Proactive Discovery of Phishing Related Domain Names

Samuel Marchal, Jérôme François, Radu State, Thomas Engel

Proactive Discovery of Phishing Related Domain Names

Samuel Marchal, Jérôme François, Radu State, and Thomas Engel

SnT - University of Luxembourg, Luxembourg,
firstname.lastame@uni.lu

Abstract. Phishing is an important security issue to the Internet, which has a significant economic impact. The main solution to counteract this threat is currently reactive blacklisting; however, as phishing attacks are mainly performed over short periods of time, reactive methods are too slow. As a result, new approaches to early identify malicious websites are needed. In this paper a new proactive discovery of phishing related domain names is introduced. We mainly focus on the automated detection of possible domain registrations for malicious activities. We leverage techniques coming from natural language modelling in order to build pro-active blacklists. The entries in this list are built using language models and vocabularies encountered in phishing related activities - "secure", "banking", brand names, etc. Once a pro-active blacklist is created, on-going and daily monitoring of only these domains can lead to the efficient detection of phishing web sites.

Keywords: phishing, blacklisting, DNS probing, natural language

1 Introduction

The usage of e-commerce, e-banking and other e-services is already current practice in the life of modern Internet users. These services handle personal and confidential user data (login, password, account number, credit card number, etc.) that is very sensitive. As a result, threats emerged for which attackers attempt to steal this data and use it for lucrative purposes. An example of these threats is phishing, a criminal mechanism employing both *technical subterfuge* and *social engineering* to abuse the naivety of uninformed users. Phishing mainly targets (75%) financial and payment activities and its cost is estimated to many billion of dollars per year[1].

Phishing attacks leverage some techniques such as e-mail spoofing or DNS cache poisoning to misdirect users to fake websites. Attackers also plant crime-ware directly onto legitimate web server to steal users data. However, the two last techniques require to penetrate web servers or change registration in DNS server, which might be difficult. Most often, phishers try to lure Internet users by having them clicking on a rogue link. This link seemed to be trustworthy because it contained a brand name or some keywords such as *secure* or *protection*.

[1] http://www.brandprotect.com/resources/phishing.pdf, accessed on 04/04/12

Current protecting approaches rely on URL blacklists being integrated in client web browsers. This prevents users from browsing malicious URLs. Google Safe Browsing[2] or Microsoft Smart Screen[3] are two examples and their efficiency has been proved in [14]. However, as reported in [21], the average uptime of phishing attacks is around 2 days and the median uptime is only 12 hours. Due to this very short lifetime, reactive blacklisting is too slow to efficiently protect users from phishing; hence, proactive techniques must be developed. The previous report also points out that some phishing attacks involve URLs containing unique number in order to track targeted victims. The only common point between these unique URLs remains their domain name; as a result domain name blacklisting should be more efficient and useful than URL blacklisting. Moreover, it emphasizes that one maliciously registered domain name is often used in multiple phishing attacks and that each of them use thousands of individual URLs. As a result, the identification of only one phishing domain name can lead to protect Internet users from tens of thousand malicious URLs.

According to recent reports [21, 1] from the Anti Phishing Working Group (APWG), the number of phishing attacks is fast growing. Between the first half of 2010 and the first half of 2011 the number of phishing attacks raised from 48,244 to 115,472 and the number of dedicated registered domains from 4,755 to 14,650. These domains are qualified as maliciously registered domains by the APWG. These counts highlight the trend that attackers prefer to use more and more their own maliciously registered domains rather than hacked named domains for phishing purposes. Moreover, observations reveal that malicious domain names and particularly phishing ones are meaningful and composed of several words to obfuscate URLs. Attackers insert some brands or keywords that are buried in the main domain name to lure victims, as for instance in `protectionmicrosoftxpscanner.com`, `google-banking.com` or `domainsecurenethp.com`. As a result, this paper focuses on the identification of such phishing domain names that are used in URL obfuscation techniques.

This paper introduces a pro-active domain monitoring scheme that generates a list of potential domain names to track in order to identify new phishing activities. The creation of the list leverages domain name features to build a natural language model using Markov chains combined with semantic associations. We evaluate and compare these features using real malicious and legitimate datasets before testing the ability of our approach to pro-actively discover new phishing related domains.

The rest of this paper is organized as follows: Section 2 describes the design of the architecture and the steps to follow to generate malicious domain names. Section 3 introduces the datasets used for the validation and experimentation. In section 4, differences between malicious and legitimate domains are analyzed and domain name generation is tested in some real case studies. Finally, related

[2] http://code.google.com/apis/safebrowsing/, accessed on 04/04/12
[3] http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter, accessed on 04/04/12

work is discussed in section 5. We conclude in section 6 and point out the further research to be done.

2 Modeling a Phisher's language

Phishers are human and will generate names for their domains using some simple patterns. They will use names that are similar to legitimate domain names, append some other words that come from a target vocabulary and leverage some domain specific knowledge and expertise. Thus, we argue that pro-active monitoring can emulate this process and generate potential domains to be tracked permanently. This tracking can be done on a daily basis and thus detect new phishing sites. This requires however to generate domain names that are or will be involved in phishing activities. These names follow a model build on statistical features. The domain names considered in our work are composed of two parts, the top level domain (TLD) and the second level domain also called *main domain*. In this approach, the TLD can be either only one level domain "`.com`" or more "`.org.br`", we refer to Public Suffix List[4] to identify this part of the URL and the *main domain* is considered as the label preceding the TLD. For the rest of the paper these domains (main domain + TLD) will be called *two-level-domains*. Assuming a dataset containing domain names and URLs such as:

- `www.bbc.co.uk`
- `wwwen.uni.lu/snt/`
- `secan-lab.uni.lu/index.php?option=com_user&view=login`

Features are extracted only from the *two-level-domains*, which are respectively `bbc.co.uk`, with `bbc` the *main domain* and `co.uk` the TLD, for the first one and `uni.lu` for the two others, with `uni` the *main domain* and `lu` the TLD. The domain names generated are also *two-level-domains*.

