Semantic based DNS Forensics

Samuel Marchal, Jérôme François, Radu State, Thomas Engel

To cite this version:

Samuel Marchal, Jérôme François, Radu State, Thomas Engel. Semantic based DNS Forensics. International Workshop on Information Forensics and Security - WIFS, Dec 2012, Tenerife, Spain. pp.91-96, 10.1109/WIFS.2012.6412631 . hal-00784965

Semantic based DNS Forensics

Samuel Marchal[1], Jérôme François[1], Radu State[1], Thomas Engel[1]

[1] *SnT, University of Luxembourg*
6 rue R. Coudenhove Kalergi, L-1359 Luxembourg
firstname.lastname@uni.lu

Abstract—**In network level forensics, Domain Name Service (DNS) is a rich source of information. This paper describes a new approach to mine DNS data for forensic purposes. We propose a new technique that leverages semantic and natural language processing tools in order to analyze large volumes of DNS data. The main research novelty consists in detecting malicious and dangerous domain names by semantic similarity with already known names. This process can provide valuable information for reconstructing network and user activities. We show the efficiency of the method on experimental real world datasets gathered from a national passive DNS system.**

I. Introduction

Due to the increasing number of attacks occurring every day in Internet and their increasing variety and complexity, being protected to all of them is quite unfeasible. Reports like [1] show such an evolution as well as recent facts highlighting the inability, even for big companies or governments, to counter some attacks [2]. Therefore, cyber-crime is now a main concern where the attacker motivation has shifted from technical challenges to financial, political or ideological reasons. In such a context, detecting and analyzing attacks is essential for system recovery and for preventing future intrusions. Identifying the responsible people and their motivation is also essential to take legal actions as well. This leads to have dedicated digital forensics techniques for gathering knowledge about intrusions to figure out relevant evidences. As digital infrastructures are ever bigger and more complex, a primary issue is the volume of data to handle [3] which is all the more important with standards tools, e.g. [4], which require a lot of manual support. Fully or semi automated analysis is so an important research field as for example [3]. From a network point of view, full packet analysis is proposed in [5]. To achieve scalability, using compressed source of information like network flow data was introduced in [6].

This paper focuses on network forensics and in particular passive DNS analysis [7]. DNS analysis is powerful for network forensics as DNS can capture evidences of attacks as popular threats usually rely on it for being efficient. Because the main functionality provides a translation of human readable names into machine addresses, it can also be used by attackers as for example for botnets which are known, as the vector of many other attacks, to be a major threat [1]. Fast-flux [8] consists into naming a phishing website or the C&C (Command and Control) with a unique DNS name which points alternatively and rapidely to distinct IP addresses.

Hence, DNS analysis is helpful in network forensics. Moreover, the amount of information is quite limited compared with traffic data as DNS traffic is only a subset. As the passive capture does not store information about request originator, it is privacy friendly which is also an issue in forensics [6], [9].

In this paper, we propose a tool to automatically analyze passive DNS data for tracking malicious related domains which should be considered as a good starting point for a deep analysis of network and user activity whereas considering entirely them is not scalable. In particular, our analysis relies on the semantic of DNS names because malicious related names are mainly constructed from a specific semantic field to lure the user by using attractive names including and combining brand names or specific keywords like *secure* or *protection* [10]: for example, `protectionmicrosoftxpscanner.com`, `securepaypal.com` or `domainsecurenethp.com`.

Starting from the fact that users complain about payment done over paypal that they never did, a first step may reveal that credentials have been stolen (connection from unusual locations). Then, a semantic based DNS analysis would reveal potential harmful websites like *www.securepaypal.com* where the users connected to. Hence, this phishing website can be identified as the original source of the problem.

Therefore, new metrics are defined to catch the semantic properties of DNS names and evaluated to measure their ability to distinguish malicious and legitimate names.

The paper is structured as follows: the overall architecture, including an overview in DNS, is described in section II. Semantic based metrics are formally defined in section III before being assessed in section IV. Section V presents related work before the conclusion in section VI.

