# Building a Dynamic Reputation System for DNS

Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster
College of Computing, Georgia Institute of Technology,
{manos,rperdisc,dagon,wenke,feamster}@cc.gatech.edu

## Abstract

*The Domain Name System (DNS) is an essential protocol used by both legitimate Internet applications and cyber attacks. For example, botnets rely on DNS to support agile command and control infrastructures. An effective way to disrupt these attacks is to place malicious domains on a "blocklist" (or "blacklist") or to add a filtering rule in a firewall or network intrusion detection system. To evade such security countermeasures, attackers have used* DNS agility*, e.g., by using new domains daily to evade static blacklists and firewalls. In this paper we propose Notos, a dynamic reputation system for DNS. The premise of this system is that malicious, agile use of DNS has unique characteristics and can be distinguished from legitimate, professionally provisioned DNS services. Notos uses passive DNS query data and analyzes the network and zone features of domains. It builds models of known legitimate domains and malicious domains, and uses these models to compute a reputation score for a new domain indicative of whether the domain is malicious or legitimate. We have evaluated Notos in a large ISP's network with DNS traffic from 1.4 million users. Our results show that Notos can identify malicious domains with high accuracy (true positive rate of 96.8%) and low false positive rate (0.38%), and can identify these domains weeks or even months before they appear in public blacklists.*

## 1 Introduction

The Domain Name System (DNS) [12, 13] maps domain names to IP addresses, and provides a core service to applications on the Internet. DNS is also used in network security to distribute IP reputation information, e.g., in the form of DNS-based Block Lists (DNSBLs) used to filter spam [18, 5] or block malicious web pages [26, 14].

Internet-scale attacks often use DNS as well because they are essentially Internet-scale malicious applications. For example, *spyware* uses anonymously registered domains to exfiltrate private information to *drop sites*. Disposable domains are used by *adware* to host malicious or false advertising content. *Botnets* make agile use of short-lived domains to evasively move their command-and-control (C&C) infrastructure. Fast-flux networks rapidly change DNS records to evade blacklists and resist take downs [25]. In an attempt to evade domain name blacklisting, attackers now make very aggressive use of *DNS agility*. The most common example of an *agile* malicious resource is a fast-flux network, but DNS agility takes many other forms including disposable domains (e.g., tens of thousands of randomly generated domain names used for spam or botnet C&C), domains with dozens of `A` records or `NS` records (in excess of levels recommended by RFCs, in order to resist takedowns), or domains used for only a few hours of a botnet's lifetime. Perhaps the best example is the Conficker.C *worm* [15]. After Conficker.C infects a machine, it will try to contact its C&C server, chosen at random from a list of 50,000 possible domain names created every day. Clearly, the goal of Conficker.C was to frustrate blacklist maintenance and takedown efforts. Other malware that abuse DNS include Sinowal (a.k.a. Torpig) [9], Kraken [20], and Srizbi [22]. The aggressive use of newly registered domain names is seen in other contexts, such as spam campaigns and malicious flux networks [25, 19]. This strategy delays takedowns, degrades the effectiveness of blacklists, and pollutes the Internet's name space with unwanted, discarded domains.

In this paper, we study the problem of *dynamically* assigning *reputation* scores to new, unknown domains. Our main goal is to automatically assign a low reputation score to a domain that is involved in malicious activities, such as malware spreading, phishing, and spam campaigns. Conversely, we want to assign a high reputation score to domains that are used for legitimate purposes. The reputation scores enable *dynamic* domain name blacklists to counter cyber attacks much more effectively. For example, with static blacklisting, by the time one has sufficient evidence to put a domain on a blacklist, it typically has been involved in malicious activities for a significant period of time. With dynamic blacklisting our goal is to decide, even for a new domain, whether it is likely used for malicious purposes. To this end, we propose Notos, a system that dynamically assigns reputation scores to domain names. Our work is based on the observation that agile malicious uses of DNS have unique characteristics, and can be distinguished from legitimate, professionally provisioned DNS services. In short, network resources used for malicious and

fraudulent activities inevitably have distinct *network characteristics* because of their need to evade security countermeasures. By identifying and measuring these features, Notos can assign appropriate reputation scores.

Notos uses historical DNS information collected passively from multiple recursive DNS resolvers distributed across the Internet to build a model of how network resources are allocated and operated for legitimate, professionally run Internet services. Notos also uses information about malicious domain names and IP addresses obtained from sources such as spam-traps, honeynets, and malware analysis services to build a model of how network resources are typically allocated by Internet miscreants. With these models, Notos can assign reputation scores to new, previously unseen domain names, therefore enabling dynamic blacklisting of unknown malicious domain names and IP addresses.

Previous work on dynamic reputation systems mainly focused on IP reputation [24, 31, 1, 21]. To the best of our knowledge, our system is the first to create a comprehensive *dynamic* reputation system around domain names. To summarize, our main contributions are as follows:

- We designed Notos, a dynamic, comprehensive reputation system for DNS that outputs reputation scores for domains. We constructed *network* and *zone* features that capture the characteristics of resource provisioning, usages, and management of domains. These features enable Notos to learn models of how legitimate and malicious domains are operated, and compute accurate reputation scores for new domains.

- We implemented a proof-of-concept version of our system, and deployed it in a large ISP's DNS network in Atlanta, GA and San Jose, CA, USA, where we observed DNS traffic from 1.4 million users. We also used passive DNS data from Security Information Exchange (SIE) project [3]. This extensive *real-world* evaluation shows Notos can correctly classify new domains with a low false positive rate (0.38%) and high true positive rate (96.8%). Notos can detect and assign a low reputation score to malware- and spam-related domain names several days or even weeks before they appear on public blacklists.

Section 2 provides some background on DNS and related works. Readers familiar with this may skip to Section 3, where we describe our passive DNS collection strategy and other whitelist and blacklist inputs. We also describe three feature extraction modules that measure key network, zone and evidence-based features. Finally, we describe how these features are clustered and incorporated into the final reputation engine. To evaluate the output of Notos, we gathered an extensive amount of network trace data. Section 4 describes the data collection process, and Section 5 details the sensitivity of each module and final output.

## 2 Background and Related Work

DNS is the protocol that resolves a domain name, like `www.example.com`, to its corresponding IP address, for example `192.0.2.10`. To resolve a domain, a host typically needs to consult a local recursive DNS server (RDNS). A recursive server iteratively discovers which Authoritative Name Server (ANS) is responsible for each zone. The typical result of this iterative process is the mapping between the requested domain name and its current IP addresses.

By aggregating all **unique**, successfully resolved `A-type` DNS answers at the recursive level, one can build a passive DNS database. This passive DNS (pDNS) database is effectively the DNS fingerprint of the monitored network and typically contains unique `A-type` resource records (RRs) that were part of monitored DNS answers. A typical RR for the domain name `example.com` has the following format: {`example.com. 78366 IN A 192.0.2.10`}, which lists the domain name, TTL, class, type, and rdata. For simplicity, we will refer to an RR in this paper as just a tuple of the domain name and IP address.

Passive DNS data collection was first proposed by Florian Weimer [27]. His system was among the first that appeared in the DNS community with its primary purpose being the conversion of historic DNS traffic into an easily accessible format. Zdrnja et al. [29] with their work in "Passive Monitoring of DNS Anomalies" discuss how pDNS data can be used for gathering security information from domain names. Although they acknowledge the possibility of creating a DNS reputation system based on passive DNS measurement, they do not quantify a reputation function. Our work uses the idea of building passive DNS information only as a seed for computing statistical DNS properties for each successful DNS resolution. The analysis of these statistical properties is the basic building block for our dynamic domain name reputation function. Plonka et al. [17] introduced Treetop, a scalable way to manage a growing collection of passive DNS data and at the same time correlate zone and network properties. Their cluster zones are based on different classes of networks (class A, class B and class C). Treetop differentiates DNS traffic based on whether it complies with various DNS RFCs and based on the resolution result. Plonka's proposed method, despite being novel and highly efficient, offers limited DNS security information and cannot assign reputation scores to records.

Several papers, e.g., Sinha et al. [24] have studied the effectiveness of IP blacklists. Zhang, et al. [31] showed that the hit rate of highly predictable blacklists (HBLs) decreases significantly over a period of time. Our work addresses the dynamic DNS blacklisting problem that makes it significantly different from the highly predictable blacklists. Importantly, Notos does not aim to create IP blacklists. By using properties of the DNS protocol, Notos can rank a domain name as potentially malicious or not. Garera et al. [8] discussed "phishing" detection predominately using properties of the URL and not sta-

tistical observations about the domains or the IP address. The statistical features used by Holz et al. [10] to detect fast flux networks are similar to the ones we used in our work, however, Notos utilizes a more complete collection of network statistical features and is not limited to fast flux networks detection.

Researchers have attempted to use unique characteristics of malicious networks to detect sources of malicious activity. Anderson et al. [1] proposed Spamscatter as the first system to identify and characterize spamming infrastructure by utilizing layer 7 analysis (i.e., web sites and images in spam). Hao et al. [21] proposed SNARE, a spatio-temporal reputation engine for detecting spam messages with very high accuracy and low false positive rates. The SNARE reputation engine is the first work that utilized statistical network-based features to harvest information for spam detection. Notos is complementary to SNARE and Spamscatter, and extends both to not only detect spam, but also identify other malicious activity such as phishing and malware hosting. Qian et al. [28] present their work on spam detection using network-based clustering. In this work, they show that network-based clusters can increase the accuracy of spam-oriented blacklists. Our work is more general, since we try to identify various kinds of malicious domain names. Nevertheless, both works leverage network-based clustering for identifying malicious activities.

Felegyhazi et al. [7] proposed a DNS reputation blacklisting methodology based on `WHOIS` observations. Our system does not use `WHOIS` information making our approaches complementary by design. Sato et al. [23] proposed a way to extend current blacklists by observing the co-occurrence of IP address information. Notos is a more generic approach than the proposed system by Sato and is not limited to botnet related domain name detection. Finally, Notos builds the reputation function mainly based upon passive information from DNS traffic observed in real networks — not traffic observed from honeypots.

