COMPUTERS & SECURITY ■■ (2018) ■■–■■

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Malicious URL protection based on attackers' habitual behavioral analysis

Sungjin Kim [a], *Jinkook Kim* [b], *Brent ByungHoon Kang* [a,*]

[a] *Graduate School of Information Security, School of Computing, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon Republic of Korea*

[b] *NCSOFT JAPAN K.K., Tokyo 106-0032, Japan*

ARTICLE INFO

Article history:
Received 1 December 2016
Received in revised form 23 December 2017
Accepted 15 January 2018
Available online

Keywords:
Behaviors
Fuzzy
Malicious URL
Similarity matching
Web-filtering

ABSTRACT

In terms of URL-based features, some studies have classified malicious URLs into a group with the same attributes. However, the malicious URLs are of two different types, each of which produces entirely different results. Thus, depending on their intention, adversaries leave slightly different behavioral traces within the malicious URLs. This paper presents an in-depth empirical study conducted based on 1,529,433 malicious URLs collected over the past two years.

In particular, we analyze attackers' tactical behavior regarding URLs and extract common features. We then divide them into three different feature pools to determine the level of compromise of unknown URLs. To leverage detection rates, we employ a *similarity matching* technique. We believe that new URLs can be identified through attackers' habitual URL manipulation behaviors. This approach covers a large set of malicious URLs with small feature sets. The accuracy of the proposed approach (up to 70%) is reasonable and the approach requires only the attributes of URLs to be examined. This model can be utilized during preprocessing to determine whether input URLs are benign, and as a web filter or a risk-level scaler to estimate whether a URL is malicious.

1. Introduction

A web server consists of tens of thousands of URL links. Adversaries may inject a short iFrame into one of the webpages to redirect users to a malicious site. If the infected domain is a popular website that receives millions of daily visitors, one malicious URL can contaminate a large number of user machines in minutes. Therefore, protecting against these malicious links helps to block daily malware infestations.

However, current antivirus software that supports URL filters provides domain-based protection. When a user sends an HTTP request, this software blocks access to a domain if the "Host" in a header packet exactly matches a domain in a blacklist database. This web-filtering is very helpful in protecting against blacklisted domains. These domain-based technologies keep up with the volume of malicious domains and IP addresses. Using this approach, Google's Safe Browsing API (Google, 2016), McAfee's SiteAdvisor (McAfee SiteAdvisor software, 2017), or Symantec's Safe Web (Norton safe web, 2017) prevents users from visiting these malicious sites.

However, these web-filtering systems induce a high False-Positive (FP) rate when malicious domains are inactive. Attackers quickly remove the attack URLs from the malicious sites. They may appear some time later, or may not be exposed again on the same websites. In this regard, if domain-based solutions refrain from updating their blacklists, users can be blocked from normal access to an entire website. For instance,

* *Corresponding author.*
E-mail addresses: r3dzon3@kaist.ac.kr (S. Kim), jinkim@ncjapan.co.jp (J. Kim), brentkang@kaist.ac.kr (B.B.H. Kang).

https://doi.org/10.1016/j.cose.2018.01.013
0167-4048/

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

Table 1 – Feature sets and features.

Feature Set	Feature Name	Malicious URL Examples
Host-based features	similar domain	wsad004.asia, wsad066.asia, wsad123.asia
	similar subdomain	webhosting50.1blu.de, webhosting52.1blu.de
	similar IP prefix	104.149.195.115, 104.149.195.116
	similar port	174.139.26.238:801, 174.139.30.222:802
	same domain	couponmoeum.com/PEG/css/1.js, couponmoeum.com/PEG/css/ad.html
	same subdomain	aa.sswangima.com, aa.wangma1q.com
	same IP	175.126.74.101/files/2.js, 175.126.74.101/files/1.js
	same port	2288.org:8832, 6600.org:8832, 8800.org:8832, 8866.org:8832
Path-based features	similar pathname	sbsjob.co.kr/PEG/js/check_38746.js, **dowmi.net/PEG/ad/check_38746.js**
	same pathname	braxico.com/images/m2dcbAnA.php, kappen-orth.de/images/iRsGKfv2.php
Filename-based features	similar query type	rcjhvqmtkmp?bdhkuyjqwmpi=6621548, rckjyihjpoggki?bxvuwwsceqt=6621548
	similar filename	199.188.107.109/yy.html, 199.188.107.109/zz.html
	same query type	175.45.4.158/index.php?id=kim031, 175.45.4.158/index.php?id=kim032
	same filename	199.188.107.112/xiaoyu.html, 198.2.221.203/xiaoyu.html

domain-based Google Safe Browsing prohibits user access to entire webpages for a grace period, even though attacks have stopped. For small online companies, a disruption of this nature can have critical consequences. Currently, most web-filtering systems cause the same problem. Hence, blocking malicious URLs instead of malicious domains can be a more acceptable solution. This enables users to access all the webpages of the domain except for the malicious links even when FPs are encountered.

However, for protection against malicious URLs, which have propagated malware with hundreds of subdomains such as athersite.com and findhere.org, a URL-based web-filtering system requires a large-scale blacklist containing all malicious links, in contrast to a domain-based system.

Thus, we propose a URL-based web-filtering model that can protect against a large number of malicious webpages with very small blacklist sets. The proposed model protects Internet users from malware contamination and allows e-service providers to achieve business continuity. As a result, the premise of our study is to implement a web-filtering model that prohibits malware proliferation with minimal FPs and small datasets.

Our main idea is based on the repeatable behavior in an adversary's URL manipulation, as shown in Table 1. Our objective is to design a detection model with *similarity matching* based on the traceable features of malicious URLs. Our method can detect large-scale malicious URLs from a small database, even though no identical match is found unlike domain-based systems. Consequently, we have two objectives: i) understanding of the important characteristics of malicious URL types, and ii) proving the validity of our proposed model based on the features of malicious URLs.

This paper describes a measurement sample of 1,529,433 malicious URLs collected from a Tier-1 ISP (SK broadband). The main contributions of this paper are as follows.

1. We introduce new features that characterize malicious URLs, and which provide useful insights for a new detection model. Further, we provide previously unreported results and statistics that are valid for future use.
2. Our behavior-based model increases detection coverage to 70% than current web-filtering systems. Our work is applicable to actual environments.

The remainder of this paper is organized as follows. We present related work in Section 2. Section 3 defines malicious URLs. Section 4 describes our URL dataset. We furnish details of our empirical research in Section 5 and provide an overview of the proposed model in Section 6. In Section 7, the details of our evaluation are explained, and the experimental setup and results are reported. We discuss the limitations of our approach in Section 8. Our conclusions are outlined in Section 9.

2. Related work

Two methods have been extensively used in the wild to detect malicious websites.

2.1. *Feature-based detection*

The methods that are presently emerging only use the URL structure for detection. Huang et al. (2013) mentioned that the lexical tokens in the URL are less effective. Instead, they used the patterns of malicious URL segments, because these are more informative. However, the main drawback of the greedy selection algorithm they used is that it cannot process polymorphic URLs, such as when the pattern `*/*paypal*.com*/*` is changed to `*/*paypaaxl*.com*/*`. The fixed patterns cannot respond to URL variations. In other words, this model is not sufficient for detecting small changes as it needs to regenerate new patterns for subsequent URL detection.

In studies on lexical properties of the URL, McGrath and Gupta (2008) selected length in the domain and URL, and the number of unique characters (e.g., dots). They used these features to determine the properties of phishing URLs. Similarly, "Beyond blacklists" (Ma et al., 2009a) performed analysis based on IP properties (e.g., blacklisted IPs), and geographical location. EXPOSURE (Bilge et al., 2011) extracted 15 features associated with passive DNS analysis. In particular, the authors defined domain name-based features. They observed the numerical characters and substring length of "domain fluxing," which is algorithmically generated. The features were mainly quantified as counts (#) or a ratio (%). Another DNS paper

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

(Antonakakis et al., 2010) proposed a dynamic reputation system for the attacker's DNS agility. Features related to DNS include the diversity of geographical locations, distinct TLD counts, and the average length of domain names. Prophiler (Canali et al., 2011) proposed the use of host and URL-based properties. They observed untrusted WHOIS registration information, domain/filename/URL lengths, and certain TLDs. They also noted the absence of subdomains and the presence of an IP address or port number in the URL. Such features were used to construct classifiers for malicious webpage detection. However, the feature sets are still weak at identifying URL polymorphism types using benign attributes, which are surpassed.

On the contrary, in our model, the presence of subdomains is a critical element for detection. The length of the URL is classified differently according to the types of malicious URLs (details in Section 3). We do not consider WHOIS information. Instead, we emphasize IP subsequences. Unlike previous prominent studies (Le et al., 2011; McGrath and Gupta, 2008; Whittaker et al., 2010), which entailed the selection of features related to abnormal behavior, our feature sets resemble the attackers' habitual behavior. These features reflect similar characteristics on URLs, and they share syntactical similarity with other malicious URLs.

2.2. *Blacklist-based detection*

In the real world, there are many blacklisting systems, such as antivirus software-based malicious URL protection systems.

A webpage contains many URLs, ranging from a few to several hundred links. Thus, finding malicious URLs in a web server that is composed of myriads of webpages may not be feasible. To this end, we learned various features from previous achievements (Antonakakis et al., 2010; Canali et al., 2011; Huang et al., 2013; Kapravelos and Shoshitaishvili, 2013; Ma et al., 2009a; Stringhini et al., 2013) and added new indicators of malicious behavior by verifying significant differences in the types of malicious URLs. These properties are based on URL-based, attackers' tactical habits.

Although a web-filtering system provides high-performance protection, it can only identify malicious domain patterns that are identical to those stored within a database. In this regard, our analysis revealed that more than 70% are closely related with other malicious URLs. For example, domain names contain similarities such as `01g.info`, `0ql.info`, and `05f.info`.

2.3. *Others*

A method to detect machines that resolve domain names to networks that have been prone to infections (e.g., malicious IP spaces), has been developed (Antonakakis et al., 2011). Segugio (Rahbarinia et al., 2015) searched for machines that were consistently queried, were active only for a very short time, or pointed to previously abused IP space. This requires a blacklist, whitelist, and passive DNS DB. These systems mainly aim to detect DNS-based malicious domains, including malware, phishing, and spam domains. In this approach, we can observe that they do not leverage the similarity of malicious URL strings itself formed by attackers. These systems behave differently from ours.

Perdisci et al. (2013) considers traffic information produced by malware. Their similarity was based on the total number of HTTP, GET, and POST requests by malware and the average length of the URLs. This study searched for semantic similarity in HTTP traffic. Krishnan et al. (2016) captured a "tree-like" malicious form on a subtree similarity. However, our similarity relates to the adversary's deployment behavior in a URL lexical format.

