Detecting Malicious Web Links and Identifying Their Attack Types

Hyunsang Choi
Korea University
Seoul, Korea
realchs@korea.ac.kr

Bin B. Zhu
Microsoft Research Asia
Beijing, China
binzhu@microsoft.com

Heejo Lee
Korea University
Seoul, Korea
heejo@korea.ac.kr

Abstract

Malicious URLs have been widely used to mount various cyber attacks including spamming, phishing and malware. Detection of malicious URLs and identification of threat types are critical to thwart these attacks. Knowing the type of a threat enables estimation of severity of the attack and helps adopt an effective countermeasure. Existing methods typically detect malicious URLs of a single attack type. In this paper, we propose method using machine learning to detect malicious URLs of all the popular attack types and identify the nature of attack a malicious URL attempts to launch. Our method uses a variety of discriminative features including textual properties, link structures, webpage contents, DNS information, and network traffic. Many of these features are novel and highly effective. Our experimental studies with 40,000 benign URLs and 32,000 malicious URLs obtained from real-life Internet sources show that our method delivers a superior performance: the accuracy was over 98% in detecting malicious URLs and over 93% in identifying attack types. We also report our studies on the effectiveness of each group of discriminative features, and discuss their evadability.

1 Introduction

While the World Wide Web has become a killer application on the Internet, it has also brought in an immense risk of cyber attacks. Adversaries have used the Web as a vehicle to deliver malicious attacks such as phishing, spamming, and malware infection. For example, phishing typically involves sending an email seemingly from a trustworthy source to trick people to click a URL (Uniform Resource Locator) contained in the email that links to a counterfeit webpage.

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. The very nature of exact match in blacklisting renders it easy to be evaded.

This weakness of blacklisting has been addressed by anomaly-based detection methods designed to detect unknown malicious URLs. In these methods, a classification model based on discriminative rules or features is built with either knowledge a priori or through machine learning. Selection of discriminative rules or features plays a critical role for the performance of a detector. A main research effort in malicious URL detection has focused on selecting highly effective discriminative features. Existing methods were designed to detect malicious URLs of a single attack type, such as spamming, phishing, or malware.

In this paper, we propose a method using machine learning to detect malicious URLs of all the popular attack types including phishing, spamming and malware infection, and identify the attack types malicious URLs attempt to launch. We have adopted a large set of discriminative features related to textual patterns, link structures, content composition, DNS information, and network traffic. Many of these features are novel and highly effective. As described later in our experimental studies, link popularity and certain lexical and DNS features are highly discriminative in not only detecting malicious URLs but also identifying attack types. In addition, our method is robust against known evasion techniques such as redirection [42], link manipulation [16], and fast-flux hosting [17].

Identification of attack types is useful since the knowledge of the nature of a potential threat allows us to take a proper reaction as well as a pertinent and effective countermeasure against the threat. For example, we may conveniently ignore spamming but should respond immediately to malware infection. Our experiments on 40,000 benign URLs and 32,000 malicious URLs obtained from real-life Internet sources show that our method has achieved an accuracy rate of more than 98% in detecting malicious URLs and an accuracy rate

This work was done when Hyunsang Choi was an intern at Microsoft Research Asia. Contact author: Bin B. Zhu (binzhu@ieee.org).

of more than 93% in identifying attack types.

This paper has the following major contributions:

- We propose several groups of novel, highly discriminative features that enable our method to deliver a superior performance and capability on both detection and threat-type identification of malicious URLs of main attack types including spamming, phishing, and malware infection. Our method provides a much larger coverage than existing methods while maintaining a high accuracy.

- To the best of our knowledge, this is the first study on classifying multiple types of malicious URLs, known as a multi-label classification problem, in a systematic way. Multi-label classification is much harder than binary detection of malicious URLs since multi-label learning has to deal with the ambiguity that an entity may belong to several classes.

The remainder of this paper is organized as follows. We present the proposed method and the learning algorithms it uses in Section 2, and describe the discriminative features our method uses in Section 3. Evaluation of our method with real-life data is reported in Section 4. We review related work in Section 5, and conclude the paper in Section 6.

2 Our Framework

2.1 Overview

Our method consists of three stages as shown in Figure 1: training data collection, supervised learning with the training data, and malicious URL detection and attack type identification. These stages can operate sequentially as in batch learning, or in an interleaving manner: additional data is collected to incrementally train the classification models while the models are used in detection and identification. Interleaving operations enable our method to adapt and improve continuously with new data, especially with online learning where the output of our method is subsequently labeled and used to train the classification models.

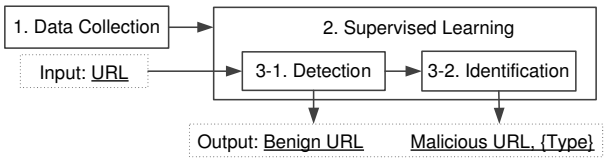

Figure 1: The framework of our method.

2.2 Learning Algorithms

The two tasks performed by our method, detecting malicious URLs and identifying attack types, need different machine learning methods. The first task is a binary classification problem. The Support Vector Machine (SVM) is used to detect malicious URLs. The second task is a multi-label classification problem. Two multi-label classification methods, (RA*k*EL [38] and ML-*k*NN [48]), are used to identify attack types.

Task1: Support Vector Machine (SVM). SVM is a widely used machine learning method introduced by Vapnik *et al.* [8]. SVM constructs hyperplanes in a high or infinite dimensional space for classification. Based on the Structural Risk Maximization theory, SVM finds the hyperplane that has the largest distance to the nearest training data points of any class, called *functional margin*. Functional margin optimization can be achieved by maximizing the following equation

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, ..., n$$

where α_i and α_j are coefficients assigned to training samples x_i and x_j. $K(x_i, x_j)$ is a kernel function used to measure similarity between the two samples. After specifying the kernel function, SVM computes the coefficients which maximize the margin of correct classification on the training set. C is a regulation parameter used for tradeoff between training error and margin, and training accuracy and model complexity.

Task2: RA*k*EL. and ML-*k*NN. RA*k*EL is a high-performance multi-label learning method that accepts any multi-label learner as a parameter. RA*k*EL creates *m* random sets of *k* label combinations, and builds an ensemble of Label Powerset (LP) [47] classifiers from each of the random sets. LP is a transformation-based algorithm that accepts a single-label classifier as a parameter. It considers each distinct combination of labels that exists in the training set as a different class value of a single-label classification task. Ranking of the labels is produced by averaging the zero-one predictions of each model per considered label. An ensemble voting process under a threshold *t* is then employed to make a decision for the final classification set. We use C4.5 [32] as the single-label classifier and LP as a parameter of the multi-label learner.

ML-*k*NN is derived from the traditional *k*-Nearest Neighbor (*k*NN) algorithm [1]. For each unseen instance, its k nearest neighbors in the training set are first identified. Based on the statistical information gained from the label sets of these neighboring instances, maximum a posteriori principle is then utilized to determine the label set for the unseen instance.