No previous work has tried to assign a dynamic domain name reputation score for any domain that traverses the edge of a network. Notos harvests information from multiple sources—the domain name, its effective zone, the IP address, the network the IP address belongs to, the Autonomous System (AS) and honeypot analysis. Furthermore, Notos uses short-lived passive DNS information. Thus, it is difficult for a malicious domain to dilute its passive DNS footprint.

## 3 Notos: A Dynamic Reputation System

The goal of the Notos reputation system is to dynamically assign reputation scores to domain names. Given a domain name $d$, we want to assign a low reputation score if $d$ is involved in malicious activities (e.g., if it has been involved with botnet C&C servers, spam campaigns, malware propagation, etc.). On the other hand, we want to assign a high reputation score if $d$ is associated with legitimate Internet services.

Notos' main source of information is a passive DNS (pDNS) database, which contains historical information about domain names and their resolved IPs. Our pDNS database is constantly updated using real-world DNS traffic from multiple geographically diverse locations as shown in Figure 1. We collect DNS traffic from two ISP recursive DNS servers (RDNS) located in Atlanta and San Jose. The ISP nodes witness 30,000 DNS queries/second during peak hours. We also collect DNS traffic through the Security Information Exchange (SIE) [3], which aggregates DNS traffic received by a large number of RDNS servers from authoritative name servers across North America and Europe. In total, the SIE project processes approximately 200 Mbit/s of DNS messages, several times the total volume of DNS traffic in a single US ISP.

Another source of information we use is a list of known malicious domains. For example, we run known malware samples in a controlled environment and we classify as suspicious all the domains contacted by malware samples that do not match a pre-compiled white list. In addition, we extract suspicious domain names from spam emails collected using a large spam-trap. Again, we discard the domains that match our whitelist and consider the rest as potentially malicious. Furthermore, we collect a large list of popular, legitimate domains from `alexa.com` (we discuss our data collection and analysis in more details in Section 4). The set of known malicious and legitimate domains represents our *knowledge base*, and is used to train our reputation engine, as we discuss in Section 4.

Intuitively, a domain name $d$ can be considered suspicious when there is evidence that $d$ or its IP addresses are (or were in previous months) associated with known malicious activities. The more evidence of "bad associations" we can find about $d$, the lower the reputation score we will assign to it. On the other hand, if there is evidence that $d$ is (or was in the past) associated with legitimate, professionally run Internet services, we will assign it a higher reputation score.

### 3.1 System Overview

Before describing the internals of our reputation system, we introduce some basic terminology. A domain name $d$ consists of a set of substrings or labels separated by a period; the rightmost label is called the *top-level* domain, or TLD. The *second-level* domain (2LD) represents the two rightmost labels separated by a period; the *third-level* domain (3LD) analogously contains the three rightmost labels, and so on. As an example, given the domain name $d$="a.b.example.com", $TLD(d)$="com", $2LD(d)$="example.com", and $3LD(d)$="b.example.com".

Let $s$ be a domain name (e.g., $s$="example.com"). We define $Zone(s)$ as the set of domains that include $s$ and all domain names that end with a period followed by $s$ (e.g., domains ending in ".example.com").

Let $D = \{d_1, d_2, ..., d_m\}$ be a set of domain names. We

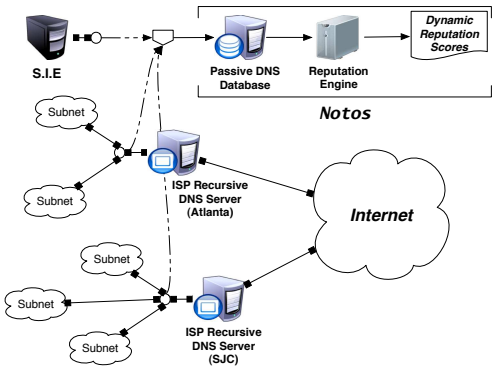


**Figure 1. System overview.**

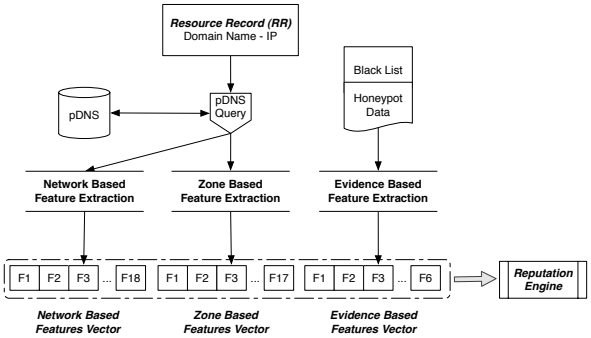


**Figure 2. Computing network-based, zone-based, evidence-based features.**

call $A(D)$ the set of IP addresses ever pointed to by any domain name $d \in D$.

Given an IP address $a$, we define $BGP(a)$ to be the set of all IPs within the BGP prefix of $a$, and $AS(a)$ as the set of IPs located in the autonomous system in which $a$ resides. In addition, we can extend these functions to take as input a set of IPs: given IP set $A = a_1, a_2, ..., a_N$, $BGP(A) = \bigcup_{k=1..N} BGP(a_k)$; $AS(a)$ is similarly extended.

To assign a reputation score to a domain name $d$ we proceed as follows. First, we consider the most current set $A_c(d) = \{a_i\}_{i=1..m}$ of IP addresses to which $d$ points. Then, we query our pDNS database to retrieve the following information:

- *Related Historic IPs (RHIPs)*, which consist of the union of $A(d)$, $A(Zone(3LD(d)))$, and $A(Zone(2LD(d)))$. In order to simplify the notation we will refer to $A(Zone(3LD(d)))$ and $A(Zone(2LD(d)))$ as $A_{3LD}(d)$ and $A_{2LD}(d)$, respectively.

- *Related Historic Domains (RHDNs)*, which comprise the entire set of domain names that ever resolved to an IP address $a \in AS(A(d))$. In other words, RHDNs contain all the domains $d_i$ for which $A(d_i) \cap AS(A(d)) \neq \emptyset$.

After extracting the above information from our pDNS database, we measure a number of statistical features. Specifically, for each domain $d$ we extract three groups of features, as shown in Figure 2:

- **Network-based features:** The first group of statistical features is extracted from the set of RHIPs. We measure quantities such as the total number of IPs historically associated with $d$, the diversity of their geographical location, the number of distinct autonomous systems (ASs) in which they reside, etc.

- **Zone-based features:** The second group of features we extract are those from the RHDNs set. We measure the average length of domain names in RHDNs, the number of distinct TLDs, the occurrence frequency of different characters, etc.

- **Evidence-based features:** The last set of features includes the measurement of quantities such as the number of distinct malware samples that contacted the domain $d$, the number of malware samples that connected to any of the IPs pointed by $d$, etc.

Once extracted, these statistical features are fed to the reputation engine. Notos' reputation engine operates in two modes: an off-line "training" mode and an on-line "classification" mode. During the off-line mode, Notos *trains* the reputation engine using the information gathered in our *knowledge base*, namely the set of known malicious and legitimate domain names and their related IP addresses. Afterwards, during the on-line mode, for each new domain $d$, Notos queries the trained reputation engine to compute a reputation score for $d$ (see Figure 3). We now explain the details about the statistical features we measure, and how the reputation engine uses them during the off-line and on-line modes to compute a domain names' reputation score.

## 3.2 Statistical Features

In this section we identify key statistical features and the intuition behind their selection.

### 3.2.1 Network-based Features

Given a domain $d$ we extract a number of statistical features from the set RHIPs of $d$, as mentioned in Section 3.1. Our network-based features describe how the operators who *own* $d$ and the IPs that domain $d$ points to, allocate their network resources. Internet miscreants often abuse DNS to operate their malicious networks with a high level of *agility*. Namely, the

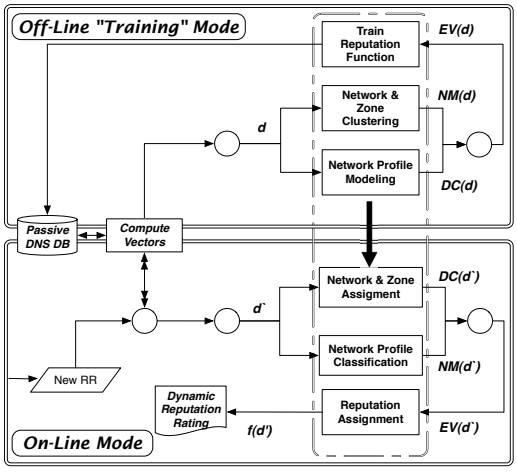


**Figure 3. Off-line and on-line modes in Notos.**

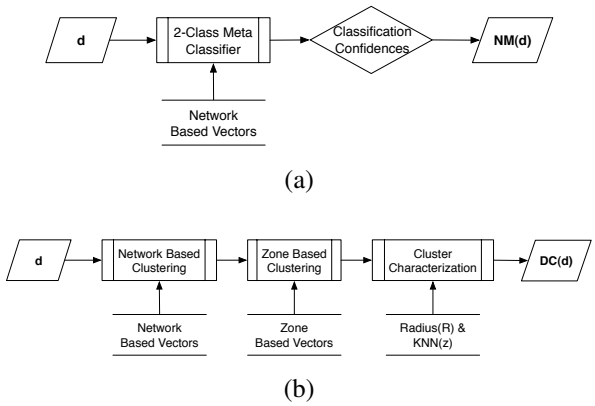


**Figure 4. (a) Network profile modeling in Notos. (b) Network and zone based clustering in Notos.**

domain names and IPs that are used for malicious purposes are often short-lived and are characterized by a high *churn* rate. This agility avoids simple blacklisting or removals by law enforcement. In order to measure the level of agility of a domain name $d$, we extract eighteen statistical features that describe $d$'s *network profile*. Our network features fall into the following three groups:

- *BGP features*. This subset consists of a total of nine features. We measure the number of distinct BGP prefixes related to $BGP(A(d))$, the number of countries in which these BGP prefixes reside, and the number of organizations that *own* these BGP prefixes; the number of distinct IP addresses in the sets $A_{3LD}(d)$ and $A_{2LD}(d)$; the number of distinct BGP prefixes related to $BGP(A_{3LD}(d))$ and $BGP(A_{2LD}(d))$, and the number of countries in which these two sets of prefixes reside.