WebWinnow (Eshete and Venkatakrishnan, 2014) classified malicious domains by comparing the features of about 40 different exploit toolkits. They considered the attack properties in retaining exploit codes such as obfuscation, vulnerable application verification, blacklist lookup, and cloaking. Similarly, Stock et al. (2016) and Eshete et al. (2015) addressed approaches to automatically detect the presence of exploit. However, we consider malicious URL types that trigger exploit kits than EKs itself.

Consequently, our URL-mining scheme emphasizes the similarities between malicious URLs. The similarity matching technique is more beneficial than simple domain matching. It shows high similarity existence without the real-time addendum of patterns. Accordingly, a major difference between our technique and previous ones is the use of different feature sets. Based on these properties, our model enhances the prior URL-based feature scope as well as the detection range of current blacklist-based detection.

3. Definition of malicious URLs

Before explaining our approach, we need to understand malicious URL types. All malicious URL types are not equal. For instance, URLs used in malicious webpages contain three types in terms of their role. Firstly, there are "landing URLs". This URL is very similar to the benign URL with the aim of concealing the identity of an attack. They resemble benign URLs in terms of the URL length and lexical format. Second, "distribution URLs" that contain an exploit toolkit exist. The lexical type of these URLs differs from that of landing URLs in terms of the length and lexical format. Lastly, there are URLs that are generated by malware after drive-by downloads. Many studies focused on the last type (Holz et al., 2008; Nazario and Holz, 2008).

In this paper, we focus on the first and second types. We do not believe that the three types of URLs have the same result. The length of the URLs and the number of dots or subdomains are not extensively applied for spam filtering (Thomas et al., 2011). Hence, it may also differ in terms of phishing URLs (Le et al., 2011), fast flux-based URLs (Holz et al., 2008; Nazario and Holz, 2008), Twitter stream (Lee and Kim, 2012), or other types of malicious URLs. For instance, Le et al. (2011) proposed a method to detect malicious URLs based on URL obfuscation and URL length. Obfuscation in landing URLs is applied in entire script code than a URL itself. In addition, the URL length is similar with benign URL length.

As mentioned above, there are two types of malicious URLs. Stokes et al. (2010) defined them as landing sites and distribution sites with hops, and Wang et al. (2013) defined the three components: landing page, redirection URL, and exploitation URL. In this paper, we categorize malicious URLs as either landing URLs or distribution URLs. Landing URLs are used to

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

redirect users to the attack code, whereas distribution URLs contain codes that exploit and compromise vulnerable software on client machines. Thus, we differentiate malicious URLs into two distinct groups.

A distribution URL manages multiple landing URLs. Eventually, all landing URLs are centralized to distribution URLs. Therefore, blocking distribution URLs is a key factor to decrease the probability of malware proliferation. Another important facet of distribution URLs is a notable difference in features (see Table 1), compared with those of landing URLs. To address this issue, we focus on finding the properties of distribution URLs constructed for malicious purposes. In particular, "drive-by downloads" caused by exploit toolkits (EKs) are directly related to distribution URLs. We then identify the relevance by analyzing adversaries' URL deployment behavior.

In general, URL-based detection employs a lexical analysis of the URL itself. However, our approach utilizes behavioristics, which examines "how attackers operate landing and distribution URLs," or "how attackers have shown similar behavior regarding their IP management." This behavioral approach can detect suspicious IPs without counting the hyphens or dots. That is, some existing models (Eshete, 2013; Ma et al., 2009b) identify traits using textual analysis, such as the URL length, hostname length, path length, number of dots, and number of hyphens. Such features are often used for classification. In contrast, our behavioral approach considers the identical attributes or attitudes that attackers utilize in malicious URLs. Of course, not all malicious URLs have homologous forms.

The behavioristic approach involves examining the properties of landing URLs that mask attackers' activities by appearing to be benign URLs. Thus, these URLs redirect users to other webpage through seemingly proper connections. Such that, landing URLs share very similar traits with benign URLs in a lexical format and this similarity is apparently used in practice. In contrast, distribution URLs behave much more aggressively. General distribution URLs share similar properties with other distribution URLs. They can be identified in various ways, such as by detecting atypical characters, duplications, or both, and IP geolocation differences. All of these URL features represent behavioral traces left by attackers when they insert or modify malicious URLs.

4. Collection of malicious URLs

The dataset for the URL pattern analysis was gathered as follows.

We designed a `Collector` to collect malicious URLs. The `Collector` parses the web sources with browser rendering, and detects the distribution webpages containing the exploit codes. More specifically, it detects whether malicious webpages create new files, modify a registry, or attempt a new network connection on a 32-bit VM image installed with Java SE 6u25, IE 6.0.2900.5512, Adobe Flash Player (AFP) 11.5.502.146, and Silverlight 3 SDK. This is because attack codes can take the form of Oracle JRE/Applet, AFP, MS IE, XML, or ActiveX.

From a hooked web browser, the `Collector` monitored drive-by download activation. It was programmed based on `Detours` (Detours, 2015) (32-bit). Malicious links that exposed malicious activities were gathered using traffic logs generated by HTTP requests. The collected malicious links were instantly sent to `Protector`, which was deployed in one of the domestic major ISPs for end-user protection. `Protector` is a web-filtering system, and it collects *referrer* URLs, generated when web accesses of ISP users are redirected to malicious URLs, and forwards them to our database system. The landing URLs were mainly collected from `Protector`'s *referrers* and `Collector`'s redirection URLs. The distribution URLs were obtained from `Collector`.

A total of 1,529,433 malicious URLs were selected for the experimental datasets. Among these, 1,509,230 landing URLs and 20,203 distribution URLs were used for measurement. They are based on fully qualified domain names.

In particular, 90.31% of 20,203 distribution URLs existed in VirusTotal (2014), and 6.27% existed in websites of other well-known blacklist information providers such as DNS-BH (Malware domain blocklist, 2016), PhishTank (2016), and Google Safe Browsing. The remaining unknown URLs constitute only 3.42%, which were labeled using Cuckoo Sandbox (Cuckoo, 2016). In this paper, we present the analysis results of these sample datasets. We provide a dataset at this address: https://drive.google.com/file/d/0Bwjmbj3-p7V_S3RIRXFxTHZkcUU/view.

5. Empirical studies

Our study is motivated by the fact that attackers rely on *habitual behaviors* to accomplish their daily attacks. In this section, we discuss the results of empirical research on the features of this behavior.

The URL specification is classified into `scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]` (Uniform resource locator, 2016). However, we define a URL as having three parts: a host, path, and filename for detection convenience (e.g., http://host/path/filename). We also handle a URL as a set of tokens that includes delimiters, numbers, and alphabetic characters. The URL contains meaningful properties such as a distinct length, a geographical property, and a semantic string.

Previous studies have proposed detection techniques using URL lexical analysis (Eshete, 2013; Huang et al., 2013; McGrath and Gupta, 2008; Wang et al., 2013). In terms of URL attributes, some studies (Eshete et al., 2012; McGrath and Gupta, 2008) have also suggested that malicious URLs have different lengths compared to benign URLs as well as more delimiters and alphanumeric characters. Although this may be true, most malicious landing URLs closely resemble benign URL attributes, because adversaries wish to hide their landing URLs without exposure by exhibiting the characteristics of benign URLs. Attackers have effectively utilized these properties.

In contrast, attackers' URL manipulation (or URL rewriting) in distribution URLs includes generating random queries, obscurely modifying the pathname and renaming the filename analogously, and substituting specific URL segments. Attackers may also use an IP with similar prefixes and the existence of identical subdomains. Traces of these forgeries have been located in various malicious URLs (see Table A1 in the Appendix). These properties are critical for our detection model;

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

Table 2 – Properties of malicious URLs.

Features	Description	Section
Alexa-based properties	Adversaries mainly use subdomains in Alexa top-ranked domains attack. High-ranked domains for landing URLs and relatively lower-ranked sites for distribution URLs are used.	4.1.1
Geographical trends	Between landing URLs and distribution URLs show high geolocation difference.	4.1.2
Code changes in EKs	The type of some EKs can be predicted via the frequency of code changes.	4.2
Landing URL	Landing URLs in terms of length and path depth resemble benign URLs.	4.3.1
Serial IP zone	New malicious IPs are often found in /16 and /24 prefixes around malicious IPs.	4.3.2
Pathname	Attackers reuse pathnames.	4.3.3
Filename	Typical filenames: very short, default and its variants, foreign-language-based, and random. Distribution URLs' size is shorter in general than that of landing URLs. Alphanumeric filenames are more popular than numeric filenames.	4.3.4

they are summarized in Table 2. The characteristics of malicious URLs in Table 2 expose the URL similarities in Table 1.

5.1. *Characteristics of malicious URLs*

5.1.1. *Alexa-based properties of malicious URLs*

In this section, our analysis methodology is outlined based on the Alexa rankings – which mainly represents domain-based ranks. Thus, we treated all their subdomains with the same rank. For example, facebook.com and apps.facebook.com are regarded as having the same rank.

Adversaries use landing URLs, which are ranked highly in Alexa, whereas distribution URLs are ranked lower. This is because the landing URL is an advantageous location in which to situate the landing points of malware proliferation on popular websites, whereas the distribution URL is beneficial for locating attack codes on unpopular websites, where they are easily exploited and less likely to be discovered because of the lack of web maintenance. Typically, unpopular websites are not well managed by the server manager. The attack codes are typically long scripts that can be discovered more easily than the landing points. In this circumstance, an adversary invokes attacks on poorly maintained websites, rendering them useful for malware distribution. On the other hand, the role of the landing page is to hide and redirect the landing URL to the exploit code, masquerading as a benign URL. Thus, adversaries manage malicious URLs of two different types for attack convenience.

As a result, we assumed that landing URLs are scattered across sites with high Alexa ranks (e.g., between 1 and 100,000), and distribution URLs are predominantly centralized in low-ranking sites. However, in our datasets, 1.43% of landing URLs have remarkably high ranks (within the top 1000); almost 3.2% are located in the top 10,000 of the Alexa ranking, more than 6.9% are ranked within the top 100,000, and about 16% are within the top 1,000,000. Almost 63% of landing URLs were out-of-rank.

On the other hand, for distribution URLs, the high-rank rate is relatively lower than for landing URLs, as per our assumption – only 0.11% within the top 1000, about 0.2% within the top 10,000, and approximately 3.47% within the top 100,000. Almost 8.38% are within the top 1,000,000, leaving some 74.5% out-of-rank. Both types of malicious URLs are therefore mainly in out-of-rank sites. In particular, 0.5% of landing URLs are within the top 100 Alexa rankings, whereas none of the distribution URLs rank this high.

The Alexa top 100 is thought to be relatively safe (Eshete et al., 2012) according to a study that provided different weights according to the rank level. They used Alexa-based features for classification, which slightly conflicts with our analysis. Accordingly, from these artifacts, we assume that the belief that high-ranked websites provide safety regarding malware propagation may be biased.

