

# Kindred Domains: Detecting and Clustering Botnet Domains Using DNS Traffic

Matthew Thomas  
Verisign Labs, VA, USA  
mthomas@verisign.com

Aziz Mohaisen  
Verisign Labs, VA, USA  
amohaisen@verisign.com

## ABSTRACT

In this paper we focus on detecting and clustering distinct groupings of domain names that are queried by numerous sets of infected machines. We propose to analyze domain name system (DNS) traffic, such as Non-Existent Domain (NXDomain) queries, at several premier Top Level Domain (TLD) authoritative name servers to identify strongly connected cliques of malware related domains. We illustrate typical malware DNS lookup patterns when observed on a global scale and utilize this insight to engineer a system capable of detecting and accurately clustering malware domains to a particular variant or malware family without the need for obtaining a malware sample. Finally, the experimental results of our system will provide a unique perspective on the current state of globally distributed malware, particularly the ones that use DNS.

## Categories and Subject Descriptors

C.2.0 [Computer Communication Networks]: General – *Security and Protection*; C.4 [Performance of Systems]: Measurement studies

## Keywords

Malware, Clustering, Automatic Analysis, DNS.

## 1. INTRODUCTION

Malware domains now commonly utilize a technique called Domain Fast-Fluxing or a Domain Generation Algorithm (DGA) [23] that enables the botmaster to more easily evade detection and their botnet being disabled [15]. For example, an infected host will generate a set of pseudo-random domain names, typically based off the system clock as a seed, and use them as points of rendezvous to receive their next set of instructions. This set of domains may vary in the number of domains and be spread across multiple TLDs [18]. Embedding the DGA instead of a list of previously-generated domains in the unobfuscated binary of the malware protects against a strings dump that could be fed into a network blacklisting appliance. This DGA/Fast-Flux technique was popularized by the Con-

ficker malware family [19] and has grown in popularity with more recent malware families such as FlashBack [20].

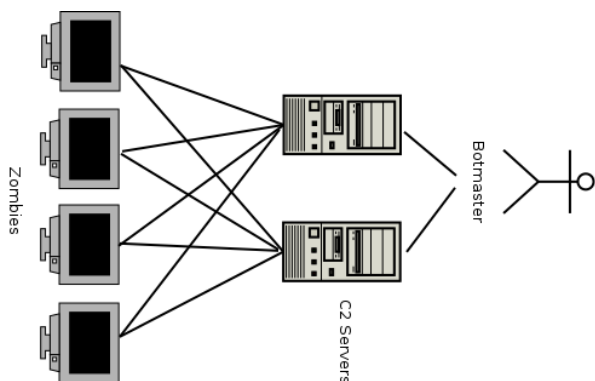
Due to the fact that the DGA is pseudo random, DNS traffic lookup patterns will emerge as all infected hosts will request the same set of domain names in a given epoch of time. Our research will aim to identify these communities of malicious domains generated by DGAs. Specifically, we will look at traffic volume metrics and similar lookup patterns within a given set of domains to possibly identify such malicious domains. As the nature of the DGA is to create a large number of domains, yet the botmaster will only register a small subset of these domains, a large portion of the DNS traffic will fall into the NXDomain (Non-Existent Domain) [14]. We observe that NXDomains have a limited caching characteristic, which make most of the queries concerning NXDomains propagate to the registry DNS servers. We hope to identify these domains and use that as a base of tracking the actively registered command and control domain in the YXDomain (Existent Domain), non-NXDomain typically resulting in NOERROR DNS responses.

To this end, the contribution of this paper is a first look at understanding the malware domain names of specific types through the lenses of authoritative DNS servers and queries. Equipped with observations on the malware and botnet domain names and their request patterns, we devise a method for traffic similarity that highlights likely related hosts/domain names. Second, using real world traces of DNS queries, including YXDomain and NXDomain traffic of known malware domains, we show how to use the traffic similarity metric for finding malicious domain names—those registered for the command and control of a botnet and variants.