- *AS features*. This subset consists of three features, namely the number of distinct autonomous systems related to $AS(A(d))$, $AS(A_{3LD}(d))$, and $AS(A_{2LD}(d))$.

- *Registration features*. This subset consists of six features. We measure the number of distinct registrars associated with the IPs in the $A(d)$ set; the diversity in the registration dates related to the IPs in $A(d)$; the number of distinct registrars associated with the IPs in the $A_{3LD}(d)$ and $A_{2LD}(d)$ sets; and the diversity in the registration dates for the IPs in $A_{3LD}(d)$ and $A_{2LD}(d)$.

While most legitimate, professionally run Internet services have a very stable network *profile*, which is reflected into low values of the network features described above, the profiles of malicious networks (e.g., fast-flux networks) usually change relatively frequently, thus causing their network features to be assigned higher values. We expect a domain name $d$ from a legitimate zone to exhibit a small values in its AS features, mainly because the IPs in the RHIPs should belong to the same organization or a small number of different organizations. On the other hand, if a domain name $d$ participates in malicious activities (i.e., botnet activities, flux networks), then it could reside in a large number of different networks. The list of IPs in the RHIPs that correspond to the malicious domain name will produce AS features with higher values. In the same sense, we measure that homogeneity of the registration information for benign domains. Legitimate domains are typically linked to address space owned by organizations that acquire and announce network blocks in some order. This means that the registration-feature values for a legitimate domain name $d$ that owned by the same organizations will produce a list of IPs in the RHIPs that will have small registration feature values. If this set of IPs exhibits high registration feature values, it means that they very likely reside in different registrars and were registered on different dates. Such registration-feature properties are typically linked with fraudulent domains.

### 3.2.2 Zone-based Features

The network-based features measure a number of characteristics of IP addresses historically related to a given domain name $d$. On the other hand, the zone-based features measure the characteristics of domain names historically associated with $d$. The intuition behind the zone-based features is that while legitimate Internet services may be associated with many different domain names, these domain names usually have strong similarities. For example, `google.com`, `googlesyndication.com`, `googlewave.com`, etc., are all related to Internet services provided by Google, and contain the string "google" in their name. On the other hand, malicious domain names related to the same spam campaign, for example, often look randomly generated and share few common characteristics. Therefore, our zone-based features aim to measure the

level of *diversity* across the domain names in the RHDNs set. Given a domain name $d$, we extract seventeen statistical features that describe the properties of the set RHDNs of domain names related to $d$. We divide these seventeen features into two groups:

- *String features*. This group consists of twelve features. We measure the number of distinct domain names in RHDNs, and the average and standard deviation of their length; the mean, median, and standard deviation of the occurrence frequency of each single character in the domain name strings in RHDNs; the mean, median and standard deviation of the distribution of 2-grams (i.e., pairs of characters); the mean, median and standard deviation of the distribution of 3-grams.

- *TLD features*. This group consists of five features. For each domain $d_i$ in the RHDNs set, we extract its top-level domain $TLD(d_i)$ and we count the number of distinct TLD strings that we obtain; we measure the ratio between the number of domains $d_i$ whose $TLD(d_i)$=".com" and the total number of TLD different from ".com"; also, we measure the mean, median, and standard deviation of the occurrence frequency of the TLD strings.

It is worth noting that whenever we measure the mean, median and standard deviation of a certain property, we do so in order to summarize the shape of its distribution. For example, by measuring the mean, median, and standard deviation of the occurrence frequency of each character in a set of domain name strings, we summarize how the distribution of the character frequency looks like.

#### 3.2.3 Evidence-based Features

We use the evidence-based features to determine to what extent a given domain $d$ is associated with other known malicious domain names or IP addresses. As mentioned above, Notos collects a *knowledge base* of known suspicious, malicious, and legitimate domain names and IPs from public sources. For example, we collect malware-related domain names by executing large numbers of malware samples in a controlled environment. Also, we check IP addresses against a number of public IP blacklists. We elaborate on how we build Notos' knowledge base in Section 4. Given a domain name $d$, we measure six statistical features using the information in the knowledge base. We divide these features into two groups:

- *Honeypot features*. We measure three features, namely the number of distinct malware samples that, when executed, try to contact $d$ or any IP address in $A(d)$; the number of malware samples that contact any IP address in $BGP(A(d))$; and the number of samples that contact any IP address in $AS(A(d))$.

- *Blacklist features*. We measure three features, namely the number of IP addresses in $A(d)$ that are listed in public IP blacklists; the number of IPs in $BGP(A(d))$ that are listed in IP blacklists; and the number of IPs in $AS(A(d))$ that are listed in IP blacklists.

Notos uses the blacklist features from the evidence vector so it can identify the re-use of known malicious network resources like IPs, BGP prefixes or even ASs. Domain names are significantly cheaper than IPv4 addresses; so malicious users tend to reuse address space with new domain names. We should note that the evidence-based features represent only part of the information we used to compute the reputation scores. The fact that a domain name was queried by malware does not automatically mean that the domain will receive a low reputation score.

### 3.3 Reputation Engine

Notos' reputation engine is responsible for deciding whether a domain name $d$ has characteristics that are similar to either legitimate or malicious domain names. In order to achieve this goal, we first need to *train* the engine to recognize whether $d$ belongs (or is "close") to a known *class of domains*. This training can be repeated periodically, in an off-line fashion, using historical information collected in Notos' *knowledge base* (see Section 4). Once the engine has been trained, it can be used in on-line mode to assign a reputation score to each new domain name $d$.

In this section, we first explain how the reputation engine is trained, and then we explain how a trained engine is used to assign reputation scores.

#### 3.3.1 Off-Line Training Mode

During off-line training (Figure 3), the reputation engine builds three different modules. We briefly introduce each module and then elaborate on the details.

- *Network Profiles Model*: a model of how well known networks behave. For example, we model the network characteristics of popular content delivery networks (e.g., Akamai, Amazon CloudFront), and large *popular* websites (e.g., google.com, yahoo.com). During the on-line mode, we compare each new domain name $d$ to these models of well-known network profiles, and use this information to compute the final reputation score, as explained below.

- *Domain Name Clusters*: we group domain names into clusters sharing similar characteristics. We create these clusters of domains to identify groups of domains that contain mostly malicious domains, and groups that contain mostly legitimate domains. In the on-line mode,

given a new domain $d$, if $d$ (more precisely, $d$'s projection into a statistical feature space) falls within (or close to) a cluster of domains containing mostly malicious domains, for example, this gives us a hint that $d$ should be assigned a low reputation score.

- *Reputation Function*: for each domain name $d_i, i = 1..n$, in Notos' knowledge base, we *test* it against the trained network profiles model and domain name clusters. Let $NM(d_i)$ and $DC(d_i)$ be the output of the Network Profiles (NP) module and the Domain Clusters (DC) module, respectively. The reputation function takes in input $NM(d_i)$, $DC(d_i)$, and information about whether $d_i$ and its resolved IPs $A(d_i)$ are known to be legitimate, suspicious, or malicious (i.e., if they appeared in a domain name or IP blacklist), and builds a model that can assign a reputation score between zero and one to $d$. A reputation score close to zero signifies that $d$ is a malicious domain name while a score close to one signifies that $d$ is benign.

We now describe each module in detail.

#### 3.3.2 Modeling Network Profiles

During the off-line training mode, the reputation engine builds a model of well-known network behaviors. An overview of the network profile modeling module can be seen in Figure 4(a). In practice we select five sets of domain names that share similar characteristics, and *learn* their network profiles. For example, we identify a set of domain names related to very popular websites (e.g., google.com, yahoo.com, amazon.com) and for each of the related domain names we extract their network features, as explained in Section 3.2.1. We then use the extracted feature vectors to train a statistical classifier that will be able to recognize whether a new domain name $d$ has network characteristics similar to the popular websites we modeled.

In our current implementation of Notos we model the following classes of domain names:

- **Popular Domains**. This class consists of a large set of domain names under the following DNS zones: google.com, yahoo.com, amazon.com, ebay.com, msn.com, live.com, myspace.com, and facebook.com.
- **Common Domains**. This class of domains includes domain names under the top one hundred zones, according to `alexa.com`. We exclude from this group all the domain names already included in the *Popular Domains* class (which we model separately).
- **Akamai Domains**. Akamai is a large content delivery network (CDN), and the domain names related to this CDN have very peculiar network characteristics. To model the network profile of Akamai's domain names, we collect a set of domains under the following zones: akafms.net, akamai.net, akamaiedge.net, akamai.com, akadns.net, and akamai.com.
- **CDN Domains**. In this class we include domain names related to CDNs other than Akamai. For example, we collect domain names under the following zones: panthercdn.com, llnwd.net, cloudfront.net, nyud.net, nyucd.net and redcondor.net. We chose not to aggregate these CDN domains and Akamai's domains in one class, since we observed that Akamai's domains have a very unique network profile, as we discuss in Section 4. Therefore, learning two separate models for the classes of *Akamai Domains* and *CDN Domains* allows use to achieve better classification accuracy during the on-line mode, compared to learning only one model for both classes (see Section 3.3.5).
- **Dynamic DNS Domains**. This class includes a large set of domain names registered under two of the largest dynamic DNS providers, namely No-IP (`no-ip.com`) and DynDNS (`dyndns.com`).

