Deepsquatting: Learning-Based Typosquatting Detection at Deeper Domain Levels

Conference Paper · November 2017

DOI: 10.1007/978-3-319-70169-1_26

7 authors, including:

Luca Piras
Università degli studi di Cagliari

Giorgio Giacinto
Università degli studi di Cagliari

Deepsquatting: Learning-based Typosquatting Detection at Deeper Domain Levels

Paolo Piredda[2], Davide Ariu[1,2], Battista Biggio[1,2], Igino Corona[1,2], Luca Piras[1,2], Giorgio Giacinto[1,2], and Fabio Roli[1,2]

[1] Pluribus One, Italy
[2] Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi 09123, Cagliari, Italy
{paolo.piredda,davide.ariu,battista.biggio,roli}@diee.unica.it

Abstract. Typosquatting consists of registering Internet domain names that closely resemble legitimate, reputable, and well-known ones (e.g., Farebook instead of Facebook). This cyber-attack aims to distribute malware or to phish the victims users (i.e., stealing their credentials) by mimicking the aspect of the legitimate webpage of the targeted organisation. The majority of the detection approaches proposed so far generate possible typo-variants of a legitimate domain, creating thus blacklists which can be used to prevent users from accessing typo-squatted domains. Only few studies have addressed the problem of Typosquatting detection by leveraging a passive Domain Name System (DNS) traffic analysis. In this work, we follow this approach, and additionally exploit machine learning to learn a similarity measure between domain names capable of detecting typo-squatted ones from the analyzed DNS traffic. We validate our approach on a large-scale dataset consisting of 4 months of traffic collected from a major Italian Internet Service Provider.

1 Introduction

The Domain Name System (DNS) is a crucial component of the Internet infrastructure. By means of the DNS, Internet nodes can be reliably identified and located by translating (resolving) a *string* (*i.e.*, a domain name), into an *integer* (*i.e.*, an IP address), through an hierarchical and distributed database. The DNS infrastructure effectively adds a layer of abstraction that allows for high-availability and agility of Internet services, while making them reachable through *human-friendly* domain names. Unfortunately, such DNS properties are also abused by miscreants for a myriad of Internet scams. Typosquatting is one among those subtle, widespread DNS scams mentioned before. In this attack, cybercriminals register (typo) domain names that closely resemble legitimate, reputable, and well-known ones (*e.g.*, `farebook.com` vs `facebook.com`). The main aim of miscreants is to harvest and monetize Internet traffic originally destined to the mimicked (legitimate) services [1], by exploiting their online *popularity* as well as *user mistakes*. Incoming traffic may be due to users who accidentally mistype browser URLs [10], destination emails [3], even HTML code [14], or who unluckily *click* on "legitimate-looking", malicious (*e.g.*, phishing) links [13]. An important point exploited by miscreants when building malicious links is the *gap* between user perception of domain names and the actual domain name resolution process. Domain names are usually composed by words which are expected to be read *from left to right*, *e.g.*, in languages derived from *latin* or *greek*. Conversely, domain names are actually resolved from *right to left*. Thus, in DNS entries like `facebook.com.xyz.fakedomain.it` the user-perceived domain name may be `facebook.com`, whereas the most important part is the *effective* second level domain name (2LD), *i.e.*, `fakedomain.it`, which is the domain name that has been actually registered by miscreants. Under such a single 2LD, miscreants may *freely* setup an *arbitrary* large number of domains with lower level, where `facebook.com.xyz.fakedomain.it` is just an instance. From an attacker perspective, this also makes typosquatting attacks very cheap.