The organization of the rest of this paper is as follows. In section 2 we introduce DNS profile of DGA domains, highlighting some background, context, and data. In section 4 we discuss a method for botnet detection and clustering using traffic similarity of the requesting recursive name servers (RNS) at the authoritative resolution level. In section 5 we provide our insight into using the method discussed earlier at a large scale using domain knowledge and pre-processing (filtering). The related work is outlined in section 6 followed by a conclusion in section 7.

## 2. DNS PROFILING OF DGA DOMAINS

A botnet is a group of infected machines controlled by a “bot herder or botmaster” [13]. The botmaster sends commands and malware updates through their command-and-control (C2) server, which is commonly referenced through a layer of indirection via a domain name. Accordingly, the C2 server plays an important role in the botnet for sending instructions to infected computers and is conceivably a single point of failure. Security researchers or law



**Figure 1: An illustration of a botnet infrastructure, with a botmaster, C2 servers, and Bots “zombies”.**

enforcement agencies can potentially shut down the entire botnet by making the C2 server unavailable [22].

Since DGAs are an effective way to evade detection systems relying on blacklist [10] or signature filters [11], there is a growing trend in malware development to implement DGAs. It can be found in some of the most popular malware, such as Conficker [21], Srizbi [8], Torpig [22], and Zeus [16]. DGAs provide the capability of generating pseudo-random domain names using a deterministic algorithm and some form of a seed, typically the date. The domains generated by the algorithm may vary in length and may be spread over multiple TLDs. However, the algorithm must be constructed in such a way that the domains generated are still available for registration. The malware author—who is fully aware of the domains to be generated at a particular point in time—registers a subset of the generated domains to serve as the C2 server prior to the domain’s generation date. The botmaster only requires one actively registered domain for a given generation date to effectively communicate with their botnet. While, infected computers will generate the same set of domains at that particular period of time and attempt to connect to those domain names, only the domain previously registered by the malware author will resolve and serve as the C2 server while the remaining domains will fail to resolve.

**A system for C2 detection:** While the straightforward method to capture the C2 using a pre-generated set of domain names is the perhaps the most effective, the method has some cost associated to it. First and foremost, one needs to reverse engineer the DGA in order to pre-compute those domain names, and take actions in advance to prevent malware propagation. This is however a time consuming task, and may not always be possible. To this end, the outcomes of this study look at alternatives for understanding domain name names that are likely used for malicious activities, namely those generated by DGA’s. The end result of this work is that one can, with some confidence, rely on the traffic similarity metric observed at the authoritative name servers, to identify malware families that use this piece of infrastructure for its operation. One can also use inherent differences in the traffic pattern within the same malware family to identify variants. Furthermore, this passive measurement technique provides the capability of detecting and organizing malicious domains solely based on the by-product of DNS traffic queries from a piece of malware and does not require any form of sandboxing specific malware samples.

### 3. DATA SETS

In this section, we describe the data used to analyze the behavior of malicious domains. The data described and referenced

Variant	Domains / Day	TLDs
A	250	biz, info, org, net, and com
B	250	biz, info, org, net, com, ws, cc, cn
C	50k	110 ccTLDs not including tv or cc

**Table 1: Conficker DGA Profiling by Variant**

herein was collected during July of 2012 from Verisign’s authoritative name servers for the COM, NET, TV and CC TLD authoritative name servers. As an operator of several of the largest TLDs, Verisign is positioned with a unique global view of DNS traffic that provides a distinctively alternative view of malware associated DNS traffic, as opposed to smaller passive DNS analysis systems. To that end, the type and source of data used in this study represent one unique aspect that illuminate better insight and support for the findings.

### 3.1 Malware Data

One of the most well known and widely distributed pieces of malware to utilize a DGA function is Conficker. Originally discovered in 2008, Conficker still infects thousands of computers worldwide and has been mutated to at least five known variants dubbed Conficker A, B, C, D, and E [21]. The Conficker Working Group, a consortium of researchers and security professionals, have successfully reverse engineered the DGA and pre-calculated the domains to be generated each day for variants A, B and C [1].

