

Flow Level Data Mining of DNS Query Streams for Email Worm Detection

Nikolaos Chatzis and Radu Popescu-Zeletin

Fraunhofer Institute FOKUS,
Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{nikolaos.chatzis,radu.popescu-zeletin}@fokus.fraunhofer.de

Abstract. Email worms remain a major network security concern, as they increasingly attack systems with intensity using more advanced social engineering tricks. Their extremely high prevalence clearly indicates that current network defence mechanisms are intrinsically incapable of mitigating email worms, and thereby reducing unwanted email traffic traversing the Internet. In this paper we study the effect email worms have on the flow-level characteristics of DNS query streams a user machine generates. We propose a method based on unsupervised learning and time series analysis to early detect email worms on the local name server, which is located topologically near the infected machine. We evaluate our method against an email worm DNS query stream dataset that consists of 68 email worm instances and show that it exhibits remarkable accuracy in detecting various email worm instances¹.

1 Introduction

Email worms remain an ever-evolving threat, and unwanted email traffic traversing the Internet steadily escalates [1]. This causes network congestion, which results in loss of service or degradation in the performance of network resources [2]. In addition, email worms populate almost exclusively the monthly top threat lists of antivirus companies [3,4], and are used to deliver Trojans, viruses, and phishing attempts.

Email worms rely mainly on social engineering to infect a user machine, and then they exploit information found on the infected machine about the email network of the user to spread via email among social contacts. Social engineering is a non-technical kind of intrusion, which depends on human interaction to break normal security procedures. This propagation strategy differs significantly from IP address scanning propagation; therefore, it renders network detection methods that look for high rates at which unique destination addresses are contacted [5], or high number of failed connections [6], or high self-similarity of packet contents [7] incapable of detecting this class of Internet worms. Likewise, Honeypot-based systems [8], which provide a reliable anti-scanning mechanism, are not effective against email worms. Commonly applied approaches like antivirus and antispam software and recipient or message based rules have two deficiencies. They suffer from poor detection time

¹The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216585 (INTERSECTION Project).

against novel email worm instances because they entail non-trivial human labour in order to develop a signature or a rule, and go to no lengths to reducing the unwanted email traffic traversing the Internet, as their target is to detect abusive email traffic in the network of the potential victim.

In the past much research effort has been devoted to analyzing the traffic email worm-infected user machines generate [9-13]. These studies share the positive contribution that email worm infection affects at application layer the Domain Name System (DNS) traffic of a user machine. Detection methods based on this observation focus on straightforward application layer detection, which makes them suitable for detecting few specific outdated email worms.

In this work we go a step beyond earlier work, and identify anomalies in DNS traffic that are common for email spreading malicious software, and as such they can serve as a strong basis for detecting various instances of email worms in the long run. We show that DNS query streams that non-infected user machines generate share at flow level many of the same canonical behaviours, while email worms rely on similar spreading methods that generate DNS traffic that share common patterns. We present a detection method that builds on unsupervised learning and time series analysis, and uses the wavelet transform in a different way than this often proposed in the Internet traffic analysis literature. We experiment with 68 worm instances that appeared in the wild between April 2004 and July 2007 to show that flow-level characteristics remain unaltered in the long run, and that our method is remarkably accurate.

Inspecting packets at flow level does not involve deep packet analysis. This ensures user privacy, renders our approach unaffected by encryption and keeps the processing overhead low, which is a strong requirement for busy, high-speed networks. Moreover, DNS query streams consist of significantly less data than the input of conventional network intrusion detection systems. This is advantageous, since high volumes of input data inevitably degrade the effectiveness of these systems [14]. Additionally, the efficiency and deployment ease of an in-network detection system increase as the topological proximity between the system and the user machines decreases. Local name servers are the first link of the chain of Internet connectivity. Moreover, detection at the local name server, which is located topologically near the infected machine, contributes to reducing unwanted traffic traversing the Internet.

The paper is organized as follows. In Section 2, we discuss related work. In Section 3, we explain our method for detecting email worms by flow-level analysis of DNS query streams. In Section 4, we validate our approach by examining its detection capabilities over various worm instances. We conclude in Section 5.