Defensive registration is the main countermeasure used by large Internet providers, banks, financial operators, and in general by all the players which are heavily targeted by typosquatting and phishing attacks. Nevertheless, such measure can mitigate only the case of typosquatting occurring at the 2LD (`farebook.com` vs. `facebook.com`) while it remains totally ineffective against typosquatting attacks where the squatting occurs at lower levels. Additionally, given the large number of domain name variations that may take place, defensive registration may be very *expensive*, and *incomplete* by definition, since it may cover only typo-variations that defenders are able to foresee. From a defender perspective, a more effective and cheaper approach against typosquatting may be to detect registered typo-domains and act against them if necessary. This is where past research work focused, the most. All the proposed approaches for typosquatting detection have in common two distinguishing features. First, they focus on detecting 2LD typosquatting, through either generative models [5–12] using legitimate 2LDs as seed, or string similarity measures and time correlation in live traffic [13]. Second, the considered typo-variants where mainly obtained with the substitution (*e.g.*, `facebo0k`), addition (*e.g.*, `faceboook`), or cancellation (*e.g.*, `facebok`) of one character. This means that typosquatting introducing more of one of these operations would go undetected (*e.g.*, `faceboooook`).

In this work, we overcome the aforementioned limits. We present a novel detection approach capable of detecting typosquatting at the 2LD, but also at lower levels. In addition, we do not leverage any generative model, but we detect typo-variations of known domain names observed in the wild in large-scale networks, at the Internet Service Provider (ISP) level, as they are requested by real (victim) users. Finally, our approach is more general than state-of-the-art methods, as it is based on n-grams, and it can thus detect typo-variations with much more than one substitution, addition or cancellation.

2 Background

DNS Basics. As shown in Fig. 1, when a user wants to resolve a domain name (**1**) (*aiia2017.di.uniba.it* in the example) the resolver (e.g. the DNS server of

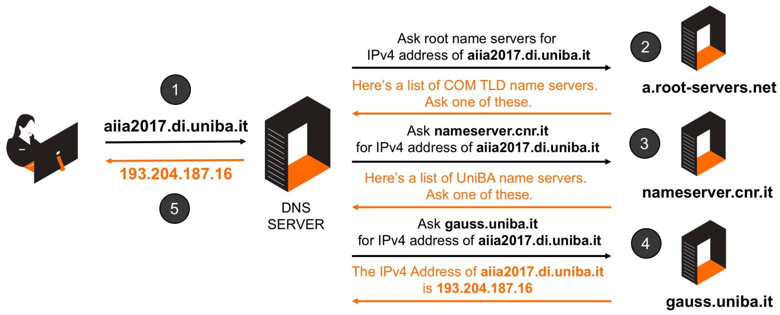

Fig. 1. An example of resolution of an internet domain name.

the Internet Service Provider) makes first a request to the root name servers (**2**), in order to obtain the list of the servers authoritatives for that **T**op-**L**evel **D**omain (*.it* in the example). Then, the resolver makes a request (**3**) to the root servers delegated for the *.it* TLD in order to get the list of nameserver(s) authoritative(s) for the *uniba* Second-Level Domain. Finally, such authoritative nameserver is queried (**4**) to obtain the IP address of *aiia2017.di.uniba.it*. The resolver finally passes such address to the user (**5**) which is then able to reach the website.

Previous work on typosquatting. Typosquatting is also known as cybersquatting. According to the United States federal law known as the Anticybersquatting Consumer Protection Act (ACPA, year 1999) [2], Cybersquatting is

> "*the registration, trafficking in, or use of a domain name that is identical to, confusingly similar to* [...] *a service mark of another that is distinctive at the time of registration of the domain name* [...] *with the bad-faith intent to profit from the goodwill of another's mark.*"

As shown in Fig. 2, typosquatting may be motivated by many different reasons, including (a) phishing scam advertisement/malware attacks; (b) collection of email messages erroneously sent to the typo domains; (c) monetization of traffic through affiliate marketing links/parked domain advertisements; (d) selling the typo domains to target brand competitors or the legitimate brand itself [3, 1]. Please note that legitimate brands may also defend themselves against cybersquatting by proactively registering, or acquiring control of, typo domains.

Points (c) and (d) were the main aim of a large-scale attack studied by Edelman [4] in 2003, while tracing back domain names registered by a unique individual. Such study highlighted more than 8,000 typo domains, most of them leading victim users (including children) to sexually-explicit websites.