2.1 Architecture

An overview of our approach is illustrated in Figure 1 where the main input is a list of known domains related to malicious activities. Based on that, the first stage (1) decomposes the name and extracts two main parts: the TLD and the *main domain*. Then, each of these two is divided into words (2). For TLD, a simple split regarding the dot character is sufficient but for the second, a real word segmentation is required to catch meaningful words. As illustrated here with a small example, *macromediasetup.com/dl.exe*, the following words are extracted:

- TLD: `com`
- *main domain*: `macro`, `media`, `set`, `up`

[4] http://publicsuffix.org, accessed on 08/03/12

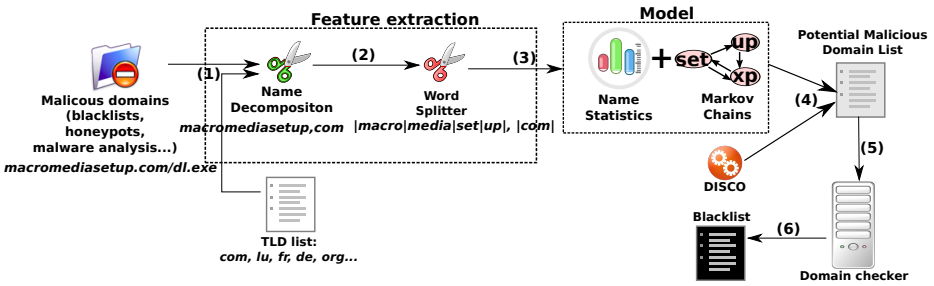


Fig. 1. Proactive Malicious Domain Name Discovery – Overview

These features are then used to build a model (3) by computing statistics, as for example the length distribution of the *main domain* in words as well as a Markov chain for representing probabilistic word transitions. The statistics and Markov chain model are computed for each level. Next, this model is combined (4) with semantic extensions. This leads to generating similar words (only for the *main domain*) and a list of potential malicious domain is built. These latter are checked online (5) for potential phishing activities. The online validation is not described in this paper, but can be done with various techniques: signature-based approach, honeypots, manual analysis, etc. Hence, our experiments are based on publicly available blacklists for cross-validation (see section 3).

2.2 Features extraction

Features : Given a set of *two-level-domains* as $D = \{d_1, ..., d_p\}$, a set of words as $W = \{w_1; ....; w_p\}$ and a set of domain levels $L = \{l_1; l_2\}$ where l_1 is the *TLD* and l_2 is the *main domain*, we define:

- $\#len_{l,n}$ the number of domains $d \in D$ having the l^{th} level ($l \in L$) composed of n words
- $\#word_{l,w}$ the number of domains $d \in D$ containing the word $w \in W$ at the level $l \in L$
- $\#fisrtword_{l,w}$ the number of domains $d \in D$ having the l^{th} level ($l \in L$) starting with the word $w \in W$
- $\#biwords_{l,w_1,w_2}$ the number of domains $d \in D$ containing the consecutive words w_1 and w_2 ($(w_1, w_2) \in W^2$) at the level $l \in L$

The following list groups the features extracted from a list of domains or URLs :

- $distlen_{l,n}$: the distribution of the length $n \in \mathbb{N}$ expressed in word for a level $l \in L$ and defined as:

$$distlen_{l,n} = \frac{\#len_{l,n}}{\sum_{i \in \mathbb{N}} \#len_{l,i}} \tag{1}$$

- $distword_{l,w}$: the distribution of the number of occurrences of a word $w \in W$ at the level $l \in L$ and defined as:

$$distword_{l,w} = \frac{\#word_{l,w}}{\sum_{i \in W} \#word_{l,i}} \tag{2}$$

- $distfirstword_{l,w}$: the distribution of the number of occurrences of a word $w \in W$ as first word for the level $l \in L$ and defined as:

$$distfirstword_{l,w} = \frac{\#fisrtword_{l,w}}{\sum_{i \in W} \#fisrtword_{l,i}} \tag{3}$$

- $distbiwords_{l,w_1,w_2}$: the distribution of the number of occurrences of a word $w_2 \in W$ following the word $w_1 \in W$ for the level $l \in L$ and defined as:

$$distbiwords_{l,w_1,w_2} = \frac{\#biwords_{l,w_1,w_2}}{\sum_{i \in W} \#biwords_{l,w_1,i}} \tag{4}$$

Word extraction : The *main domain* of DNS names can be composed of several words like `computeraskmore` or `cloud-anti-malware`. Using a list of separating characters, as for instance "-" is too restrictive. We have thus used a word segmentation method, similar to the one described in [22]. The process is recursive by successively dividing the label in 2 parts that give the best combination, *i.e.* with the maximum probability, of the first word and the remaining part. Therefore, a label l is divided in 2 parts for each position i and the probability is computed:

$$P(l,i) = P_{word}(pre(l,i))P(post(l,i)) \tag{5}$$

where $pre(l,i)$ returns the substring of l composed of the first i characters and $sub(l,i)$ of the remaining part. $P_{word}(w)$ returns the probability of having the word W equivalent to its frequency in a database of text samples.

TLDs are split in different labels using the separating character ".".

2.3 Domain names generation model

The generator designed for domain generation is mainly based on an n-gram model. Coming from natural language processing an n-gram is a sequence of n consecutive *grams*. These *grams* are usually characters, but in our approach, *grams* are words. We especially focus on bigrams of words that are called *biwords*. These couples of words are further used to build a Markov chain through which *two-level-domains* are generated.

Markov Chain : A Markov chain is a mathematical system that undergoes transitions from one state to another. Each possible transition between two states can be taken with a transition probability. Two Markov chains are defined in the domain generation model, one for each level, l_1 and l_2. The states of the Markov chains are defined as the words $w \in W$ and the probability of transition between

two words w_1 and w_2 for the level $l \in \{1; 2\}$ is given by $distbiwords_{l,w_1,w_2}$. A part of a created Markov chain is given in Table 1 for some transitions, and the associated probabilities, starting from the word *pay*. In order to generate new names the Markov chain is completed with additional transitions that have never been observed - this technique is called additive smoothing or Laplace smoothing. For each state s, a small probability (0.05) is assigned for transitions to all the words that have been observed at the level l and for which s does not have any transition yet. This probability is shared between the words of the level l according to the distribution $distword_{l,w}$. The same method is applied for states s that do not have any existing transitions. In this case, their transitions follow the probability given by the distribution $distword_{l,w}$.

Transition	per	z	for	secure	bucks	bill	process	pay	account	soft	page	...
Probability	0.13	0.1	0.06	0.06	0.06	0.03	0.03	0.03	0.03	0.03	0.03	...