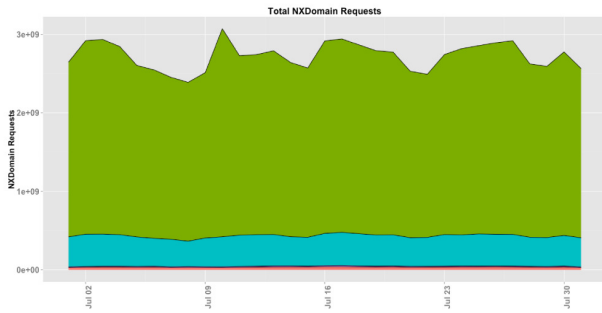
Table 1 shows the various TLDs for variants A, B and C and the number of domains to be generated on a daily basis. Variants A and B utilize the com, net and ccTLDs which are operated by Verisign allowing us to analyze the DNS traffic for these domains. By April of 2009, all domains generated by variant A of Conficker were successfully locked or preemptively registered in order to mitigate the proliferation and upgrade capabilities of the variant. Of the 15,500 domains to be generated by variants A and B in July of 2012 (which corresponds to 500 domains per day—as shown in Table 1—over 31 days), 30 of the domains were registered in either com or net with active name servers resulting in YXDomain traffic while the remaining DGA domains resulted in NXDomain traffic.

### 3.2 NXDomain Data

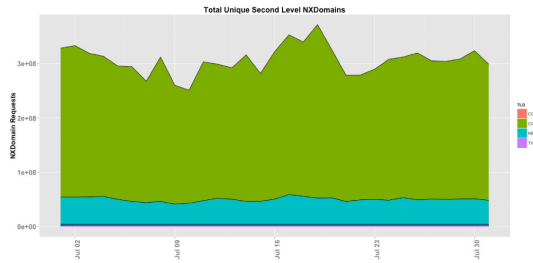
NXDomain is a commonly used term for a domain name that is unable to resolve because the domain name is not registered or a name server problem occurred during resolution. The term was originally used to represent DNS response code 3 and is formally referenced in both RFC 1035 (Domain names - implementation and specification) [14] and RFC 2308 (Negative Caching of DNS Queries) [2]. All of the data below corresponds to the state of the DNS resolution system operated by Verisign in middle of 2012. We note that while we are not able to use DNS traffic for some of the TLDs listed in Table 1, the ones using TLDs operated by Verisign (cc, tv, net and com) were captured, measured and analyzed.

Figure 2 illustrates the daily NXDomain traffic as observed for the com, net, tv and cc zones, whereas Figure 3 illustrates the number of second level domain names for the same set of TLDs over the same period of time. A typical day within the com zone will consist of over 2.5 billion NXDomain requests for more than 350 million unique second-level domains (“secondlevel.com”) while net receives around 500 million NXDomain requests for more than 60 million unique second-level domains. However, smaller zones such as tv and cc receive several magnitudes of less NXDomain traffic than com and net.

While the daily volume of total requests and unique domains is



**Figure 2: Daily NXDomain traffic volume – The total number of queries per day over a period of month**



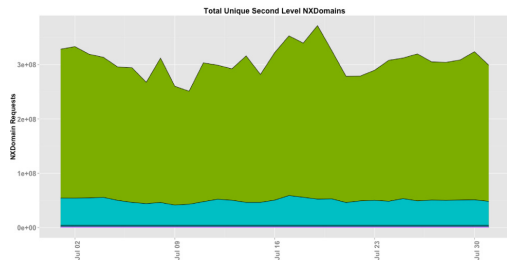
**Figure 3: Daily NXDomain Traffic Volumes – The number of second level domain names per day**

extremely high, the vast majority of individual NXDomains observed receive very few requests within a given epoch of time. Figure 4 shows a cumulative distribution function of the number of requests a given NXDomain receives within a 24 hour span of time. As depicted, more than 95% of the unique second-level NXDomains receive less than 10 requests within the 24 hour epoch.