2 Related Work

Since our work builds on DNS traffic analysis for detecting email worms and security oriented time series analysis of Internet traffic signals using the wavelet transform we provide below the necessary background on these areas.

Previously published work provides evidence that the majority of today's open Internet operational security issues affect DNS traffic. In this section we concentrate solely on email worms and refer the interested reader to [15] for a thorough analysis. Wong et al. [11] analyze DNS traffic captured at the local name server of a campus network during the outbreak of SoBig.F and MyDoom.A. Musashi et al. [12] present

similar measurements for the same worms, and extend their work by studying Net-sky.Q and Mydoom.S in [13]. Whyte et al. [9] focus on enterprise networks and measure the DNS activity of NetSky.Q. Despite, their positive contribution in proving that there exists a correlation between email worm infection and DNS traffic, the efficacy of the detection methods these studies propose would have been dwarfed, if the methods had been evaluated against various worm instances. Indeed, the methods presented in [9,11,12,13] are straightforward and focus on application layer detection. They propose that many queries for Mail eXchange (MX) resource records (RR) or the relative numbers of queries for pointer (PTR) RR, MX and address (A) RR a user machine generates give a telltale sign of email worm infection. Although this observation holds for the few email worm instances studied in each paper, it can not be generalized for detecting various email worm instances. Furthermore, these methods neglect that DNS queries carry user sensitive information, and due to the high processing overhead they introduce by analyzing packet payloads, they are not suitable for busy, high speed networks. Moreover, as any other volume based method they require an artificial boundary as threshold on which the decision whether a user machine is infected or not is taken. In [10] the authors argue that anomaly detection is a promising solution to detect email worm-infected user machines. They use Bayesian inference assuming a priori knowledge of worm signature DNS queries to detect email worm. However, such knowledge is not apparent, and if it was it would allow straightforward detection.

The wavelet transform is a powerful tool, since its time and scale localization abilities make it ideally suited to detect irregular patterns in traffic traces. Although, many research papers appear that analyze Denial of Service (DoS) attack traffic [16,17] by means of wavelets, only a handful of papers deals with applying the wavelet transform on Internet worm signals [18,19]. Inspired by similar methodology, used to analyze DoS traffic, these papers concentrate solely on looking at worm traffic for repeating behaviors by means of the self-similarity parameter Hurst.

In this work we present a method that accurately detects various email worms by analyzing DNS query streams at flow level. We show that flow-level characteristics remain unaltered in the long run. Flow-level analysis does not violate user privacy, renders our method unaffected by encryption, eliminates the need for not anonymized data; and makes our method suitable for high-speed network environments. In our framework, we see DNS query streams from different hosts as independent time series and use the wavelet transform as a dimensionality reduction tool rather than a tool for searching self-similar patterns on a single signal.

3 Proposed Approach

Our approach is based on time series analysis of DNS query streams. Given the time series representation, we show by similarity search over time series using clustering that user machines' DNS activity fall into two canonical profiles: *legitimate user* behaviour and *email worm infected* behaviour. DNS query streams generated by user machines share many of the same canonical behaviours. Likewise, email worms rely on similar spreading methods generating query streams that share common patterns.

3.1 Data Management

As input, our method uses the complete set of DNS queries that a local name server received within an observation interval. Since we are not interested in application level information, we retain for each query the time of the query and the IP address of the requesting user machine. We group DNS queries per requesting user machine. For each user machine we consider successive time bins of equal width, and we count the DNS queries in each bin. Thereby, we get a set of univariate time series, where each one of them expresses the number of DNS queries of a user machine through time. The set of time series can be expressed as an $n \times p$ time series matrix; n is the number of user machines and p the number of time bins.

3.2 Data Pre-processing

A time series of length p can be seen as a point in the p -dimensional space. This allows using multivariate data mining algorithms directly to time series data. However, most data mining algorithms, and in particular, clustering algorithms do not work well for time series. Working with each and every time point, makes the significance of distance metrics, which are used to measure the similarity between objects, questionable [20]. To attack this problem numerous time series representations have been proposed that facilitate extracting a feature vector from the time series. The feature vector is a compressed representation of the time series, which serves as input to the data mining algorithms.