Table 1. Example of Markov chain transitions for the state *pay*

For *two-level-domains* generation, the first state is randomly initialized using $distfirstword_{l,w}$, the number of transitions that must be completed in the Markov chain is randomly determined using $distlen_{l,n}$. Given these two parameters by applying n steps from word w in the Markov chain, a label is generated for the level l.

Semantic Exploration : The words composing the *main domains* of different malicious domains often belong to the one or more shared semantic fields. Given some malicious domain names such as *xpantiviruslocal.com*, *xpantivirusplaneta.com*, *xpantivirusmundo.com* and *xpantivirusterra.com*, it clearly appears that they are related. Applying the word extraction process, from all of these domains, the words "xp", "anti" and "virus" will be extracted and the four words "local", "planeta", "mundo" and "terra" will be extracted from each of them. These four words are closely related, particularly the three last ones. As a result, given one of these domains, the remaining three could be found as well. However, even if this intuitive conclusion is obvious for human, it is more complicated to implement it in an automatic system.

For this purpose, DISCO [12] is leveraged, a tool based on efficient and accurate techniques to automatically give a score of relatedness between two words. To calculate this score, called similarity, DISCO defines a sliding window of four words. This window is applied to the content of a dictionary such as Wikipedia[5] and the metric $\|w, r, w'\|$ is calculated as the number of times that the word w' occur r words after the word w in the window, therefore $r \in \{-3; 3\} \setminus \{0\}$. Table 2 highlights an example of the calculation of $\|w, r, w'\|$ for two sample pieces of text. Afterwards the mutual information between w and w', $I(w, r, w')$ is defined as:

$$I(w, r, w') = log \frac{(\|w, r, w'\| - 0.95) \times \|*, r, *\|}{\|w, r, *\| \times \|*, r, w'\|} \tag{6}$$

[5] http://www.wikipedia.org, accessed on 04/04/12

Finally, the similarity $sim(w_1, w_2)$ between two words w_1 and w_2 is given by the formulae:

$$sim(w_1, w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} I(w_1, r, w) + I(w_2, r, w)}{\sum_{(r,w)\in T(w_1)} I(w_1, r, w) + \sum_{(r,w)\in T(w_2)} I(w_2, r, w)} \quad (7)$$

where $T(w)$ is all the pairs $(r, w') \mid I(w, r, w') > 0$.

Using this measure and given a word w_1, DISCO returns the x most related words ordered by their respective similarity score $sim(w_1, w_2)$. Based on the words extracted from the *main domain*, DISCO is used to compose new labels for the *main domain*.

position	-3	-2	-1	0	+1	+2	+3
sample 1	a	client	uses	services	of	the	platform
sample 2	the	platform	provides	services	to	the	client

$\|\|services, -3, a\|\| = 1$	$\|\|services, -3, the\|\| = 1$
$\|\|services, -2, client\|\| = 1$	$\|\|services, -2, platform\|\| = 1$
$\|\|services, -1, uses\|\| = 1$	$\|\|services, -1, provides\|\| = 1$
$\|\|services, 1, of\|\| = 1$	$\|\|services, 1, to\|\| = 1$
$\|\|$**services,2,the**$\|\| =$ **2**	$\|\|services, 3, client\|\| = 1$
$\|\|services, 3, platform\|\| = 1$	

Table 2. Example of co-occurrence counting (2 windows centered on *services*)

A complete example of label generation is illustrated in Figure 2 for the level 2 (*main domain*) with (1), the selection of the length of the label in words, and (2) the selection of the first word that starts the label. The Markov chain is applied for the remaining words to generate (3). For each word at the step (2) and (3), DISCO is applied to generate other words. The same scheme generates TLD for the level 1 without using DISCO.

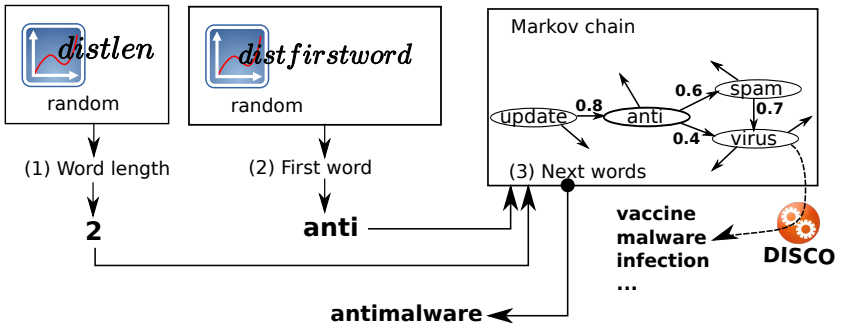


Fig. 2. Main domain generator

3 Dataset

For assessing our approach, two datasets are selected. The first one is a malicious dataset composed of domain names from which maliciousness has been confirmed. The second dataset is a legitimate dataset containing non-malicious domain names. In a first step, these will be used to show that the features introduced in section 2.2 allow to discriminate phishing domain names from legitimate ones. In a second step, the malicious dataset will be used to show the efficiency of the generation of phishing domain names.

3.1 Malicious Dataset

To compose the dataset of malicious domain names, three freely downloadable blacklists are used. These have been selected because each of them proposes an historical list of blacklisted domains ordered by their discovery date. These have been collected during at least the last three years. This is an essential dataset requirement in order to test the predictability of the approach.

- **PhishTank**[6]: PhishTank is a community website where anybody can submit a suspicious phishing URL that will be further checked and verified. The downloaded historical blacklist contains 3,738 phishing URLs.
- **DNS-BH**[7]: DNS-Black-Hole aims to maintain an up-to-date list of domains that spread malware and spyware. A list of 17,031 malicious domains is available.
- **MDL**[8]: Malware Domain List is another community project aimed at creating and maintaining a blacklist of domains involved in malware spreading. This list contains 80,828 URL entries.

DNS-BH and MDL are not only dedicated to phishing, but also to malware diffusion. These two lists have been chosen because as described in [1], diffusion of malware designed for data-stealing and particularly crimeware is a big part of phishing activities. This various dataset allows also to strengthen the validation of our approach (introduced in section 4). Following the extraction of the distinct domain names from the 101,597 URLs and the deletion of duplicated entries between the three lists, the final dataset contains 51,322 different *two-level-domains*. Out of these 51,322 domain names, 39,980 have their *main domain* divisible in at least two parts.