For each class of domains, we train a statistical classifier to distinguish between one of the classes and all the others. Therefore, we train five different classifiers. For example, we train a classifier that can distinguish between the class of *Popular Domains* and all other classes of domains. That is, given a new domain name $d$, this classifier is able to recognize whether $d$'s network profile looks like the profile of a well-known popular domain or not. Following the same logic we, can recognize network profiles for the other classes of domains.

#### 3.3.3 Building Domain Name Clusters

In this phase, the reputation engine takes the domain names collected in our pDNS database during a *training period*, and builds clusters of domains that share similar network and zone based features. The overview of this module can be seen in Figure 4(b). We perform clustering in two steps. In the first step we only use the network-based features to create coarse-grained clusters. Then, in the second step, we split each coarse-grained cluster into finer clusters using only the zone-based features, as shown in Figure 5.

**Network-based Clustering** The objective of network-based clustering is to group domains that share similar levels of *agility*. This creates separate clusters of domains with "stable" network characteristics and "non-stable" networks (like CDNs and malicious flux networks).

**Zone-based Clustering** After clustering the domain names according to their network-based features, we further split the network-based clusters of domain names into finer groups. In this step, we group domain names that are in the same

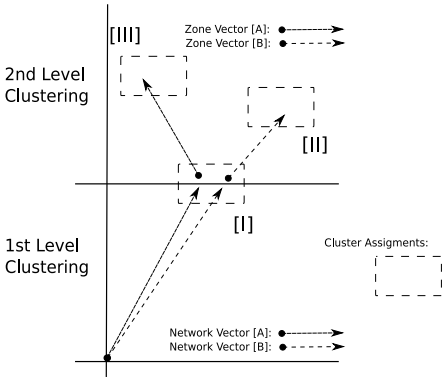


**Figure 5. Network & zone based clustering process in Notos, in the case of a Akamai [A] and a malicious [B] domain name.**

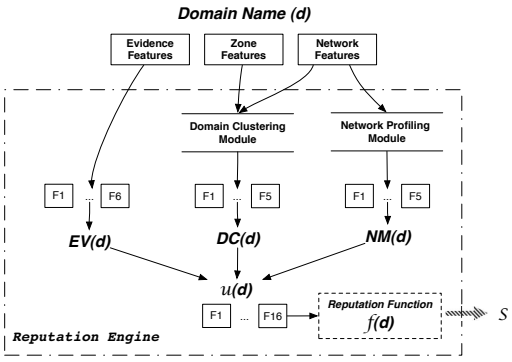


**Figure 6. The output from the network profiling module, the domain clustering module and the evidence vector will assist the reputation function to assign the reputation score to the domain $d$.**

network-based cluster and also share similar zone-based features. To better understand how the zone-based clustering works, consider the following examples of zone-based clusters:

**Cluster 1**:

..., **72.247.176.81 e55.g.akamaiedge.net,** 72.247.176.94 e68.g.akamaiedge.net, 72.247.176.146 e120.g.akamaiedge.net, 72.247.176.65 e39.na.akamaiedge.net, 72.247.176.242 e216.g.akamaiedge.net, 72.247.176.33 e7.g.akamaiedge.net, 72.247.176.156 e130.g.akamaiedge.net, 72.247.176.208 e182.g.akamaiedge.net, 72.247.176.198 e172.g.akamaiedge.net, 72.247.176.217 e191.g.akamaiedge.net, 72.247.176.200 e174.g.akamaiedge.net, 72.247.176.99 e73.g.akamaiedge.net, 72.247.176.103 e77.g.akamaiedge.net, 72.247.176.59 e33.c.akamaiedge.net, 72.247.176.68 e42.gb.akamaiedge.net, 72.247.176.237 e211.g.akamaiedge.net, 72.247.176.71 e45.g.akamaiedge.net, 72.247.176.239 e213.na.akamaiedge.net, 72.247.176.120 e94.g.akamaiedge.net, ...

**Cluster 2**:

..., **90.156.145.198 spzr.in,** 90.156.145.198 vwui.in, 90.156.145.198 x9e.ru, 90.156.145.50 v2802.vps.masterhost.ru, 90.156.145.167 www.inshaker.ru, 90.156.145.198 x71.ru, 90.156.145.198 c3q.at, 90.156.145.198 ltkq.in, 90.156.145.198 x7d.ru, 90.156.145.198 zdlz.in, 90.156.145.159 www.designcollector.ru, 90.156.145.198 x7o.ru, 90.156.145.198 q5c.ru, 90.156.145.159 designtwitters.com, 90.156.145.198 u5d.ru, 90.156.145.198 x9d.ru, 90.156.145.198 xb8.ru, 90.156.145.198 xg8.ru, 90.156.145.198 x8m.ru, 90.156.145.198 shopfilmworld.cn, 90.156.145.198 bigappletopworld.cn, 90.156.145.198 uppd.in, ...

Each element of the cluster is a *domain name - IP address* pair. These two groups of domains belonged to the same network cluster, but were separated into two different clusters by the zone-based clustering phase. *Cluster 1* contains domain names belonging to Akamai's CDN, while the domains in *Cluster 2* are all related to malicious websites that distribute malicious software. The two clusters of domains share similar network characteristics, but have significantly different zone-based features. For example, consider domain names $d_1$="e55.g.akamaiedge.net" from the first cluster, and $d_2$="spzr.in" from the second cluster. The reason why $d_1$ and $d_2$ were clustered in the same network-based cluster is because the set of RHIPs (see Section 3.1) for $d_1$ and $d_2$ have similar characteristics. In particular, the *network agility* properties of $d_2$ make it look like if it was part of a large CDN. However, when we consider the set of RHDNs for $d_1$ and $d_2$, we can notice that the zone-based features of $d_1$ are much more "stable" than the zone-based features of $d_2$. In other words, while the RHDNs of $d_1$ share strong domain name similarities (e.g., they all share the substring "akamai") and have low variance of the *string features* (see Section 3.2.2), the strong *zone agility* properties of $d_2$ affect the zone-based features measured on $d_2$'s RHDNs and make $d_2$ look very different from $d_1$.

One of the main advantages of Notos is the reliable assignment of low reputation scores to domain names participating in "agile" malicious campaigns. Less agile malicious campaigns, e.g., Fake AVs campaigns may use domain names structured to resemble CDN related domains. Such strategies would not be beneficial for the FakeAV campaign, since domains like `virus-scan1.com`, `virus-scan2.com`, etc., can be trivially blocked by using simple regular expressions [16]. In other words, the attackers need to introduce more "agility" at both the network and domain name level in order to avoid simple domain name blacklisting. Notos would only require a few labeled domain names belonging to the malicious campaign for training purposes, and the reputation engine would then generalize to assign a low reputation score to the remaining (previously unknown) domain names that belong to the same malicious campaign.

### 3.3.4 Building the Reputation Function

Once we build a model of well-known network profiles (see Section 3.3.2) and the domain clusters (see Section 3.3.3), we can build the reputation function. The reputation function will assign a reputation score in the interval $[0, 1]$ to domain names, with 0 meaning low reputation (i.e., likely malicious) and 1 meaning high reputation (i.e., likely legitimate). We implement our reputation function as a statistical classifier. In order to train the reputation function, we consider all the domain

names $d_i, i = 1, .., n$ in Notos' *knowledge base*, and we feed each domain $d_i$ to the *network profiles* module and to the *domain clusters* module to compute two output vectors $NM(d_i)$ and $DC(d_i)$, respectively. We explain the details of how $NM(d_i)$ and $DC(d_i)$ are computed later in Section 3.3.5. For now it sufficient to consider $NM(d_i)$ and $DC(d_i)$ as two feature vectors. For each $d_i$ we also compute an *evidence features* vector $EV(d_i)$, as described in Section 3.2.3. Let $v(d_i)$ be a feature vector that combines the $NM(d_i)$, $DC(d_i)$, and $EV(d_i)$ feature vectors. We train the reputation function using the labeled dataset $L = \{(v(d_i), y_i)\}_{i=1..n}$, where $y_i = 0$ if $d_i$ is a known malicious domain name, otherwise $y_i = 1$.

#### 3.3.5 On-Line Mode

After training is complete; the reputation engine can be used in on-line mode (Figure 3) to assign a reputation score to new domain names. For example, given an input domain name $d$, the reputation engine computes a score $S \in [0, 1]$. Values of $S$ close to zero mean that $d$ appears to be related to malicious activities and therefore has a low reputation. On the other hand, values of $S$ close to one signify that $d$ appears to be associated with benign Internet services, and therefore has a high reputation. The reputation score is computed as follows. First, $d$ is fed into the *network profiles* module, which consists of five statistical classifiers, as discussed in Section 3.3.2. The output of the *network profiles* module is a vector $NM(d) = \{c_1, c_2, ..., c_5\}$, where $c_1$ is the output of the first classifier, and can be viewed as the probability that $d$ belongs to the class of *Popular Domains*, $c_2$ is the probability that $d$ belongs to the class of *Common Domains*, etc. At the same time, $d$ is fed into the *domain clusters* module, which computes a vector $DC(d) = \{l_1, l_2, ..., l_5\}$. The elements $l_i$ of this vector are computed as follows. Given $d$, we first extract its network-based features and identify the closest network-based cluster to $d$, among the network-based clusters computed by the *domain clusters* module during the off-line mode (see Section 3.3.3). Then, we extract the zone-based statistical features and identify the zone-based cluster closest to $d$. Let this closest domain cluster be $C_d$. At this point, we consider all the zone-based feature vectors $v_j \in C_d$, and we select the subset of vectors $V_d \subseteq C_d$ for which the two following conditions are verified: i) $dist(z_d, v_j) < R$, where $z_d$ is the zone-based feature vector for $d$, and $R$ is a predefined *radius*; ii) $v_j \in KNN(z_d)$, where $KNN(z_d)$ is the set of $k$ nearest-neighbors of $z_d$.