3 Discriminative Features

Our method uses the same set of discriminative features for both tasks: malicious URL detection and attack type identification. These features can be classified into six groups: lexicon, link popularity, webpage content, DNS, DNS fluxiness, and network traffic. They can effectively represent the entire *multifaceted* properties of a malicious URL and are robust to known evasion techniques.

3.1 Lexical Features

Malicious URLs, esp. those for phishing attacks, often have distinguishable patterns in their URL text. Ten lexical features, listed in Table 1, are used in our method. Among these lexical features, the average domain/path token length (delimited by '.', '/', '?', '=', '-', '_') and brand name presence were motivated from a study by McGrath and Gupta [24] that phishing URLs show different lexical patterns. For example, a phishing URL likely targets a widely trusted brand name for spoofing, thus contains the brand name. Therefore, we employ a binary feature to check whether a brand name is contained in the URL tokens but not in its SLD (Second Level Domain)[1].

Table 1: Lexical features (LEX)

No.	Feature	Type
1	Domain token count	Integer
2	Path token count	Integer
3	Average domain token length	Real
4	Average path token length	Real
5	Longest domain token length	Integer
6	Longest path token length	Integer
7∼9	Spam, phishing and malware SLD hit ratio	Real
10	Brand name presence	Binary

In our method, the detection model maintains two lists of URLs: a list of benign URLs and a list of malicious URLs. The identification model breaks the list of malicious URLs into three lists: spam, phishing, and malware URL lists. For a URL, our method extracts its SLD and calculates the ratio of the number that the SLD matches SLDs in the list of malicious URLs or a list of specific type of malicious URLs (e.g., spam URL list) to the number that the SLD matches SLDs in the list of benign URLs. This ratio is called the *malicious* or a specific attack type (e.g., *spam*) *SLD hit ratio* feature, which is actually an *a priori* probability of the URL to be malicious or of a specific malicious type (e.g., spam) based on the precompiled URL lists.

Previous methods use URL tokens as the "bag-of-words" model in which the information of a token's position in a URL is lost. By examining a large set of malicious and benign URLs, we observed that the position of a URL token also plays an important role. SLDs are relatively hard to forge or manipulate than URL tokens at other positions. Therefore, we discard the widely used "bag-of-words" approach and adopt several new features differentiating SLDs from other positions, resulting in a higher robustness against lexical manipulations by attackers. Lexical features No. 1 to No. 4 in Table 1 are from previous work. Feature No. 10 is different from the "bag-of-words" model used in previous work by excluding the SLD position. The other lexical features in Table 1 are novel features never used previously.

3.2 Link Popularity Features

One of the most important features used in our method is "link popularity", which is estimated by counting the number of incoming links from other webpages. Link popularity can be considered as a reputation measure of a URL. Malicious sites tend to have a small value of link popularity, whereas many benign sites, especially popular ones, tend to have a large value of link popularity. Both link popularity of a URL and link popularity of the URL's domain are used in our method. Link popularity (LPOP) can be obtained from a search engine[2]. Different search engines may produce different link popularity due to different coverage of webpages each has crawled. In our method, five popular search engines, `Altavista`, `AllTheWeb`, `Google`, `Yahoo!`, and `Ask`, are used to calculate the link popularity of a URL and the link popularity of its domain, corresponding to LPOP features No. 1 to 10 in Table 2.

One problem in using link popularity is "link-farming [16]", a link manipulation that uses a group of webpages to link together. To address this problem, we develop five additional LPOP features by exploiting different link properties between link-manipulated malicious websites and popular benign websites. The first feature, the distinct domain link ratio, is the ratio of the number of unique domains to the total number of domains that link to the targeted URL. The second feature, the max domain link ratio, is the ratio of the maximum number of links from a single domain to the total number of domains that link to the targeted URL. Link-manipulated malicious URLs tend to be linked many times with a few domains, resulting in a low score on the distinct domain link ratio and a high score on the max domain link ratio. A study by Castillo *et al.* [4] indicates that spam pages tend to be linked mainly by spam pages. We believe that a hypothesis to assume that not only spam pages, but also phishing and malware pages tend to be linked by phishing and malware pages, respectively, is plausible. Therefore, we develop the last three features: spam link ratio, phishing link ratio, and malware link ratio. Each represents the ratio from domains of a specific malicious type that link to the targeted URL. To measure these three features, we use the malicious URL lists described in Section 3.1. The link popularity features described in this subsection are all novel

[1] Brand names can be taken from the SLDs of the Alexa [2] top 500 site list.

[2] For example, we can use *Yahoo! site explorer* to get inlinks of target URLs.

features.

Table 2: Link popularity features (LPOP)

No.	Feature	Type
1∼5	5 LPOPs of the URL	Integer
6∼10	5 LPOPs of the domain	Integer
11	Distinct domain link ratio	Real
12	Max domain link ratio	Real
13∼15	Spam, phishing and malware link ratio	Real

3.3 Webpage Content Features

Recent development of the dynamic webpage technology has been exploited by hackers to inject malicious code into webpages through importing and thus hiding exploits in webpage content. Therefore, statistical properties of client-side code in the Web content can be used as features to detect malicious webpages. To extract webpage content features (CONTs), we count the numbers of HTML tags, iframes, zero size iframes, lines, and hyperlinks in the webpage content. We also count the number for each of the following seven suspicious native JavaScript functions: escape(), eval(), link(), unescape(), exec(), link(), and search() functions. As suggested by a study of Hou *et al.* [18], these suspicious JavaScript functions are often used by attacks such as cross-site scripting and Web-based malware distribution. For example, unescape() can be used to decode an encoded shellcode string to obfuscate exploits. The counts of these seven suspicious JavaScript functions form features No. 6 to No. 12 in Table 3. The last feature in this table is the the sum of these function counts, i.e., the total count of these suspicious JavaScript functions. All the features in Table 3 are from the previous work [18].

Table 3: Webpage content features (CONT)

No.	Feature	Type
1	HTML tag count	Integer
2	Iframe count	Integer
3	Zero size iframe count	Integer
4	Line count	Integer
5	Hyperlink count	Integer
6∼12	Count of each suspicious JavaScript function	Integer
13	Total count of suspicious JavaScript functions	Integer

The CONTs may not be effective to distinguish phishing websites from benign websites because a phishing website should have similar content as the authentic website it targets. However, this very nature of being sensitive to one malicious type but insensitive to other malicious types is very much desired in identifying the type of attack that a malicious URL attempts to launch.

3.4 DNS Features

The DNS features are related to the domain name of a URL. Malicious websites tend to be hosted by less reputable service providers. Therefore, the DNS information can be used to detect malicious websites. Ramachandran *et al.* [33] showed that a significant portion of spammers came from a relatively small collection of autonomous systems. Other types of malicious URLs are also likely to be hosted by disreputable providers. Therefore, the Autonomous System Number (ASN) of a domain can be used as a DNS feature.