II. Architecture overview

For sake of clarity, DNS is introduced in this section but further details may be found in [11] (caching, reverse DNS, etc). The primary objective of DNS is to translate a human understandable address into an IP address. DNS names are organized in a hierarchical manner. For example, `www.uni.lu` is a domain name[1] which has 3 levels with `lu` the first level also called the top level domain (TLD), `uni` the second level, `www` the third one. Of course there are other subdomains. For example, all `*.lu` domains are subdomains of `lu` TLD.

[1] A domain name can be either a final host or a domain including several final hosts

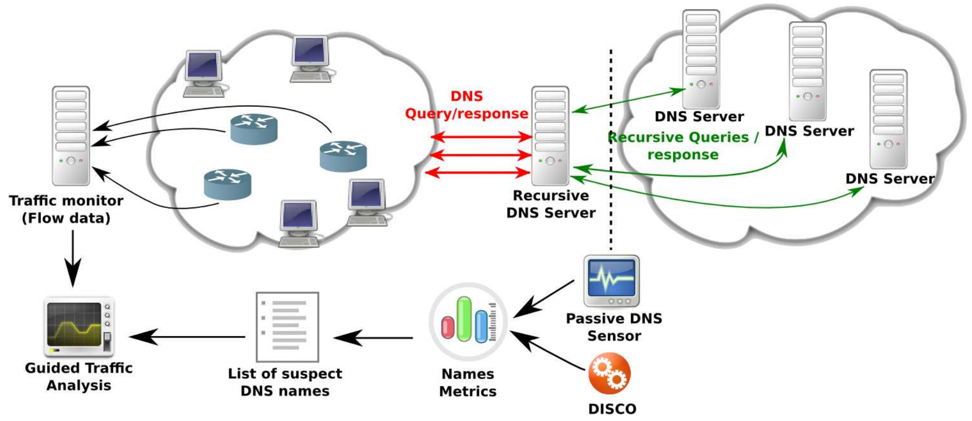

Fig. 1: Architecture overview

When a DNS client wants to access `www.uni.lu`, he sends a request to a recursive DNS server, commonly provided by his operator. The latter then resolves the query by contacting iteratively the authoritative servers of the different levels starting first with a root server which indicates the authoritative server for the `lu` domain, which can then resolve the query `uni.lu` and so on. Except for the first query initiated by the client, all others are requested by the recursive DNS server itself before returning the final answer, the IP address of the requested name, to the client. In fact, DNS messages include Ressource Records (RRs) which reflect the types of queries and answers. For example, `A` is the type representing an IPv4 address.

As highlighted in Figure 1, a passive DNS probe is deployed to gather all traffic between the recursive server and the others to keep confidential the original IP address of the DNS client. Then, metrics are computed on the collected names based on a semantic tool, DISCO [12], before being used to figure out suspect domain names that should be tracked more attentively for forensics. Even if out of the scope of the paper, figure 1 shows one standard use case where all traffic is collected in parallel, as for instance with a flow based solution [13], and then analyzed using the suspect host list to guide the investigation.

III. Semantic metrics

Unlike standard texts, domain names are composed of few words and so deriving a global semantic is hard. To make it easier, grouping multiple names to increase the number of words before determining the overall semantic is possible. It is also compatible with forensics since infected or attacked machines will probably communicate with multiple malicious hosts or also because a malicious domain can support several malicious subdomains or an IP subnet can belong to an AS known to host malware [14].

We assume that this grouping is already performed using a state-of-the-art technique like [15]. Hence, the goal is to analyse if domain names from two different set types, legitimate or malicious, are composed of words that share semantic similarities or disclose semantic differences. To do so, we need to perform first an extraction of the words that compose a domain, second, find a metric characterizing semantic relatedness between two words and, third, develop metrics to give a score of similarity between two sets composed of several domain names.