Subsequent work focused on the detection of typo variations of popular domain names, according to a set of *generative* models. Such models typically receive a legitimate domain name as *seed* and then generate a set of candidate

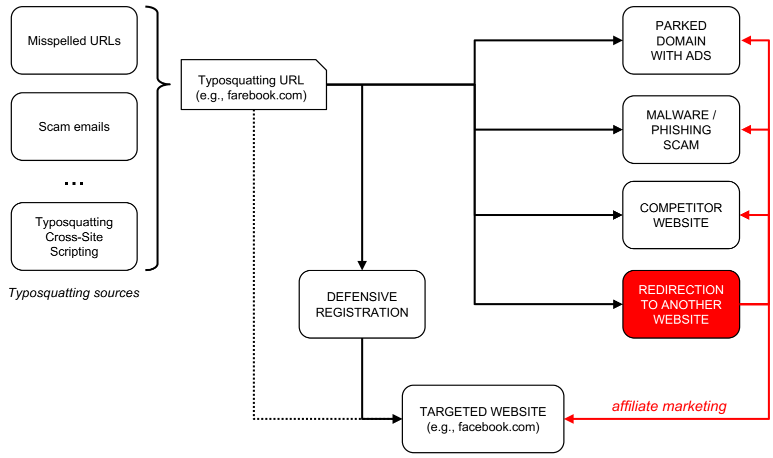

Fig. 2. Main sources of typosquatting, including defensive registrations.

typo domain names. Each domain name in such set is investigated through *active* approaches, *e.g.*, resolving it and retrieving web content [5–12].

Differently from the aforementioned approach, Khan *et al.* [13] propose a *passive* approach for detecting typosquatting domain names, by passively looking for domain resolutions and HTTP traffic within a live network (University Campus). The main assumption is that target (legitimate) domain names typically appear *close in time* with their typo versions, since users may correct their errors, *e.g.*, correct the typed URL. Under such assumption, typosquatting domain names as well as their legitimate counterparts are clustered together using time-based metrics and a Damerau-Levenshtein edit distance of *one*.

Typosquatting can be also exploited to acquire control of and exfiltrate data from websites relying on third-party (external) JavaScript libraries, thanks to typographical errors in the implementation of web pages. Nikiforakis *et al.* [14] named this threat as Typosquatting Cross-site Scripting (TXSS). The impact of this threat may be very significant as demonstrated by the authors registering several typo domains against popular domain names serving third-party JavaScript libraries (*e.g.*, `googlesyndicatio.com` vs `googlesyndication.com`).

Contributions of this work. Similarly to the work by Khan *et al.* [13], we employ a passive approach to the detection of typosquatting domain names. However, to the best of our knowledge, this is the first typosquatting detection approach that operates at the ISP level. We perform an extensive evaluation that involves traffic about hundreds of thousands of real users. Additionally, we do not rely on any assumption about temporal correlation between legitimate domains and their typo variations. Finally, our similarity measure can operate in realtime (in the sense of detecting malicious domains as they start being observed in DNS traffic, with the purpose of subsequent blacklisting) and it is more general, as it is based on *n-grams* and considers *multiple levels* of the domain name (not only the 2LD). This approach allows us to detect many typo domains in the wild that would be very difficult (if not impossible) to detect with generative approaches, and that involve manipulation at levels lower than the effective 2LD. In this study, we focus our detection on typo variations of two very popular domain names. By using n-grams as machine-learning features, we were able to get useful insights into the strategies currently employed by miscreants in the typosquatting landscape.

3 Typosquatting Domain Detection

The underlying idea used in this work is to use n-gram-based representations to detect typosquatting domains. We adapted this idea from [15], where n-gram-based representations were used to detect misspelled nouns in databases. The rationale behind our idea is that such representations may enable detecting typosquatting domains that are not necessarily within small edit distances from the targeted domain name, *i.e.*, they enable the detection of a wider set of potential *typosquatting patterns*.

Let us consider a simple example to clarify this concept. Consider the 2LD name `google` and its bi-gram representation, using also a special character to denote the beginning (#) and the end of the string ($):

$$\texttt{\#google\$} \rightarrow \texttt{\#g go oo og gl le e\$}. \quad (1)$$

Now, consider the typosquatting domain `goooooogle.com`, for which the bi-gram representation of the 2LD is `#g go oo oo ... oo og gl le e$`. By computing the intersection of this bi-grams with the previous ones obtained for `google`, one finds that all the seven bi-grams present in `google` are also present in the typosquatting domain. In practice, by assigning a binary feature to each bi-gram of the targeted domain (`google` in our running example), we can construct a numerical feature vector, suitable to train a machine-learning algorithm. In our case, the feature vector associated to `goooooogle.com` consists of seven 1s, and it is thus likely that it will be classified correctly. To yield a more complete n-gram-based representation, we also consider tri-grams and non-consecutive bi-grams (*i.e.*, skip-grams) skipping one character.