The feature vectors in $V_d$ are related to domain names extracted from Notos' *knowledge base*. Therefore, we can assign a label to each vector $v_i \in V_d$, according to the nature of the domain name $d$ from which $v_i$ was computed. The domains in Notos' *knowledge base* belong to different classes. In particular, we distinguish between eight different classes of domains, namely *Popular Domains*, *Common Domains*, *Akamai*, *CDN*, and *Dynamic DNS*, which have the same meaning as explained in Section 3.3.2, and *Spam Domains*, *Flux Domains*, and *Malware Domains*.

In order to compute the output vector $DC(d)$, we compute the following five statistical features: the *majority class* label $L$ (e.g., $L$ may be equal to *Malware Domain*), i.e., the label that appears the most among the vectors $v_i \in V_d$; the standard deviation of label frequencies, i.e., given the occurrence frequency of each label among the vectors $v_i \in V_d$ we compute their standard deviation; given the subset $V_d^{(L)} \subseteq V_d$ of vectors in $V_d$ that are associated with label $L$, we compute the *mean*, *median* and *standard deviation* of the distribution of distances between $z_d$ and the vectors $v_j \in V_d^{(L)}$.

#### 3.3.6 Assigning Reputation Scores

Given a domain $d$, once we compute the vectors $NM(d)$ and $DC(d_i)$ as explained above, we also compute the evidence vector $EV(d)$ as explained in Section 3.2.3. At this point, we concatenate these three feature vectors into a sixteen dimensional feature vector $v(d)$, and we feed $v(d)$ in input to our *trained* reputation function (see Section 3.3.4). The reputation function computes a score $S = 1 - f(d)$, where $f(d)$ can be interpreted as the probability that $d$ is a malicious domain name. $S$ varies in the $[0, 1]$ interval, and the lower the value of $S$, the lower $d$'s reputation.

## 4 Data Collection and Analysis

This section summarizes observations from passive DNS measurements, and how professional, legitimate DNS services are distinguished from malicious services. These observations provided the ground truth for our dynamic domain name reputation system. We also provide an intuitive example to illustrate these properties, using a few major Internet zones like Akamai and Google.

### 4.1 Data Collection

The basic building block for our dynamic reputation rating system is the historical or "passive" information from successful `A-type` DNS resolutions. We use the DNS traffic from two ISP-based sensors, one located on the US east coast (Atlanta) and one located on the US west coast (San Jose). Additionally we use the aggregated DNS traffic from the different networks covered by the SIE [3]. In total, our database collected 27,377,461 unique resolutions from all these sources over a period of 68 days, from $19^{th}$ of July 2009 to $24^{th}$ September 2009.

Simple measurements performed on this large data set demonstrate a few important properties leveraged by our selected features. After just a few days the rate of new, unique pDNS entries leveled off. The graph in Figure 7(b) shows only about 100,000 to 150,000 new domains/day (with a brief outage issue on the $53^{rd}$ day), despite very large numbers of

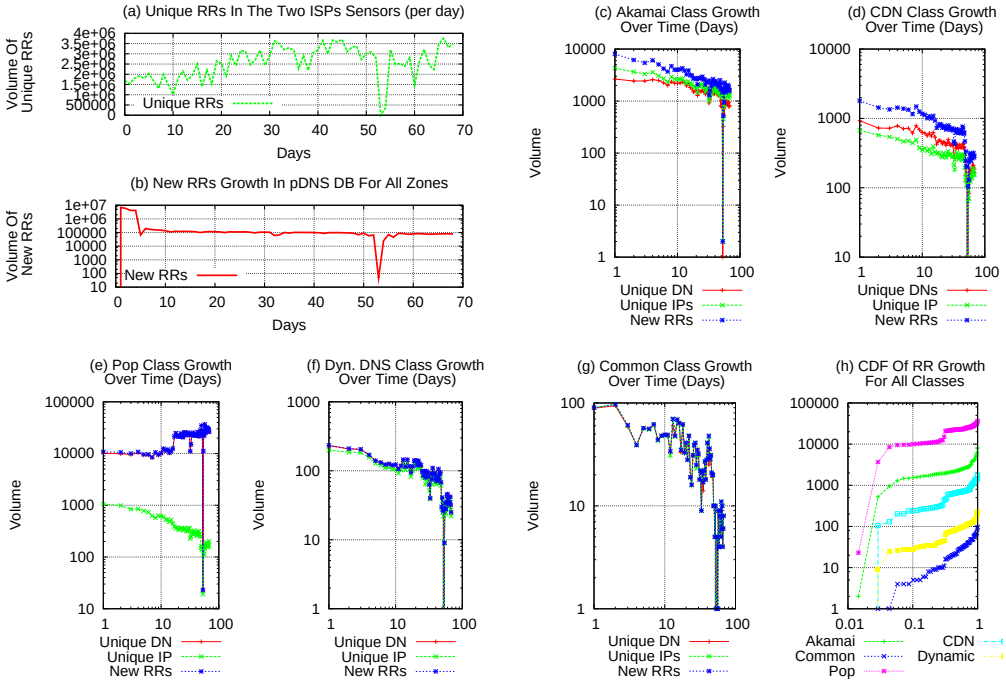


**Figure 7. Various RRs growth trends observed in the pDNS DB over a period of 68 days**

RRs arriving each day (shown in Figure 7(a)). This suggests that most RRs are duplicates, and approximately after the first few days, 94.7% – on average – from the unique RRs observed in daily base at the sensor level are already recorded by the passive DNS database. Therefore, even a relatively small pDNS database may be used to deploy Notos. In Section 5, we measure the sensitivity of our system to traffic collected from smaller networks.

The remaining plots in Figure 7 show the daily growth of our passive DNS database, from the point of view of five different zone classes. Figure 7(c) and (d) show the growth rate associated with CDN networks (Akamai, and all other CDNs). The number of unique IPs stays nearly constant with the number of unique domains (meaning that each new RR is a new IP and a new child domain of the CDN). In a few weeks, most of the IPs became known—suggesting that one can fully map CDNs in a modest training set. This is because CDNs, although large, always have a fixed number of IP addresses used for hosting their high-availability services. Intuitively, we believe this would not be the case with malicious CDNs (e.g., flux networks), which use randomly spreading infections to continually recruit new IPs.

The ratio of new IPs to domains diverges in Figure 7(e), a plot of the rate of newly discovered RRs for popular websites (e.g., Google, Facebook). Facebook notably uses unique child domains for their Web-based chat client, and other top Internet sites use similar strategies (encoding information in the domain, instead of the URI), which explains the growth in domains shown in Figure 7(e). These popular sites use a very small number of IPs, however, and after a few weeks of training our pDNS database identified all of them. Since these popular domains make up a large portion of traffic in any trace, our intuition is that simple whitelisting would significantly reduce the workload of a classifier.

Figure 7(f) shows the rate of pDNS growth for zones in Dynamic DNS providers. These services, sometimes used by botmasters, demonstrate a nearly matched ratio of new IPs to new domains. The data excludes non-routable answers (e.g., dynamic DNS domains pointing to 127.0.0.1), since this contains no unique network information. Intuitively, one can think of dynamic DNS as a nearly complete bijection of domains to IPs. Figure 7(g) shows the growth of RRs for `alexa.com` top 100 domains. Unlike dynamic DNS domains, these points to a small set of unique addresses, and most can be identified in a few weeks' worth of training.

A comparison of all the zone classes appears in Figure 7(h), which shows the cumulative distribution of the unique RRs detailed in Figure 7(c) through (g). The different rates of change illustrate how each zone class has a distinct pattern of RR use: some have a small IP space and highly variable domain names; some pair nearly every new domain with a new IP. Learning approximately 90% of all the unique RRs in each zone class, however, only requires (at most) tens of thousands of distinct RRs. The intuition from this plot is that, despite the very large

data set we used in our study, Notos could potentially work with data observed from much smaller networks.

### 4.2 Building The Ground Truth

To establish ground truth, we use two different labeling processes. First, we assigned labels to RRs at the time of their discovery. This provided an initial static label for many domains. Blacklists, of course, are never complete and always dynamic. So our second labeling process took place during evaluation, and monitored several well-known domain blacklists and whitelists.

The data we used for labeling came from several sources. Our primary source of blacklisting came from services such as `malwaredomainlist.com` and `malwaredomains.com`. In order to label IP addresses in our pDNS database we also used the Sender Policy Block (SBL) list from Spamhaus [18]. Such IPs are either known to send spam or distribute malware. We also collected domain name and IP blacklisting information from the Zeus tracker [30]. All this blacklisting information was gathered before the first day of August 2009 (during all the 15 days in which we collected passive DNS data). Since blacklists traditionally lag behind the active threat, we continued to collect all new data until the end of our experiments.

Our limited whitelisting was derived from the top 500-`alexa.com` domain names, as of the $1^{st}$ of August 2009. We reasoned that, although some malicious domains become popular, they do not stay popular (because of remediation), and never break into the top tier of domain rankings. Likewise, we used a list of the 18 most common 2LDs from various CDNs, which composed the main corpus of our CDN labeled RRs. Finally a list of 464 dynamic DNS second level domains allowed us to identify and label domain name and IPs coming from zones under dynamic DNS providers. We label our evaluation (or testing) data-set by aggregating updated blacklist information for new malicious domain names and IPs from the same lists.

To compute the honeypot features (presented in Section 3.2.3) we need a malware analysis infrastructure that can process as many "new" malware samples as possible. Our honeypot infrastructure is similar to "Ether" [4] and is capable of processing malware samples in a queue. Every malware sample was analyzed in a controlled environment for a time period of five minutes. This process was repeated during the last 15 days of July 2009. After 15 days of executions we obtained a set of successful DNS resolutions (domain names and IPs) that each malware looked up. We chose to execute malware and collect DNS evidence through the same period of time in which we aggregate the passive DNS database. Our virtual machines are equipped with five popular commercial anti-virus engines. If one of the engines identifies an executable as malicious, we capture all domain names and the corresponding IP mappings that the malware used during execution. After excluding all domain names that belong to the top 500 most popular `alexa.com` zones, we assemble the main corpus of our "honeypot data". We automated the crawling and collection of black list information and honeypot execution.