A. Word extraction

The first requirement is to split the domain names in order to extract all words that composed them as highlighted in Figure 2. These words are supposed to belong to a particular semantic field in order to give a similarity score with other words. Domain names are first split by level according to the separating dots, ".", and the TLD is excluded based on the list from *iana.org*[2], as using the related semantic is too naive (names registered in Russia for example). Since hyphens ("-") are allowed in DNS names, a second split is done accordingly. Furthermore, the digits are removed and considered also as separating characters. This leads to have remaining parts composed of letters only ([a-z] as DNS does not differentiate the character case), which can still be composed of several words like *computeraskmore* or *cloudantimalware*.

The segmentation of alphabetical parts is inspired from [16] that consists in successively dividing a label in 2 parts until finding the combination that gives the maximum probability. Hence, assuming a label l, for each position $i \in [1; len(l)]$, l is divided in 2 parts and the probability $P(l,i)$ given in equation (1) is calculated:

$$P(l,i) = P_{word}(pre(l,i)) \times P_{word}(post(l,i)) \qquad (1)$$

where $pre(l,i)$ returns the substring of l composed of the first i characters and $post(l,i)$ 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. This process is applied to all newly split parts $pre(l,i)$ and $post(l,i)$ as long as $\exists i \in [1; len(l)-1], P(l,i) \geq P_{word}(l)$.

Finally, we propose to count the occurrences of words because a word appearing more frequently than others should have a bigger impact in the global semantic of set of words.

[2] http://data.iana.org/TLD/tlds-alpha-by-domain.txt accessed on 10/06/12

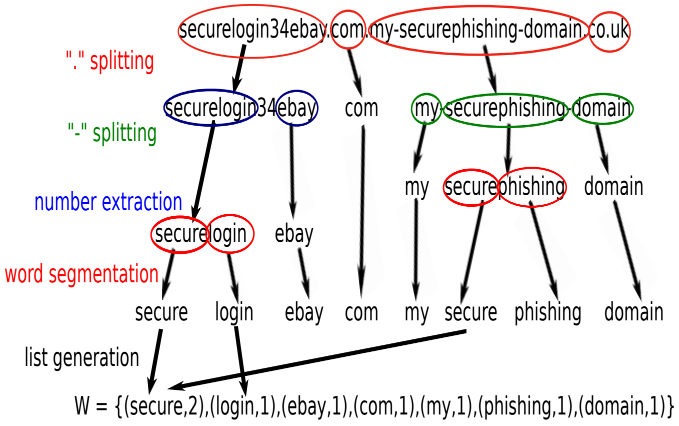

Fig. 2: Example of words extraction for *securelogin34ebay.com.my-securephishing-domain.co.uk*

As shown in Figure 2, couples $[word, occurrence]$ are stored in a vector W for further analysis

B. Semantic score

Once all words that are likely to be meaningful are extracted from domain names, evaluating the semantic similarity between sets of domains and so sets of words is required. For this purpose, DISCO [12] is leveraged, a tool based on efficient and accurate techniques to automatically give a score of the 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[3] and the metric $\|w, r, w'\|$ is calculated as the number of times that the word w' occurs r words after the word w in the window, therefore $r \in \{-3; 3\} \setminus \{0\}$. Table I 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:

$$I(w,r,w') = log\frac{(\|w,r,w'\| - 0.95) \times \|*,r,*\|}{\|w,r,*\| \times \|*,r,w'\|} \qquad (2)$$

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

$$sim(w_1,w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} I(w_1,r,w) + I(w_2,r,w)}{\sum_{(r,w)\in T(w_1)} I(w_1,r,w) + \sum_{(r,w)\in T(w_2)} I(w_2,r,w)} \qquad (3)$$

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 can either give a similarity score with another word w_2 or return $Disco(w_1, n)$, the n most related words, w_i, ordered by their decreasing respective similarity score $sim(w_1, w_i)$.