Another relevant difference with state-of-the-art approaches is that we aim to detect whether typosquatting also occurs at lower domain levels than the 2LD. To this end, we consider the aforementioned n-gram-based representations and look for *typosquatting patterns* at lower domain levels, by concatenating such patterns to form a unique feature set. In particular, it is worth remarking two aspects. First, we ignore the TLD, since for most of popular, legitimate websites (*e.g.*, google and facebook), registrations are existing at each national level. Second, to keep the number of features fixed and compact, we concatenate features extracted from the 2LD, 3LD and 4LD. Then, we consider an additional set of n-grams to identify potential typosquatting at lower level domains, from the 5LD up to the 10LD. This set is simply the set of n-grams corresponding to the level (among the 5LD, 6LD, . . . , 10LD) in which most of the n-grams match those of the targeted domain (*i.e.*, the sum of the corresponding features is maximum). For example, consider the domain

$$\underbrace{\texttt{google-974}}_{\text{3LD}}.\underbrace{\texttt{zone-one}}_{\text{2LD}}.\underbrace{\texttt{com}}_{\text{TLD}} \tag{2}$$

which has three domain levels. In this case, our feature representation is obtained by concatenating the n-grams of the targeted domain `google` found at each level:

$$\underbrace{}_{[0...0]}\underbrace{}_{[0...0]}\underbrace{\texttt{google-974}}_{[1...10]}.\underbrace{\texttt{zone-one}}_{[0...0]}.\texttt{com}\,, \tag{3}$$

namely, $[0 \dots 0, 1 \dots 10, 0 \dots 0]$, where we have fourteen 0 at the beginning (since there is neither match at the 5LD and below, nor at the 4LD), six 1 and one 0 at the 3LD (since `google` is completely matched except for the termination character), and then we have further seven 0 at the 2LD.

4 Experimental Analysis

We report here an experimental analysis to evaluate the soundness of the proposed approach. In particular, the goal of our experiments is to understand whether a learning algorithm trained on the aforementioned n-gram representation can detect typo-squatting at the 2LD and also at lower domain levels, overcoming the limits of the existing typo-squatting detection techniques [5, 7].

We conduct our experiments using real DNS data collected from an Italian ISP. We focus on detecting typo-squatting against two popular web services, *i.e.*, Google and Facebook. To this end, we built two datasets (one per service) as described below.