The reader should note that we chose to label our data in as transparent way as possible. We used public blacklisting information to label our training dataset before we build our models and train the reputation function. Then we assigned the reputation scores and validated the results again using the same publicly available blacklist sources. It is safe to assume that private IP and DNS blacklist will contain significant more complete information with lower FP rates than the public blacklists. By using such type of private blacklist the accuracy of Notos' reputation function should improve significantly.

## 5 Results

In this section, we present the experimental results of our evaluation. We show that Notos can identify malicious domain names sooner than public blacklists, with a low false positive rate (FP%) of 0.38% and high true positive rate (TP%) of 96.8%. As a first step, we computed vectors based on the statistical features (described in Section 3.2) from 250,000 unique RRs. This volume corresponds to the average volume of new – previously unseen – RRs observed at two recursive DNS servers in a major ISP in one day, as noted in Section 4, Figure 7(b). These vectors were computed based on historic passive DNS information from the last two weeks of DNS traffic observed on the same two ISP recursive resolvers in Atlanta and San Jose.

### 5.1 Accuracy of Network Profile Modeling

The accuracy of the Meta-Classification system (Figure 4(a)) in the network profile module is critical for the overall performance of Notos. This is because, in the on-line mode, Notos will receive unlabeled vectors which must be classified and correlated with what is already present in our knowledge base. For example, if the classifier receives a new RR and assigns to it the label Akamai with very high confidence, that implies the RR which produced this vector will be part of a network similar to Akamai. However, this does not necessarily mean that it is part of the actual Akamai CDN. We will see in the next section how we can draw conclusions based on the proximity between labeled and unlabeled RRs within the same zone-based clusters. Furthermore, we discuss the accuracy of the Meta-Classifier when modeling each different network profile class (profile classes are described in Section 3.3.2).

Our Meta-Classifier consists of five different classifiers, one for each different class of domains we model. We chose to use a Meta-Classification system instead of a traditional single classification approach because Meta-Classification systems typically perform better than a single statistical classi-

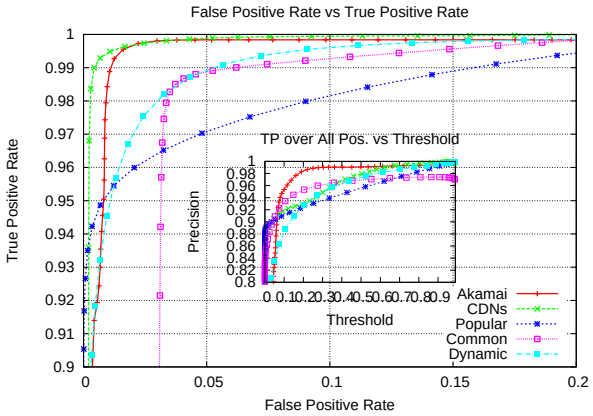


**Figure 8. ROC curves for all network profile classes shows the Meta-Classifier's accuracy.**

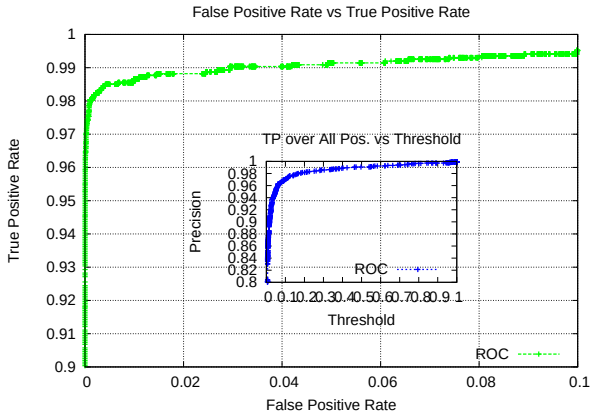


**Figure 9. The ROC curve from the reputation function indicating the high accuracy of Notos.**

fier [11, 2]. Throughout our experiments this proved to be also true. The ROC curve in Figure 8, shows that the Meta-Classifier can accurately classify RRs for all different network profile classes.

The training dataset for the Meta-Classifier is composed of sets of 2,000 vectors from each of the five network profile classes. The evaluation dataset is composed of 10,000 vectors, 2,000 from each of the five network profile classes. The classification results for the domains in the Akamai, CDN, dynamic DNS and Popular classes showed that the supervised learning process in Notos is accurate, with the exception of a small number of false positives related to the Common class (3.8%). After manually analyzing these false positives, we concluded that some level of confusion between the vectors produced by Dynamic DNS domain names and the vectors produced by domain names in the Common class still remains. However, this minor misclassification between network profiles does not significantly affect the reputation function. This is because the zone profiles of the Common and Dynamic DNS domain names are significantly different. This difference in the zone profiles will drive the network-based and zone-based clustering steps to group the RRs from Dynamic DNS class and Common class in different zone-based clusters.

Despite the fact that the network profile modeling process provides accurate results, it doesn't mean this step can independently designate a domain as benign or malicious. The clustering steps will assist Notos to group vectors not only based their network profiles but also based on their zone properties. In the following section we show how the network and zone profile clustering modules can better associate similar vectors, due to properties of their domain name structure.

## 5.2 Network and Zone-Based Clustering Results

In the domain name clustering process (Section 3.3.3, Figure 4(b)) we used X-Means clustering in series, once for the network-based clustering and again for the zone-based clustering. In both steps we set the minimum and maximum number of clusters to one and the total number of vectors in our dataset, respectively. We run these two steps using different numbers of zone and network vectors. Figure 11 shows that after the first 100,000 vectors are used, the number of network and zone clusters remains fairly stable. This means that by computing at least 100,000 network and zone vectors—using a 15-day old passive DNS database—we can obtain a stable population of zone and network based clusters for the monitored network. We should note that reaching this network and cluster equilibrium does not imply that we do not expect to see any new type of domain names in the ISP's DNS recursive. This just denotes that based on the RRs present in our passive DNS database, and the daily traffic at the ISP's recursive, 100,000 vectors are enough to reflect the major network profile trends in the monitored networks. Figure 11 indicates that a sample set of 100,000 vectors may represent the major trends in a DNS sensor. It is hard to safely estimate the exact minimum number of unique RRs that is sufficient to identify all major DNS trends. An answer to this should be based upon the type, size and utilization of the monitored network. Without data from smaller corporate networks it is difficult for us to make a safe assessment about the minimum number of RR necessary for reliably training Notos.

The evaluation dataset we used consisted of 250,000 unique domain names and IPs. The cluster overview is shown in Figure 10 and in the following paragraphs we discuss some in-

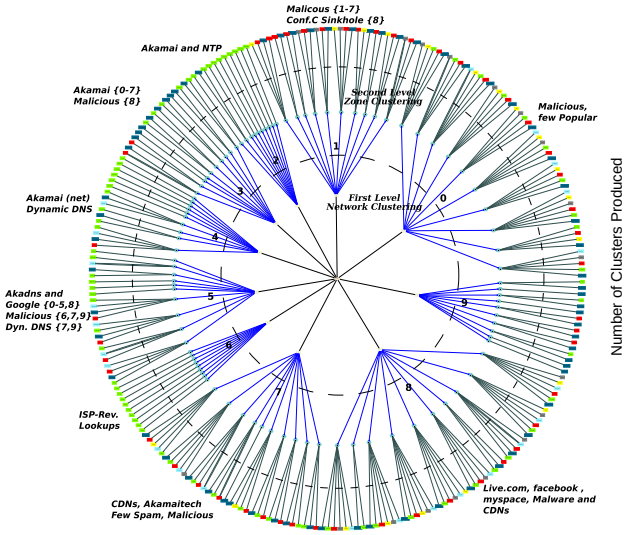


**Figure 10. With the 2-step clustering step, Notos is able to cluster large trends of DNS behavior.**

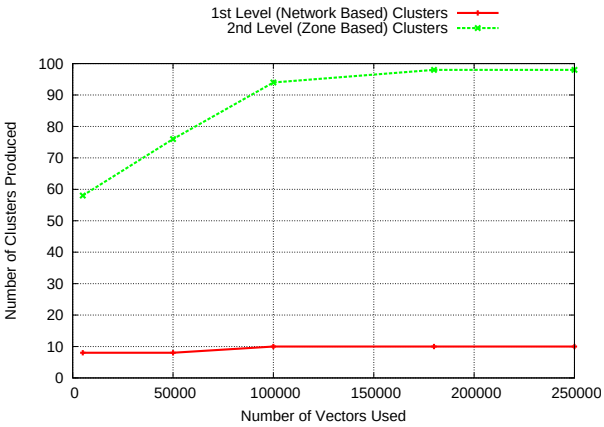


**Figure 11. By using different number of network and zone vectors we observe that after the first 100,000, there is no significant variation in the absolute number of produced clusters during the $1^{st}$ and $2^{nd}$ level clustering steps.**

teresting observations that can be made from these network-based and zone-based cluster assignments. As an example, network clusters 0 and 1 are predominantly composed of zones participating in fraudulent activities like spam campaigns (yellow) and malware dropping or C&C zones (red). On the other hand, network clusters 2 to 5 contain Akamai, dynamic DNS, and popular zones like Google, all labeled as benign (green). We included the unlabeled vectors (blue) based on which we evaluated the accuracy of our reputation function. We have a sample of unlabeled vectors in almost all network and zone clusters. We will see how already labeled vectors will assist us to characterize the unlabeled vectors in close proximity.