C. Similarity metrics

Given a set of p domain names $D = \{d_1, ..., d_p\}$, words are extracted from each domain name following the technique given in section III-A. They form a set of n words with their corresponding number of occurrences in all the set of domains $W_D = \{(w_1, o_{w_1}), ..., (w_n, o_{w_n})\}$. $distword_{w_i,W_D}$ is the frequency of a word w_i in W_D:

position	-3	-2	-1	0	$+1$	$+2$	$+3$
sample1	a	client	uses	services	of	the	platform
sample2	the	platform	provides	services	to	the	client

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

TABLE I: Example of co-occurrence counting (2 windows centered on *services*)

$$distword_{w_i,W_D} = \frac{o_{w_i}}{\sum_{j\in\{1,n\}} o_{w_j}} \qquad (4)$$

Following this formula, three metrics are defined to quantify the semantic similarities between two sets of domain names, A and B.

$Sim_1(A, B)$ considers all the words $w_A \in W_A$ and $w_B \in W_B$ and compares them pair-wise using DISCO. By summing over all similarities, a global score of similarity between two sets is obtained:

$$Sim_1(A,B) = \sum_{w_A\in W_A}\sum_{w_B\in W_B} sim(w_A,w_B) \qquad (5)$$

The second metric is close to the first one except that it takes into account the number of occurrences of the words into each dataset A and B. Logically, this is done in order that the similarity obtained from words appearing more frequently would have a bigger impact on the metric than the one computed from words that appear once or few times.

Therefore, when calculating $Sim_2(A, B)$, $sim(w_A, w_B)$ is multiplied by the frequency of occurrences of w_A and w_B in their respective dataset to give a weight to each similarity:

$$Sim_2(A,B) = \sum_{w_A\in W_A}\sum_{w_B\in W_B} sim(w_A,w_B) \times distword_{w_A,W_A} \times distword_{w_B,W_B} \qquad (6)$$

Since preliminary experiments show that calculating $sim(w_A, w_B)$ is time consuming (around 0.5 sec with a standard desktop PC: Intel Core 2 Quad 2.83Ghz, 4GB RAM), pair-wise comparisons in equations (5) and (6) are not efficient. Getting the top ten most related words of w using $Disco(w, n)$ requires approximatively the same amount of time than $sim(w_A, w_B)$. Thus, we propose to keep the semantic analysis by extracting the most related words of each word of the set w_A before searching them into the set w_B:

$$Sim'_3(A,B) = \sum_{w\in W_A}\sum_{w'\in Disco(w,n)} sim(w,w') \times distword_{w',W_B} \qquad (7)$$

As highlighted, the score is still weighted by the frequency of occurrences. This equation avoids pairwise comparisons

[3] http://www.wikipedia.org, accessed on 10/06/12

using Disco. Moreover, $Sim'_3(A, B)$ is not a symmetric and so we have the equivalent symmetric with:

$$Sim_3(A, B) = Sim'_1(A, B) + Sim'_1(B, A) \qquad (8)$$

Hence, there are three metrics Sim_1, Sim_2 and Sim_3 defined in equations (5), (6) and (8) to give a score between two sets of domains in terms of semantic relatedness.

IV. Evaluation

A. Datasets

To test our method two sets of domains are formed: a legitimate one, containing non-malicious domain names and a malicious one, containing domain names from which maliciousness has been proved.

Malicious Dataset: Three freely downloadable blacklists are used to construct the dataset containing malicious domain names. These have been selected because each of them proposes an historic of blacklisted domains including different types of related malicious activities like spamming, phishing, worm spreading, fake anti-virus, etc. Since they contain several entries, a large dataset of malicious domain names is obtained. Thus, it clearly strengthens the validation of our method. The details are as follows:

- **PhishTank**[4]: PhishTank is a community website where contributors can add potential phishing URLs which will be then tested to update a global blacklist. 5,521 phishing URLs have been gathered for our experiments.
- **DNS-BH**[5]: DNS-Black-Hole maintains an up-to-date list of domains known to support malware and spyware diffusion. A list of 17,035 malicious domains is available.
- **MDL**[6]: like Phishtank, Malware Domain List is based on a community approach and maintains a blacklist constructed from proposed inputs from contributors. This list contains 82,480 URL entries.

After removing duplicate entries, the final dataset contains 66,633 different domain names.