Table 4: DNS features (DNS)

No.	Feature	Type
1	Resolved IP count	Integer
2	Name server count	Integer
3	Name server IP count	Integer
4	Malicious ASN ratio of resolved IPs	Real
5	Malicious ASN ratio of name server IPs	Real

All the five DNS features listed in Table 4 are novel features. The first is the number of IPs resolved for a URL's domain. The second is the number of name servers that serves the domain. The third is the number of IPs these name servers are associated with. The next two features are related to ASN. As we have mentioned in Section 3.1, our method maintains a benign URL list and a malicious URL list. For each URL in the two lists, we record its ASNs of resolved IPs and ASNs of the name servers. For a URL, our method calculates hit counts for ASNs of its resolved IPs that matches the ASNs in the malicious URL list. In a similar manner, it also calculates the ASN hit counts using the benign URL list. Summation of malicious ASN hit counts and summation of benign ASN hit counts are used to estimate the malicious ASN ratio of resolved IPs, which is used as an *a priori* probability for the URL to be hosted by a disreputable service provider based on the precompiled URL lists. ASNs can be extracted from MaxMind's database file [14].

3.5 DNS Fluxiness Features

A newly emerging fast-flux service network (FFSN) establishes a proxy network to host illegal online services with a very high availability [17]. FFSNs are increasingly employed by attackers to provide malicious content such as malware, phishing websites, and spam campaigns. To detect URLs which are served by FFSNs, we use the discriminative features proposed by Holz *et al.* [17], as listed in Table 5.

Table 5: DNS fluxiness features (DNSF)

No.	Feature	Type
1∼2	φ of N_{IP}, N_{AS}	Real
3∼5	φ of N_{NS}, N_{NSIP}, N_{NSAS}	Real

We lookup the domain name of a URL and repeat the DNS lookup after TTL (Time-To-Live value in a DNS packet) timeout given in the first answer to have consecutive lookups of the same domain. Let N_{IP} and N_{AS} be

the total number of unique IPs and ASNs of each IP, respectively, and N_{NS}, N_{NSIP}, N_{NSAS} be the total number of unique name servers, name server IPs, and ASNs of the name server IPs in all DNS lookups. Then, we can estimate *fluxiness* using the acquired numbers. For example, *fluxiness* of the resolved IP address is estimated as follows,

$$\varphi = N_{IP}/N_{single},$$

where φ is the *fluxiness* of the domain and N_{single} is the number of IPs that a single lookup returns. Similarly, all of the other *fluxiness* features are estimated.

3.6 Network Features

Attackers may try to hide their websites using multiple redirections such as iframe redirection and URL shortening. Even though also used by benign websites, the distribution of redirection counts of malicious websites is different from that of redirection counts of benign websites [31]. Therefore, redirection count can be a useful feature to detect malicious URLs. In a HTTP packet, there is a content-length field which is the total length of the entire HTTP packet. Hackers often set malformed (negative) content-length in their websites in a buffer overflow exploit. Therefore, content-length is used as a network discriminative feature. Benign sites tend to be more popular with a better service quality than malicious ones. Web technologies tend to make popular websites quick to look up and faster to download. In particular, benign domains tend to have a higher probability to be cached in a local DNS server than malicious domains, esp. those employing FFSNs and dynamic DNS. Therefore, domain lookup time and average download speed are also used as features to detect malicious URLs. The network features listed in Table 6 except the third and fifth features are novel features.

Table 6: Network features (NET)

No.	Feature	Type
1	Redirection count	Integer
2	Downloaded bytes from content-length	Real
3	Actual downloaded bytes	Real
4	Domain lookup time	Real
5	Average download speed	Real

4 Evaluation

In this section, we evaluate the performance of our method for both malicious URL detection and attack type identification. We also study the effectiveness of different groups of features. The main findings of our experiments include:

- **Link popularity.** Link popularity first used in our method is highly discriminative for both malicious URL detection (over 96% accuracy) and attack type identification (over 84% accuracy). `Google`'s search engine was not suitable to estimate link popularity since it reported just a partial list of link popularity.
- **Link distribution.** Malicious URLs are mainly linked by malicious URLs of the same attack type: about 56% of malicious URLs were found to be linked only by the malicious URLs of the same attack type.
- **Multi-labels.** In our collected malicious URLs, over 45% belong to multiple types of threat. Therefore, malicious URLs should be classified with a multi-label classification method in order to produce a more accurate result on the nature of attack.
- **Identification.** Our method has achieved an accuracy rate of over 93% in attack type identification. It is worth mentioning that novel features used in our method including malicious SLD hit ratio in LEX, three malicious link ratios in LPOP, two malicious ASN ratios in DNS were found to be highly effective in distinguishing different attack types.

4.1 Methodology and Data Sets

Real-life data was collected from various sources to evaluate our method:

- **Benign URLs.** 40,000 benign URLs were collected from the following two sources as used in previous work [49, 43, 21, 22]: 1) randomly selected 20,000 URLs from the DMOZ Open Directory Project [10] (manually submitted by users), 2) randomly selected 20,000 URLs from `Yahoo!`'s directory (generated by visiting `http://random.yahoo.com/bin/ryl`)[3].
- **Spam URLs.** The spam URLs were acquired from jwSpamSpy [19] which is known as an e-mail spam filter for Microsoft Windows. We also used a publicly available Web spam dataset [3].
- **Phishing URLs.** The phishing URLs were acquired from `PhishTank` [29], a free community site where anyone can submit, verify, track and share phishing data.
- **Malware URLs.** The malware URLs were obtained from `DNS-BH` [11], a project creates and maintains a list of URLs that are known to be used to propagate malware.

The data set of malicious URLs is simply the union of the three individual data sets of malicious types. A total of 32,000 malicious URLs was collected. A malicious URL may launch multiple types of attack, *i.e.*, belongs to multiple malicious types. The malicious data sets collected above were marked with only single labels. URLs

[3]Many URLs from 1) and 2) did not have any sub-path. We adjusted the ratio of benign URLs with a sub-path to be half of benign URLs.

of multi-labels were found by querying both McAfee SiteAdvisor[4] [23] and WOT[5] (Web of Trust) [41] for each URL in the malicious URL data set. The two sites provide reputation of a submitted website URL including the detailed malicious types it belongs to. Their information was relatively accurate, although they had made errors (*e.g.*, SiteAdvisor has incorrectly labeled websites[6] and WOT was manipulated by attackers to generate incorrect labels[7]). We use (λ_i) with a single index i to represent a single type: spam (λ_1), phishing (λ_2), malware (λ_3). Multi-labels are represented by the set of their associated indexes, *e.g.*, $\lambda_{1,3}$ represents a URL of both spam and malware. Table 7 shows the resulting distribution of multi-label URLs, where L_{SAd} and L_{WOT} represent the results reported by SiteAdvisor and WOT, respectively, and L_{Both} denotes their intersection. From Table 7, about half of the malicious URLs were classified to be multi-labels: 45% by SiteAdvisor and 46% by WOT. Comparing the labeling results by both L_{SAd} and L_{WOT}, 91% of the URLs were labeled consistently whereas 9% of URLs were labeled inconsistently by the two sites.