In particular, the malicious rate that occurred in subdomains of the top 1000 ranks covered 79.28% and 90.87% in landing and distribution URLs, respectively. The frequency used in top-level domains (TLDs) is very low in the ranks. Thus, we infer that attacks mainly occur in subdomains rather than TLDs. For instance, some domains such as UglyAs.com and AtHerSite.com show that 201 and 261 subdomains were used to disseminate malware.

Landing URLs tend to be centralized in highly ranked domains at the start of malware infiltration to increase the attack success rate. The *Spearman's coefficient* between the rank distribution of the two malicious URL types was found to be 0.945. This result is a strong indication of a correlation between the two groups. However, as shown in Fig. 1, the cumulative distribution function (CDF) of the two malicious URL types is similar, but they exhibit different traits at high ranks (less than 1M). The figure shows the tendency of landing URLs to be centralized at a high rank, and for distribution URLs to be centralized at a low rank, even though it is trivial. The average distribution rate of landing URLs at high ranks exceeds that of distribution URLs by at least a factor of two.

Among highly ranked domains, such as Facebook, Naver, and Baidu, many were distributing malware through their search engine (see Table A2 in the Appendix). These sites were used as landing URLs that were redirected to distribution URLs. Amazon and Youtube hosted malicious advertisements, and googleusercontent.com transferred malicious links through Google translation.

We summarize the findings from this section as follows.

Finding 1: Attacks mainly occurred in subdomains rather than TLDs in Alexa top ranked domains even though malicious URLs are rarely found at the top of the Alexa ranking; however, once all of the top ranks were unaffected in our datasets, we confirm some known facts that the Alexa top-ranked domains were used as landing points. This does not necessarily mean that the top-ranked domains were compromised. Rather, it denotes that adversaries craftily manipulate the websites' functions to mount an attack.

Finding 2: The Spearman's correlation and CDF results show that the two types of malicious URLs are similarly distributed. However, we verified that attackers targeted relatively highly ranked domains for locating landing URLs, and

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

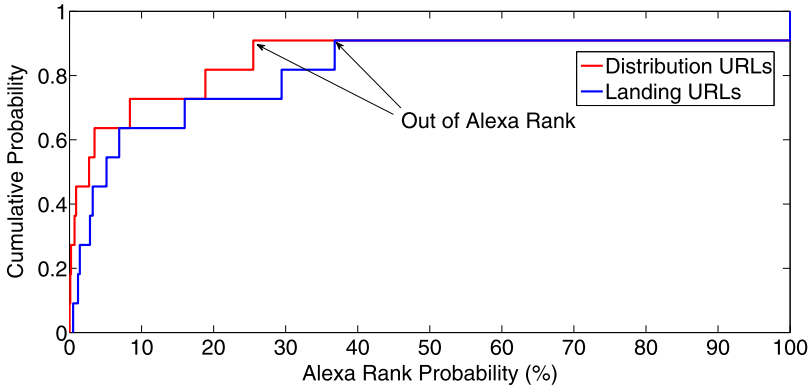

Fig. 1 – CDF of malicious URLs based on Alexa rank.

depended on lower-ranked sites to a greater extent for locating distribution URLs.

5.1.2. *Geographical trends*

Another interesting factor concerns the use of generic top-level domains (gTLDs) and country code top-level domains (ccTLDs).

We investigated the geographical differences by examining ccTLDs in both landing and distribution URLs. Previous studies (Klien and Strohmaier, 2012; Stringhini et al., 2013) used geographical properties, whereas we adopted location difference between landing and distribution URLs. In ccTLDs, the rate at which landing URLs are located domestically but distribution URLs are situated overseas is 98.67% and 31.88%, respectively. This means that many domestic users are exploited by attack servers abroad. Moreover, gTLDs with domestic IPs constitute 93.18% of landing URLs and 16.12% of distribution URLs. This significant difference is approximately 77% (see Table 3).

In terms of the locations of ccTLD and gTLD, landing URLs are more likely to be domestic, whereas distribution URLs are more likely to be overseas. This discrepancy in ccTLD/gTLD between the two malicious URL types is frequently observed in real environments. Regarding location conflicts, Table 3 highlights that the rate of gTLDs in landing URLs is 93.18%; however, the overseas proportion of distribution URLs is 83.88%. An analysis of gTLDs yields the differences in location from where attacks are launched and where the exploitation/distribution occurs.

The landing URLs are located in 31 countries. Distribution URLs are spread across 46 countries. In particular, for distribution URLs, the countries with the highest ccTLD/gTLD ratio are the US and China, which cover 60.66% and 18.80%, respectively.

Finally, gTLDs in landing URLs are predominantly from domestic IPs. However, distribution URLs are located on foreign IPs. Distribution URLs are highly dependent on compromised IPs that are not located in the target country. Many IPs are therefore operated by attackers from other countries. It appears that attackers hack domains and operate them remotely.

Finding 3: Table 3 proves that landing URLs mainly begin in domestic webpages; however, the exploit servers that are used for malware distribution are located overseas. Hence, the difference in the location of the starting and ending domain is a highly malicious trait. Accordingly, the discordance of locations between ccTLD/gTLD in the two URL types is a feature that increases the likelihood of detection. In the influence of geographical location, gTLD gives more information than ccTLD.

Table 3 – Location correlation between ccTLD and gTLD based on GeoIP (Maxmind, 2016).

Location	Landing URLs		Distribution URLs	
	% ccTLD	% gTLD	% ccTLD	% gTLD
Domestic	98.67	93.18	68.12	16.12
Oversea	1.33	6.82	31.88	83.88
# URLs	19,587	20,101	22,921	24,583

5.2. *Code changes in EKs*

Adversaries distribute malware by employing automated attack tools that are effective and easy to use. There are many well-known tools, such as Sweet Orange and RedKit. Remarkably, attackers accomplish their objectives by frequently changing the EK code to avoid detection.

We measured the code changes of EKs and the frequencies of those changes. We also examined the changes of distribution URLs. We investigated whether this frequency could be used to determine the type of EK during their lifetimes. We observed a code changing frequency of 200 EKs (36 Gongda, 101 JS NB VIP, 3 RIG, and 60 Sweet Orange) that were directly related to malware distribution. Four types of EKs were randomly selected for the experiment. The graph in Fig. 2 shows the code changes of EKs (denoted as "○") whenever an EK is detected by antivirus software or when avoidance is required. In reality, some EKs routinely use anonymous virus verification services such as scan4you.net (Scan4you, 2016) to check whether they can be detected by antivirus software (Eshete and Venkatakrishnan, 2014). The average lifetime of 200 EKs represents 13.19 days for Gondad, 7.31 days for JS NB VIP, 1 day for RIG, and 8.76 days for Sweet Orange. In this test, most EKs have a short life span. Nappa et al. (2013) identified that the exploit server lifetime was mostly short.

The average number of code changes varies from 4.31 to 92.98 over their lifetime, whereby more than 55% of EKs are involved in multiple changes. Indeed, one of the EKs exhibited a maximum of 311 code changes. The number of code changes experienced by EKs increases randomly over time.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

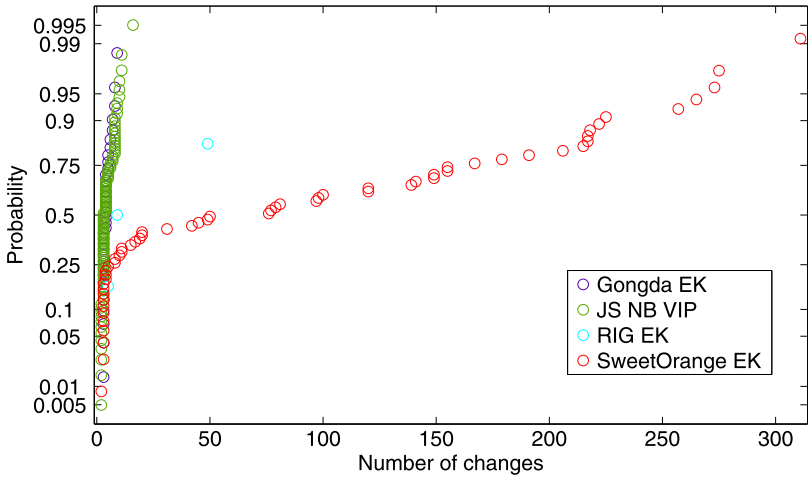

Fig. 2 – Rate of code changes of EKs. The x- and y-axes denote the number of changes and the normal probabilities of each EK, respectively.

We consequently assessed whether EKs obey a normal distribution, which would enable us to infer the types of each EK. Casual EKs were differentiated by an unusual EKs (irregular distribution). We found the accumulated changes in Sweet Orange EKs to suddenly increase and found them to be distributed extensively over a short period. Similar patterns were found between the same EKs. Thus, the extent to which thresholds are exceeded and patterns are correlated can enable the EK type to be predicted. Another tactic used by attackers is to substitute EKs with other attack types, such as iFrame, or JavaScript. The probability of these tactics being used is more than 11% for our dataset.

By analyzing webpages with 20,203 distribution URLs, we realized that many webpages had a short lifespan. Meanwhile, there were cases in which they were active for more than six months. As shown in Fig. 3, the lifetime of EKs is typically brief, but in our experiment over 1 year, exploit pages of approximately 34.10% survived for a day and 67.05% for 1 week, respectively. Further, 16.87% of exploit pages only appeared after 30 days, and 94.26% of exploit pages disappeared for 6 months. About 99.9% of exploit pages disappeared within 230 days. Only one exploit page survived for 1 year.

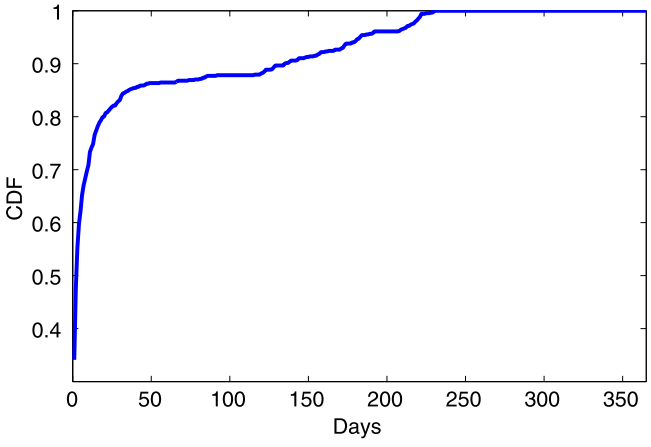

Fig. 3 – Lifetime of EKs based on one year period.

We denote that the exploit webpages mostly contained exploit kits, but they were also changed to webpages with analytics, attack scripts, or any other exploit links. Thus, these webpages do not always have EKs. They also try to change different EK types, unlike prior EK. Most EKs disappear after a short-term attack, but some EKs constantly change attack types for a long-term period. EKs disappeared after launching attacks and reappeared sometimes later.