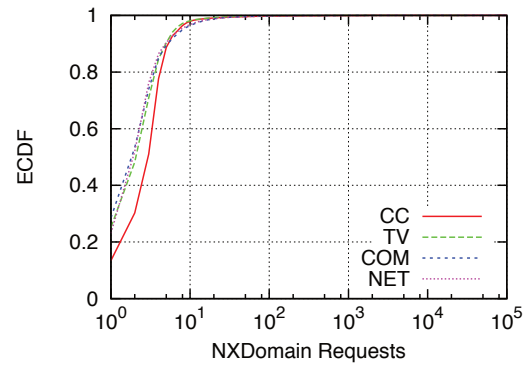
We next measure the amount of churn within the second-level NXDomains by measuring the number of days a unique second-level domain is observed within a week. As Figure 5 shows, unique NXDomains are typically only observed once within a week and only a small percentage of them reoccurs over multiple days. The large volume of NXDomain requests and low re-occurrence of unique second-level domains suggests an extremely high rate of entropy within the NXDomain DNS traffic ecosystem. This insight will help us tune our system to filter and analyze only relevant subsets of the NXDomain traffic and develop timely and performant systems using parallel computational techniques.

### 3.3 Conficker NXDomain DNS

As the vast majority of the domains generated by the Conficker

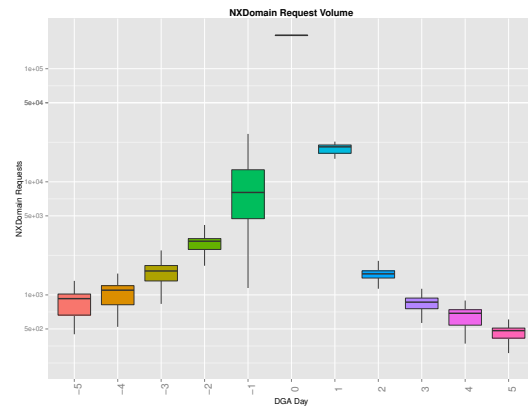


**Figure 4: Global NXDomain traffic volumes**



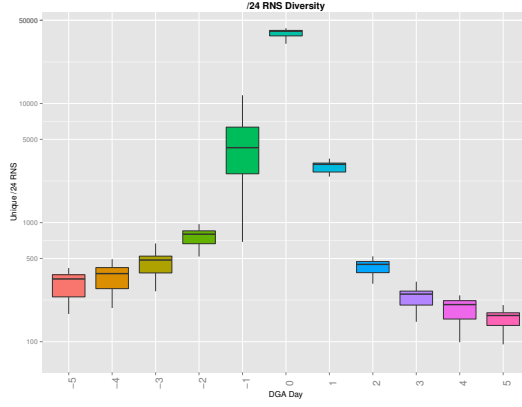
**Figure 5: Global NXDomain traffic volumes**

DGA falls into the NXDomain category, a detailed analysis of the NXDomain traffic patterns for those DGA domains was conducted. We analyzed various aspects of the DNS traffic prior to the domains generation date, the day of generation, and post generation in order to better understand the lifecycle of a DGA domain in terms of DNS traffic. Utilizing the 2012 Conficker Domain list of pre-calculated DGA domains, we were able to group domains by their generation date and measure their DNS traffic. Specifically, for a given domain to be generated on day  $x$ , we measured the domain's DNS traffic on days  $x - 5$  to  $x + 5$ .



**Figure 6: Conficker DNS lookups – total number of requests**

Figure 6 depicts the pre, during and post DNS traffic patterns for Conficker's variant B (and Figure 7 shows the number of /24 Recursive Name Servers). These figures utilize bar plots to better depict the range of DNS traffic observed on a given day. It is evident that despite a specific generation date, DGA domains receive significant volumes of traffic pre and post its specific generation date. Contrasted to the whole population of NXDomain traffic observed in figures 2 and 3, this volume and daily reoccurrence of the domain makes DGA domains statistically abnormal. Significant traffic uptake for a given DGA domain occurs both one day prior and post the specific generation date. We believe this to be a side effect of global clock skew. Traffic volumes on the exact generation date soar several magnitudes higher than the  $\pm 5$  days baseline to 42,887 unique /24 recursive name servers consisting of 199,097 total NXDomain requests from 211 unique countries for an average DGA domain. We note that /24 is used as a level of aggregation because it is small enough not to mix various networks



**Figure 7: Conficker's DNS lookup volume – /24 RNS requests** and traffic originated from them, and large enough to substantially reduce the number of entities used later on for clustering and similarity measurements.