Data and ground-truth labels. We first extracted all domain names requested and successfully resolved (along with the corresponding server IP addresses) by the users of the considered ISP between August 1, 2016 and November 30, 2016. Then, to establish the ground-truth labels reliably, *i.e.*, to label each domain as typo-squatting or not, we adopted the following strategy. We started by considering all domain names for which the 2LD is in the Alexa Top 50 as legitimate. Malicious typo-squatting domains were identified by first extracting all domain names whose 2LD has a Damerau-Levenshtein distance equal to 1 from the string *google* (for the Google dataset) and *facebook* (for the Facebook dataset). This includes all domains that one would find using the state-of-the-art generative approaches proposed in [5, 7] along with other domains for which the Damerau-Levenshtein distance is 1 but that are not encompassed by the aforementioned generative approaches. To find suspicious typo-squatting attempts beyond the 2LD, we used the approach in [15], originally proposed to find misspellings in databases. This technique detects words (from a given list) that are potential typing errors of a source word (`google` or `facebook` in our case). In particular, we used a simplified version that simply counts how many bigrams the source word has in common with each other word in the list (using special characters to denote the beginning and the end of each word, and considering the order of each gram, as described in the previous section). Once this measure of *overlap* between each word and the source word was computed, as in [15], a simple clustering procedure was used to separate potential typos from words that are clearly not typos of the source word. In our case, we considered as a word the content of each domain level (note that a single domain can consist of more words, *e.g.*, "abc.dot.gooogle.bizz.com" consists of four different words, excluding the TLD ".com", where only `gooogle` is effectively a typo of `google`). The list of potential word typos was then matched against the domain list (at each level) to find all suspicious typo-squatting domain names. However, recall that finding a domain name which is relatively close to the name of a legitimate domain is not enough to declare it as a typo-squatting attempt. As mentioned in previous work [10], roughly half of these suspicious domain names is in fact legitimate (think, *e.g.*, to *defensive registrations*). We thus checked whether the resolved IPs corresponding to the suspicious domains identified as potential typo-squatting host effectively some malicious activity or scam. To this end, we used the API service provided by VirusTotal,[3] and labeled a suspicious domain as malicious only if the resolved IP is known to be at least in a public blacklist. To reduce the probability of labeling errors, we collected such labels in January 2017, some months after our DNS traffic was observed. Eventually, we labeled a domain as typo-squatting only if (*i*) it was identified as similar to `google` or `facebook` (both in terms of Damerau-Levenshtein distance and of the clustering approach discussed in [15]), and if (*ii*) the resolved IP address of the corresponding server was known to be malicious from publicly-available blacklisting services, using the interface provided by VirusTotal (as mentioned before).

However, these services often label legitimate domains as malicious, since they simply report whether a server has been contacted by malware, and malware typically contact also legitimate services for different reasons (*e.g.*, to mislead reverse-engineering analyses, check connectivity, *etc.*). We thus further refine the ground-truth labels with a thorough manual analysis. In particular, we found that most of the domains associated to blacklisting services are labeled as malicious, as they have been probably contacted by malware. This may happen simply when a malware sample checks whether a domain is already known to be malicious or not, to avoid connecting to it. In this way, the operations performed by the malware sample may remain undetected. For example, the DNS query `facebook.com.sbl-xbl.spamhaus.org` checks whether `facebook.com.sbl` has been blacklisted by Spamhaus. If not, the malware sample can contact it incurring a lower risk of detection. These kinds of queries are definitely not typosquatting attempts but rather legitimate queries to blacklisting services. We therefore change their label to legitimate, even if they may be easily misclassified as potential typosquatting domains by our algorithm.

[3] `https://www.virustotal.com/it/documentation/public-api/#getting-ip-reports`

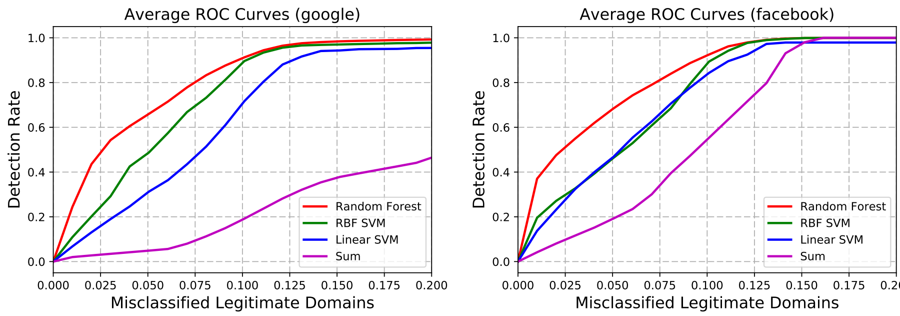

Fig. 3. Average ROC curves exhibited by the considered learning algorithms on `google` (*left*) and `facebook` (*right*) datasets.

Classifiers. We trained different state-of-the-art learning algorithms on the n-gram-based feature representation proposed in previous section (separately on each dataset, *i.e.*, for each monitored domain). In particular, we considered Support Vector Machines (SVMs) with linear and Radial Basis Function (RBF) kernels, and Random Forests (RFs). We tuned their parameters using a 5-fold cross-validation procedure on the training data, in order to minimize the classification error. For the RF classifiers, we optimized the number of base decision-tree classifiers $k \in \{10, 15, 20, \ldots, 100\}$; for the linear SVM, we optimized the regularization parameter $C \in \{10^{-2}, \ldots, 10^{3}\}$ and for the RBF SVM we additionally tuned the kernel parameter $\gamma \in \{10^{-3}, \ldots, 10^{3}\}$. We also considered a baseline algorithm that corresponds to the sum of the n-gram feature values (denoted with "Sum" for short).