During the latent period, attackers often check user visits using analytics, e.g., Yandex, 51.la, 51yes, Baidu, and cnzz. Among 20,203 distribution URLs, 12,830 were accompanied with these analytics. Apart from that, we examined the changes of distribution URLs that indicate EKs and found that many distribution URLs with iFrame or script types were changed more than twice.

This feature selection includes routine hourly checking for malicious content changes connected to the collected malicious links. We executed this routine check on 20,203 distribution URLs during the study period.

Finding 4: We may predict the type of some EKs based on the frequency of code changes. In addition, the average lifetime of an EK is less than one month; however, their websites still remain latent. These malicious websites can be reactivated after several months. The EK changes affect in the change of other attack types for detection difficulties.

5.3. *Lexical analysis*

5.3.1. *Malicious URL path depth*

Previous studies (Eshete et al., 2012; Stokes et al., 2010) have classified URLs as benign URLs and malicious URLs. Some of them have used the URL length and delimiters for detection features. They suggest that malicious URLs are longer than benign URLs. Eshete et al. (2012) noted that the average URL length of the Alexa top 500 (Alexa.com, 2015) was 15 and that of PhishTank (2016) was 45 at a 1:3 rate. However, our dataset exhibits different traits. In short, the average length of a URL (excluding duplicate URLs) in the Alexa top 500 was found to be 71.10. Landing URLs have similar lengths to those of the Alexa top URLs with an average length of approximately 80.21. However, distribution URLs are

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

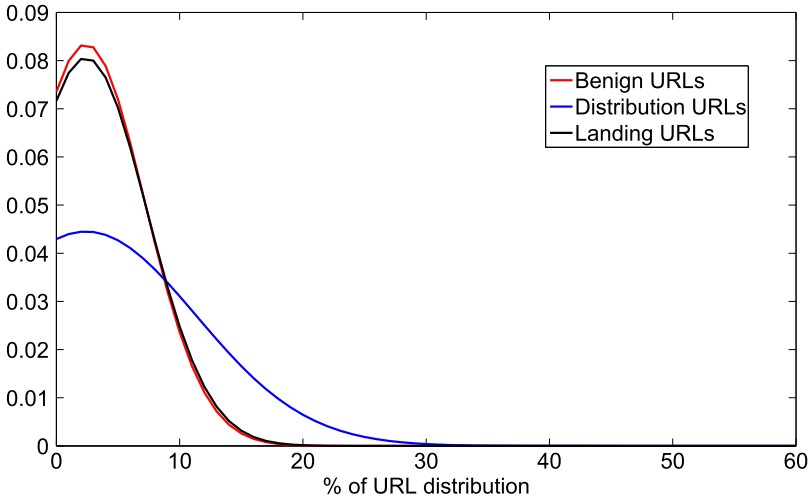

Fig. 4 – Normal distribution of URL length.

shorter, with an average length of only 39.03, which includes "http://" and "https://," even though URL shortening services, such as Twitter and spam, are not applied.

In terms of URL lengths, Fig. 4 shows that the typical length distributions of benign and malicious landing URLs are almost the same. The figure shows that distribution URLs have larger standard deviations, because the differences between the ranges are large and provide greater differentiation. Hence, the two types of URLs are not aligned in terms of their length being an indication of malicious properties. We believe that using the length of a landing URL for detection is not useful.

An examination of path depth (the count of "/" delimiters) shows that distribution URLs have depths of 1–3, with depths of more than three being much less prevalent. In contrast, this is clearly distinguishable with landing and benign URLs. The path depths of landing URLs and benign URLs are extensively distributed, unlike that of distribution URLs. Thus, this feature can be indirectly utilized in determining malicious symptoms. More than 90% of distribution URLs were located at depths of less than three, and they had a maximum depth of six. Distribution URL lengths range from 14 to 89 including delimiters (e.g., /, _, =, ?, and -).

On the other hand, the length of landing URLs varies widely. Landing URLs have depths of 1–29. The depth distribution of benign URLs is similar to that of landing URLs. This property significantly affects the decision of whether a URL is benign. The average path depths of distribution and landing URLs are each 1.672 and 2.667, respectively. The landing URLs have lengths ranging from 15–916 characters, with 83.6% in the range 21–110. A similar scenario can be extensively observed in benign URLs.

Finding 5: As shown in Fig. 4, the attributions of landing and distribution URLs differ significantly. Landing URLs resemble benign URLs in terms of URL length and path depth. Thus, adopting landing URLs as a malicious feature is not rational, whereas distribution URLs yield more precise results.

5.3.2. *Centralized IP zone and serial IP zone*

Distribution URLs exhibit similar traits with a probability of approximately 70%. The features are of various types, such as the same IP zones, same domains, same pathnames, or similar filenames. The properties of these types are often revealed somewhere in the dataset.

Mavrommatis and Provos (2008) analyzed the relationships among IPs. The study described the /8 prefix distribution for malicious IPs, which was plotted against the cumulative site fraction. Similarly, in our dataset, there are four centralized districts related to the /8 prefix of IPs: 50–70, 110–121, 173–175, and 204–222. These IP areas are associated with a high probability of containing malicious IPs. The 204–222 IP zone accounted for 39% of all malicious IPs, the 50–70 region comprised 19%, the 110–121 zone included 17%, and the 173–175 region had 10%. These IP zones formed 85% of all malicious IPs. We obtained a similar result by examining IP space concentrations for malware sites, although the datasets differed. A similar phenomenon was observed with /16 and /24.

Besides, when a malicious IP is detected, we conclude that another malicious IP is highly likely to be found in the same IP zone. The correlation between malicious IPs with the same /16 or /24 prefix is as strong as for Provos's centralized malicious IP zones. This enables us to distinguish maliciousness in candidate URLs. For example, we observed the existence of malicious IP ranges, and discovered attackers' IP management habits, such as http://198.1.x.34/index.html, http://198.1.x.35/index1.html, and http://198.1.x.36/index2.html. Many malicious IPs in the same zone exhibit similar tendencies. They have been centralized with respect to the /24 or /16 IP prefix.

Hence, the next malicious IP in such cases can be easily predicted. This assumption helps to determine the malicious attributes of an unknown URL. In our dataset, the closeness between malicious IPs with the same /24 prefix has a relatively higher probability than that of a /16 prefix. In particular, if a distribution URL is IP-based, serial IPs with the same prefix are most likely to be malicious. For example, if `110.34.196.113` is a malicious IP, then `110.34.196.114` and `110.34.196.115` are likely to be malicious IPs. Moreover, on the inetnum from `110.34.192.*` to `110.34.207.*`, we found IP zones that are likely to be highly malicious.

In the real world, it is difficult to detect such zones without this property. The web servers with these malicious IPs

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

generally return 404 errors when accessed and contain only a few webpages. They are only open for a limited duration when the attacks are launched. Thus, the malicious IP management habits of attackers appear unexposed. In this regard, our IP prefix-based feature can help to predict unknown short-lived malicious IPs.

Our empirical results for VirusTotal in May 2015 substantiate this point. We identified that malicious inetnums from VirusTotal were extensively distributed. The probability that serial malicious IPs are detected in an equal IP class significantly increases. Accordingly, if "110.34.197.243", "110.34.197.244," and "110.34.197.246" were used as malicious IPs, predictably, 110.34.197.245 has a relatively high probability of being malicious. This approach may predict unused malicious IPs that are exposed later. Consequently, current clean sites may be used as malicious sites at a later date. Based on these circumstances, we distinguished these subnet properties by classifying 2383 malicious IPs in the 20,203 distribution URLs. A total of 94.13% of the malicious IPs showed serial IP properties. However, we have not yet confirmed the other 5.87% of malicious IPs.

These serial IPs can be easily used to detect pharming websites. To this end, we programmed a `Pharming Collector` that takes a snapshot of the web content of an IP address. First, we searched pharming websites that communicate with malware downloaded from a CK VIP attack. We crawled the IP subnet of the pharming website based on WHOIS, and finally found another unused pharming website. In the IP zone, there existed many pharming and suspicious gambling sites being prepared for use. These sites were alternatives that could be rapidly replaced when the current pharming websites became blocked. However, http://43.249.82.136 was not detected by VirusTotal (2015-5-8). We realized that attackers were preparing redundant pharming websites in addition to those currently being used. This feature may be effective in detecting newly malicious IPs, as in the above example. Our model thus adopted this approach.

Prior studies (Holz et al., 2008; Nazario and Holz, 2008; Stringhini et al., 2013) considered malicious IP spaces such as Provos's centralized malicious IP zones. However, we adopt serial IP zones in same IP class.

Finding 6: Attackers gradually utilize their management IPs in bulk. Accordingly, a series of malicious IPs marks a suspicious trait. Detecting new malicious IPs requires us to conduct surveillance around malicious IPs with /16 and /24 prefixes.

5.3.3. *Characteristics of pathname*

Attackers exhibit specific patterns or leave considerable traces on distribution URLs. In particular, these similarities appear in pathnames. For example, a distribution URL with the `/.errordocs` pathname is also found in other distribution URLs. This property often surfaces because attackers show similar behavior. This modus operandi is highly related to the habits of adversaries.

In general, attackers show similar behavior when they distribute their custom-built malware on the web. They reuse similar or identical pathnames with a similar or the same query in the URLs, because these share the characteristics needed to execute extensive dissemination for a short time. In our dataset, some pathnames are found relatively frequently, such as /pop, /image, /data, /update, /updir, /upfiles, /upload, and /file. The pathnames are referenced in the detection model.

Finding 7: Attackers habitually reuse their pathnames.

5.3.4. *Filename, query string and extensions*

There is a high frequency of filename similarities within distribution URLs. For example, http://110.34.196.117/cake.php was regarded as a malicious IP by VirusTotal. We also found that http://110.34.196.125/cake.php included an EK on 2014-11-27 16:09 (VirusTotal did not detect this during the test period, but identified malicious activities on the VM (Cuckoo, 2016)). These URLs share identical segments in their filenames, which were changed to similar filenames on other IPs (e.g., index, index2, and index3). The malicious IPs exhibit similar filenames (see Table A1 in the Appendix).

Filenames in distribution URLs were broadly distributed with short lengths. The filename is a serial number or closely relates to a character or a string, such as 1.html, 1.js, 2.js, 3.js, a.js, a.html, b.html, and c.js. In particular, extremely short filenames comprised 11.17% in total, such as 1, 2, 3, a, b, c, d, and x. Most of the other filenames had short names such as ad.html. Moreover, meaningless filenames with fixed alphanumeric sizes, such as `RCkWbqGd.php`, were often found. In our datasets, most filenames consisted of fewer than eleven characters; 22.4% could not be decoded, and the remaining 77.6% were readable. They combined upper- and lower-case alphanumeric characters. In particular, the random filenames contained approximately seven to nine characters. Alphanumeric filenames appeared more often than purely numeric names, with a ratio of approximately 4:1, and are mostly accompanied with .php and .html extensions.