## 4. DETECTION AND CLUSTERING

Before delving into the details of designing or operating a system that detects malware based on the C2 domain names they use, we explore the similarity aspects of traffic, and utilize this similarity measure to cluster domains based on their traffic. The resulting clusters may prove to be meaningful underpinnings in detecting botnets, and their variants, once utilized operationally (§5).

### 4.1 Computing Traffic Similarity

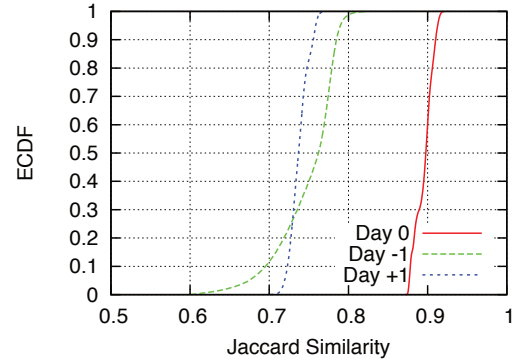
Calculating similarity between two or more entities has several standard approaches. The Jaccard index is a very simple measure that reflects the intersection of two sets over their union and provides a convenient and simple metric reflecting the similarity of two entities:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

The end goal of the similarity measurement is to determine the set of recursive name servers that have requested two domain names at the same time. This is, let  $X$  and  $Y$  be two domain names with the sets  $A$  and  $B$  being the sets of the distinct RNS that requested the domain name over a period of time  $T$ . The model in (1) calculates a normalized value for the intersection of  $A$  and  $B$ .

Figure 8 shows the cumulative distribution function of the pairwise domain similarities for a set of DGA domains based on their /24 RNS traffic for a given day. The domains on a specific DGA generation date have very high similarity measures, most measuring higher than 0.9. While the days prior and post the generation have lower similarity patterns, they are still very significant when contrasted to the whole NXDomain population's DNS traffic.

A few of the domains generated by Conficker as described in §3 were registered and had active name servers resulting in YX-Domain traffic. These domains were compared on their specific DGA generation date to the domains that received NXDomain traffic. We found that the YXDomain traffic displayed a slightly less pairwise similarity, with median similarity at about 0.7, as opposed to 0.75 and 0.9 for NXDomains the days of registration and afterwards. We hypothesize this may be influenced by DNS caching differences of YXDomains and NXDomains at the recursive name server level. However, this concept also validates the capability of using the NXDomain similarity patterns of a given DGA to find the

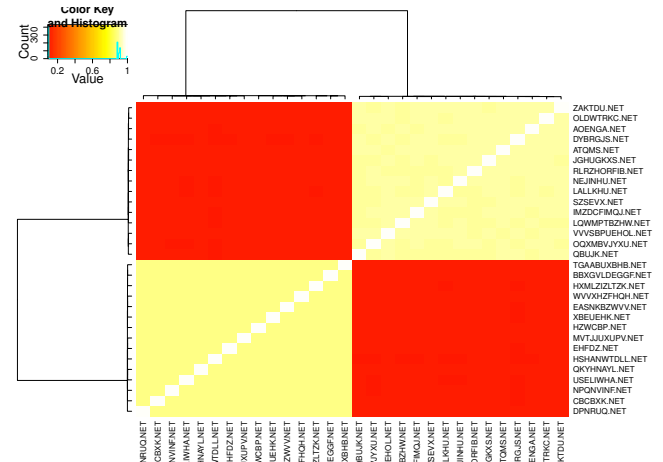


**Figure 8: Conficker's Traffic Similarity**

active and registered C2 domain in the YXDomain traffic based on DNS traffic similarities of /24 RNS.

### 4.2 Clustering

Hierarchical clustering is a popular technique in which a hierarchy of clusters is constructed by merging items into clusters based on some measure of similarity—a typical similarity measure that is widely used is the Jaccard index. Using such an approach could potentially group domains from a specific DGA or piece of malware into distinctive clusters based on DNS traffic similarity patterns. The general complexity of the algorithm is  $O(n^3)$ ; however more efficient techniques for single-linkage clustering can be accomplished in  $O(n^2)$ . The typical output of a hierarchical clustering algorithm can be depicted in the form of a dendrogram.