3.2 Legitimate Dataset

The objective is to faithfully represent realistic normal domain names. This dataset is selected to show that even if malicious domains use some brands

[6] http://www.phishtank.com, accessed on 15/03/12
[7] http://www.malwaredomains.com, accessed on 15/03/12
[8] http://www.malwaredomainlist.com, accessed on 15/03/12

included in the URLs of famous websites in order to mimic them, they still disclose differences. Two sources are chosen to compose this "legitimate" dataset.

- **Alexa**[9]: Alexa is a company that collects browsing behavior information in order to report statistics about web traffic and websites ranking. From Alexa's "top 1000000 sites" list, 40,000 domain names are randomly picked in the top 200,000 domains.
- **Passive DNS**: To diversify this dataset and in order to have the same amount of domain names in each dataset, we had it completed with 11,322 domain names extracted from DNS responses. DNS responses were passively gathered from DNS recursive servers of some Luxembourg ISPs. We ensure that these domain names are not present in the initial dataset from Alexa.

The normal dataset contains 51,322 entries. 38,712 names have their *main domain* divisible in at least two parts. Hence, we have two datasets: a *legitimate* one and a *malicious* one of equivalent size.

4 Experiments

4.1 Datasets analysis

In this section metrics and statistical parameters extracted from each dataset are compared to demonstrate that features described before are able to distinguish malicious from legitimate domains. A first proposition is to analyze the number of words that composes the *main domain* name $\#len_{2,n}$. *Main domains* that can be split in at least two parts are considered. The *malicious* dataset contains 39,980 such domain names and the *legitimate* dataset 38,712. Figure 3 shows the distribution of the ratio of *main domains* that are composed from 2 to 10 words ($distlen_{2,n} \mid n \in \{2; 10\}$) in the *legitimate* dataset and in the *malicious* dataset.

69% of legitimate *main domains* are composed of two words whereas only 50% of malicious are. For all upper values, the ratio for malicious domains is higher than for legitimate ones. This shows that malicious *main domains* tend to be composed of more words than legitimate *main domains*.

The following analysis studies the composition similarity between the domain names of the different datasets. Two probabilistic distributions are extracted from the domain names:

- the different labels of the TLDs: $\forall w \in W, P_1(w) = distword_{1,w}$
- the different words that compose the *main domains*: $\forall w \in W, P_2(w) = distword_{2,w}$

We used the Hellinger Distance to evaluate the similarity in each dataset and dissimilarity between datasets. The Hellinger Distance is a metric used to quantify the similarity (or dissimilarity) between two probabilistic distributions P

[9] http://www.alexa.com, accessed on 15/03/12

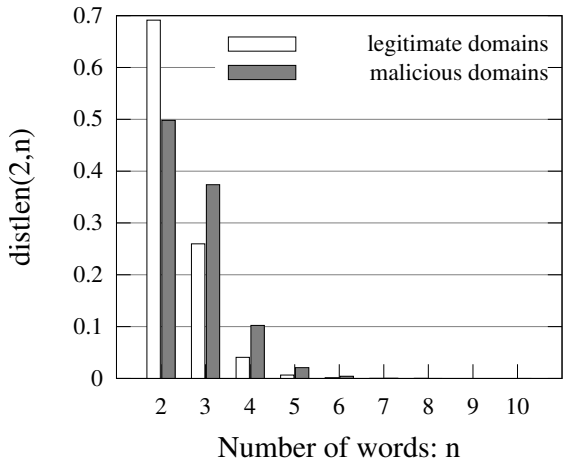


Fig. 3. $distlen_{2,n} \mid n \in \{2;10\}$ for malicious and legitimate dataset

and Q. In continuous space, the definition is:

$$H^2(P,Q) = \frac{1}{2}\int\left(\sqrt{\frac{dP}{d\lambda}} - \sqrt{\frac{dQ}{d\lambda}}\right)^2 d\lambda \tag{8}$$

The equivalent function in discrete space distribution is given by:

$$H^2(P,Q) = \frac{1}{2}\sum_{x\in P\cup Q}\left(\sqrt{P(x)} - \sqrt{Q(x)}\right)^2 \tag{9}$$

It's an instance of f-divergence as well as KL-divergence metric. Hellinger Distance is symmetric and bounded on $[0;1]$ where 1 is a total dissimilarity ($P \cap Q = \emptyset$) and where 0 means that P and Q have the same probabilistic distribution.

This metric is preferred rather than more usual metric such as Jaccard Index or KL-divergence. Jaccard Index only considers the presence or not of an element in two datasets but never considers the probability associated to an element. KL-divergence metric is an non-symmetric measure as well as unbounded function ($[0;+\infty]$). Finally, KL-divergence requires that Q includes at least the same elements of P: $\forall i P(i) > 0 \Rightarrow Q(i) > 0$. This constraint may not be satisfied with our datasets.

The *malicious* dataset and *legitimate* datasets are randomly split in five smaller subsets, respectively mal-x and leg-$x \mid x \in \{1;5\}$, of equivalent size ($\sim$ 10000 domains). Table 3 shows the Hellinger Distance for TLDs distribution between all the subset $P_1(w)$. Globally all the TLDs are quite the same in all subsets ($0 < H(P,Q) < 0.15$), a clear difference is although present in $H(P,Q)$ when P and Q are picked from the same dataset (leg/leg or mal/mal, $H(P,Q) \sim 0.015$) or from two different datasets (leg/mal, $H(P,Q) \sim 0.130$).

	leg-5	leg-4	leg-3	leg-2	leg-1	mal-5	mal-4	mal-3	mal-2
mal-1	0.133	0.136	0.133	0.129	0.134	0.014	0.012	0.013	0.014
mal-2	0.134	0.140	0.135	0.131	0.135	0.014	0.012	0.013	
mal-3	0.135	0.139	0.134	0.131	0.136	0.013	0.013		
mal-4	0.130	0.136	0.131	0.127	0.132	0.013			
mal-5	0.134	0.138	0.132	0.129	0.134				
leg-1	0.017	0.017	0.018	0.019					
leg-2	0.018	0.020	0.018						
leg-3	0.016	0.019							
leg-4	0.017								

Table 3. Hellinger Distance for TLDs (leg=legitimate, mal=malicious)

Table 4 considers the words-in-main-domain distribution $P_2(w)$. Here, the distributions are more scattered ($0.4 < H(P,Q) < 0.6$); however, difference is higher between subsets created from distinct datasets ($H(P,Q) \sim 0.56$) and subsets of the same dataset. Moreover, we can see that malicious *main domains* show more similarity between them ($H(P,Q) \sim 0.44$) than legitimate *main domains* between them ($H(P,Q) \sim 0.50$).