Legitimate Dataset: The objective is to faithfully represent normal domain names in a realistic manner relying on two sources:

- **Alexa**[7]: Alexa provides a ranking of websites based on browsing statistics. 50,000 domains in the top 200,000 have been extracted for our experiments.
- **Passive DNS** from a Luxembourg ISP (Internet Service Provider): this dataset was obtained using a passive DNS infrastructure similar to the one described in section II in collaboration with a Luxembourg ISP. It is composed of 16,633 DNS names after having deleted duplicate entries from Alexa and known malicious domains (from the previously described blacklists)

Finally, 66,633 entries are contained in the legitimate datasets. Hence, we have two datasets, *legitimate* and *malicious* of equivalent size.

	leg-5	leg-4	leg-3	leg-2	leg-1	mal-5	mal-4	mal-3	mal-2
mal-1	19.3	19.3	20.1	18.7	20.1	29.7	30.0	30.3	31.0
mal-2	19.4	19.3	20.2	18.8	20.2	29.4	29.6	30.0	
mal-3	19.2	19.2	19.9	18.5	19.9	28.6	28.9		
mal-4	18.5	18.4	19.2	17.9	19.1	28.4			
mal-5	18.3	18.3	19.0	17.8	19.0				
leg-1	25.7	25.6	26.1	25.1					
leg-2	24.5	24.4	25.0						
leg-3	25.5	25.5							
leg-4	24.8								

15
19
23
27
31
35

TABLE II: Values of Sim_1 between malicious and legitimate subsets

	leg-5	leg-4	leg-3	leg-2	leg-1	mal-5	mal-4	mal-3	mal-2
mal-1	1.034	1.069	1.033	0.989	1.010	1.366	1.332	1.385	1.332
mal-2	1.033	1.053	1.035	0.994	1.013	1.369	1.298	1.365	
mal-3	1.085	1.137	1.089	1.046	1.058	1.388	1.347		
mal-4	1.013	1.041	1.002	0.964	0.964	1.370			
mal-5	1.010	1.051	1.002	0.972	0.972				
leg-1	1.475	1.481	1.489	1.455					
leg-2	1.455	1.452	1.507						
leg-3	1.508	1.525							
leg-4	1.472								

0.9
1.04
1.18
1.32
1.46
1.6

TABLE III: Values of $Sim_2 \times 10^3$ between malicious and legitimate subsets

B. Results

To assess the relevancy of semantic approach in identification of malicious domain names, this section shows that words composing the malicious domain names belong to different semantic fields than those from legitimate domain names even if malicious domains use pattern or brands to mimic composition of famous URLs.

To have an extensive evaluation, initial malicious and legitimate datasets defined in previous section are split in five subsets of equivalent size (13,326 domains). These ten subsets called mal-i, $i \in \{1; 5\}$ for malicious domains, and leg-i, $i \in \{1; 5\}$ for legitimate domains, are then compared one to the other by computing similarity scores. In fact, the following comparisons are done:

- Sets composed of malicious domains vs. sets composed of malicious domains
- Sets composed of malicious domains vs. sets composed of legitimate domains
- Sets composed of legitimate domains vs. sets composed of legitimate domains

In order to keep meaningful words for this study, only words extracted from subsets that are composed of at least 4 characters are considered. It avoids considering generic words such as *the*, *of*, *www*, etc.

Table II shows the score of Sim_1 computed between

[4] http://www.phishtank.com, accessed on 15/06/12

[5] http://www.malwaredomains.com, accessed on 15/06/12

[6] http://www.malwaredomainlist.com, accessed on 15/06/12

[7] http://www.alexa.com, accessed on 15/06/12

	leg-5	leg-4	leg-3	leg-2	leg-1	mal-5	mal-4	mal-3	mal-2
mal-1	0.776	0.795	0.793	0.789	0.785	0.955	0.962	0.965	0.975
mal-2	0.782	0.800	0.798	0.797	0.797	0.965	0.968	0.973	
mal-3	0.772	0.796	0.793	0.788	0.784	0.951	0.962		
mal-4	0.783	0.804	0.804	0.800	0.796	0.953			
mal-5	0.769	0.785	0.784	0.782	0.772				
leg-1	0.946	0.948	0.952	0.938					
leg-2	0.915	0.924	0.922						
leg-3	0.936	0.934							
leg-4	0.935								