**Figure 9: Clustering Domains by Traffic Similarity**

A subset of domains from variants A and B were extracted and clustered via the single-linkage algorithm based on their traffic. Figure 9 shows a heat map plotting the pairwise similarity of the domains as well as a dendrogram which clustered the domains based on their similarity. This figure clearly shows high similarity patterns of domains within intra-variants (e.g. A to A and B to B) and low similarity patterns for inter-variants (e.g. A to B). The resulting clusters grouped the DGA domains to distinctive clusters based on their DNS traffic and contained only domains relevant to that particular variant, thus supporting our hypothesis. Furthermore, by not taking the visual into account, one can end up with multiple

clusters that vary based on the threshold used for the cutting phase in the hierarchical clustering.

## 5. GLOBAL MALWARE DETECTION

Based on our analysis and understanding of DGA domains and NXDomain traffic, we constructed a system that computes pairwise DNS similarity measures and subsequently performs agglomerative clustering. However, as shown in Figure 3, the number of unique second-level domains observed within a day is several magnitudes too large to fit within a pragmatic and operationally acceptable running time. Our understanding of DGA domains has shown that wide spread malware will result in thousands of unique /24 RNS requests for a given domain on its generation date. Accordingly, we utilize this knowledge to apply some filtering techniques to the set of domains over which our system will operate. Our initial thresholding criteria was set at to a minimum of 1,000 unique /24 RNS requests for a domain during a given day and resulted an average daily count of 50,000 NXDomains.

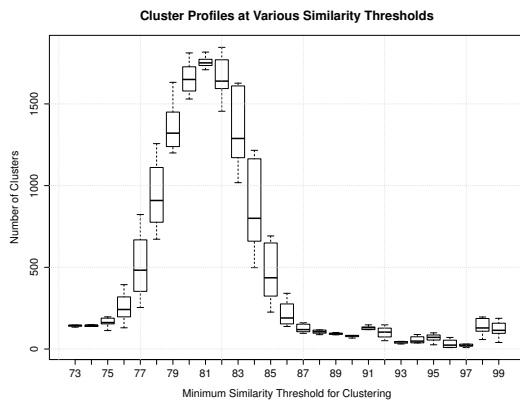


Figure 10: Number of clusters with various thresholds.

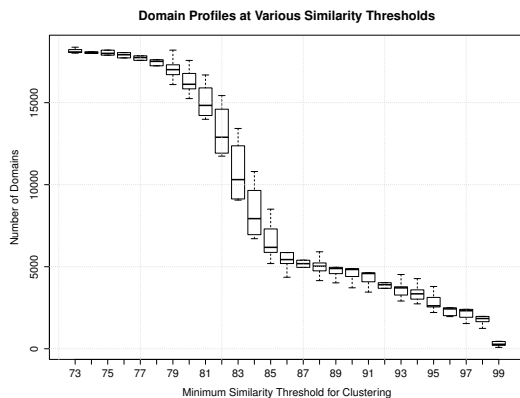


Figure 11: Number of domains with various thresholds.

As mentioned earlier, setting the minimum threshold similarity will dramatically affect the number of clusters created when performing any type of clustering. Figure 10 illustrates the various levels of a minimum similarity threshold and the corresponding number of clusters formed in the entire resolution stream at the authoritative level. The number of clusters experiences an increase when the similarity threshold is set between the levels of 75% and

82%, after which the number of clusters decreases demonstrating a convergence in the number to a fixed set of closely related types of domain names. Alternatively, figure 11 measures the number of domains that are contained in the set of clusters at various similarity threshold levels. As expected, the number of domains increases as the similarity threshold is decreased. The same growth pattern is observed when the threshold level is set between 75% and 85%.