Although many representations have been proposed in the time series literature, only few of them are suitable for our framework. The fundamental requirement for our work is that clustering the feature vectors should result in clusters containing feature vectors of time series, which are similar in the time series space. The authors in [26] proved that in order for this requirement to hold the representation should satisfy the *lower bounding lemma*. In [21] the authors give an overview of the state of art in representations and highlight those that satisfy the lower bounding lemma.

We opt to use the Discrete Wavelet Transform (DWT). Motivation for our choice is that the DWT representation is intrinsically multi-resolution and allows simultaneous time and frequency analysis. Furthermore, for time series typically found in practice, many of the coefficients in a DWT representation are either zero or very small, which allows for efficient compression. Moreover, DWT is applicable for analyzing non-stationary signals, and performs well in compressing sparse spike time series.

We apply the DWT independently on each time series of the time series matrix using Mallat's algorithm [22]. The Mallat algorithm is of $O(p)$ time complexity and decomposes a time series of length p , where p must be power of two, in $\log_2 p$ levels. The number of wavelet coefficients computed after decomposing a time series is equal to the number of time points of the original time series. Therefore, applying the DWT on each line of the time series matrix gives an $n \times p$ wavelet coefficient matrix.

To reduce the dimensionality of the wavelet coefficient matrix, we apply a compression technique. In [23] the author gives a detailed description of four compression techniques. Two of them operate on each time series independently and suggest retaining the k first wavelet coefficients or the k largest coefficients in terms of absolute normalized value. Whereas the rest two are applicable to a set of n time series, so directly to the wavelet coefficient matrix. The first suggests retaining the k columns of

the matrix that have the largest mean squared value. The second suggests retaining for a given k the $n \times k$ largest coefficients of the wavelet coefficient matrix. In the interest of space, we present here experimental results only retaining the first k wavelet coefficients. This produces an $n \times k$ feature vector matrix.

3.3 Data Clustering

We validate our hypothesis that DNS query streams generated by non-infected user machines share similar characteristics, and that these characteristics are dissimilar to those of email worm-infected user machines in two steps. First we cluster the rows of the feature vector matrix in two clusters. Then we examine if one cluster contains only feature vectors of non-infected user machines and the other only feature vectors of email worm-infected machines.

We use hierarchical clustering, since it produces relatively good results, as it facilitates the exploration of data at different levels of granularity, and is robust to variations of cluster size and shape. Hierarchical clustering methods require the user to specify a dissimilarity measure. We use as dissimilarity measure the Euclidean distance, since it produces comparable results to more sophisticated distance functions [24]. Hierarchical clustering methods are categorized into *agglomerative* and *divisive*. Agglomerative methods start with each observation in its own cluster and recursively merge the less dissimilar clusters into a single cluster. By contrast, the divisive scheme starts with all observations in one cluster and subdivides them into smaller clusters until each cluster consists of only one observation. Since, we search for a small number of clusters we use the divisive scheme.

The most commonly cited disadvantage of the divisive scheme is its computational cost. A divisive algorithm considers first all divisions of the entire dataset into two non-empty sets. There are $2^{(n-1)} - 1$ possibilities of dividing n observations in two clusters, which is intractable for many practical datasets. However, DIvisive ANALysis (DIANA) [25] uses a splitting heuristic to limit the number of possible partitions, which results in $O(n^2)$. Given its quadratic time on the number of observations, DIANA scales poor with the number of observations, however we use it because it is the only divisive method generally available and in our framework it achieves clustering on a personal computer in computing time in the order of milliseconds.

4. Experimental Evaluation

We set up an isolated computer cluster, and launch 68 out of a total of 164 email worms, which have been reported between April 2004 and July 2007 in the monthly updated top threat lists of Virus Radar [3] and Viruslist [4]. We capture over a period of eight hours the DNS query streams the infected machines generate to create – to the best of our knowledge – the largest email worm DNS dataset that has been up to date used to evaluate an in-network email worm detection method. As our isolated computer cluster has no real users, we merge the worm traffic with legitimate DNS traffic captured at the primary name server of our research institute, which serves daily between 350 and 500 users. We use three recent DNS log file fragments that we split in eight hour datasets and present here experiments with four datasets.