	leg-5	leg-4	leg-3	leg-2	leg-1	mal-5	mal-4	mal-3	mal-2
mal-1	0.564	0.571	0.561	0.566	0.565	0.446	0.439	0.443	0.438
mal-2	0.565	0.569	0.566	0.571	0.565	0.445	0.447	0.446	
mal-3	0.561	0.566	0.563	0.569	0.564	0.448	0.444		
mal-4	0.563	0.567	0.558	0.564	0.561	0.447			
mal-5	0.564	0.565	0.554	0.555	0.558				
leg-1	0.501	0.494	0.490	0.493					
leg-2	0.493	0.497	0.496						
leg-3	0.490	0.491							
leg-4	0.489								

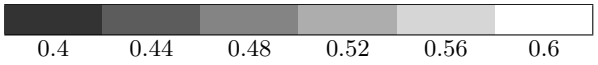


Table 4. Hellinger Distance for words (leg=legitimate, mal=malicious)

Table 5 provides the statistics of the Markov chains for each dataset for the *main domain* level. The number of initial states is given by $Card(V) \mid \forall w \in V, \#fisrtword_{l_2,w} > 0$, the number of states corresponds to $Card(W) \mid \forall w \in W, \#words_{l_2,w} > 0$ and the number of transitions before implementation of Laplace smoothing is $Card(U^2) \mid \forall (w_1, w_2) \in U^2, \#biwords_{l_2,w_1,w_2} > 0$. This

table strengthens the assertion that words present in malicious *main domains* are more related together than those present in legitimate *main domains*, because Hellinger Distance is lower between malicious subsets compared to legitimate subsets despite the higher number of words (states) in the Markov chain created from the malicious dataset.

These experiments show that our model built on top of blacklist will be able to generate proactively maliciously registered domains with a limited impact regarding legitimate ones.

Metrics	*Legitimate*	*Malicious*
# initial states	14079	14234
# states	23257	26987
# transitions	48609	56286

Table 5. Markov Chain statistics for *main domain*

4.2 Types of generated domains

The dataset chosen for the rest of the experiments is the whole *malicious dataset* introduced in section 3.1. This dataset is split in two subsets and depending on the experiment performed, the domains selection technique to compose the subsets and the number of domains in each subset vary. One of these subsets is called the *training set*, from which the features described in section 2.2 are extracted in order to build the word generation system depicted in Figure 2, section 2.1. Based on it, new domain names are generated and their maliciousness is confirmed only if they belong to the second subset called the *testing set*.

The term *probing campaign* is defined as the generation of one million of unique *two-level-domains* that are checked in term of existence and maliciousness. A domain name is considered as existing if it is actually reachable over the Internet, *i.e.* it is mapped to an IP address. For each generated domain, a DNS A request is performed and according to the DNS response status, the domain name is considered as existing (status = `NOERROR`) or non-existing (status = `NXDOMAIN`). For more information about DNS and its operation, the reader must refer to [17–19].

The first step of the experiments aims at analyzing the existence and the type of generated domains. Figure 4 is an histogram depicting the run of five probing campaigns using a generation model trained on 10% of the *malicious dataset*, each of the five complete rectangle represents the number of existing domains generated. We can see that over one million unique domains probed, between 80,000 and 110,000 so around 10% are potentially reachable over the Internet. These existing domains are divided in three categories represented distinct filling pattern in the histograms.

The white one represents the number of wildcarded domains. Domain wildcarding is a technique that consists in associating an IP address to all possible subdomains of a domain by registering a domain name such as `*.yahoo.com`. As a result all DNS queries sent for a domain containing the suffix `yahoo.com` will be

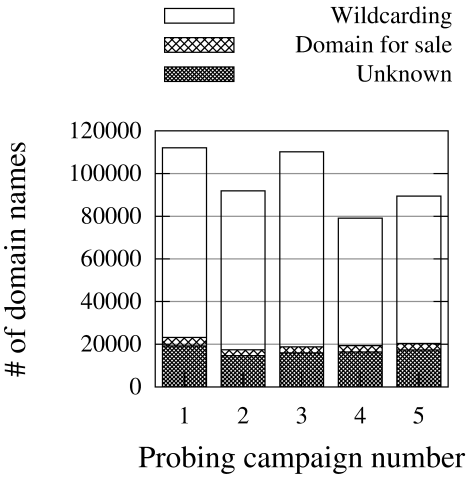


Fig. 4. Distribution of domains discovered regarding their types

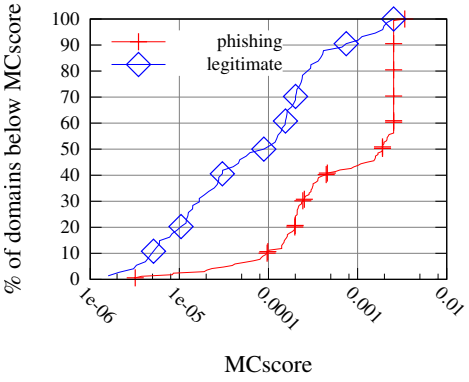


Fig. 5. Cumulative ratio of domains regarding their score

answered with a `NOERROR` status DNS response containing always the same IP address. This technique is useful to tolerate Internet users typing mistakes, or misspelling of subdomain without any consequence. For instance, DNS requests for domain such as `wwe.yahoo.com`, `snt.yahoo.com` or `anyotherlabel.yahoo.com` will return the same IP address. However some TLDs such as `.ws`, `.tk`, `.us.com`, etc. apply also wildcarding. As a result these TLDs have been identified in order to discard all generated *two-level-domains* that contain one of them. We can see that these domains represent between 75% and 85% (from 60,000 to 90,000) of the domains discovered.

The remaining part is composed of two other categories. First some domains are registered but lead to websites of domain name resellers such as GoDaddy or Future Media Architect. A lot of meaningful domain names belong to this category, around 4,000 per campaign. Some examples of such domains are `freecolours.com` or `westeurope.com`. Regarding a probing campaign, the IP addresses obtained through DNS responses are stored and sorted by their number of occurrences. The IP addresses having more than fifty occurrences are manually checked to see if they are either related to real hosting or domain selling. Around fifty IP addresses and ranges have been identified as leading to domain name resellers. These domains are also discarded in our study, as they are not likely to be malicious domains. Finally the black part represents the domains that are unknown and have to be checked to confirm if they are related to phishing or not. As highlighted in Figure 4, the remaining potential malicious domains represent only between 15,000 and 20,000 domains out from one million of generated ones. This reduction is automated and allows discarding a lot of domain names, which will reduce the overhead of the checking process.