Table 7: The collected data set of multi-labels

Label	Attribute	L_{SAd}	L_{WOT}	L_{Both}
λ_1	spam	6020	6432	5835
λ_2	phishing	1119	1067	899
λ_3	malware	9478	8664	8105
$\lambda_{1,2}$	spam, phishing	4076	4261	3860
$\lambda_{1,3}$	spam, malware	2391	2541	2183
$\lambda_{2,3}$	phishing, malware	4729	4801	4225
$\lambda_{1,2,3}$	spam, phishing, malware	2219	2170	2080

Once the URL data sets were built, three crawlers were used to crawl features from different sources. A webpage crawler crawled the webpage content features and the network features by accessing each URL in the data sets. We implemented a module to the webpage crawler using the cURL library [9] to detect redirections (including URL shortening) and find original URLs automatically. A link popularity crawler crawled the link popularity features from the five search engines, Altavista, AllTheWeb, Google, Yahoo!, and Ask, for each URL and collected inlink information. A DNS crawler crawled and calculated the DNS features and DNS fluxiness features by sending queries to DNS servers.

Two-fold cross validation was performed to evaluate our method: the URLs in each data set were randomly split into two groups of equal size: one group was selected as the training set while the other was used as the testing set. Ten rounds of two-fold cross validation were used to obtain the performance for both malicious URL detection and attack type identification. The SVM-light [35] software package was used as the support vector machine implementation in our evaluation.

[4]The SiteAdvisor is a service to report safety of websites using a couple of webpage analysis algorithms.

[5]The WOT is a community-based safe surfing tool that calculates the reputation of a website through a combination of user ratings and data from trusted sources.

[6]http://en.wikipedia.org/wiki/McAfee_SiteAdvisor

[7]http://mashable.com/2007/12/04/web-of-trust/

4.2 Malicious URL Detection Results

The following metrics were used to evaluate the detection performance: 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.

4.2.1 Detection Accuracy

By applying all the discriminative features on the data sets described in Section 4.1, our malicious URL detector produced the following results: 98.2% for the accuracy, 98.9% for the true positive rate, 1.1% for the false positive rate, and 0.8% for the false negative rate. We also conducted the same experiments using only the 20,000 benign URLs collected from Yahoo!'s directory. The results were similar: 97.9% for the accuracy, 98.2% for the true positive rate, 0.98% for the false positive rate, and 1.08% for the false negative rate.

To study the effectiveness of each feature group, we performed detection using only each individual feature group. The resulting accuracy and true positive rate are shown in Figure 2. We can clearly see in this figure that LPOP is the most effective group of features in detecting malicious URLs in terms of both detection accuracy and true positive rate.

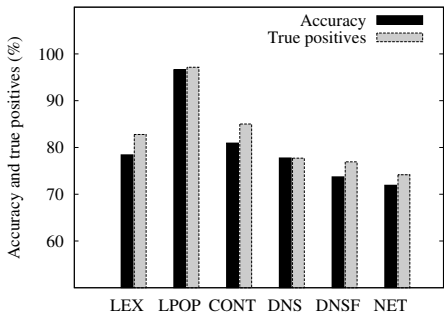

Figure 2: Detection accuracy and true positives for each group of features.

We also compared the performance of each feature group on detecting each type of malicious URLs by mixing the corresponding malicious URL data set with the benign URL data set. The resulting accuracies and true positive rates are shown in Table 8.

As expected, the lexical features (LEX) are effective on detecting phishing URLs, but did a poor job to detect spam and malware URLs. This is because the latter types do not show very different textual patterns as

Table 8: Detection accuracy and true positive rate (%) of individual feature groups for each malicious type

Dataset	Metric	Feature group					
		LEX	LPOP	CONT	DNS	DNSF	NET
Spam	ACC	73.0	97.2	82.8	77.4	**87.7**	72.1
	TP	72.4	97.4	74.2	75.9	**86.3**	77.4
Phishing	ACC	**91.6**	98.1	77.3	76.3	71.8	77.2
	TP	**86.1**	95.1	82.8	76.9	70.1	78.2
Malware	ACC	70.3	96.2	**86.2**	78.6	68.1	73.3
	TP	74.5	93.2	**88.4**	75.1	74.2	78.2

compared with benign URLs. A different sensitivity to a different malicious type is exactly what we want to distinguish one malicious type from other malicious types (phishing from spam and malware for the specific case of lexical features) in the attack type identification to be reported in Section 4.3. These partially discriminative features (effective only for some types of attack) and the features that are effective for all the malicious types form the set of discriminative features for our malicious URL detector.

The link popularity features (LPOP) outperformed all the other groups of features for detecting any type of malicious URLs. Table 8 shows that the webpage content features (CONT) are useful in distinguishing malware URLs from benign ones. This is because malware URLs usually have malicious tags or scripts in their Web content to infect visitors. From Table 8, it seems that the webpage content features are also effective in detecting spam and phishing URLs as malicious URLs from a mixture of malicious and benign URLs. That might be partially due to the fact that many spam or phishing URLs also belonged to malware, as we have seen in Section 4.1. Note that a URL is claimed to be malicious no matter which malicious type it is detected to belong to.

From Table 8, the DNS fluxiness features (DNSF) were effective to detect spam URLs. This should be due to the fact that FFSNs were widely used by spam campaigns, as shown by Moore *et al.* [25]. Malicious network behaviors such as redirections using multiple proxies can be employed by any type of threat. That can explain similar performance of the network features (NET) for detecting each type of malicious URLs.

4.2.2 Link Popularity Feature Analysis

In this section, we study the effectiveness of the link popularity features in detail, and show the effectiveness of our method for two unfavorable scenarios when the link popularity features are not effective:1) the case when malicious websites have high manipulated popularity scores; and 2) the case when newly-setup benign websites do not have high popularity scores.

First, we studied the distribution of the link popularity for each data set. In our data sets, malicious URLs had typically much smaller LPOP than benign URLs. A majority, more precisely 60.35%, of the malicious URLs had 0 link popularity. On the other hand, only a very small portion of benign URLs had almost 0 link popularity. This confirms the observation in Section 4.2.1 that LPOP is effective to differentiate malicious URLs from benign URLs.

Next we studied the quality of the link popularities retrieved from the five different search engines: `Altavista`, `AllTheWeb`, `Google`, `Yahoo!`, and `Ask`. The distribution of LPOP for each search engine over 20,000 benign URLs randomly selected from the collected 40,000 benign URLs is shown in Figure 3, and the distribution over the 32,000 malicious URLs is shown in Figure 4. The x-axis in both figures is the index of the URLs sorted by the link popularity.

