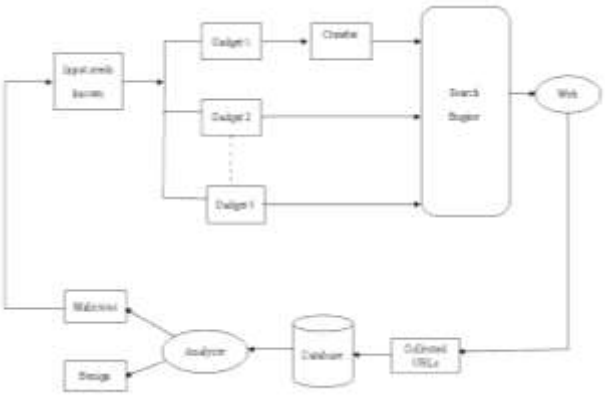


Fig1. Proposed System architecture

B. Developing Processes:

In this section, methods are developed to find correlated suspicious links. These five methods perform the processing steps which gives result as additional malignant contents from search engine through web.

- Following are five methods involved in first methods:

1. Links Process
2. Content Dorks Process
3. Search Engine Optimization Process
4. Domain Registration Process
5. DNS Queries Process

The working of all these methods is different and explained as below:

1. Links Process:

A given link as suspicious URL to this method, it finds the correlated links of it. Later it submitted to analyzer for classification purpose.

2. Content Dork Method:
Content dorks method searches the suspicious URLs by using known malicious keywords. These malicious keywords are taken from Google Hacking Database. All content dorks are yield to search engine for obtaining their links and applying analyzer for classification.

3. Search Engine Optimization Method:
The objective of search engine optimization (SEO) method is to identify a cloaking technique as well as links which show malicious page and its corresponding cloaked page which is set up by attackers.

4. Domain Registration Process:
Domain registration is the process where all the domains must be registered, when they are newly created and uploaded on Internet. This method discovers registration history of input URL. If any link find it as suspicious then this method flagged to not only that domain but domains which are before and after of that domain also.

5. DNS Queries Process:
Domain name system is the process which provides global mapping between IP addresses and domain names. it discovers the path of DNS requests to acquire a suspicious or malicious domain or URL from traced path.

III. Development of Analyzer for Classifying URLS

In this section, analyzer is developed to classify URLs into benign and malicious. For this classification purpose lexical and host- based features are utilized. Finally machine learning classifier is used for analysis like Naive Bayes.

A. URL Lexical features:

Malicious URLs look different to users who see them. Hence URL parsed to retrieve hostname and path name for classification purpose. It consist of two lists i.e. malicious URLs and benign URLs.
Following are the characteristics which have chosen and included in lexical features:
1. URL parsed to retrieve hostname and the path.
2. Delimiters used like (strings delimited by '.', '/', '?', '.', '=', '-' and '_').
3. Length of URL
4. Length of hyphen
5. Length of domain
6. Domain name extension
7. Length of max-length in domain name
8. Length of directory

After this, comparison takes place by calculating above properties of malicious as well as benign URLs with each other. It gives result as input URL is malicious or benign.

- **Naïve Bayes Theorem:**

It is commonly used in spam filters. It is a probabilistic method based on applying Bayes theorem. It describes the probability of an event, based on conditions that might be related to the event. Specifically, to compute the class of a program given that the program contains a set of features F, C to be a random variable over the set of classes: benign and malicious executables. To compute P , the probability that a program is in a certain class given the program contains the set of features. Applying Bayes rule and express the probability is

$$P(C/F) = \frac{\mathrm{P(F/C)} * \mathrm{P(C)}}{P(F)}$$

B. URL Host- based features:

There are number of host based features but here following are considered:
1. Check for blacklist IP addresses.
2. Date of registration, update and expiration.
3. Check value of the time-to-live (TTL) for the DNS records associated with the hostname.
4. Geographical location of the IP address belongs.
5. Check for who is registrar and registrant.

This approach gives appropriate result for classifying URLs as either malicious or benign based on both lexical and host-based features.

By applying both lexical and host based features on URLs, they appropriately analyzed into malicious and benign URLs.
After this implementation of analyzer we have collected final malicious URLs dataset which we have submitted to our implemented proxy server. This proxy server acts as firewall between user and malicious URLs. Proxy servers have mainly two purposes:

1. ***Improve performance:***
   Proxy servers can dramatically improve performance for groups of users. This is because it saves the results of all requests for a certain amount of time.
2. ***Filter Requests:***
   Proxy servers can also be used to filter requests. For example, a company might use a proxy server to prevent its employees from accessing a specific set of Web sites.

- **Working of Proxy Server:**

- The proxy server evaluates the request means the proxy provides the resource by connecting to the relevant server and requesting the service on behalf of the client.
- Proxy sites enable you to bypass your own Internet provider and browse through the proxy web site.
- If the URL matches a pattern or a site that requires a proxy, it will connect to the proxy server rather than going directly to the site.
- Check it in to the dataset as well as submit to the analyzer to classify the URLs as malicious and benign.
- After above step identified malicious URLs submit to all methods to generate all corresponding URLs for gathering more malicious executables.