For domains of the unknown part that are known to be really legitimate or phishing, a score, $MCscore$, is calculated. This latter measures the similitude with the underlying training dataset, which have been used for building the model. Assuming a *two-level-domain* $w_1 w_2 \dots w_n.tld$ where w_i is the i^{th} word composing the name, w_i may have been generated using DISCO from an original word observed w'_i. $MCscore$ is computed as follows:

$$MCscore = distfirstword_{2,w'_1} \times sim_{w_1,w'_1} \times \prod_{i=1}^{n} distbiwords_{2,w_i,w'_{i+1}} \times sim_{w_{i+1},w'_{i+1}} \tag{10}$$

The first word probability is multiplied by each probability of crossed transition in the Markov chain. If some parts are found using DISCO, the similarity score given in equation (7) is used ($sim_{w_i,w_i} = 1$ else).

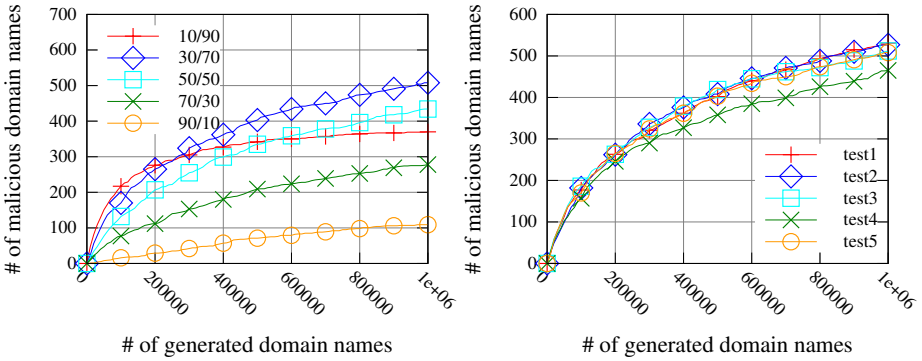
Figure 5 represents the cumulative sums of the ratio of domains (in %) that have a score lower than x for each kind of label. These curves show that globally phishing related domain names have a higher $MCscore$ than legitimate ones, around ten times higher. Even if a high number of domains are labeled as unknown and some of them are legitimate, it is easy to discard a lot of them in order to keep a set containing a main part of malicious domains. If we consider as malicious only the generated domains having a $MCscore$ higher than 0.001, then 93% of the legitimate domains will be discarded while 57% of the malicious domains will be kept. This technique can be used to avoid the use of a domain checking technique, as introduced in 2.1, or to reduce its workload.

4.3 Efficiency and steadiness of generation

This section assesses the variation of the efficiency of the malicious domain discovery regarding the ratio of domains in the *testing set* and in the *training set*. Five probing campaigns are performed with a ratio that varies from 10% training/90% testing (10/90) to 90% training/10% testing (90/10), the subsets are randomly made up. Figure 6(a) shows the number of malicious domains generated regarding the total number of probed names.

On one hand the best result is given by 30% training/70% testing with a total number of 508 phishing domains discovered. When the *testing set* size decreases, there are less domain names candidates that can be found, which implies that more domains are discovered with 30/70 than 90/10. On the other hand, the curve representing 10% training/90% testing grows faster, and after only 100,000 probes more than the half (217 domains) of the total number of phishing domains generated are found. Following the curve's trend, if more probes are performed, a reasonable assumption is that more malicious domain names can be discovered.

Figure 6(b) depicts the steadiness of the discovery results. Five probing campaigns are performed for the ratio that yields the best result: 30% training/70% testing. The training and testing sets are randomly made up and are different for each campaign. Observations are similar for every tests which lead to discover around 500 phishing domains. Moreover, half of the discovered phishing domains



(a) Variation of testing/training set ratio (b) 5 probing campaigns with 30% training & 70% testing

Fig. 6. Number of domains discovered regarding the number of probes

are generated during the first 200,000 generations, highlighting the ability of our system to generate the most likely malicious domains in priority before being discarded for next probes.

4.4 Predictability

This experiment evaluates the time between the date when a malicious domain name can be generated using the generator and the date it is actually blacklisted. The training set is composed of the 10% oldest blacklisted domains and the remaining 90% belong to the *testing set*. The *testing set* represent 34 months of blacklisted domains and the *training set* 4 months. Figure 7 depicts the number of malicious domain names generated regarding the number of months they are actually blacklisted after their generation (m+x). A large quantity of generated malicious domains appears in first four months after their generation, 14 in the two following months and 23 more in the next two months. This shows that domain name composition follows fashion schemes because more malicious domains that are discovered appear just after the ones that are used to train the model. However, it is worth noting that such domains continue to be discovered in the present showing that even old datasets can be useful to generate relevant malicious domains

4.5 Strategy evaluation

We have described in section 2 the two core building blocks for generating domain names: the Markov chain model and the semantic exploration module. The impact of each module is assessed with respect to four strategies:

- MC: the Markov chain model alone.
- MC + 5 DISCO: the Markov chain model and for each selected state of the Markov chain the five most related words, regarding DISCO, are tested.

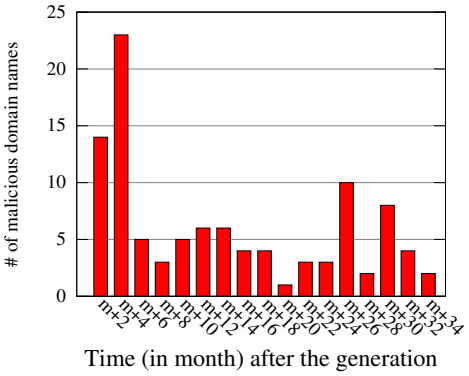


Fig. 7. Distribution of domain discovered regarding the time they are blacklisted

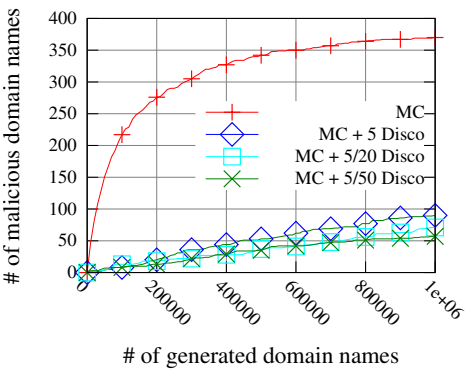


Fig. 8. Number of domain names discovered regarding the method

- MC + 5/20 DISCO: the Markov chain model and for each selected state of the Markov chain five words randomly picked from the twenty most related words, regarding Disco, are tested.
- MC + 5/50 DISCO: the Markov chain model and for each selected state of the Markov chain five words randomly picked from the fifty most related words, regarding Disco, are tested.