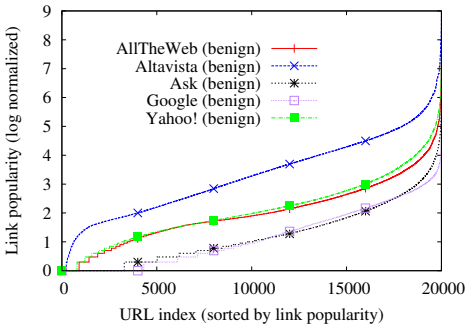

Figure 3: LPOP of benign URLs for each search engine.

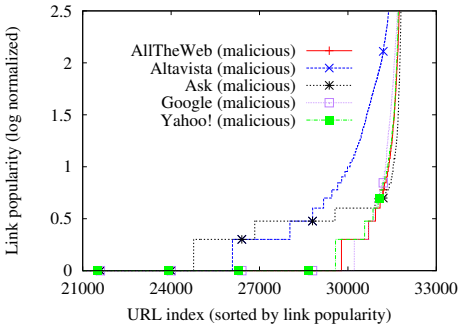

Figure 4: LPOP of malicious URLs for each search engine.

The larger the gap between benign URLs and malicious URLs a search engine reports, the more accurate that the link popularity is in distinguishing malicious URLs from benign URLs. `Google` tends to report a lower link popularity for both benign and malicious URLs and thus should produce higher false positives and lower false negatives. Table 9 shows the measured metrics for the malicious URL detection using only LPOP reported by each individual search engine. From the table, `Google` yielded high false positives (12.3%) and low false negatives (2.1%). `AllTheWeb` showed a link popularity distribution similar to that of `Yahoo!`. They had similar performance on malicious URL detection. This is not a surprise since `AlltheWeb` started to

use Yahoo!'s database since March 2004[8].

The result using Google was a surprise to us. We expected that Google would report the same, if not higher, link polarity than other search engines since it should have more comprehensive information of the Web. It turned out that Google just reported a partial list of link popularity, as their official website described[9]. The Google Webmaster Tool provides more comprehensive external link information, but we could not use it since it is available only for the owner's website.

Table 9: Detection accuracy, false positives and false negatives using only LPOP reported by each individual search engine (%)

Metric	AllTheWeb	Altavista	Ask	Google	Yahoo!
ACC	95.1	95.6	84.0	85.7	95.9
TP	95.3	96.3	85.7	86.7	95.7
FP	2.7	2.7	8.4	**12.3**	2.1
FN	2.2	1.6	7.6	**2.1**	2.1

Unpopular legitimate link classification. From the results reported above, we can conclude that LPOP is the most effective discriminative feature for detecting malicious URLs. It outperforms all the other feature groups by a large margin. However, LPOP alone may be ineffective for certain types of URLs, for example, to distinguish malicious URLs from a group of unpopular or newly setup benign URLs which also have low LPOP scores. This is the worst scenario for our malicious URL detector since the most effective feature, LPOP, is ineffective in this case. To conduct a test on the performance for this worst scenario, we used only the benign and malicious URLs which had zero LPOP to evaluate the performance of our detector. We obtained the following results on malicious URL detection: 91.2% for the accuracy, 4.0% for false positives, and 4.8% for false negatives. The accuracy remains high even under this worst scenario.

Popularity-manipulated link classification. As described in Section 3.2, some malicious URLs have high LPOP scores because their links are manipulated using a link farm [16]. We have developed five features, i.e., distinct domain link ratio, max domain link ratio, spam link ratio, phishing link ratio, and malware link ratio, to detect link manipulated malicious URLs. To make our detector light-weight and feasible in real-time applications, we used sampled link information instead of the whole link information to calculate each of these features. To evaluate the performance when the links are manipulated, we collected malicious URLs which had high LPOP scores (LPOP $>$ 10). Among the 32,000 malicious URLs we collected, only 622 URLs could be selected. Their distinct domain link ratio and max domain link ratio are shown against those of benign URLs in Figure 5. This figure indicates that the popularity-manipulated malicious URLs show a different pattern from those of benign URLs. Moreover, about 90% of these malicious URLs have more than 10% malicious link ratio (spam link ratio, phishing link ratio, and malware link ratio), whereas about 5% of benign URLs have more than 10% malicious link ratio. About 56% of these malicious URLs were linked exclusively by malicious URLs of the same type. Consequently, we obtained 90.03% accuracy in detecting link-manipulated malicious URLs with the aforementioned five features.

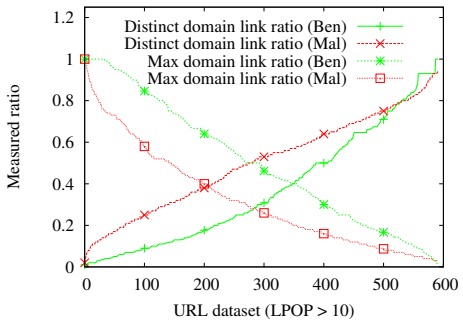

Figure 5: Distinct domain link ratio and max domain link ratio for benign and malicious URLs.

4.2.3 Error Analysis

In this section, both false positives and negatives are further studied to understand why these errors happened in order to further improve our method.

False positives. A false positive is when a benign URL is misclassified as malicious. False positives can be broadly categorized as follows:

- **Disreputable URL.** 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. More than 90% of the false positives belonged to the disreputable case (*e.g.*, 208.43.27.50/~mike).

- **Contentless URL.** Some benign URLs had no content on their webpages. In this case, CONT would fail (*e.g.*, 222.191.251.167, 1traf.com, and 3gmatrix.cn).

- **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).

- **Abnormal token URL.** We observed several benign URLs which had unusual long domain tokens typically appearing in phishing URLs (*e.g.*,

[8] AlltheWeb was taken over by Yahoo!.

[9] http://sites.google.com/site/webmasterhelpforum/en/faq--crawling--indexing---ranking\#links

centraldevideoscomhomensmaduros.blogspot.com).

False negatives. A false negative is when a malicious URL is undetected. Most false negatives were hosted by popular social networking sites which had a high link popularity and most URLs they hosted were benign. Most of the false negative URLs were of spam or phishing type. They generated features similar to those of benign URLs. More than 95% of the false negatives belonged to this case (*e.g.*, blog.libero.it/matteof97/ and digilander.libero.it/Malvin92/?). This will be further discussed in Section 4.4.

4.3 Attack Type Identification Results

To evaluate the performance of attack type identification, the following metrics given in [37] for multi-label classification were used: 1) micro and macro averaged metrics, and 2) ranking-based metrics with respect to the ground truth of multi-label data.

Identification metrics. Additional notation is first introduced. Assume that there is an evaluation data set of multi-label examples $(x_i, Y_i), i = 1, ...m$, where x_i is a feature vector, $Y_i \subseteq L$ is the set of true labels, and $L = \{\lambda_j : j = 1...q\}$ is the set of all labels.