**Labeling:** Guided by some information on the a subset of domain in some of the clusters, one can extrapolate a label in a semi-supervised learning system and determine their type: legit or malware serving domain names. We notice with high similarity, it was easy to figure out the proper use of the domains and filter them out without having to pre-compute malicious domains—it is enough to use a previously known set of malicious domain names, see where they are located in the clusters, and extrapolate the rest of them with the proper label.

A temporal analysis of the clusters from day to day was conducted to better understand the dynamics and makeup of the domains within the identified clusters. For a set of domains identified in a cluster on a given day  $x$ , we measured the percent of domains observed on day  $x + i$  where  $i = 0$  (i.e., that day) through 5. We observe that the number of domain names in a cluster exhibit a strong temporal pattern of decay as the time goes, but also 50% of the domain names initially clustered in a cluster remain in that cluster even after 5 days. This illustrates that the average half-life of a domain name in a cluster is about 5 days, after which the clustering yields less meaningful results from its initial set. This tells that one can perhaps compute the proper time parameter after which not acting (by perhaps blocking malicious domain names) a C2's registration (by the botmaster) would be considered a success.

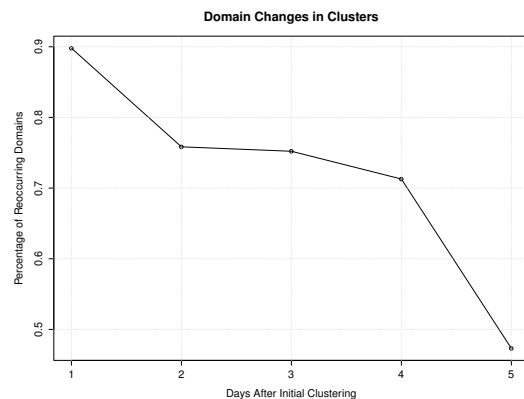
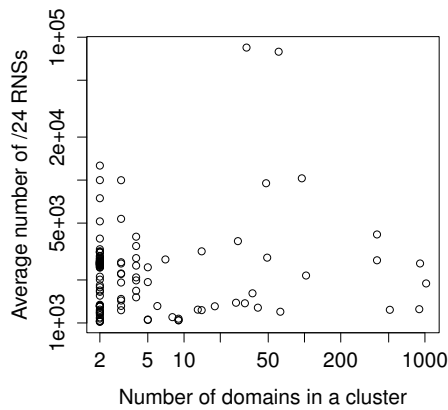


Figure 12: Temporal Variance of Domain Clusters

Figure 13 shows the resulting clusters identified via our system with a similarity threshold set to 0.9. This figure plots the number of domains contained within the cluster against the average number of unique /24 RNS requesting the domains within that cluster on a log-log scale. The results show that many of the detected clusters or malware samples/variants use a relatively small amount of domain names, observable in Verisign operated TLDs, in their DGA cycle to be typically less than 10 — while a few samples generated several hundreds of domains. This is interesting as the number of domains generated may influence the evasiveness and resilience of a botnet to a takedown. It is also interesting to note how prevalent many of these clusters are with hundreds to thousands of distinct /24 RNS addresses querying them, which may hint at the infection rate and infection diversity of a specific piece of malware.



**Figure 13: Identified Clusters – the number of clusters vs the diversity and number of querying recursive name servers.**

## 6. RELATED WORKS

Several systems have been previously constructed to detect DGA based malware systems by analyzing passive DNS traffic collected at smaller and more localized networks [23, 5, 3, 4, 9, 17, 7]. The vast majority of these systems utilize various statistical features to detect malicious domains using advanced machine learning algorithms like binary classifiers (e.g., support vector machines, decision trees, etc). While effective, those systems do not scale to the size of traffic observed at the authoritative resolution sites of a major TLD operator, like com and net. Furthermore, algorithms used in the prior literature fall under the supervised machine learning category, which requires active training and labeling.

Most related to this work is the work of Jiang et al. [12], in which the authors considered identifying malicious domain names in the view point of an enterprise network, unlike our work which detects malicious domains at the authoritative level. The work in [12] uses all codes other than “NoError” [14] (code 0) in the DNS response as an indicator of failure, while we use the “NXDomain” (code 3) only as the indicator of failure. The dataset used in both works differ in context and size. A similar idea is used in [6] utilizing NXDomain traffic at the recursive level, which also gives a limited visibility to the traffic originated for a given domain name.