For each DNS query stream we consider a time series of length 512 with one minute time bins. We decompose the time series, compress the wavelet coefficient matrix, retain k wavelet coefficients to make the feature vector matrix, and cluster rows of the feature vector matrix in two clusters. In this paper we focus on showing that our method detects various instances of email worms; therefore, we assume that only one user machine is infected at each time. The procedure is shown in Fig.1.

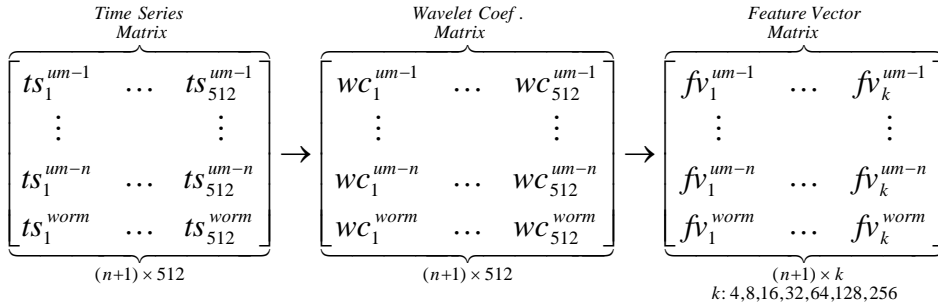


Fig. 1. We append one infectious time series to the non-infected user machines time series; decompose the time series to form the wavelet coefficient matrix, which we compress to get the feature vector matrix. This is input to the clustering analysis, which we repeat 4 Datasets \times 7 Feature vector lengths \times 68 Email Worms = 1904 times.

We examine the resulting two-cluster scheme, and find that it comprises of one dense populated cluster and a low populated cluster. Our method detects an email worm, when its feature vector belongs to the low populated cluster. In Fig. 2 we present the false positive and false negative rates. Our method erroneously reports legitimate user activity as suspicious with less than 1% for every value of k , whereas worms are misclassified with less than 2%, if at least 16 wavelet coefficients are used.

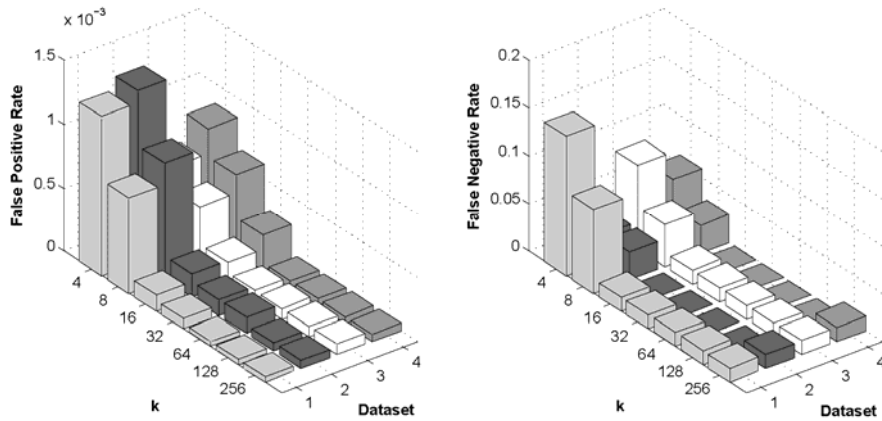


Fig. 2. False negative and false positive rates for detecting various instances of email worms over four different DNS datasets (DS1, DS2, DS3 and DS4), while retaining 4, 8, 16, 32, 64, 128 or 256 wavelet coefficients. With 16 or more wavelet coefficients both rates fall below 2%.