Filenames with a query were repeatedly used, such as `dftuegni?hirgnsuriq=6621548`, and the "6621548" string was found in more than 200 different URLs. In distribution URLs, "index.html" was one of the most extensively used filenames. Derivations with default names were also used, such as index1.html, index2.html, inde.htm, index.html., index.html, and iindex.php. These index.html types comprised 23.4% of our distribution URLs and the 93 different types of index.html existed. This means that attackers insert their malicious code in a default webpage and its variants.

The filename feature is largely classified into four categories. First, there are short filenames, such as 1, 2, 3, m, s, x, y, and ww. Second, there are frequently used filenames and derivational filenames, such as index, indexa, and indexb. Third, the filename is rendered in a foreign language, such as shifu, xiaomao, luku, maomi, meigui, and xiaoyu. Fourth, there are random numbers or alphanumeric characters with a fixed length, such as 13212323, RCkWbqGd, as well as query patterns (e.g., `01oQJqSl.php?id=18584046`, `02aqlXuJ.php?id=47043301`, and `crpy.html?j=1958545`) (This is a known fact). In addition, the most frequently used extensions are html, js, php, and asp. The other extensions are aspx, exe, css, gif, htm, jpg, enc, swf, phtml, apk, and jhtm.

Finding 8: There are four types of filenames: very short, well known and its variants, foreign-language-based, and random. Their respective sizes are predominantly less than 11. Alphanumeric names are more popular than purely numeric ones. More than 20% of filenames are unreadable. Consequently, to exhibit the effectiveness of these URLs' similarities, a model should be cooperated with distribute URLs.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

6. Building classification models

In this section, we give the details of the proof-of-concept model based on the features selected in the previous section.

6.1. Model motivation and basic idea

Attackers create malicious URLs by randomly generating a fixed string of alphanumeric characters, replacing a segment with a random numeric string, or both. Unfortunately, those URLs can bypass current detection systems.

Distribution URLs are the source of malware propagation. Such URLs have been used to launch attacks to disrupt critical systems and to upload sensitive information. Our approach is to classify URLs of this type by adopting a *similarity matching* technique to protect against malicious URLs that circumvent a blacklist database. This technique compares the similarity of strings in specific segment regions of a URL with feature sets.

Current string matching, such as web-filtering systems, covers only certain cases of exact string matching. In fact, current commercial antivirus products mostly provide 1:1 matching. However, *similarity matching* provides more extensive detection ranges. In this regard, string *similarity matching* should consider several preconditions for extensive detection. First, the feature set of distribution URLs is needed. Second, newly found distribution URLs are constantly needed to detect new types of malicious URLs. Third, the similarity method may not detect malicious URLs that avoid predefined features. Nonetheless, *similarity matching* based on well-defined features can cover a large portion of malicious URL sets. To advance the properties, our model provides scoring factors based on the predefined features. This helps to identify the malicious symptoms.

In this paper, we implemented our similarity matching approach for a real-time web-filtering system.

6.2. Model definition

In this subsection, we present our model setup based on the proposed features. Our detection model procedures are outlined in Fig. 5.

6.2.1. Dataset selection

The URLs are divided into benign URLs and malicious URLs. The malicious URLs we collected are classified as being either landing URLs or distribution URLs, which together are a subset of malicious URLs. Let $M = \{(l_1, \ldots, l_n), (d_1, \ldots, d_n)\}(n \geq 1)$ be a set of malicious URLs, where l denotes a landing URL and d denotes a distribution URL. d_n expresses the n^{th} distribution URL. Let $D = \{d_1, \ldots, d_n\}(n \geq 1)$ be a set of distribution URLs. $M \supset D$, but $M \not\subset D$. We only select distribution URLs for *feature selection* from this dataset. This is because all malicious landing URLs are centralized to distribution URLs.

6.2.2. Feature extraction and grouping

A feature is a set of meaningful strings used in segments of distribution URLs. A string is a set of tokens. It shares the same meaning or represents repeated attempts with delimiters and/or alphanumeric characters in the distribution URLs.

Trivially, $F = \{f_1, \ldots, f_n\}(n \geq 1)$ is a set of features on D, where f_n is said to be the n^{th} feature string of a unit in D. In fact, we have three different feature sets: host features, path features, and filename features. We have a feature pool for each of these, and individually name them F_{host}, F_{path}, and $F_{filename}$. To generate the most suitable pattern, our model compares the features from each of the pools.

To extract features from D, we parse d_i $(1 \leq i \leq n)$ into a host, a pathname and a filename portion, and classify s_i $(1 \leq i \leq n)$ (which is a partial string of a URL and not necessarily a feature) as a host, path, and filename (temporal) group. We then sort these respective groups, remove duplicates from ordered sequences, and randomly choose one of them when there is a high similarity between serial s_i. For example, we select one among 103.14.114.44, 103.14.114.45, and 103.14.114.47. We eradicate "*www*" and "*filenames*" from the default webpages of domains to increase the accuracy of similarity matching. Accordingly, a function f: $F \rightarrow S$, where $S = \{s_1, \ldots, s_n\}(n \geq 1)$ is a set of strings, satisfies the set f_i $(1 \leq i \leq n)$ in set S. Therefore, F_{host}, F_{path} and $F_{filename}$ are optimized feature sets that exist exclusively on any f_i.

Attackers intentionally or unintentionally mix various behavioral characteristics in creating malicious URLs. They use serial IP lists, serial numeric or alphabetic strings, meaningless filenames with alphanumeric strings, and pathnames with fixed-size variables. These traits are exposed as a combined figure.

In this regard, our model generates malicious patterns with various combinations from three different pools such as a "mesh structure." f_i contains one feature from three sets. From

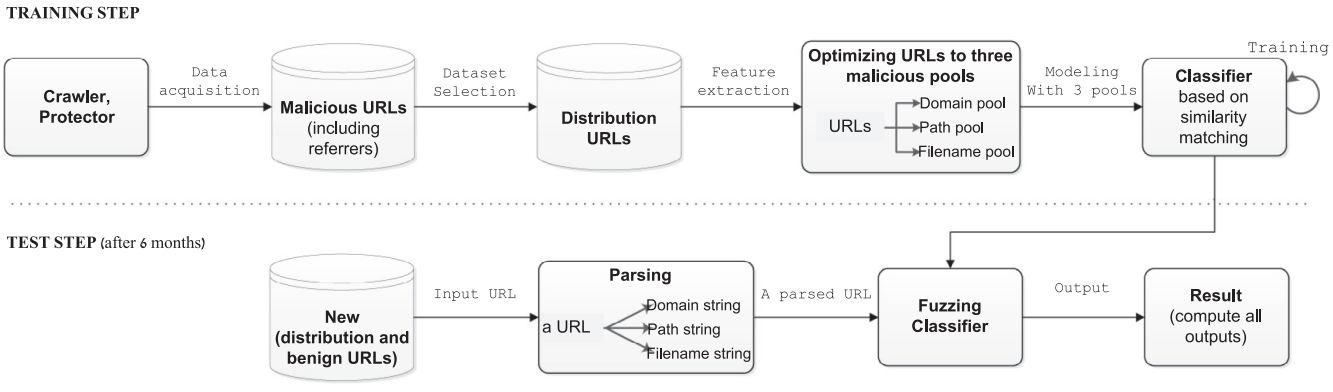

Fig. 5 – Overview of *fuzzy-based similarity matching*.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

the given f_i, we combine some or all of the features to generate a new pattern. Let $P = \{p_1, \ldots, p_n\}$ be a set of URL patterns, where p_i $(1 \le i \le n)$ is generated by combining f_i elements from F. Similarly, the combination of some features selected from $f_1, \ldots, f_n$ generates $p_1, \ldots, p_n$ such as $p_i =$ "http://f1/f4/f9". At this point, the hostname is denoted as f_1, f_4 is the pathname, and f_9 is the filename. A pattern is at least 10 characters long, including *http*. In general, P produces the following pattern rules:

$$P_{number\ of\ pattern} = \prod_{i=1}^{3} F_i + \sum F_{host}$$

F_i is sequentially $\#F_{host}$, $\#F_{path}$ and $\#F_{filename}$, where # means the cardinality of set F_i, and $P_{numberofpattern}$ is the product of # of all F_i, which should be greater than or equal to one. In particular, $\#F_{host}$ can be utilized independently as a pattern. In similarity matching, $P_{numberofPattern}$ can be countless. A feature can be used as a duplicate with features in other pools.

6.2.3. *Similarity measure and modeling*

In model design, to formalize the scoring algorithm for measuring the maliciousness of a candidate URL, we define C_L based on the scoring factors in (1) and $\sum FR$ as the sum of malicious probabilities from *Fuzzy-based similarity matching*. L_R and P_R denote the geolocation and URL length result, respectively. The malicious probability ratio F_W of a candidate URL is $\frac{\sum FR}{|\#FR|}$, where $|\#FR|$ is *the number of pools* that participated in the similarity matching.

Our URL *similarity matching* model first parses a candidate URL as host, pathname, and filename length intervals. The model then searches for similar IP prefixes (/16, /24) or similar hostnames from the set of F_{host}. The pathname and filename also correspond with F_{path} and $F_{filename}$. Next, the model measures the best fitted similarity of each host, pathname, and filename, and computes the overall relationship FR. However, if F_W threshold does not reach the level required for a malicious decision ($F_W \ge 7$) and returns a suspicious extent of $6 \le F_W < 7$, the model searches for additional evidence from L_R and P_R ("7" is the minimum control value computed by the classifier). Similarly, it checks the IP location (or inconsistency thereof) based on the ccTLD/gTLD as well as the URL length.

Our model calculates ten phases of risk level; however, each candidate URL has a different weight, because of the difference in similarity. Similarity in the IP/domain incurs a high severity. Some similar pathnames and filenames, such as "`.errordocs`," "`?click=1110828`," and "`zB0ypBk9.php`," are also treated with a high suspicion. These malicious indications help to form the malicious relationship.

$$C_L = F_W + \sum \begin{cases} if\ F_W < 7, \\ L_R,\ 0\ or\ 1\ (if\ \ foreign\ IP) \\ P_R,\ takeup\ or\ giveup \end{cases} \quad (1)$$

FR comprises the *Levenshtein distance*, given by $Lev_{C,F}$ (|C|, |F|), where |C| and |F| are regarded as candidate URLs and feature sets that create arbitrary URL patterns. This gives a similarity value of up to ten. If $max(C, F)$ is less than seven, the proposed model calculates the extra value. We define a similarity level of more than seven as malicious. L_R is used to relieve suspicious URLs, but P_R is applied to eliminate suspicious URLs based on the URL length. We assume that long URL forms (*length* > 100) are rarely used in distribution URLs; the longest length was 89 in our empirical study (see Section 5.3.1).