Performance Evaluation. Performance was evaluated in terms of Receiver Operating Characteristic (ROC) curves, averaged on 5 random training-test splits, using 80% of the data for training (and 20% for testing) in each split.

Experimental Results. Results are reported in Fig. 3, for both `google` and `facebook` datasets. First, note that Sum is outperformed by all learning algorithms used in our experiments. This witnesses that using machine learning in this case is really helpful to find some specific *registration patterns* corresponding to malicious typosquatting domains, *i.e.*, the only presence of some specific n-grams in the domain name is not sufficient to classify it as a potential typosquatting domain. Another interesting observation is that Random Forests outperform significantly the SVM-based classifiers. This may be due to the fact that they leverage bagging and the random subspace method to build a classifier ensemble of decision trees, which typically improves the performance over baseline, monolithic learning algorithms.

In Table 1 we additionally report the detection rates of the RF classifier (which performed best) for typosquatting occuring at different domain levels and

Table 1. Detection rates of the Random Forest (RF) classifier for typosquatting at different domain levels (from 2LD to 7LD, 8LD+ denotes the grouping of 8LD, 9LD and 10LD) and Damerau-Levenshtein (DL) distances, for the `google` and `facebook` data. In both cases, the operating point of the RF is set to achieve a 2.5% false positive rate, which roughly corresponds to a detection rate of 50%, as also shown in Fig. 3.

`google`	**DL = 0**			**DL = 1**			**DL >1**			**Overall (DL ≥0)**		
	True	*Detected*		*True*	*Detected*		*True*	*Detected*		*True*	*Detected*	
2LD	0	0		576	458	79,5%	412	328	79,6%	988	786	79,6%
3LD	305	162	53,1%	17	5	29,4%	97	63	64,9%	419	230	54,9%
4LD	483	50	10,4%	13	10	76,9%	193	43	22,3%	689	103	14,9%
5LD	161	27	16,8%	0	0		54	31	57,4%	215	58	27,0%
6LD	55	24	43,6%	1	0	0,0%	34	16	47,1%	90	40	44,4%
7LD	17	11	64,7%	0	0		4	1	25,0%	21	12	57,1%
8LD+	8	4	50,0%	0	0		11	7	63,6%	19	11	57,9%
Total										2441	1240	50,8%

`facebook`	**DL = 0**			**DL = 1**			**DL >1**			**Overall (DL ≥0)**		
	True	*Detected*		*True*	*Detected*		*True*	*Detected*		*True*	*Detected*	
2LD	0	0		387	314	81,1%	928	134	14,4%	1315	448	34,1%
3LD	347	26	7,5%	16	14	87,5%	334	282	84,4%	697	322	46,2%
4LD	216	69	31,9%	2	0	0,0%	352	342	97,2%	570	411	72,1%
5LD	83	31	37,3%	0	0		7	2	28,6%	90	33	36,7%
6LD	22	0	0,0%	0	0		0	0		22	0	0,0%
7LD	1	1	100,0%	0	0		89	89	100,0%	90	90	100,0%
8LD+	11	5	45,5%	0	0		0	0		11	5	45,5%
Total										2795	1309	46,8%

at different edit distances. This shows that our approach is capable of detecting typosquatting attempts beyond the state-of-the-art techniques proposed so far.