The objective of this assessment is to identify the best tradeoff between the success rates in discovering rogue domain names with respect to the computational effort.

Figure 8 shows the number of actual generated malicious domain names with respect to the number of probes performed over a probing campaign. The same training set is used to build the generation model for the four different probing campaigns. It clearly comes out that the Markov chain model alone yields the best results in term of number of malicious domain names discovered with a total of 370. However even if DISCO strategies are able to generate only between 57 and 90 malicious domain names over these campaigns depending on the technique, between 79% and 85% of these generated domains are unique, *i.e.* none of the other strategies are able to find them. If a global probing is targeted all the part of the generation module must be used in order to discover the maximum of phishing related domain names. However, the Markov chain model is sufficient to find out domains over a short period of time.

5 Related Work

Because of their essential role, anti-phishing and identification of malicious domain are increasingly popular and addressed in several previous works. Two major approaches exist: methods based on blacklists and heuristics-based methods.

Heuristic-based approaches rely on classification algorithms to identify whether a domain is malicious or not, based on features extracted from dif-

ferent sources. The leveraging of machine learning techniques to classify on the fly, domains as malicious or benign is widely used, with either batch methods such as SVM, Naive Bayes, Logistic Regression(like in [2–4]), or on-line classification algorithms such as Confidence Weighted (CW), Adaptive Regularization of Weight (AROW), Passive-Aggressive (PA) (see [5, 9, 15]).

Their differences are mainly related to the feature set. In [2–4], the building of classification models relies on passively gathered DNS queries to figure out predominantly malware domains involved in botnet communications. However for phishing detection purposes, it can be either host-based features (WHOIS info, IP prefix, AS number) in [15] or web page content-based features in [25].

The majority of features are however extracted directly from URLs. These can be for instance the type of protocol, hostname, TLD, domain length, length of URL, etc. In [5, 9, 11], the authors particularly focus on the tokens that compose a complete URL, which includes the domain as well as the path and the query. In these studies, the classification is based on the relative position of these tokens (domain or path level for instance) or the combination of these tokens (*token1.token2/token3/token4?token5=token6*). The conclusion is that tokens that occur in phishing URLs belong to a limited dictionary and tend to get reused in other URLs. Moreover, Garera *et al.*, in [8], are the first to use the occurrences of manually defined words (secure, banking, login, etc.) in URL as features. In [13], Le *et al.* use both batch and on-line classification techniques to show that lexical features extracted from URLs are sufficient to detect phishing domains. Even also based on lexical features, our work is different as we consider meaningful words that compose the same label of a domain name. Moreover, our work consists in a predictive and active discovery rather than classification of domain names observed on network traffic.

There have been other works taken advantage of URL based lexical analysis for different purposes. In [27], statistical measures are applied to alphanumeric characters distribution and bigrams distribution in URLs in order to detect algorithmically generated fluxing domains. The same technique is used in [6] to detect DNS tunnels and, in [26], Xie *et al.* generate signature for spamming URLs using Regular Expressions. URLs related to the same spam campaign are grouped for creating a signature based on regular expression.

This work is close to our approach but only lead to disclose domain names related to a specific spam campaign from which some domain names have been already observed. Our approach is more general and allows discovering new phishing domains that have no apparent relations with previous ones.

Blacklisting approaches consist in the partially manual construction of a list of malicious URLs that will be used by web browser or e-mail client in order to prevent the users to access them. Due to their short lifetime, the early identification of phishing websites is paramount, as a result several methods have been proposed to avoid reactive blacklisting and develop more proactive methods. In [10], Hao *et al.* analyze early DNS behavior of newly registered domains. It is demonstrated that they are characterized by DNS infrastructure pattern and DNS lookup patterns monitored as soon as they are registered, such

as either a wide scattering of resource records across the IP address space in only few regions, or resource records that are often hosted in tainted autonomous systems. A very close approach is used in [7] to build proactive domain blacklists. Assuming a known malicious domain, zone information is mined to check if other domains are registered at the same time, on the same name server and on the same registrar. However, this approach cannot be widely applied because domain zone information is not always available and needs a prior knowledge.

Predictive blacklisting is also addressed in [28] where it's assumed that a host should be able to predict future attackers from the mining of its logs. Hence, a host can create its own customized blacklist well fitted to its own threats. The composition of this blacklist relies also on other machines' logs that are considered as similar. This similarity score is calculated using the number of common attackers between two victims and stored into a graph explored using a PageRank algorithm to estimate whether a domain is likely to conduct an attack against a particular victim or not. The similarity calculation is refined in[23].

Another proactive blacklisting approach to detect phishing domains is addressed through Phishnet in [20]. This work is the closest to ours, the idea is to discover new phishing URLs based on blacklist of existing phishing URLs. As in [26], URLs are clustered based on their shared common domain names, IP addresses or directory structures, then a regular expression is extracted from each cluster. The variable part of regular expression is exploited to generate new URLs instead of only compare existing URLs to extracted patterns like in [26]. Though this method is more proactive than the previous ones, it is only able to disclose URLs related to already blacklisted URLs that are likely to belong to the same phishing attack. Whereas these URLs are part of a very small pool of domains, our approach is capable to extend the knowledge about distinct phishing attacks. Therefore these approaches are quite complementary.

Finally we have already treated and proved the efficacy of algorithmic domain generation based on Markov chain in [24]. We apply this technique on bigrams in order to perform a discovery of all the subdomains (`www`, `mail`, `ftp`) of a given domain (`example.com`). We extend this approach in [16], based on an existing list of subdomains, we leverage semantic tools and incremental techniques to discover more subdomains.