- **Security Advantages:**

- Blocking of dangerous URLs
- Filter dangerous content
- Eliminate need for transport layer routing between networks
- Single point of access, control and logging

IV. Experimental Evaluation

In this section, we experimentally validate our initial hypothesis used in our project is effective by identifying as well as blocking malignant URLs and it does so in an effective manner.
We have used two key metrics to establish the effectiveness of our system is toxicity and accuracy.
Toxicity is calculated by number of suspicious URLs submitted by all methods to analyzer.
Accuracy is measured by accuracy (ACC) which is the proportion of true results (both true positives and true negatives) over all data sets; true positive rate (TP, also referred to as recall) which is the number of the true positive classifications divided by the number of positive examples; false positive rate (FP) and false negative rate (FN) which are defined similarly.

A. Links Process:

Total 2684 suspicious URLs are collected by this method, from which 233 URLS are identified as pure malignant by our analyzer. Hence toxicity of this process is 0.0868.

B. Content dorks Process:

This method contains all malign keywords which were taken from Google Hacking Database. Hence all corresponding URLs from this method are malicious. This process have found total 1012 URLs and submitted to our analyzer, from which 669 URLs are found as malignant. 0.66.

C. S.E.O Process:

This process found 35 web sites which are cloacked and after submitted to these URLs to analyzer, it classified as 15 URLs are cloacked web links.

D. Domain Registration Process:

At the first time this process did not find any URL, but after some days we examined that this process found 18 URLs as suspicious from which 7 URLs are classified as malicious.

E. DNS Queries Process:

This process found 16 URLs from which 5 URLs are malign URLs.
Hence we examined that these count of malicious URLs is changed as these processes found URLs and their count also changed as input given to them.

Overall, this experiment shows that this system clearly outperforms in toxicity 1.34%.

- **Detection Accuracy:**

1. True Positives (TP), the number of malicious executable examples classified as malicious executables.
2. True Negatives (TN), the number of benign programs classified as benign.
3. False Positives (FP), the number of benign programs classified as malicious executables.
4. False Negatives (FN), the number of malicious executables classified as benign binaries.

By applying the discriminative features on the datasets our malicious URL detector produced the following results:

- **Lexical feature Analysis:**

Lexical features are the textual properties of the URL itself. These properties include the length of the hostname, the length of the entire URL, as well as the number of dots in the URL, maximum length of domain, count of hyphens in domain name, length of directories.
After measurement of these properties, input URL is trained and classified into benign or malignant by applying Naive Bayes classifier.
Naive Bayes is a classifier that sees wide-spread use in spam filters and related security applications, in part because the training and testing performance of Naive Bayes is fast.
However, the benefit of reduced training time is outweighed in this case by the benefit of using classifiers whose explicit goal is to minimize errors. This trade off is particularly worthwhile when dealing with a large feature set.
Following explanations show both true positive and negative as well as false positives and negative rates:

- **True positive and Negative:**

Total 100 malicious and benign URLs have submitted to the lexical analyzer from which it successfully classified these URLs into benign and malicious URLs with true positive and negative rates.

- **False positives:**

A false positive is when a benign URL is misclassified as malicious.
Apart from these datasets, we have found other types of URLs on Internet which were misclassified as malicious by other detectors but in actual they are benign. There are different types of URLs are categorized to trap the users, like disreputable, contentless, abnormal token, brand name URLs. Such types of URLs we submitted to our Lexical analyzer to check false positive rate.

For this we have chosen URLs from each category and submitted to Lexical analyzer hence we got following result:

- **Disreputable URLs:**

A benign URL is likely misclassified by our detector if it fits into two or more of the following three cases: 1) the URL's domain has a very low link popularity (LPOP errors), 2) the URL contains a malicious SLD (LEX errors), and 3) the URL's domain is hosted by malicious ASNs (DNS errors). In this case, a benign URL can be considered as a disreputable URL. (ex: : 208.43.27.50/˜mike).
But our analyzer shows correct result as expected.

- **Contentless URLs:**

Some benign URLs had no content on their web pages. In this case (*e.g.*, 222.191.251.167, 1traf.com, and 3gmatrix.cn). But our analyzer shows correct result as expected for this contentless URLs.

- **Brand name URL:**

Some benign URLs contained a brand name keyword even they were not related to the brand domain. These URLs could be misclassified as malicious (*e.g.*, twitterfollower.wikispaces.com).
But for this type of URL analyzer misclassified.

- **Abnormal token URL:**

We observed several benign URLs which had unusual long domain tokens typically appearing in phishing URLs (*e.g.*,8centraldevideoscomhomensmaduros.blogspot.com).
But for abnormal token URL analyzer misclassified.

➢ **False Negative:**

A false negative is when a malicious URL is undetected.

Most of the false negative URLs are of spam or phishing type URLs. They generated features similar to those of benign URLs.