Besides the aforementioned considerations, the reported results clearly show that the proposed method is not ready to be deployed on a large scale, *e.g.*, to monitor the DNS traffic of an ISP, due to a rather high false positive rate (*i.e.*, fraction of misclassified legitimate domains). Nevertheless, the reason is simply that the structure of the domain name does not suffice to correctly identify a typosquatting domain hosting malicious or suspicious activities. To confirm this issue, we report some examples of misclassified domains by our algorithm in Table 2. By a deeper inspection of the misclassified legitimate domains, we have discovered that, in practice, some may host malicious activities or even malware although VirusTotal labeled them as legitimate (*e.g.*, this happens for `googllee.co.uk`, while the suspicious `6ooogoogle.ru` is inactive). Note also that defensive registrations are not always correctly labeled by VirusTotal. This witnesses that the false positive rate may be even lower than the one effectively reported in our experiments (due to the fact that our ground-truth labeling

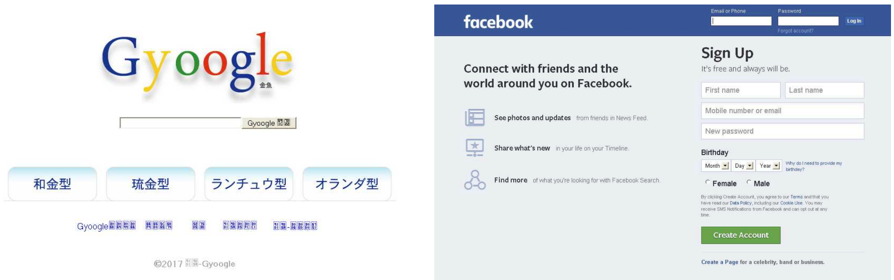

Fig. 4. Screenshot of the typosquatting domain `www.gyoogle.net` (*left*), and of the defensive registration at `www.faceboo.com` (*right*).

source is not very reliable). Furthermore, it should be clear from the reported set of examples that categorizing a malicious typosquatting domain by only looking at the structure of its name is an ill-posed problem; *e.g.*, finding "google" at the beginning of a domain name beyond the 2LD is a typosquatting pattern recognized correctly by our algorithm in most of the cases. For this reason, to reduce the false positive rate, more characteristics should be taken into account, as done in previous work aimed to detect malicious domains from DNS traffic [16–18]. Despite this, our analysis shows that characterizing the domain name using n-grams and machine learning may improve the detection of typosquatting domains over the state of the art, *i.e.*, beyond the 2LD and small Damerau-Levenshtein distance values. We thus believe that our approach may be particularly useful to improve the aforementioned existing systems aimed to detect malicious domains while passively monitoring the DNS traffic [16–18], especially since typosquatting makes sense only if the domain name retains some degree of similarity with respect to the targeted website; in other words, this is a constraint for the attack to successfully mislead most of the unexperienced Internet users. To summarize, using n-gram-based representations and machine learning as advocated in this work can be thus deemed an interesting research direction to improve systems that detect malicious domains from DNS traffic.

5 Conclusions and Future Work

In this work, we proposed a passive DNS analysis approach to the detection of typosquatted Internet domain names. The proposed approach provides an advancement with respect to the solutions proposed so far in the literature as it enables the detection of a typosquatting patterns beyond the 2LD and for values of the Damerau-Levenshtein distance higher than 1, which is the kind of typosquatting usually consideres also by preventive registration mechanisms. The main limitation of our approach is currently represented by the false positive rate, which may be reduced using whitelisting; however, we strongly believe that our work may be useful to improve previous work for the detection of malicious

Table 2. Some examples of domain names correctly-classified as typosquatting by our algorithm (using the RF classifier) along with some misclassified legitimate ones. Defensive registrations and misclassified queries to blacklisting services are highlighted with (*) and (**), respectively.

google	
correctly detected	*misclassified legitimate domains*
google.com-prize4you.com	www.goolge.de (*)
google.com–support.info	news.gogle.it (*)
google.com-updater.xyz	googlehouse.com
google.com-62.org	googllee.co.uk
google.itoogle.it	www.sxgoogle.net
www.gyoogle.net	www.googlel.com
google.com-1prize4you.com	6ooogoogle.ru

facebook	
correctly detected	*misclassified legitimate domains*
facebook.com-winner.me	www.tai-facebook.xyz
ww25.facebook.comfacebook.com	facebook-jaegermeister.syzygy.de
facebook.com-feed.top	facebook.feargames.it
facebook.com-iii.org	facebook.fantatornei.com
facebook.com-prize4you.com	faceboock.ddns.net (**)
facffebook.com	faceslapbook.blogspot.com
www.faceboo.com (*)	facebook.fantatornei.com

domains from DNS traffic [16–18]. Our research work on this area is currently ongoing, and future enhancements will include both the analysis of the contents hosted by the detected domains, as well as the analysis of features extracted at domain registration time and DNS features especially to correctly categorize defensive registrations.