4. Kaspersky Lab Viruslist, <http://www.viruslist.com>
5. Roesch, M.: Snort - Lightweight Intrusion Detection for Networks. In: LISA '99, 13th USENIX Systems Administration Conference, pp. 229–238. USENIX (1999)
6. Paxson, V.: Bro: A System for Detecting Network Intruders in Real-Time. In: 7th Conference on USENIX Security Symposium. USENIX (1998)
7. Singh, S., Estan, C., Varghese, G., Savage, S.: The Earlybird System for Real-time Detection of Unknown Worms. Tech. Report CS2003-0761, University of California (2003)
8. Provos, N., Holz, T.: Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison Wesley Professional (2007)
9. Whyte, D., van Oorschot, P., Kranakis, E.: Addressing Malicious SMTP-based Mass Mailing Activity within an Enterprise Network. Technical Report TR-05-06, Carleton University, School of Computer Science (2005)
10. Ishibashi, K., Toyono, T., Toyama, K., Ishino, M., Ohshima, H., Mizukoshi, I.: Detecting Mass-Mailing Worm Infected Hosts by Mining DNS Traffic Data. In: MineNet'05 ACM SIGCOMM Workshop, pp. 159–164. ACM Press, NY, USA (2005)
11. Wong, C., Bielski, S., McCune, J., Wang, C.: A Study of Mass-Mailing Worms. In: WORM '04 ACM Workshop, pp. 1–10, ACM Press, NY, USA (2004)
12. Musashi, Y., Matsuba, R., Sugitani, K.: Indirect Detection of Mass Mailing Worm-Infected PC Terminals for Learners. In: 3rd International Conference on Emerging Telecommunications Technologies and Applications, pp. 233–237 (2004)
13. Musashi, Y., Rannenber, K.: Detection of Mass Mailing Worm-Infected PC Terminals by Observing DNS Query Access. IPSJ SIG Notes, pp. 39–44 (2004)
14. Schaelicke, L., Slabach, T., Moore, B., Freeland, C.: Characterizing the Performance of Network Intrusion Detection Sensors. In: Recent Advances in Intrusion Detection, 6th International Symposium, RAID. LNCS, pp. 155–172. Springer (2003)
15. Chatzis, N.: Motivation for Behaviour-Based DNS Security: A Taxonomy of DNS-related Internet Threats. In: International Conference on Emerging Security Information Systems, and Technologies, pp. 36–41. IEEE, Los Alamitos, CA, USA (2007)
16. Dainotti, A., Pescape, A., Ventre, G.: Wavelet-based Detection of DoS Attacks. In: Global Telecommunications Conference, GLOBECOM '06, pp. 1–6. IEEE (2006)
17. Li, L., Lee, G.: DDoS Attack Detection and Wavelets. In: 12th International Conference on Computer Communications and Networks, ICCCN 2003, pp. 421–427. IEEE (2003)
18. Chong, K., Song, H., Noh, S.: Traffic Characterization of the Web Server Attacks of Worm Viruses. In: Int. Conference on Computational Science, pp. 703–712. Springer (2003)
19. Dainotti, A., Pescape, A., Ventre, G.: Worm Traffic Analysis and Characterization. In: International Conference on Communications, ICC '07. pp. 1435–1442. IEEE (2007)
20. Aggarwal, C., Hinneburg, A., Keim, D.: On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: 8th Int. Conf. on Database Theory, LNCS, pp. 420–434, Springer (2001)
21. Bagnall, A., Ratanamahatana, C., Keogh, E., Lonardi, S., Janacek, G.: A Bit Level Representation for Time Series Data Mining with Shape Based Similarity. *Data Min. and Knowl. Discovery*, pp. 11–40, 13(1) (2006)
22. Mallat, S.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 674–693, 11(7), IEEE (1989)
23. Mörchen, F.: Time Series Feature Extraction for Data Mining Using DWT and DFT. Technical Report No. 33, Dept. of Maths and CS, Philipps-U. Marburg (2003)
24. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A survey and empirical demonstration. *Data Min. and Knowl. Discovery.*, pp. 349–371, 7(4) (2003)
25. Kaufman, L., Rousseeuw, P.: *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley (1990)
26. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time-Series Databases. In: *ACM SIGMOD International Conference on Management of Data*, pp. 419–429, ACM Press (1994)