6 Conclusion and future work

This paper introduces an efficient monitoring scheme for detecting phishing sites. The main idea consists in generating a list of potential domain names that might be used in the future by an attacker. This list can be checked on a daily basis to detect the apparition of a new phishing site. The list is generated using language models applied to known ground truth data. We have proposed a novel technique to generate domain names following a given pattern that can be learned from existing domain names. This domain generation leverages a Markov chain model and relevant lexical features extracted from a semantic splitter. Domain specific knowledge is added from semantic tools. The efficiency of this generation tool is

tested on the real world datasets of phishing domain names. We proved that our method is able to generate hundreds of new domain names that are actually related to phishing and appear to be in use in the period following their generation. To the best of our knowledge, our approach is the only one to propose proactive generation and discovery of malicious domains, which is complementary to state of the art approaches that addressed proactive blacklisting of URLs.

In future works, the remaining part of the architecture, the domain checker, will be implemented shortly. Furthermore, the feedback from this checker will be used to adapt the Markov chain transition probability through reinforcement learning in order to strengthen the generation model. The code is available on request.

Acknowledgements. This work is partly funded by OUTSMART, a European FP7 project under the Future Internet Private Public Partnership programme. It is also supported by MOVE, a CORE project funded by FNR in Luxembourg. The authors would like to thank P. Bedaride for discussions and advice on natural language processing tools.

References

1. Anti-Phishing Working Group and others: Phishing Activity Trends Report - 1H2011. Anti-Phishing Working Group (2011)
2. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for dns. In: Proceedings of the 19th USENIX conference on Security. pp. 18–18. USENIX Security'10, USENIX Association, Berkeley, CA, USA (2010)
3. Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, II, N., Dagon, D.: Detecting malware domains at the upper dns hierarchy. In: Proceedings of the 20th USENIX conference on Security. pp. 27–27. SEC'11, USENIX Association, Berkeley, CA, USA (2011)
4. Bilge, L., Kirda, E., Kruegel, C., Balduzz, M.: EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In: NDSS 2011. Internet Society (Feb 2011)
5. Blum, A., Wardman, B., Solorio, T., Warner, G.: Lexical feature based phishing url detection using online learning. In: Proceedings of the 3rd ACM workshop on Artificial Intelligence and Security. pp. 54–60. ACM (2010)
6. Born, K., Gustafson, D.: Detecting dns tunnels using character frequency analysis. Arxiv preprint arXiv:1004.4358 (2010)
7. Felegyhazi, M., Kreibich, C., Paxson, V.: On the potential of proactive domain blacklisting. In: Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more. pp. 6–6. USENIX Association (2010)
8. Garera, S., Provos, N., Chew, M., Rubin, A.: A framework for detection and measurement of phishing attacks. In: Proceedings of the 2007 ACM workshop on Recurring malcode. pp. 1–8. ACM (2007)
9. Gyawali, B., Solorio, T., Wardman, B., Warner, G., et al.: Evaluating a semisupervised approach to phishing url identification in a realistic scenario. In: Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. pp. 176–183. ACM (2011)

10. Hao, S., Feamster, N., Pandrangi, R.: Monitoring the initial DNS behavior of malicious domains. In: Proceedings of the ACM SIGCOMM Internet Measurement Conference. pp. 269–278. IMC '11, ACM, New York, NY, USA (Nov 2011)
11. Khonji, M., Iraqi, Y., Jones, A.: Lexical url analysis for discriminating phishing and legitimate websites. In: Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. pp. 109–115. ACM (2011)
12. Kolb, P.: DISCO: A Multilingual Database of Distributionally Similar Words. In: Storrer, A., Geyken, A., Siebert, A., Würzner, K.M. (eds.) KONVENS 2008 – Ergänzungsband: Textressourcen und lexikalisches Wissen. pp. 37–44 (2008)
13. Le, A., Markopoulou, A., Faloutsos, M.: Phishdef: Url names say it all. In: INFOCOM, 2011 Proceedings IEEE. pp. 191–195. IEEE (2011)
14. Ludl, C., Mcallister, S., Kirda, E., Kruegel, C.: On the effectiveness of techniques to detect phishing sites. In: Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 20–39. DIMVA '07 (2007)
15. Ma, J., Saul, L., Savage, S., Voelker, G.: Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 681–688. ACM (2009)
16. Marchal, S., François, J., Wagner, C., Engel, T.: Semantic exploration of DNS. In: IFIP/TC6 Networking 2012. Prague - Czech Republic (may 2012)
17. Mockapetris, P.: Rfc 1035: Domain names - implementation and specification
18. Mockapetris, P.: Rfc 1034: Domain names - concepts and facilities (1987)
19. Mockapetris, P., Dunlap, K.: Development of the domain name system. In: Proceedings of the 1988 ACM SIGCOMM. pp. 123–133. IEEE Computer Society, Stanford, CA, USA (1988)
20. Prakash, P., Kumar, M., Kompella, R., Gupta, M.: Phishnet: predictive blacklisting to detect phishing attacks. In: INFOCOM, 2010 Proceedings IEEE. pp. 1–5. IEEE (2010)
21. Rasmussen, R., Aaron, G.: Global phishing survey: trends and domain name use in 1h2011. Anti-Phishing Working Group (2011)
22. Segaran, T., Hammerbacher, J.: Beautiful Data: The Stories Behind Elegant Data Solutions, chap. 14. O'Reilly Media (2009)
23. Soldo, F., Le, A., Markopoulou, A.: Predictive blacklisting as an implicit recommendation system. In: INFOCOM, 2010 Proceedings IEEE. pp. 1–9. IEEE (2010)
24. Wagner, C., François, J., State, R., Engel, T., Dulaunoy, A., Wagener, G.: SDBF: Smart DNS Brute-Forcer. In: Proceedings of IEEE/IFIP Network Operations and Management Symposium - NOMS. IEEE Computer Society (2012)
25. Xiang, G., Hong, J.: A hybrid phish detection approach by identity discovery and keywords retrieval. In: Proceedings of the 18th international conference on World wide web. pp. 571–580. ACM (2009)
26. Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., Osipkov, I.: Spamming botnets: signatures and characteristics. In: ACM SIGCOMM Computer Communication Review. vol. 38, pp. 171–182. ACM (2008)
27. Yadav, S., Reddy, A.K.K., Reddy, AL, Ranjan, S.: Detecting algorithmically generated malicious domain names. In: Proceedings of the 10th annual conference on Internet measurement. pp. 48–61. ACM (2010)
28. Zhang, J., Porras, P., Ullrich, J.: Highly predictive blacklisting. In: Proceedings of the 17th conference on Security symposium. pp. 107–122. USENIX Association (2008)