References

1. Spaulding, J., Upadhyaya, S.J., Mohaisen, A.: The landscape of domain name typosquatting: Techniques and countermeasures. In: The 11th International Conference on Availability, Reliability and Security. Volume abs/1603.02767. (2016)
2. Senate, U.: The anticybersquatting consumer protection act (August, 5 1999)
3. Zetter, K.: Researchers' typosquatting stole 20 gb of e-mail from fortune 500. Wired.com (August 2011)
4. Edelman, B.: Large-scale registration of domains with typographical errors. Technical report, Berkman Center for Internet & Society - Harvard Law School (2003)
5. Wang, Y.M., Beck, D., Wang, J., Verbowski, C., Daniels, B.: Strider typo-patrol: Discovery and analysis of systematic typo-squatting. In: Proceedings of the 2Nd Conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2. SRUTI'06, Berkeley, CA, USA, USENIX Association (2006) 5–5

6. Holgers, T., Watson, D.E., Gribble, S.D.: Cutting through the confusion: A measurement study of homograph attacks. In: Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference. ATEC '06, Berkeley, CA, USA, USENIX Association (2006) 24–24
7. Banerjee, A., Barman, D., Faloutsos, M., Bhuyan, L.N.: Cyber-fraud is one typo away. In: IEEE INFOCOM 2008 - The 27th Conference on Computer Communications. (April 2008)
8. Moore, T., Edelman, B.: Measuring the perpetrators and funders of typosquatting. In: Proceedings of the 14th International Conference on Financial Cryptography and Data Security. FC'10, Berlin, Heidelberg, Springer-Verlag (2010) 175–191
9. Nikiforakis, N., Acker, S.V., Meert, W., Desmet, L., Piessens, F., Joosen, W.: Bitsquatting: exploiting bit-flips for fun, or profit? In: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013. (2013) 989–998
10. Szurdi, J., Kocso, B., Cseh, G., Spring, J., Felegyhazi, M., Kanich, C.: The long "taile" of typosquatting domain names. In: Proceedings of the 23rd USENIX Conference on Security Symposium. SEC'14, Berkeley, CA, USA, USENIX Association (2014) 191–206
11. Nikiforakis, N., Balduzzi, M., Desmet, L., Piessens, F., Joosen, W.: Soundsquatting: Uncovering the use of homophones in domain squatting. In: in Proceedings of the 17th Information Security Conference (ISC). (2014)
12. Agten, P., Joosen, W., Piessens, F., Nikiforakis, N.: Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In: 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2015. (2015)
13. Khan, M.T., Huo, X., Li, Z., Kanich, C.: Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In: 2015 IEEE Symposium on Security and Privacy. (May 2015) 135–150
14. Nikiforakis, N., Invernizzi, L., Kapravelos, A., Van Acker, S., Joosen, W., Kruegel, C., Piessens, F., Vigna, G.: You are what you include: Large-scale evaluation of remote javascript inclusions. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. CCS '12, New York, NY, USA, ACM (2012) 736–747
15. Mazeika, A., Böhlen, M.H.: Cleansing databases of misspelled proper nouns. In: Proceedings of the First Int'l VLDB Workshop on Clean Databases, CleanDB 2006, September 11, 2006, Seoul, Korea (Co-located with VLDB 2006). (2006)
16. Perdisci, R., Corona, I., Giacinto, G.: Early detection of malicious Flux networks via large-scale passive DNS traffic analysis. IEEE Transactions on Dependable and Secure Computing **9**(5) (August 2012) 714–726
17. Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C.: Exposure: A passive dns analysis service to detect and report malicious domains. ACM Trans. Inf. Syst. Secur. **16**(4) (April 2014) 14:1–14:28
18. Hao, S., Kantchelian, A., Miller, B., Paxson, V., Feamster, N.: Predator: Proactive recognition and elimination of domain abuse at time-of-registration. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16, New York, NY, USA, ACM (2016) 1568–1579