## 7. CONCLUSION AND FUTURE WORK

In this paper we looked at a simple operational method of detecting domain names used for the command and control of botnets. We examined the unique characteristics of traffic sent by recursive name servers to the authoritative name servers for pre-computed C2 domain names of the Conficker malware family. We found that one not only can characterize a C2 based on the similarity measure of the requesting RNS’s, but also can find variants of a family using their C2 traffic patterns. In this paper we limited ourselves to Conficker, which is among the most prevalent families. Our preliminary studies show that findings reported in this work also apply to other families, such as Flashback and other popular DGA based malware families. Future work also includes designing a detection system based on previously clustered known malware domains to discover unknown botnets in their early phases that span both positive and negative DNS responses.

## 8. REFERENCES

- [1] —. The conficker working group.  
<http://bit.ly/1kAYsJA>, Nov 2012.

- [2] M. Andrews. Negative caching of DNS queries (DNS NCACHE). RFC 2308, 1998.
- [3] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, and J. Bellmor. A centralized monitoring infrastructure for improving dns security. In *RAID*, pages 18–37, 2010.
- [4] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a dynamic reputation system for dns. In *USENIX Security Symposium*, pages 273–290, 2010.
- [5] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou II, and D. Dagon. Detecting malware domains at the upper dns hierarchy. In *USENIX Security Symposium*, 2011.
- [6] M. Antonakakis, R. Perdisci, Y. Nadjji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of dga-based malware. In *USENIX Security*, 2012.
- [7] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. Exposure: Finding malicious domains using passive dns analysis. In *NDSS*. The Internet Society, 2011.
- [8] D. Bleaken. Botwars: the fight against criminal cyber networks. *Comp. Fraud & Sec.*, 2010.
- [9] D. Dagon, M. Antonakakis, K. Day, X. Luo, C. P. Lee, and W. Lee. Recursive dns architectures and vulnerability implications. In *NDSS*. The Internet Society, 2009.
- [10] M. Felegyhazi, C. Kreibich, and V. Paxson. On the potential of proactive domain blacklisting. In *USENIX LEET*, 2010.
- [11] K. Griffin, S. Schneider, X. Hu, and T.-c. Chiueh. Automatic generation of string signatures for malware detection. In *RAID*, pages 101–120. Springer, 2009.
- [12] N. Jiang, J. Cao, Y. Jin, E. L. Li, and Z.-L. Zhang. Identifying suspicious activities through dns failure graph analysis. In *ICNP*, pages 144–153, 2010.
- [13] M. Lelarge and J. Bolot. Economic incentives to increase security in the internet: The case for insurance. In *IEEE INFOCOM*, pages 1494–1502, 2009.
- [14] P. Mockapetris. Domain names: implementation and specification (november 1987). RFC 1035, 2004.
- [15] A. Mohaisen and O. Alrawi. Amal: High-fidelity, behavior-based automated malware analysis and classification. Technical Report, VeriSign Labs, 2013.
- [16] A. Mohaisen and O. Alrawi. Unveiling zeus: automated classification of malware samples. In *WWW (Companion Volume)*, pages 829–832, 2013.
- [17] Y. Nadjji, M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee. Beheading hydras: performing effective botnet takedowns. In *ACM CCS*, pages 121–132, 2013.
- [18] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. In *IEEE MALWARE*, pages 24–31, 2008.
- [19] P. Porras, H. Saidi, and V. Yegneswaran. Conficker c analysis. *SRI International*, 2009.
- [20] C. Roach. Flashback and mac malware.  
<http://bit.ly/1aC1qar>, April 2012.
- [21] S. Shin and G. Gu. Conficker and beyond: a large-scale empirical study. In *ACSAC*, 2010.
- [22] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: analysis of a botnet takeover. In *ACM CCS*, pages 635–647, 2009.
- [23] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan. Detecting algorithmically generated malicious domain names. In *ACM IMC*, pages 48–61, 2010.