Before we describe two sample cases of dynamic characterization within zone-based clusters, we need to discuss our radius $R$ and $k$ value selection (see Section 3.3.5). In Section 3.3.5, we discuss how we build domain name clusters. At that point we introduced the dynamic characterization process that gives Notos the ability to utilize already label vectors in order to characterize a newly obtained unlabeled vector by leveraging our prior knowledge. After looking into the distribution of Euclidean distances between unlabeled and labeled vectors within the same zone clusters, we concluded that in the majority of these cases the distances were between 0 and 1000. We tested different values of the radius $R$ and the value of $k$ for the K-nearest neighbors (KNN) algorithm. We observed that the experiments with radius values between 50 and 200 provided the most accurate reputation rating results, which we describe in the following sections. We also observed that if $k > 25$ the accuracy of the reputation function is not affected for all radius values between 50 and 200. Based on the results of these pilot experiments, we decided to set $k$ equal to 50 and the radius distance equal to 100.

Figures 12 and 13 show the effect of this radius selection on two different types of clustering problems. In Figure 12, unknown RRs for `akamaitech.net` are clustered with a labeled vector `akamai.net`. As noted in Section 4, CDNs such as Akamai tended to have new domain names with each RR, but to also reuse their IPs. By training with only a small set of labeled `akamai.net` RRs, our classifier put the new, unknown RRs for `akamaitech.net` into the existing Akamai class. IP-specific features therefore brought the new RRs close to the existing labeled class. Figure 12 compresses all of the dimensions into a two-dimensional plot (for easier visual representation), but it is clear the unknown RRs were all within a distance of 100 to the labeled set.

This result validates the design used in Section 4, where just a few weeks' worth of labeled data was necessary for training. Thus, one does not have to exhaustively discover all whitelisted domains. Notos is resilient to changes in the zone classes we selected. Services like CDNs and major web sites can add new IPs or adjust domain formats, and these will be *automatically* associated with a known labeled class.

The ability of Notos to associate new RRs based on limited labeled inputs is demonstrated again in Figure 13. In this case, labeled Zeus domains (approximately 2,900 RRs from three different Zeus-related BLs) were used to classify new RRs. Figure 13 plots the distance between the labeled Zeus-related RRs and new (previously unknown) RRs that are also related Zeus botnets. As we can see from Section 4, most of the new (unlabeled) Zeus RRs lay very

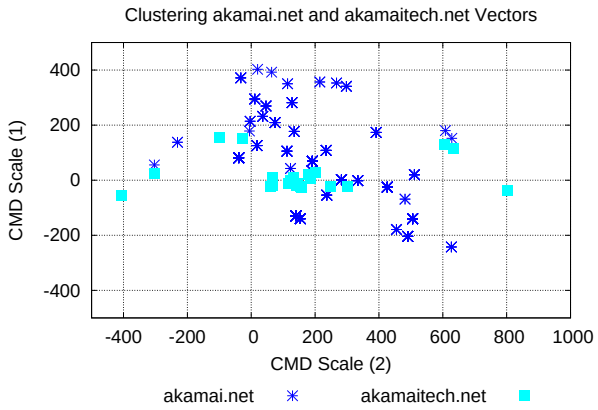


**Figure 12. An example of characterizing the akamaitech.net unknown vectors as benign based on the already labeled vectors (akamai.net) present in the same cluster.**

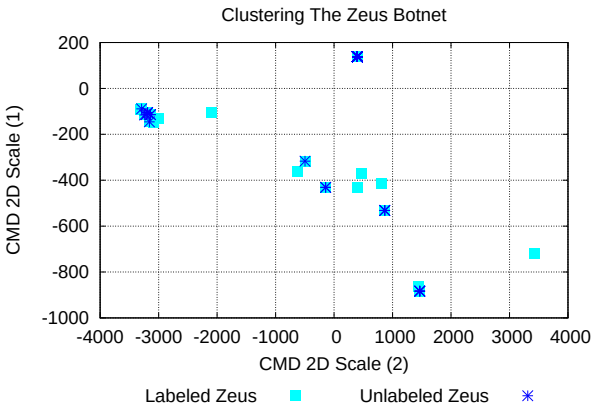


**Figure 13. An example of how the Zeus botnet clusters during our experiments. All vectors are in the same network cluster and in two different zone clusters.**

close, and often even overlap, to known Zeus RRs. This is a good result, because Zeus botnets are notoriously hard to track, given the botnet's extreme agility. Tracking systems such as `zeustracker.abuse.ch` and `malware-domainlist.com` have limited visibility into the botnet, and often produce disjoint blacklists. Notos addresses this problem, by leveraging a limited amount of training data to correctly classify new RRs. During our evaluation set, Notos correctly detected 685 new (previously unknown) Zeus RRs.

### 5.3 Accuracy of the Reputation Function

The first thing that we address in this section is our decision to use a Decision Tree using Logit-Boost strategy (LAD) as the reputation function. Our decision is motivated by the time complexity, the detection results and the precision (true positives over all positives) of the classifier. We compared the LAD classifier to several other statistical classifiers using a typical model selection procedure [6]. LAD was found to provide the most accurate results in the shortest training time for building the reputation function. As we can see from the ROC curve in Figure 9, the LAD classifier exhibits a low false positive rate (FP%) of 0.38% and true positive rate (TP%) of 96.8%. It is was noting that these results were obtained using 10-fold cross-validation, and the detection threshold was set to 0.5. The dataset using for the evaluation contained 10,719 RRs related to 9,530 *known bad* domains. The list of *known good* domains consisted of the top 500 most popular domains according to Alexa.

We also benchmarked the reputation function on other two datasets containing a larger number of *known good* domain names. We experimented with bot the top 10,000 and top 100,000 Alexa domain names. The detection results for these experiments are as follows. When using the top 10,000 Alexa domains, we obtained a true positive rate of 93.6% and a false positive rate of 0.4% (again using 10-fold cross-validation and a detection threshold equal to 0.5). As we can see, these results are not very different from the ones we obtained using only the top 500 Alexa domains. However, when we extended our list of *known good* domains to include the top 100,000 Alexa domain names, we observed a significant decrease of the true positive rate and an increase in the false positives. Specifically, we obtained a TP% of 80.6% and a FP% of 0.6%. We believe this degradation in accuracy may be due to the fact that the top 100,000 Alexa domains include not only professionally run domains and network infrastructures, but also include less good domain names, such as file-sharing, porn-related websites, etc., most of which are not run in a professional way and have disputable reputation[1].

We also wanted to evaluate how well Notos performs, compared to static blacklists. To this end, we performed a number of experiments as follows. Given an instance of Notos trained with data collected up to July 31, 2009, we fed Notos with 250,000 distinct RRs found in DNS traffic we collected on August 1, 2009. We then computed the reputation score for each of these RRs. First, we set the detection threshold to 0.5, and with this threshold we identified 54,790 RRs that had a low reputation (lower than the threshold). These RRs where

[1] A quick analysis of the top 100,000 Alexa domains reported that about 5% of the domains appeared in the SURBL (`www.surbl.org`) blacklist, at certain point in time. A more rigorous evaluation of these results is left to future work.

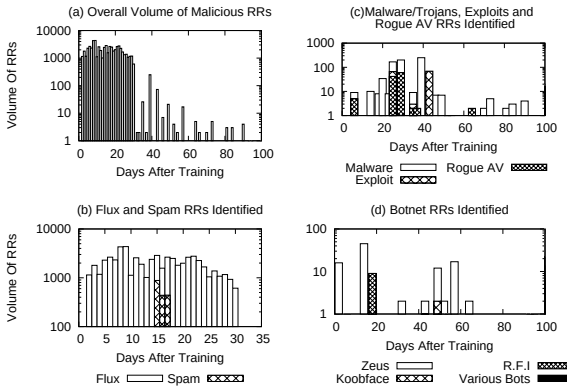


**Figure 14. Dates in which various blacklists confirmed that the RRs were malicious after Notos assigned low reputation to them on the $1^{st}$ of August.**

related to a total of 10,294 distinct domain names (notice that a domain name may map to more than one IP, and this explains the higher number of RRs). Of these 10,294 domains, 7,984 (77.6%) appeared in at least one of the public blacklists we used for comparison (see Section 4) within 60 day after August 1, and were therefore confirmed to be malicious. Figure 14(a) reports the number and date in which RRs classified as having low reputation by Notos appeared in the public blacklists. The remaining three plots (Figure 14(b), (c) and (d)), report the same results organized according to the type of malicious domains. In particular, it is worth noting that Notos is able to detect never-before-seen domain names related to the Zeus botnet several days or even weeks before they appeared in any of the public blacklists.

For the remaining 22.4% of the 10,294 domains we considered, we were not able to draw a definitive conclusion. However, we believe many of those domains are involved in some kind of more or less malicious activities. We also noticed that 7,980 or the 7,984 *confirmed bad* domain names were assigned a reputation score lower or equal to 0.15, and that none of the other non-confirmed suspicious domains received a score lower than this threshold. In practice, this means that an operator who would like to use Notos as a stand-alone dynamic blacklisting system while limiting the false positives to a negligible (or even zero) amount may fine-tune the detection threshold and set it around 0.15.

### 5.4 Discussion

This section discusses the limits of Notos, and the potential for evasion in real networks. On of the main limitations is the fact that Notos is unable to assign reputation scores for domain names with very little historic (passive DNS) information. Sufficient time and a relatively large passive DNS collection are required to create an accurate passive DNS database. Therefore, if an attacker always buys new domain names and new address space, and never reuses either resource for any other malicious purposes, Notos will not be able to accurately assign a reputation score to the new domains. In the IPv4 space, this is very unlikely to happen due to the impending exhaustion of the available address space. Once IPv6 becomes the predominant protocol, however, this may represent a problem for the statistical features we extract based on IP granularity. However, we believe the features based on BGP prefixes and AS numbers would still be able to capture the agility typical of malicious DNS hosting behavior.

As long as newly generated domain names share some network properties (e.g., IPs or BGP prefixes) with already labeled RRs, Notos will be able to assign an accurate reputation score. In particular, since network resources are finite and more expensive to renew or change, even if the domain properties change, Notos can still identify whether a domain name may be associated with malicious behavior. In addition, if a given domain name for which we want to know the reputation is not present in the passive DNS DB, we can actively probe it, thus forcing a related passive DNS entry. However, this is possible only when the domain successfully maps to a non-empty set of IPs.