- **Micro-averaged and macro-averaged metrics.** To evaluate the average performance across multiple categories, we apply two conventional methods: micro-average and macro-average [45]. The micro-average gives an equal weight to every data set, while the macro-average gives an equal weight to every category, regardless of its frequency. Let tp_λ, tn_λ, fp_λ, and fn_λ denote the number of true positives, true negatives, false positives, and false negatives, respectively, after evaluating binary classification metrics B (accuracy, true positives, etc.) for a label λ. The micro-averaged and macro-averaged version of B can be calculated as follows:

$$B_{micro} = B(\sum_{\lambda=1}^{M} tp_\lambda, \sum_{\lambda=1}^{M} tn_\lambda, \sum_{\lambda=1}^{M} fp_\lambda, \sum_{\lambda=1}^{M} fn_\lambda),$$

$$B_{macro} = \frac{1}{M}\sum_{\lambda=1}^{M} B(tp_\lambda, tn_\lambda, fp_\lambda, fn_\lambda).$$

- **Ranking-based metrics.** Among several ranking-based metrics, we employ the ranking loss and average precision for the evaluation. Let $r_i(\lambda)$ denote the rank predicted by a label ranking method for a label λ. The most relevant label receives the highest rank, while the least relevant label receives the lowest rank. The *ranking loss* is the number of times that irrelevant labels are ranked higher than relevant labels. The ranking loss, denoted as R_{Loss}, is calculated as follows:

$$R_{loss} = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|Y_i||\overline{Y}_i|}|\{(\lambda_a, \lambda_b) : r_i(\lambda_a) > r_i(\lambda_b),$$

$$(\lambda_a, \lambda_b) \in Y_i \times \overline{Y}_i\}|$$

where $\overline{Y}_i$ is the complementary set of Y_i with respect to L. The *average precision*, denoted by P_{avg}, is the average fraction of labels ranked above a particular label $\lambda \in Y_i$ which are actually in Y_i. It is calculated as follows:

$$P_{avg} = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|Y_i|}|\sum_{\lambda \in Y_i} \frac{|\{\lambda' \in Y_i : r_i(\lambda') \leq r_i(\lambda)\}|}{r_i(\lambda)}.$$

Table 10: Multi-label classification results (%)

	Label	Averaged			Ranking-based	
		ACC	micro TP	macro TP	R_{loss}	P_{avg}
RA*k*EL	L_{SAd}	90.70	87.55	88.51	3.45	96.87
	L_{WOT}	90.38	88.45	89.59	4.68	93.52
	L_{Both}	**92.79**	**91.23**	**89.04**	**2.88**	**97.66**
ML-*k*NN	L_{SAd}	91.34	86.45	87.93	3.42	95.85
	L_{WOT}	91.04	88.96	89.77	3.77	96.12
	L_{Both}	**93.11**	**91.02**	**89.33**	**2.61**	**97.85**

Identification accuracy. We performed the multi-label classification by using three label sets, L_{SAd}, L_{WOT} and L_{Both} mentioned in Section 4.1. The results for two different learning algorithms, RA*k*EL algorithm and ML-*k*MN, are shown in Table 10, where micro TP and macro TP are micro-averaged true positives and macro-averaged true positives, respectively. The following results were obtained: the average accuracy was 92.95%, whereas the average precision of ranking of the two algorithms was 97.76%. The accuracy on the label set L_{Both} was always higher than that on either L_{SAd} or L_{WOT}. This implies that more accurate label set produces a more accurate result for identifying attack types.

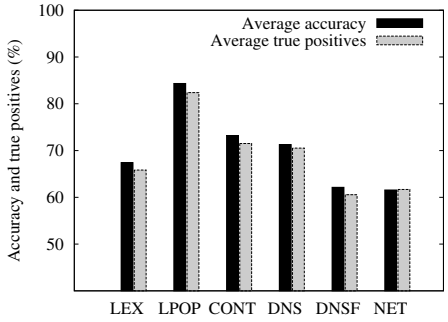

Figure 6: Average accuracy and micro-averaged true positives (%).

Fig. 6 shows effectiveness of each feature group in identifying attack types. Among the top ten most effective features, eight are novel features. They are three SLD hit ratio features in LEX, three malicious link ratios in LPOP, and two malicious ASN ratios in DNS. From this figure, even the link popularity features were also rather effective in distinguishing different attack types. In addition, no single feature group was highly effective in identifying attack types: they all yielded an accuracy lower than 85%. The combination of all the groups of features, however, yielded a much improved performance.

4.4 Evadability Analysis

Existing methods can be evaded by capable attackers. Similarly, our features are also evadable to a certain degree. However, it is an improvement if we can raise the bar of evasion difficulty by either increasing the evasion cost or decreasing the effectiveness of threat. To study evadability of our method, we discuss in this subsection the robustness of our method against known evasions and also possible evasion tactics.

Robust against known evasions. 1) Redirection: One possible evasion tactic is to hide the original URL using multiple redirections (also known as a "drive-by website attack" such as Iframe redirection) or a URL shortening service which makes a webpage available under a very short URL in addition to the original URL. Our method is robust against this kind of URL hiding and embedding evasions because our webpage crawler can automatically detect redirections and find the original URLs. 2) Link manipulation: As mentioned in Section 4.2.2, our method is robust against the link manipulation attack (more than 90% of link-manipulated URLs were detected). 3) Fast-flux hosting: The DNSF features used in our method can detect fast-fluxed domains.

URL obfuscation. If an attacker (or a domain generation algorithm in malware, *e.g.*, Conficker Worm) generates a domain name and path tokens with random length and counts, most statistical features in LEX will be evaded. Therefore, it is easy to evade the statistical features in LEX except our unique feature "malicious SLD hit ratio" since a plenty of domains have to be registered to evade the malicious SLD hit ratio. Evading brand name presence feature is easy but such an evasion will make a malicious URL less likely to be clicked, resulting in a reduced effectiveness of attack. URL obfuscation using IDN (Internationalized Domain Names) spoofing can also be used to evade our detector. For example, `http://www.p&#1072;ypal.com` represents `http://www.paypal.com`. Such an evasion can be easily prevented by adding a module to deobfuscate a URL to find the resulting URL in our webpage crawler.

JavaScript obfuscation. Malicious javascript often utilizes obfuscation to hide known exploits, embed redirection URLs, and evade signature-based detection methods. Particularly, JavaScript obfuscation can make the webpage crawler mislead webpage content features (CONT). To extract webpage content features accurately, the webpage crawler should have an automated deobfuscation functionality. The Firefox JavaScript deobfuscator add-on[10] inspired by "The Ultimate Deobfuscator" [5] can be used in our webpage content crawler as a JavaScript deobfuscation module.