0.7 0.76 0.82 0.88 0.94 1.00

TABLE IV: Values of Sim_3 between malicious and legitimate subsets

all the malicious and legitimate subsets. For a purpose of performance only the 100 most occurring words are considered in each subset. A gray shaded key is used for improving the readability and three main areas are easily separable: mal/leg, leg/leg, mal/mal. Therefore the first metric, Sim_1 is helpful for distinguishing malicious and legitimate DNS names. The lowest values are logically obtained when the types of subsets are different (mal/leg). The corresponding values are mainly under 20 and are clearly different when two sets of the same types are compared. Comparing two legitimate sets, a value around 25 is reached while malicious datasets exhibit a higher semantic similarity with a maximum of 31. Thus, the semantic field of words using in malicious words has a smaller scope than normal ones confirming that attackers targets specific services when luring users. Finally the score between malicious subsets is 50% higher than for mal/leg.

Regarding Sim_2 in table III (the score is multiply by 1,000 for readability purposes), legitimate domains look more specific than malicious ones as clearly shown by a lighter area for leg/leg in table. The difference with Sim_1 is that frequencies of occurrences are taken in account. Therefore, even if semantic scope of words is larger for normal names (lower Sim_1), there are some of them which are often used explaining a higher value for Sim_2. Nevertheless, table III discloses the same properties when two different types of datasets (mal/leg) are compared. Hence, Sim_2 is also a good alternative to distinguish malicious and legitimate domains.

Table IV represents the value of Sim_3 between the subsets. After preliminary tests, n was set to 100 in equation (7). We can see similar score for mal/mal and leg/leg comparison ($Sim_3 \sim 0.95$) that are higher than mal/leg ($Sim_3 \sim 0.75$). Like previously, this metric is able to differentiate malicious names from legitimate ones.

These experiments highlight that the three metrics Sim_1, Sim_2 and Sim_3 lead to efficiently discriminate malicious from legitimate datasets. An application of this result is to consider one of this metric and, given a set of malicious domain names, identify if an unknown set is either malicious or not. Even if the differences in values are proportionally lower than Sim_1 and Sim_2, Sim_3 is more computational efficient as mentioned in previous section. That is why, Sim_3 is computed between the subset mal-1, the model, and two other subsets, one of mal-2 (mal) and an other of leg-5 (leg). Figure 3 depicts the evolution of the average value according to the number of domain names in each subset (mal and leg) that is compared to mal-1. Ideally, the tested subset should contain a unique name, meaning that we are able to identify if a unique DNS name is malicious or not. However, as explained in section III, a single domain name is composed of few words for which it is hard to derive a relevant semantic metric and that is why domains have to be grouped prior using some state of the art technique (IP subnet, autonomous systems, registration address, etc).

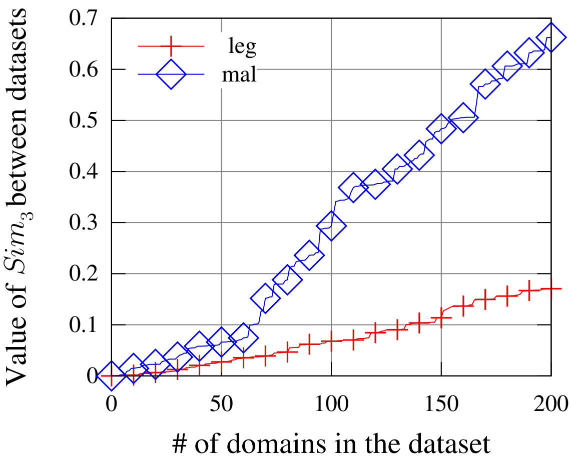

Fig. 3: Similarity score Sim_3 regarding the number of domains in the set

Thus, for small size of subsets (from 1 to 60 domains), the curves are really close even if $Sim_3(mal-1, mal) \simeq 2 \times Sim_3(mal-1, leg)$. But for more than 60 domains, a real gap is observed. Hence, malicious domains could be easily identified using a simple threshold technique. This result shows that a set of names can be identifiedf as legitimate or malicious.