Our experimental results using the top 10,000 Alexa domain names as *known good* domains, report a false positive fate of 0.4%. While low in percentage, the absolute number of false positives may become significant in those cases in which very large numbers of new domain names are fed to Notos on a daily bases (e.g., in case of deployment in a large ISP network). However, we envision our Notos reputation system to be use not as a stand-alone system, but rather in cooperation with other defense mechanisms. For example, Notos may be used in collaboration with spam-filtering system. If an email contains a link to a website whose domain name has a low reputation score according to Notos, the spam filter can increase the total spam-score of the email. However, if the rest of the email appears to be benign, the spam filter may still decide to accept the email.

During our manual analysis of (a subset of) the false positives encountered in our evaluations we were able to draw some interesting observation. We found that a number of legitimate sites (e.g., goldsgym.com) are being hosted in networks that host large volumes of malicious domain names in them. In this cases Notos will tend to penalize the reputation of this legitimate domains because they reside in a *bad neighborhood*. In time, the reputation score assigned to these domains score may change, if the administrators of the network in which the benign domain name are hosted take actions to "clean up" their networks and stop hosting bad domain names within their address space.

| Domain Name | IP | Date |
|---|---|---|
| google-bot004.cn | 213.182.197.229 | 08-15 |
| analf.net | 222.186.31.169 | 08-15 |
| pro-buh.ru | 89.108.67.83 | 08-15 |
| ammdamm.cn | 92.241.162.55 | 08-15 |
| briannazfunz.com | 95.205.116.55 | 08-15 |
| mybank-of.com | 59.125.229.73 | 08-15 |
| oc00co.com | 212.117.165.128 | 08-15 |
| avangadershem.com | 195.88.190.29 | 08-19 |
| securebizccenter.cn | 122.70.145.140 | 08-19 |
| adobe-updating-service.cn | 59.125.231.252 | 09-02 |
| 0md.ru | 219.152.120.118 | 09-19 |
| avrev.info | 98.126.15.186 | 09-27 |
| g00glee.cn | 218.93.202.100 | 09-02 |

**Table 1. Sample cases form Zeus domains detected by Notos and the corresponding days that appeared in the public BLs. All evidence information in this table were harvested from zeustracker.abuse.ch.**

| Domain Name | IP | Type | Src | Date |
|---|---|---|---|---|
| lzwn.in | 94.23.198.97 | MAL | [1] | 08-26 |
| 3b9.ru | 213.251.176.169 | MAL | [2] | 08-30 |
| antivirprotect.com | 64.40.103.249 | RAV | [3] | 09-05 |
| 1speed.info | 212.117.163.165 | CWS | [2] | 09-05 |
| spy-destroyer.com | 67.211.161.44 | CWS | [4] | 09-05 |
| free-spybot.com | 63.243.188.110 | RAV | [2] | 09-05 |
| a3l.at | 89.171.115.10 | MAL | [2] | 09-09 |
| gidromash.cn | 211.95.79.170 | BOT | [2] | 09-13 |
| iantivirus-pro.com | 188.40.52.180 | KBF | [5] | 09-19 |
| ericwanhouse.cn | 220.196.59.19 | EXP | [6] | 09-22 |
| 1165651291.com | 212.117.165.126 | RAV | [2] | 10-06 |

**Table 2. Anecdotal cases of malicious domain names detected by Notos and the corresponding days that appeared in the public BLs .**[1]: hosts-file.net, [2]: malwareurl.com, [3] siteadvisor.com, [4] virustotal.com, [5] ddanchev.blogspot.com, [6] malwaredomainlist.com

## 6 Conclusion

In this paper, we presented Notos, a dynamic reputation system for DNS. To the best of our knowledge, Notos is the first system that can assign a dynamic reputation score to any domain name in a DNS query that traverses the edge of a monitored network. Notos harvests information from multiple sources such as the DNS zone domain names belongs to, the related IP addresses, BGP prefixes, AS information and honeypot analysis to maintain up-to-date DNS information about legitimate and malicious domain names. Based on this information, Notos uses automated classification and clustering algorithms to model network and zone behaviors of legitimate and malicious domains, and then applies these models to compute a reputation score for a (new) domain name.

Our evaluation using real-world data, which includes traffic from large ISP networks, demonstrates that Notos is highly accurate in identifying new malicious domains in the monitored DNS query traffic, with a true positive rate of 96.8% and false positive rate of 0.38%. In addition, Notos is capable of identifying these malicious domain weeks or even months before they appear in public blacklists, thus enabling proactive security countermeasures against cyber attacks.

## 7 Acknowledgments

We thank Steven Gribble, our shepherd, for helping us to improve the quality of the final version of this paper, and the anonymous reviewers for their constructive comments. We also thank Gunter Ollmann and Robert Edmonds for their valuable comments. Additionally, we thank the Internet Security Consortium Security Information Exchange project (ISC@SIE) for providing portion of the DNS data used in our experiments.

This material is based upon work supported in part by the National Science Foundation under grant no. 0831300, the Department of Homeland Security under contract no. FA8750-08-2-0141, the Office of Naval Research under grants no. N000140710907 and no. N000140911042. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Department of Homeland Security, or the Office of Naval Research.

## References

[1] D. Anderson, C. Fleizach, S. Savage, and G. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In *Proceedings of the USENIX Security Symposium*, 2007.

[2] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[3] Internet Systems Consortium. SIE@ISC : Security Information Exchange. `https://sie.isc.org/`, 2004.

[4] A. Dinaburg, R. Royal, M. Sharif, and W. Lee. Ether: malware analysis via hardware virtualization extensions. In *ACM CCS*, 2008.

[5] SORBS DNSBL. Fighting spam by finding and listing Exploitable Servers. `http://www.us.sorbs.net/`, 2007.

[6] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.

[7] M. Felegyhazi, C. Keibich, and V. Paxson. On the potential of proactive domain blacklisting. In *Third USENIX LEET Workshop*, 2010.

[8] S. Garera, N. Provos, M. Chew, and A. Rubin. A framework for detection and measurement of phishing attacks. In *Proceedings of the ACM WORM*. ACM, 2007.

[9] B. 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 09*, New York, NY, USA, 2009. ACM.

[10] T. Holz, C. Gorecki, K. Rieck, and F. Freiling. Measuring and detecting fast-flux service networks. In *Proceedings of NDSS*, 2008.

[11] T. Hothorn and B. Lausen. Double-bagging: Combining classifiers by bootstrap aggregation. *Pattern Recognition*, 36(6):1303–1309, 2003.

[12] P. Mockapetris. Domain names - concepts and facilities. `http://www.ietf.org/rfc/rfc1034.txt`, 1987.

[13] P. Mockapetris. Domain names - implementation and specification. `http://www.ietf.org/rfc/rfc1035.txt`, 1987.

[14] OPENDNS. OpenDNS — Internet Navigation And Security. `http://www.opendns.com/`, 2010.

[15] P. Porras, H. Saidi, and V. Yegneswaran. An Analysis of Conficker's Logic and Rendezvous Points. `http://mtc.sri.com/Conficker/`, 2009.

[16] R. Perdisci, W. Lee, and N. Feamster. Behavioral clustering of http-based malware and signature generation using malicious network traces. In *USENIX NSDI*, 2010.

[17] D. Plonka and P. Barford. Context-aware clustering of DNS query traffic. In *Proceedings of the 8th IMC*, Vouliagmeni, Greece, 2008. ACM.

[18] The Spamhaus Project. ZEN - Spamhaus DNSBLs. `http://www.spamhaus.org/zen/`, 2004.

[19] R. Perdisci, I. Corona, D. Dagon, and W. Lee. Detecting malicious flux service networks through passive analysis of recursive DNS traces. In *Proceedings of ACSAC*, Honolulu, Hawaii, USA, 2009.

[20] P. Royal. Analysis of the kraken botnet. `http://www.damballa.com/downloads/r_pubs/KrakenWhitepaper.pdf`, 2008.

[21] S. Hao, N. Syed, N. Feamster, A. Gray and S. Krasser. Detecting spammers with SNARE: Spatio-temporal network-level automatic reputation engine. In *Proceedings of the USENIX Security Symposium*, 2009.

[22] S. Shevchenko. Srizbi Domain Generator Calculator. `http://blog.threatexpert.com/2008/11/srizbis-domain-calculator.html`, 2008.

[23] K. Sato, K. Ishibashi, T. Toyono, and N. Miyake. Extending black domain name list by using co-occurrence relation between dns queries. In *Third USENIX LEET Workshop*, 2010.

[24] S. Sinha, M. Bailey, and F. Jahanian. Shades of grey: On the effectiveness of reputation-based blacklists. In *3rd International Conference on MALWARE*, 2008.

[25] The Honeynet Project & Research Alliance. Know Your Enemy: Fast-Flux Service Networks. `http://old.honeynet.org/papers/ff/fast-flux.html`, 2007.

[26] URIBL. Real time URI blacklist. `http://uribl.com`.

[27] F. Weimer. Passive DNS replication. In *Proceedings of FIRST Conference on Computer Security Incident*, Hand ling, Singapore, 2005.

[28] Z. Qian, Z. Mao, Y. Xie and F. Yu. On network-level clusters for spam detection. In *Proceedings of the USENIX NDSS Symposium*, 2010.

[29] B. Zdrnja, N. Brownlee, and D. Wessels. Passive monitoring of DNS anomalies. In *Proceedings of DIMVA Conference*, 2007.

[30] Zeus Tracker. Zeus IP & domain name block list. `https://zeustracker.abuse.ch`, 2009.

[31] J. Zhang, P. Porra, and J. Ullrich. Highly predictive blacklisting. In *Proceedings of the USENIX Security Symposium*, 2008.