Social network site. Utilizing social network sites (*e.g.*, `Twitter`) to attack can reduce the effectiveness of LEX, LPOP, DNS, and NET features. A possible solution against this evasion tactic is to adopt features which can differentiate hacker's fake accounts from normal users. For example, we can use the number of incoming linked accounts (*e.g.*, "followers" in `Twitter`) as a feature to detect faked accounts. Such a feature is still evadable with more sophisticated attacks which build a fake social network to link each other. Like the five link ratio features in LPOP to deal with the link popularity manipulation, similar linked account ratio features can be used to deal with a fake social network. Other countermeasures against social spam and phishing [20] can also be combined with our detector.

As mentioned in this section, it may cost little to evade a single feature group. However, evading all the features in our method would cost much more and also reduce the effectiveness of attack.

5 Related Work

This section reviews the related work of our method. They can be classified into two categories depending on how the classifier is built: machine learning methods which use machine learning to build classifiers, and other methods which build classifiers with a priori knowledge.

5.1 Non-machine learning approaches

Blacklisting. One of the most popular approaches is to build a blacklist to block malicious URLs. Several websites provide blacklists such as jwSpamSpy [19], PhishTank [29], and DNS-BH [11]. Several commercial products construct blacklist using user feedbacks and their proprietary mechanisms to detect malicious URLs, such as McAfee's SiteAdvisor [23], WOT Web of Trust [41], Trend Micro Web Reputation Query Online System [36], and Cisco IronPort Web Reputation [7]. URL blacklisting is ineffective for new malicious URLs. The very nature of exact match in URL blacklisting renders it easy to be evaded. Moreover, it takes time to analyze malicious URLs and propagate a blacklist to end users. Zhang *et al.* [46] proposed a more effective blacklisting approach, "predictive blacklists", which uses a relevance ranking algorithm to estimate the likelihood that an IP address is malicious.

VM execution. Wang *et al.* [39] detected drive-by exploits on the Web by monitoring anomalous state changes in a Virtual Machine (VM). SpyProxy [26] also uses a VM-based Web proxy defense to block suspicious

[10]The Firefox add-on shows JavaScript runs on a webpage, even if the JavaScript is obfuscated and generated on the fly [28].

Web content by executing the content in a virtual machine first. The VM-based approaches detect malicious webpages with a high accuracy, but only malware exploiting pages can be detected.

Rule-based anti-phishing. Several rule-based anti-phishing approaches have been proposed. Zhang *et al.* [49] proposed a system to detect phishing URLs with a weighted sum of 8 features related to content, lexical and WHOIS data. They used the Google Web search as a filter for phishing pages. Garera *et al.* [13] used logistic regression over manually selected features to classify phishing URLs. The features include heuristics from a URL such as Google's page rank features. Xiang and Hong [43] proposed a hybrid phishing detection method by discovering inconsistency between a phishing identity and the corresponding legitimate identity. PhishNet [30] provides a prediction method for phishing attacks using known heuristics to identify phishing sites.

5.2 Machine learning-based approaches

Detection of single attack type. Machine learning has been used in several approaches to classify malicious URLs. Ntoulas *et al.* [27] proposed to detect spam webpages through content analysis. They used site-dependent heuristics, such as words used in a page or title and fraction of visible content. Xie *et al.* [44] developed a spam signature generation framework called AutoRE to detect botnet-based spam emails. AutoRE uses URLs in emails as input and outputs regular expression signatures that can detect botnet spam. Fette *et al.* [12] used statistical methods to classify phishing emails. They used a large publicly available corpus of legitimate and phishing emails. Their classifiers examine ten different features such as the number of URLs in an e-mail, the number of domains and the number of dots in these URLs. Provos *et al.* [31] analyzed the maliciousness of a large collection of webpages using a machine learning algorithm as a pre-filter for VM-based analysis. They adopted content-based features including presence of obfuscated javascript and exploit sites pointing iframes. Hou *et al.* [18] proposed a detector of malicious Web content using machine learning. In particular, we borrow several webpage contents features from their features. Whittaker *et al.* [40] proposed a phishing website classifier to update Google's phishing blacklist automatically. They used several features obtained from domain information and page contents.

Detection of multiple attack types. The classification model of Ma *et al.* [21, 22] can detect spam and phishing URLs. They described a method of URL classification using statistical methods on lexical and host-based properties of malicious URLs. Their method detects both spam and phishing but cannot distinguish these two types of attack.

Existing machine learning-based approaches usually focus on a single type of malicious behavior. They all use machine learning to tune their classification models. Our method is also based on machine learning, but a new and more powerful and capable classification model is used. In addition, our method can identify attack types of malicious URLs. These innovations contribute to the superior performance and capability of our method.

Other related work. Web spam or spamdexing aims at gaining an undeservedly high rank from a search engine by influencing the outcome of the search engine's ranking algorithms. Link-based ranking algorithms, which our link popularity is similar to, are widely used by search engines. Link farms are typically used in Web spam to affect link-based ranking algorithms of search engines, which can also affect our link popularity. Researches have proposed methods to detect Web spams by using propagating trust or distrust through links [15], detecting bursts of linking activity as a suspicious signal [34], integrating link and content features [4], or various link-based features including modified PageRank scores [6]. Many of their techniques can be borrowed to thwart evading link popularity features in our detector through link farms.

6 Conclusion

The Web has become an efficient channel to deliver various attacks such as spamming, phishing, and malware. To thwart these attacks, we have presented a machine learning method to both detect malicious URLs and identify attack types. We have presented various types of discriminative features acquired from lexical, webpage, DNS, DNS fluxiness, network, and link popularity properties of the associated URLs. Many of these discriminative features such as link popularity, malicious SLD hit ratio, malicious link ratios, and malicious ASN ratios are novel and highly effective, as our experiments found out. SVM was used to detect malicious URLs, and both RA*k*EL and ML-*k*NN were used to identify attack types. Our experimental results on real-life data showed that our method is highly effective for both detection and identification tasks. Our method achieved an accuracy of over 98% in detecting malicious URLs and an accuracy of over 93% in identifying attack types. In addition, we studied the effectiveness of each group of discriminative features on both detection and identification, and discussed evadability of the features.

References

[1] AHA, D. W. Lazy learning: Special issue editorial. *Artifiial Intelligence Review* (1997), 7–10.

[2] ALEXA. The web information company. `http://www.alexa.com`, 1996.

[3] CASTILLO, C., DONATO, D., BECCHETTI, L., BOLDI, P., LEONARDI, S., SANTINI, M., AND VIGNA, S. A reference collection for web spam. *SIGIR Forum 40*, 2 (2006), 11–24.

[4] CASTILLO, C., DONATO, D., GIONIS, A., MURDOCK, V., AND SILVESTRI, F. Know your neighbors: web spam detection using the web topology. In *ACM SIGIR: Proceedings of the conference on Research and development in Information Retrieval* (2007).

[5] CHENETTE, S. The ultimate deobfuscator. http://securitylabs.websense.com/content/Blogs/3198.aspx, 2008.