6.3. *Model implementation*

Our model consisted of 185 lines of code, and composed pattern sets called to F_{ip}, F_{domain}, F_{path}, and $F_{filename}$, and whitelist for high Alexa-ranked benign domains, which were unused as malicious URLs in the past. In our approach, we give a low value (1.0) because they can also contain pathnames and filenames in our malicious patterns due to the existing possibility of malicious URLs.

To be a malicious URL, this model compares geolocation using geoIP except for host, pathname and filename similarity. It also uses the URL length. We designed this model based on the rules of blacklists, and we built the core components using Python. The code for similarity matching using *Levenshtein distance* is described as follows:

Algorithm 1: Malicious URL Similarity Algorithm

```
Input  : U_i and p_i, which are the i^th candidate URL and pattern
Output: Malicious or Benign
procedure FUZZY_MATCH(pattern, test)
lev = fuzz.ratio(s1, s2)
return lev
procedure BEGIN( )
n = 3
i = 0
for x != empty do
    // u_i is the i^th test URL in file x
    U_i ← x(u_i)
    i += 1
    T_host ← host value from U_i
    T_path ← path value from U_i
    T_filename ← filename value from U_i
    while not at end of host pattern do
        P_host ← F_host
        max(H_n) ← FUZZY_MATCH(P_host, T_host)
    if 0.9 ≤ H_n then
        F_W = H_n;
        print "Malicious";
        break;
    else
        // compute max lev in F_path and F_filename
        max(P_n) ← FUZZY_MATCH(F_path, T_path);
        max(F_n) ← FUZZY_MATCH(F_filename, T_filename);
        FR = H_n + P_n + F_n;
        // n is the number of pools participated
        F_W = FR/n;
        if threshold ≤ F_W then
            print "Malicious";
            break;
        if threshold - α ≤ F_W then
            country ← read a geoip of T_host ;
            if country == domestic then
                print "Benign";
            else
                if URL_length < 100 then
                    print "Malicious";
                else
                    print "Benign";
                    break;
        else
            print "Benign";
```

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

In this code, the threshold is the minimum acceptable *lev* value. In our model, the threshold is *lev* (*7*). This is a real example between a test URL, http://yjfishing.kr/yjhome/bbs/data/result/1415260280/index.html, and pattern URLs, fishingnews.co.kr/yjhome/bbs/data/result/1415260280/index.html.

7. Evaluation

7.1. Setup of experimental environment

We evaluated the proposed model by designing a test environment based on a platform composed of an Intel Core i7-3610QM 2.30-GHz CPU with 8 GB RAM, 200-GB HDD, and running 64-bit Microsoft Windows 7 as the operating system. Assuming that the user accesses to a malicious webpage, we used randomly mixed candidate URLs. The lengths of the benign and malicious candidate URLs varied randomly from 14 to 722 characters.

The benign URLs were collected by crawling the websites of the Alexa top 1000 by modifying example code from PhantomJS (2016). All benign URLs were labeled as non-malicious by VirusTotal and dynamic analysis (Cuckoo, 2016). Among a collected total of 1612 URLs, 1039 benign URLs with high Alexa ranks of between 101 and 1,000 were selected for our model training. The other 573 benign URLs were selected for a real test.

During the real test, we monitored the maximum detection rate of the proposed model while generating 1874 URLs, which comprised approximately 573 benign and 1301 malicious URLs collected over a period of six months. We repeated this test without the new feature updates.

The evaluation objectives were 1) to verify the effectiveness of our features from our experimental setup, and 2) to prove a method to overcome the coverage drawback of the current blacklist database.

7.2. Evaluation of string matching algorithm

To measure the similarity between URLs, we tested various string-matching algorithms, such as *n-gram*, *Levenshtein distance*, *semantic similarity*, and *Jaccard*. The *semantic similarity*, which uses a thesaurus such as `WordNet` (WordNet, 2015), exhibited a high matching rate in the URL structure, even when some segments had complex alphanumeric names. However, it also produced the same trend in benign candidate URLs, leading to high FP rates. The *Jaccard similarity coefficient* gave a low matching rate. Its index required the intersection of finite URL sets divided by the size of the union of the candidate sets. The intersection requires exact URL string matching. Therefore, in the detection of malicious URLs that show subtle distinctions, *Jaccard* is ineffective. Other similar matching algorithms, such as the *dice coefficient*, give the same results. Therefore, from the results listed in Table A3 in the Appendix, we chose the *fuzzy* method based on the *Levenshtein distance*, which shows a high similarity index.

7.3. Training on our classifier

7.3.1. Making three feature sets from distribution URLs

We define a typical URL as: `< protocol>://<host>/<path>/<filename>`. From the distribution URLs, we extract the `< host>`, `<path>`, and `<filename>` strings, and insert them into three separate feature sets after optimization. We call these pools F_{host}, F_{path} and $F_{filename}$, respectively. Each feature set contained 6105 domain patterns and 799 IP patterns in F_{host}, 284 path patterns in F_{path}, and 603 filename patterns in $F_{filename}$. Hereafter, we measured the severity of each candidate URL by comparing {`<host>`, `<path>`, `<filename>`} with the features in the three pools. The file size of patterns was reduced up to 77.68% (778688 byte to 173771 byte).

7.3.2. Extracting three strings for a candidate URL

The model parses a candidate URL to `< host>`, `<path>`, `<filename>` strings. After tokenizing the URL, the model searches for the highest FR_X from $X_{max} := max(X_i, \ldots, X_n)$ by examining the similarity of F_X sets; in this regard, X can be `<host>`, `<path>`, and `<filename>`. $\sum_{X \in F} FR_X$ is the sum of FR_X over each element X of the feature sets. In fact, the model selects the fittest pattern for which $\sigma_{F_X = \langle X \rangle}(feature\ sets)$. $< X >$ is a partial string of a candidate URL. It can be `<host>`, `<path>`, or `<filename>`.

7.3.3. Model training

To train the proposed model, we tested our sampled 20,203 distribution URLs. The model then verified all datasets as malicious. However, in reality, C_L does not always satisfy the condition of $F_W \geq 7$ as malicious. The decision as to whether a URL is malicious is determined by the feature sets in order of priority. For example, the features of F_{host} carry the most weight, followed by those of F_{path} and $F_{filename}$. Thus, the features of $F_{filename}$ that have high similarity do not influence the result of a real test, because the filename has a high probability of being an FP. $F_{filename}$ only matters when accompanied by high FR from F_{host} and F_{path}. It does not affect others by itself. Therefore, unlike the impact of Fig. 6, our priority is $F_{host} > F_{path} \approx F_{filename}$.

Hence, our model gives a differentiated weight to each feature set based on preference. In the priority index, the high FR from {F_{host}, F_{path}, $F_{filename}$}, {F_{host}, F_{path}}, {F_{host}, $F_{filename}$}, and {F_{host}} provides a high malicious probability even though $\sum FR \neq 7$. That is, FR from {F_{host}} or {F_{host}, $F_{filename}$} satisfies ≥7, but FR from F_{path} or $F_{filename}$ may be <7. Next, the high FR from {F_{path}, $F_{filename}$} can be selected for malicious URLs. However, {F_{path}} and {$F_{filename}$} are

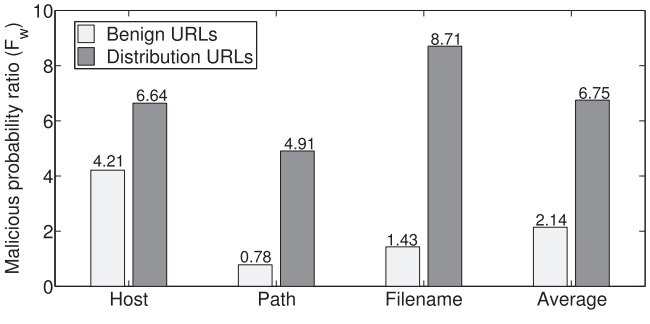

Fig. 6 – The average of the similarity probability ratio related to three finite feature sets.

too obscure to be used independently because they depend on other feature sets due to high FP; however, the likeliness of *filenames with a similar (or same) query* being malicious is so high that a future model would independently categorize them as being malicious. In other cases, the model determines when C_L is ≥7 as malicious. Accordingly, our selective C_L model does not determine malicious URLs, even though $FR_{(filename)} = 10$. After all, C_L is determined by the above priority order. The model demonstrated a true positive rate of 100.0% in training.

In summary, the similarity measurement applies selective weights to feature sets based on priority.

7.4. *Real test on proposed model*

We conducted an evaluation using new distribution URLs gathered from our `Collector` for a period of six months. There were no duplicates. The 573 benign URLs were chosen from the top 100 Alexa websites.

After classifying our entry URLs as either benign or malicious, we tested the proposed model. The experiment demonstrated the effectiveness of our *fuzzy-based similarity model* in terms of detection rate of unknown malicious URLs. Table 4 provides a brief summary of the actual test information. It exhibits an effective detection rate without new patterns being added for the past six months. The results show that our approaches were validated for malicious URL detection. That is, current web-filtering systems cannot detect new malicious URLs without new blacklist updates.

Table 4 – FP/FN detection rate based on a *fuzzy model* without new pattern updates.

URL Type	# URLs	FP	FN	C_L
Benign	573	0.00%	–	100%
Distribution	1301	–	29.98%	70.02%

Table 5 – Performance results.

	Test	Fuzzy	Prophiler
# benign	573	6.885 s	-
# malicious	1301	56.083 s	-
Total	1874	62.968 s (0.034 s)	3.297 s/page

In Table 4, the success rates are 100%, and 70.02% each, with a total of 573 and 1301 cases (the malicious rate of F_W and L_R/P_R is 61.26% and 8.76% each). The FP rate for benign URLs is 0%. We assume that the success rate would have been much higher if a greater diversity had been set up, such as new patterns. The model detected over 911 out of 1031 attempts. 311 `< host>` and 486 { `< path>`, `<filename>`} were confirmed as malicious. 114 earned with the features of geolocation and URL length (Table 5).

This model achieves high accuracy in URL mutation. From this result, we identified meaningful correlations between attackers' URL manipulation habits related to their habitual behaviors. Our features alleviate false-positive problems and augment detection coverage to 70% more than current web-filtering systems.

In terms of this trend, Fig. 7 shows the changes of TP and FP in the detection rate with respect to F_W threshold changes. Over the last six months, the average FN rate gradually de-

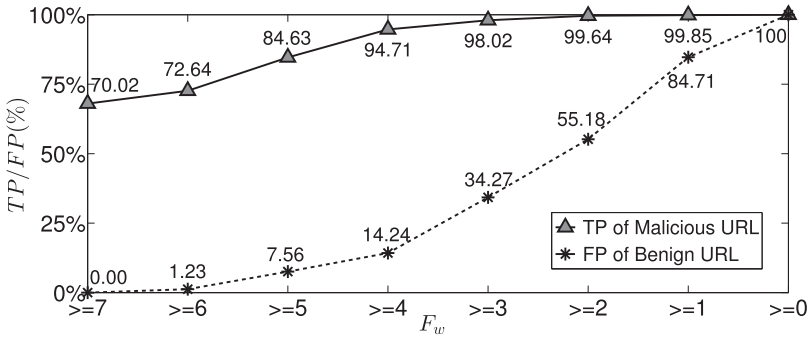

Fig. 7 – Variation in detection rate according to manipulation of F_W threshold.