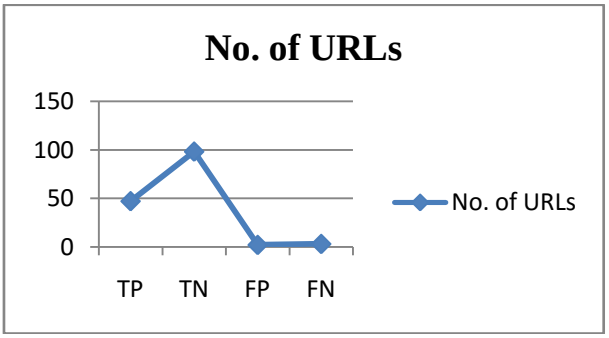
Hence we have taken some examples and submitted to our analyzer. Analyzer gave accurate results. More than 95% detectors failed to identify such URLs like blog.libero.it/matteof97/ and digilander.libero.it/Malvin92/?For analysis we have given these URLs to our lexical analyzer and we got correct results.

- **Lexical Analyzer Accuracy:**

Accuracy	**No of URLs**
TP	48
TN	97
FP	2
FN	3

Table 2.shows accuracy of lexical Analyzer

- **Lexical Analyzer Graph:**



Graph1. Lexical Analyzer

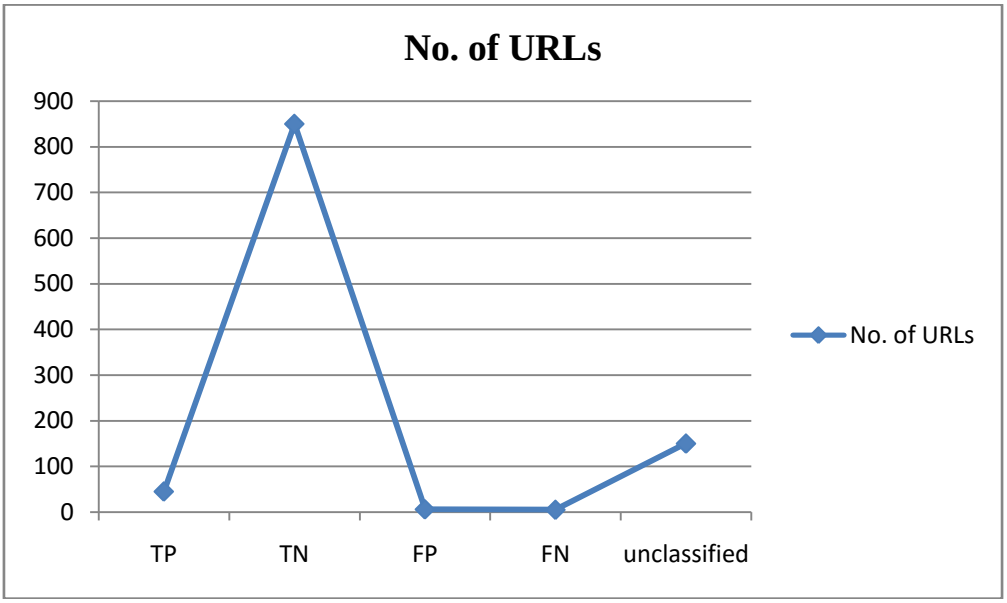
- **Host based analyzer Accuracy:**

This analyzer searched all the host features of URLs. We have collected 3190 blacklisted URLs, from which 1205 URLs are submitted to the analyzer to check. According to this analyzer it scrutinized 850 URLs as malignant and 6 URLs are misclassified as malicious and five URLs are misclassified as benign. Remained 150 URLs are found as their access was denied.

Accuracy	**No of URLs**
TP	44
TN	850
FP	6
FN	5

Table 3.shows accuracy of host-based Analyzer

Host based analyzer Graph:



Graph2. Host- based Analyzer

Conclusion

Using one malicious URL, the web is crawled so as to obtain a big dataset of suspicious URLs. The suspicious URLs are retrieved using googlebot, keywords, S.E.O techniques and domain name services.
These suspicious URLs are analyzed and are declared whether they are benign or malicious, and the results are found to be mostly correct on observation.

References

[1]. ClamAV. Clam AntiVirus. http://www.clamav.net
[2]. J. Nazario. PhoneyC: A Virtual Client Honeypot.In Proceedings of the USENIX Workshop on Large Scale Exploits and Emergent Threats, 2009.
[3]. A. Ikinci, T. Holz, and F. Freiling. Monkey Spider: Detecting Malicious Websites with Low-Interaction Honeyclients. In Proceedings of Sicherheit, Schutz und Zuverlässigkeit, April 2008.
[4]. C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages", In Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS'10),San Diego, CA, Mar 2010.
[5]. Luca Invernizzi, Santa Barbara,Stefano Benvenuti,Paolo Milani,Comparetti Lastline,Vienna, "EVILSEED: A Guided Approach to Finding Malicious Web Pages," in IEEE Symposium on Security and Privacy 20-23 May 2012, pp 428 – 442.