[6] CHUNG, Y.-J., TOYODA, M., AND KITSUREGAWA, M. Identifying spam link generators for monitoring emerging web spam. In *WICOW: Proceedings of the 4th workshop on Information credibility* (2010).

[7] CISCO IRONPORT. IronPort Web Reputation: Protect and defend against URL-based threat. http://www.ironport.com.

[8] CORTES, C., AND VAPNIK, V. Support vector networks. *Machine Learning* (1995), 273–297.

[9] CURL LIBRARY. Free and easy-to-use client-side url transfer library. http://curl.haxx.se/, 1997.

[10] DMOZ. Netscape open directory project. http://www.dmoz.org.

[11] DNS-BH. Malware prevention through domain blocking. http://www.malwaredomains.com.

[12] FETTE, I., SADEH, N., AND TOMASIC, A. Learning to detect phishing emails. In *WWW: Proceedings of the international conference on World Wide Web* (2007).

[13] GARERA, S., PROVOS, N., CHEW, M., AND RUBIN, A. D. A framework for detection and measurement of phishing attacks. In *WORM: Proceedings of the Workshop on Rapid Malcode* (2007).

[14] GEOIP API, MAXMIND. Open source APIs and database for geological information. http://www.maxmind.com.

[15] GYÖNGYI, Z., AND GARCIA-MOLINA, H. Link spam alliances. In *VLDB: Proceedings of the international conference on Very Large Data Bases* (2005).

[16] GYONGYI, Z., AND GARCIA-MOLINA, H. Web spam taxonomy, 2005.

[17] HOLZ, T., GORECKI, C., RIECK, K., AND FREILING, F. C. Detection and mitigation of fast-flux service networks. In *NDSS: Proceedings of the Network and Distributed System Security Symposium* (2008).

[18] HOU, Y.-T., CHANG, Y., CHEN, T., LAIH, C.-S., AND CHEN, C.-M. Malicious web content detection by machine learning. *Expert Systems with Applications* (2010), 55–60.

[19] JWSPAMSPY. E-mail spam filter for Microsoft Windows. http://www.jwspamspy.net.

[20] LEE, K., CAVERLEE, J., AND WEBB, S. Uncovering social spammers: social honeypots + machine learning. In *ACM SIGIR: Proceeding of the international conference on Research and development in Information Retrieval* (2010).

[21] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *KDD: Proceedings of the international conference on Knowledge Discovery and Data mining* (2009).

[22] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Identifying suspicious URLs: an application of large-scale online learning. In *ICML: Proceedings of the International Conference on Machine Learning* (2009).

[23] MCAFEE SITEADVISOR. Service for reporting the safety of web sites. http://www.siteadvisor.com/.

[24] MCGRATH, D. K., AND GUPTA, M. Behind phishing: An examination of phisher modi operandi. In *LEET: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats* (2008).

[25] MOORE, T., CLAYTON, R., AND STERN, H. Temporal correlations between spam and phishing websites. In *LEET: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats* (2009).

[26] MOSHCHUK, A., BRAGIN, T., DEVILLE, D., GRIBBLE, S. D., AND LEVY, H. M. Spyproxy: Execution-based detection of malicious web content. In *Security: Proceedings of the USENIX Security Symposium* (2007).

[27] NTOULAS, A., NAJORK, M., MANASSE, M., AND FETTERLY, D. Detecting spam web pages through content analysis. In *WWW: Proceedings of international conference on World Wide Web* (2006).

[28] PALANT, W. JavaScript Deobfuscator 1.5.6. https://addons.mozilla.org/en-US/firefox/addon/javascript-deobfuscator/, 2011.

[29] PHISHTANK. Free community site for anti-phishing service. http://www.phishtank.com/.

[30] PRAKASH, P., KUMAR, M., KOMPELLA, R. R., AND GUPTA, M. PhishNet: Predictive Blacklisting to Detect Phishing Attacks. In *INFOCOM: Proceedings of the IEEE Conference on Computer Communications* (2010).

[31] PROVOS, N., MAVROMMATIS, P., RAJAB, M. A., AND MONROSE., F. All your iFRAMEs point to us. In *Security: Proceedings of the USENIX Security Symposium* (2008).

[32] QUINLAN, J. R. C4.5: Programs for machine learning. *Morgan Kaufmann Publishers* (1993).

[33] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In *SIGCOMM* (2006).

[34] SHEN, G., GAO, B., LIU, T.-Y., FENG, G., SONG, S., AND LI, H. Detecting link spam using temporal information. *IEEE International Conference on Data Mining 0* (2006), 1049–1053.

[35] T. JOACHIMS. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.). *MIT-Press* (1999).

[36] TREND MICRO. Web reputation query - online system. http://reclassify.wrs.trendmicro.com/.

[37] TSOUMAKAS, G., KATAKIS, I., AND VLAHAVAS, I. *Mining Multi-label Data*. Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition, 2010.

[38] TSOUMAKAS, G., KATAKIS, I., AND VLAHAVAS, I. Random k-labelsets for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering* (2010).

[39] WANG, Y.-M., BECK, D., JIANG, X., ROUSSEV, R., VERBOWSKI, C., CHEN, S., AND KING, S. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In *NDSS: Proceedings of the Symposium on Network and Distributed System Security* (2006).

[40] WHITTAKER, C., RYNER, B., AND NAZIF, M. Large-scale automatic classification of phishing pages. In *NDSS: Proceedings of the Symposium on Network and Distributed System Security* (2010).

[41] WOT. Web of Trust community-based safe surfing tool. http://www.mywot.com/.

[42] WU, B., AND DAVISON, B. D. Cloaking and redirection: A preliminary study. In *AIRWeb: Proceedings of the 1st Workshop on Adversarial Information Retrieval on the Web* (2005).

[43] XIANG, G., AND HONG, J. I. A hybrid phish detection approach by identity discovery and keywords retrieval. In *WWW: Proceedings of the international conference on World Wide Web* (2009).

[44] XIE, Y., YU, F., ACHAN, K., PANIGRAHY, R., HULTEN, G., AND OSIPKOV, I. Spamming botnets: signatures and characteristics. In *SIGCOMM* (2008).

[45] YANG, Y. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval* (1999), 67–88.

[46] ZHANG, J., PORRAS, P., AND ULLRICH, J. Highly predictive blacklisting. In *Security: Proceedings of the USENIX Security Symposium* (2008).

[47] ZHANG, M.-L., AND ZHOU, Z.-H. A k-Nearest Neighbor based algorithm for multi-label classification. In *IEEE International Conference on Granular Computing* (2005), vol. 2.

[48] ZHANG, M. L., AND ZHOU, Z. H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition 40*, 7 (July 2007), 2038–2048.

[49] ZHANG, Y., HONG, J., AND CRANOR, L. CANTINA: A content-based approach to detecting phishing web sites. In *WWW: Proceedings of the international conference on World Wide Web* (2007).