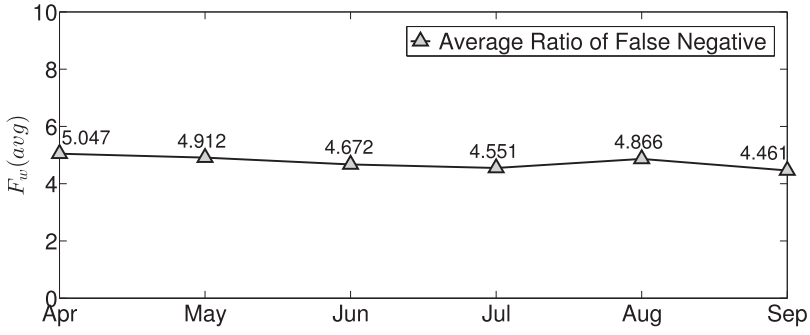

Fig. 8 – Monthly $F_{W(avg)}$ of malicious URLs that shows a false negative. $F_{W(avg)}$ means the overall average ratio of F_W.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

creased over time, as seen in Fig. 8. However, the gap is very small. This indicates that attackers leave continuously similar traces. The FN rate of the model shows its effectiveness when compared with current web-filtering systems, achieving high FPs even in the absence of a list of new patterns. The feature updates enable us to perform identification with higher accuracy.

7.5. *Performance*

Our experimental results showed a processing rate of 29.78 malicious URLs per second. This is faster than general machine learning detection (Canali et al., 2011). ML-based detection requires trade-off in extracting features. In the current Python mode, the model only executes 1874 URLs in 62.968 s as shown in Table 1. This algorithm's performance is O(N), where N is the number of potential features to be compared. The memory usage was less than 1.5%.

This model saves resources when deployed on a practical system, because the datasets of numerous blacklists are reduced to approximately 77% of their previous size. Thus, this model can help detection even when using a small-sized system that requires minimal memory consumption. Employing prior large-sized features is computationally expensive, and thus degrades overall performance.

Our dataset and this experiment might be effective only in our situation, in which we routinely monitored 0.4M domains in our local area and gathered datasets from the source. However, we believe that attackers reuse the lexical tokens of malicious URLs in terms of hostname, pathname, and filename. These properties offer alternatives to current blacklisting systems with limitations of 1:1 matching. In this regard, the benefits of our model help extend the coverage of blacklisting based on exact matching.

8. Discussion

Our model has several limitations. For example, the similarity-based matching extends the probability of detection to suspicious URLs, but it requires additional updated features because the malicious URLs can be fabricated into new types that bypass detection. In this circumstance, our URL-centric approach can determine the validity with feature updates, which yield even more precise results. That is, the detection rate can be consistently improved by incorporating additional feature data.

This model may also detect some default webpages as malicious (despite FPs) such as http://www.example.com/ or http://www.example.com/index.html. In this case, our model decides the maliciousness only with the host and a default filename. (The test result was 1.05% FP before exception processing.) Thus, we removed default filenames from F_{host}. Except for this case, our model normally analyzes the remainder of the domains.

Malicious URLs without repetition in URL segments increase the size of the feature sets, even though the patterns are definitely reduced in our model. Individual URL changes of adversaries are possible, but the execution of a massive attack leaves some footprints somewhere within the URLs. Currently, attacks that collect huge amounts of private information for pharming and phishing or ransomware are industrialized and systemized. Hence, this tendency is often observed. Naturally, our statistical results include unique malicious URLs.

Another issue concerns the overfitting of our dataset, which was collected from a Tier-1 ISP. Our dataset reflects Malware domain Malware domain blocklist (2016), PhishTank (2016), and other global malicious information. 96.58% of distribution URLs reflect information about the above malicious providers. Thus, we conclude that the experimental results are reasonable. The model also needs improved processing power and would involve the use of parallel processing in the future.

Finally, in terms of Net neutrality (2016), in collecting malicious URLs from ISP users, we have considered their privacy issues. We filtered data to minimize privacy information such as the referer, host, and request URL on inflow packets.

9. Conclusions

Adversaries leave an identifiable landscape within malicious URLs. Because attackers exhibit the same or similar behavior when they launch attacks, these traces provide information to cover a large subset of malicious URLs using relatively few features. In this regard, we have described concise concepts for two types of malicious URLs, which are broadly applicable and independent of prior URL-based analyses. The properties of the habit-centric behavior show high detection rate in similarity matching, which is based on host, path, and filename features. We conducted a comprehensive evaluation using a real-world dataset. This demonstrated that our approach is highly effective at finding unknown malicious URLs. The model is capable of performing blacklisting detection and measurements for pre-filtering of malicious URLs. It is especially helpful when checking massive URL validations. Above all, our approach overcomes the scope of 1:1 matching coverage of current web-filtering systems. We believe that our model enriches URL-only based detection scopes.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

Appendix

Table A1 – Malicious URL examples.

Malicious IP Examples	Malicious Domain Examples
110.34.196.117/cake.php	www.zilphotography.com/.errordocs/FOZ4fFXK.php
110.34.196.125/cake.php	ropeholders.info/.errordocs/sQodNA2k.php
103.251.36.92/index.html	iambrucehan.com/.errordocs/75yYodGX.php
103.251.38.103/index.html	www.wigsislandstudio.com/.errordocs/GQsvQJzH.php
103.251.37.211/index.html	www.cleanenergyhi.com/.errordocs/6UhfUL3J.php
103.251.37.213/sb.html	ozactivity.com/.errordocs/wEXfiNFD.php
103.251.37.214/sb.html	perivaleproductions.com/.errordocs/zB0ypBk9.php
103.251.37.213/main.html	live-counter.net/?click=13950265
103.251.37.214/main.html	hosttracker.net/?click=6621593
103.251.37.213/index.html	hostverify.net/?click=1110828
103.251.37.214/index.html	webexperience13.com/?click=85009921
1.226.83.40/ts/index.html	thedeadpit.com/?click=341881
1.226.83.40/ts/dy.html	internetcountercheck.com/?click=13218787
113.10.187.41/oacs19/29.html	google-ana1yticz.com/?click=172078
113.10.187.42/oacs19/29.html	coaipr.org/aqgy.html?i = 1958545
113.10.187.41/oacs19/man.html	petalconsultancy.info/aqgy.html?i = 1958545
113.10.187.42/oacs19/man.html	lindsethcpas.com/aqgy.html?i = 1958545
126.19.87.31/2222/tiancai.html	innerbath.com.au/crpy.html?j = 1958545
126.19.87.31/2222/index.html	stevebeam.com/wrpy.html?i = 1958545
116.81.235.128/3333/tiancai.html	ptsolutionsgroup.com/crgt.html?i = 1958545
116.81.235.128/3333/index.html	morehead-motorsports.com/eqgy.html?i = 1958545
126.114.226.40/3333/shifu.html	petalconsultancy.info/aqgy.html?i = 1958545
126.114.226.40/3333/index.html	cronicadelcorrugado.com/mqpt.html?i = 1958545
103.240.197.28/b.html	tvpasiones.com/arpy.html?i = 1958545
103.240.197.30/c.html	abinnetsol.ca/eqgy.html?i = 1958545
103.240.197.33/c.html	eastmead1.ipower.com/hrpt.html?i = 1958545
103.240.197.35/k.html	wheresweems.com/argt.html?i = 1958545
103.240.197.36/j.html	bestdeckshoes.com/oqpt.html?i = 1958545
103.240.197.37/v.html	lindsethcpas.com/aqgy.html?i = 1958545
113.10.187.41/live3/qq.html	burtcasey.net/wrpt.html?i = 1958545
113.10.187.42/live3/qq.html	3diporn.com/eqgy.html?i = 1958545
113.10.187.41/live2/qq.html	kirtidan.com/mqpt.html?i = 1958545
113.10.187.42/live2/qq.html	budgetcancun.com/oqpy.html?i = 1958545
113.10.187.41/code0002/qq.html	prmd.biz/wrpy.html?i = 1958545

Table A2 – Landing URLs used in Alexa top 100.

Landing URL	Distribution URL
http://l.facebook.com/lsr.php?u=http%3A%2F%2Fwww.duzonbiz.co.kr%2Fkey%2Fv3k.html&ext=1424074132&hash=AcnLdlvGrTbcTzTqzyVNC7QqyiAk4ctqMDAzdmxsPHkZEjv-	(2015-02-16) http://www.duzonbiz.co.kr/key/v3k.html
http://www.google.com/url?url=http://www.korailtour.com/UserFiles/gg/index.html&rct=j&frm=2&q=&esrc=s&sa=U&ei=VcHCVO2KDYf8ygOZ0YCICQ&ved=0CBQQFjAA&usg=AFQjCNE_7SOlEFqdS9l_cxgNPcFU6Et1yg	(2016-02-01) http://www.korailtour.com/UserFiles/gg/index.html
http://webcache.googleusercontent.com/search?q=cache:f3h3Cq-Aer4J:www.hyunjinsn.com/+&cd=1&hl=ko&ct=clnk&gl=kr	(2015-12-14) http://www.wonartschool.com/xe/libs/PEAR/view.html
http://yandex.ru/clck/jsredir?from=yandex.ru%3Bsearch%3Bweb%3B%3B&text=&etext=908.1vNg2-G_cw8lJCLRBthLM0kjY94_4GOVdgllaUV2Iyd3SZMpI_s09gbsMXpaQObG.bfe5ebfa9ebea90c7c0fe4daacc03ed318203d59&uuid=&state=AiuY0DBWFJ4ePaEse6rgeAjgs2pI3DW9J0KiE5XNXd0dp0ZMwFHoviUoYa6nzP7MFsomsouu4qcHbQqcq9usxGGO7RUCBA3CQOuv8Jg-Hj9QrokjqARXAhk_ZBegv2NHoKEopnuLoVMWYZiPqM4fPRV81es3G38m59_Blx_owikL3-IrlWDWd7PQ5JPN3hN4-uArkW6PSOiSO-qJhCUR3Q&data=UlNrNmk5WktYejR0eWJFYk1LdmtxdERJZnBDcW5pYTVPTjE4Mm42NHdyZmZPWDV4cnByVVNBTVFpS29iUkFhaE9NYkp4TFQ1WXZ2RzZCSXFlOU95emw3VnVPMFNvckxUY3ZzczI0Ukp1TEZiNlJvcm5rVVBLUQ&b64e=2&sign=cdaa450e91e8a20642b38a47b1c1c638&keyno=0&l10n=ru&cts=1450587773490&mc=5.24638133345	(2015-12-20) http://infobank.kit.ac.kr/yy/1.html

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

Table A3 – Matching rate based on similarity algorithms and defined features.