V. Related work

In [17], the author figures out valuable data provided by DNS in forensics context but rely on a manual analysis. Active monitoring [18], [19], [20] is based on gathering information about domains to detect DNS anomalies but is a bit out of the scope in the context of forensics. Therefore passive DNS proposed in [7] has gained a lot of interests. The authors in [21], [22] leverage such a technique for applying classification algorithms to detect malicious domains. Different features can be used, or evaluated through statistical metrics, like the number of IP addresses, the TTL (time-to-live) or the longest meaningful word of a DNS entry, etc. In [23], the principle is applied at a higher level in DNS hierarchy. Character and gram frequencies may also reveal misbehaviors like DNS tunneling [24] or fast-flux [15]. In [25], similar statistics are considered to build a model for exploring a domain by generating subdomains automatically. TreeTop [26] employs aggregation techniques on both domain names and IP addresses for an efficient visualization of the DNS traffic.

Close to the challenges in DNS analysis, phishing URL detection has been addressed in many works as for example [27], [28]. Such methods focus more on lexical features but does not take in account the semantic of words. Thus, we introduce the use of semantic in DNS [29] for generating automatically domain names, using a markov chain model, to probe in the context of security assessment. In [30], this approach is extended for identifying phishing domains proactively for further checking, *i.e.* for creating proactively a blacklist similarly to [31], [32]. This paper clearly differs from our previous works [25], [30] which propose active techniques generating a lot of DNS traffic. For instance, generating potential phishing domains needs an additional step to check the validity of the result in real-time which is incompatible with an offline post-mortem analysis. In addition, markov chain model is discarded to improve the scalability. Thus, this work is based on the same key finding, *i.e.* DNS names have a relevant semantic, but proposes new metrics adapted to the forensics context.

VI. Conclusion

We have presented in this paper a new approach for mining DNS data. DNS is the glue that holds the Internet together and thus is a natural candidate for early digital forensics. The target application domain of our work is the reconstruction of network and user or host level activity. Our approach combines state of the art semantic and natural processing paradigms in order to build and use models for spotting malicious domain names. The obtained names can be used as enablers and guiding lines for an initial network forensic analysis. We have shown, on real world data captured from a national passive DNS system, that our approach is practical in real scenarios. The developed tool is published under a GPL license and is available on request. We plan to extend the current tool with additional support for cross linking with additional IP Flow records and thus have an integrated tool for the forensics reconstruction of network activities.

References

[1] A. networks, “Worldwide infrastructure security report (2011 report),” Tech. Rep., 2012.

[2] D. P. Fidler, “Was stuxnet an act of war? decoding a cyberattack,” *IEEE Security and Privacy*, vol. 9, no. 4, pp. 56–59, 2011.

[3] W. Wang and T. E. Daniels, “A graph based approach toward network forensics analysis,” *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 1, pp. 1–33, 2008.

[4] “Safeback,” http://www.forensics-intl.com/safeback.html.

[5] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg, and J. V. Bokkelen, “Network forensics analysis,” *IEEE Internet Computing*, vol. 6, pp. 60–66, 2002.

[6] J. McHugh, R. McLeod, and V. Nagaonkar, “Passive network forensics: behavioural classification of network hosts based on connection patterns,” *SIGOPS*, vol. 42, no. 3, pp. 99–111, 2008.

[7] F. Weimer, “Passive dns replication,” 2005.

[8] C. Castelluccia, M. A. Kaafar, P. Manils, and D. Perito, “Geolocalization of proxied services and its application to fast-flux hidden servers,” in *Internet Measurement conference*. ACM SIGCOMM, 2009.