Type	Malicious URL Case 1	Malicious URL Case 2	*n-gram*	*Fuzzy*
Domain 1	05f.info	0lg.info	0.60	0.75
Domain 2	aa.sswangima.com	aa.wangma1q.com	0.72	0.84
Subdomain 1	fae.UglyAs.com	faf.UglyAs.com	0.87	0.93
Subdomain 2	aetaavaa.stuppoint.com:8000	aengopho.stuppoint.com:8000	0.64	0.78
Subdomain 3	01030242424.kt.io	01033383304.kt.io	0.55	0.71
Similar filename	75yYodGX.php	wEXfiNFD.php	0.17	0.42
Path 1 with filename	/.errordocs/75yYodGX.php	/.errordocs/wEXfiNFD.php	0.43	0.7
Path 2 with filename	01030242424.kt.io/sms.php	01033383304.kt.io/head_bak.php	0.41	0.55
Path 3 with query	live-counter.net/?click=13950265	hosttracker.net/?click=6621593	0.58	0.54
Path 4 with query	consultancy.info/aqgy.html?i = 1958545	coaipr.org/aqgy.html?i = 1958545	0.61	0.7
Path 5 with query	innerbath.com.au/crpy.html?j = 1958545	stevebeam.com/wrpy.html?i = 1958545	0.64	0.72
Same IP, different filename	103.251.37.213/index.html	103.251.37.213/main.html	0.79	0.88
Same IP, different path	113.10.187.41/live3/qq.html	113.10.187.41/code0002/qq.html	0.66	0.79
/24 IP prefix, same filename	110.34.196.117/cake.php	110.34.196.125/cake.php	0.84	0.91
/24 IP prefix, different filename	103.240.197.28/b.html	103.240.197.37/v.html	0.76	0.86
/24 IP prefix, different path	1.226.83.219/image/dav.html	1.226.83.40/ts/dy.html	0.53	0.69
/16 IP prefix	103.251.37.243	103.251.38.100	0.56	0.71
/16 IP prefix, same port	174.139.104.78:5920	174.139.165.116:5920	0.63	0.77
/16 IP prefix, same filename	1.234.27.81/k/index.html	1.234.3.185:1234/index.html	0.70	0.71
/16 IP prefix, different filename	1.234.51.238/ad.htm	1.234.91.43/good/good18.htm	0.53	0.61

REFERENCES

Alexa.com; 2015. Available from: http://www.alexa.com/topsites. [Accessed 17 August 2015].

Antonakakis M, Perdisci R, Dagon D, Lee W, Feamster N. Building a dynamic reputation system for DNS. USENIX Secur Proc 19th USENIX Conf Secur 2010;117.

Antonakakis M, Perdisci R, Lee W, Vasiloglou N II, Dagon D. Detecting malware domains at the upper DNS hierarchy. In: USENIX security symposium, vol. 11. 2011. p. 1–16.

Bilge L, Kirda E, Kruegel C, Balduzzi M. EXPOSURE: finding malicious domains using passive DNS analysis. NDSS 117. 2011.

Canali D, Cova M, Vigna G, Kruegel C. Prophiler: a fast filter for the large-scale detection of malicious web pages. Proc Int World Wide Web Conf 2011;197–206. doi:10.1145/1963405.1963436.

Cuckoo; 2016. Available from: http://www.cuckoosandbox.org/. [Accessed 5 February 2016].

Detours; 2015. Available from: http://research.microsoft.com/en-us/projects/detours/. [Accessed 10 August 2015].

Eshete B. Effective analysis, characterization, and detection of malicious webpages. In: Proceedings of the 22nd international conference on World Wide Web companion. 2013.

Eshete B, Venkatakrishnan VN. Webwinnow: leveraging exploit kit workflows to detect malicious URLs. In: Proceedings of the 4th ACM conference on data and application security and privacy. ACM; 2014.

Eshete B, Villafiorita A, Weldemariam K. BIN-SPECT: holistic analysis and detection of malicious webpages. In: Proceedings of the international conference on Security and privacy in Communication networks (SECURECOMM). Springer-Verlag; 2012. p. 149–66.

Eshete B, Alhuzali A, Monshizadeh M, Porras P, Venkatakrishnan VN, Yegneswaran V. EKHunter: a counter-offensive toolkit for exploit kit infiltration. NDSS; 2015.

Google. Safe browsing API; 2016. Available from: http://code.google.com/apis/safebrowsing/.

Holz T, Gorecki C, Rieck K, Freiling F. Measuring and detecting fast-flux service networks. NDSS; 2008.

Huang D, Xu K, Pei J. Malicious URL detection by dynamically mining patterns without pre-defined elements. World Wide Web 17 2013;1375–94. doi:10.1007/s11280-013-0250-4.

Kapravelos A, Shoshitaishvili Y. Revolver: an automated approach to the detection of evasive web-based malware. USENIX Security Symposium. 2013.

Klien F, Strohmaier M. Short links under attack: geographical analysis of spam in a URL shortener network. In: Proceedings of the 23rd ACM conference on hypertext and social media. ACM; 2012.

Krishnan S, Taylor T, Monrose F, McHugh J. Detecting malicious exploit kits using tree-based similarity searches. In: Proceedings of the sixth ACM conference on data and application security and privacy. ACM; 2016.

Le A, Markopoulou A, Faloutsos M. PhishDef: URL names say it all. In: 2011 Proceedings – IEEE INFOCOM. IEEE; 2011. p. 191–5. doi:10.1109/INFCOM.2011.5934995.

Lee S, Kim J. WarningBird: detecting suspicious URLs in Twitter stream, vol. 12. NDSS; 2012.

Ma J, Saul LK, Savage S, Voelker GM. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. World Wide Web Internet Web Inf Syst 2009a;1245–53. doi:10.1145/1557019.1557153.

Ma J, Saul LK, Savage S, Voelker GM. Identifying suspicious URLs: an application of large-scale online learning. In: International conference on machine learning. 2009b. p. 681–8. doi:10.1145/1553374.1553462.

Malware domain blocklist; 2016. Available from: http://www.malwaredomains.com/. [Accessed 19 January 2016].

Mavrommatis P, Provos N. All your iFRAMEs point to us. USENIX Security Symposium. 2008.

McAfee SiteAdvisor software; 2017. Available from: http://www.siteadvisor.com/. [Accessed 20 April 2017].

McGrath DK, Gupta M. Behind phishing: an examination of phisher modi operandi. Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET). 2008.

Maxmind; 2016. Available from: https://www.maxmind.com/en/geoip2-services-and-databases. [Accessed 14 June 2016].

Nappa A, Zubair Rafique M, Caballero J. Driving in the cloud: An analysis of drive-by download operations and abuse reporting. In: International Conference on detection of intrusions and malware, and vulnerability assessment. Berlin Heidelberg: Springer; 2013.

Nazario J, Holz T. As the net churns: fast-flux botnet observations. In: Malicious and unwanted software, 2008. MALWARE 2008. 3rd international conference on. IEEE; 2008.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013

Net neutrality; 2016. Available from: https://en.wikipedia.org/wiki/Net_neutrality. [Accessed 31 August 2016].
Norton safe web; 2017. Available from: https://safeweb.norton.com/. [Accessed 20 April 2017].
Perdisci R, Ariu D, Giacinto G. Scalable fine-grained behavioral clustering of http-based malware. Comput Netw 2013;57(2):487–500.
PhantomJS; 2016. Available from: http://phantomjs.org/. [Accessed 13 June 2016].
PhishTank; 2016. Available from: https://www.phishtank.com/. [Accessed 19 January 2016].
Rahbarinia B, Perdisci R, Antonakakis M. Segugio: efficient behavior-based tracking of malware-control domains in large ISP networks. In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE; 2015. p. 403–14.
Scan4you; 2016. Available from: http://scan4you.net/index1.php. [Accessed 1 August 2016].
Stock B, Livshits B, Zorn B. Kizzle: a signature compiler for detecting exploit kits. DSN; 2016.
Stokes JW, Andersen R, Seifert C, Chellapilla K. WebCop: locating neighborhoods of malware on the web. LEET '10 3rd USENIX Work Large-Scale Exploit Emergent Threat 2010;55. doi:10.1.1.188/9226.
Stringhini G, Kruegel C, Vigna G. Shady paths: leveraging surfing crowds to detect malicious web pages. In: Proceedings of the 2013 ACM SIGSAC conference on computer communications security. ACM; 2013.
Thomas K, Grier C, Ma J, Paxson V, Song D. Design and evaluation of a real-time URL spam filtering service. In: 2011 IEEE symposium on security and privacy. IEEE; 2011.
Uniform resource locator; 2016. Available from: https://en.wikipedia.org/wiki/Uniform_Resource_Locator. [Accessed 4 August 2016].
VirusTotal; 2014. Available from: https://www.virustotal.com/. [Accessed 27 November 2014].
Wang G, Stokes JW, Herley C, Felstead D. Detecting malicious landing pages in malware distribution networks. In: Proceedings of the international conference on dependable systems and networks. 2013. doi:10.1109/DSN.2013.6575316.
Whittaker C, Ryner B, Nazif M. Large-scale automatic classification of phishing pages. NDSS '10. 2010.
WordNet; 2015. Available from: https://wordnet.princeton.edu/. [Accessed 17 June 2015].

Sungjin Kim is currently a PhD student at KAIST. He received the BS and MS in computer science from Ohio State University and Sogang University respectively. His current research interests include various security issues such as malware, forensic, big data analysis, risk analysis, and network-based security systems.

Jinkook Kim received his BS and MS degrees from Arizona University and Cornell. His research interests include big data analysis and data mining for handing unorganized data. In particular, his research intent is to study the abnormal behaviors of people and adversaries on networks or online games.

Brent Byunghoon Kang is currently an associate professor at the Graduate School of Information Security at Korea Advanced Institute of Science and Technology (KAIST). Before KAIST, he has been with George Mason University as an associate professor. Dr. Kang received his PhD in Computer Science from the University of California at Berkeley, and MS from the University of Maryland at College Park, and BS from Seoul National University. He has been working on systems security area including botnet defense, OS kernel integrity monitors, trusted execution environment, and hardware-assisted security. He is currently a member of the IEEE, the USENIX and the ACM.

Please cite this article in press as: Sungjin Kim, Jinkook Kim, Brent ByungHoon Kang, Malicious URL protection based on attackers' habitual behavioral analysis, computers & security (2018), doi: 10.1016/j.cose.2018.01.013