[9] M. Afanasyev, T. Kohno, J. Ma, N. Murphy, S. Savage, A. C. Snoeren, and G. M. Voelker, “Privacy-preserving network forensics,” *Commun. ACM*, vol. 54, no. 5, pp. 78–87, 2011.

[10] Anti-Phishing Working Group and others, “Phishing Activity Trends Report - 1H2011,” *Anti-Phishing Working Group*, 2011.

[11] P. Mockapetris, “Rfc 1035: Domain names - implementation and specification.”

[12] P. Kolb, “DISCO: A Multilingual Database of Distributionally Similar Words,” in *KONVENS 2008 – Ergänzungsband: Textressourcen und lexikalisches Wissen*, 2008.

[13] B. Claise, “Cisco systems netflow services export version 9,” 2004. [Online]. Available: http://tools.ietf.org/html/rfc3954

[14] C. A. Shue, A. J. Kalafut, and M. Gupta, “Abnormally Malicious Autonomous Systems and Their Internet Connectivity,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 220–230, 2012.

[15] S. Yadav, Reddy, A.K.K., Reddy, AL, and S. Ranjan, in *Conference on Internet measurement*.

[16] T. Segaran and J. Hammerbacher, *Beautiful Data: The Stories Behind Elegant Data Solutions*. O’Reilly Media, 2009, ch. 14.

[17] B. J. Nikkel, “Domain name forensics: a systematic approach to investigating an internet presence,” *Digital Investigation*, vol. 1, no. 4, pp. 247–255, 2004.

[18] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, and J. Bellmor, “A centralized monitoring infrastructure for improving dns security,” in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, vol. 6307. Springer Berlin / Heidelberg, 2010.

[19] S. Hao, N. Feamster, and R. Pandrangi, “Monitoring the initial DNS behavior of malicious domains,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, ser. IMC ’11. New York, NY, USA: ACM, Nov. 2011, pp. 269–278.

[20] M. Felegyhazi, C. Kreibich, and V. Paxson, “On the potential of proactive domain blacklisting,” in *Conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*. USENIX Association, 2010.

[21] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, “Building a dynamic reputation system for dns,” in *SENIX Security’10*, 2010.

[22] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzz, “EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis,” in *NDSS 2011*. Internet Society, Feb. 2011.

[23] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, II, and D. Dagon, “Detecting malware domains at the upper dns hierarchy,” in *Proceedings of the 20th USENIX conference on Security*, ser. SEC’11. Berkeley, CA, USA: USENIX Association, 2011, pp. 27–27.

[24] K. Born and D. Gustafson, “Detecting dns tunnels using character frequency analysis,” *Arxiv preprint arXiv:1004.4358*, 2010.

[25] C. Wagner, J. François, R. State, T. Engel, A. Dulaunoy, and G. Wagener, “SDBF: Smart DNS Brute-Forcer,” in *Proceedings of IEEE/IFIP Network Operations and Management Symposium - NOMS*, 2012.

[26] D. Plonka and P. Barford, “Context-aware clustering of dns query traffic,” in *Conference on Internet Measurement*. ACM SIGCOMM, 2008.

[27] B. Gyawali, T. Solorio, B. Wardman, G. Warner *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*. ACM, 2011, pp. 176–183.

[28] A. Le, A. Markopoulou, and M. Faloutsos, “Phishdef: Url names say it all,” in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 191–195.

[29] S. Marchal, J. François, C. Wagner, and T. Engel, “Semantic exploration of DNS,” in *IFIP/TC6 Networking 2012*, Prague - Czech Republic, may 2012.

[30] S. Marchal, J. François, R. State, and T. Engel, “Proactive discovery of phishing related domain names,” in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science. Springer, 2012. [Online]. Available: http://lorre.uni.lu/~jerome/files/raid12.pdf

[31] J. Zhang, P. Porras, and J. Ullrich, “Highly predictive blacklisting,” in *Proceedings of the 17th conference on Security symposium*. USENIX Association, 2008, pp. 107–122.

[32] P. Prakash, M. Kumar, R. Kompella, and M. Gupta, “Phishnet: predictive blacklisting to detect phishing attacks,” in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.