Detecting Malware Domains at the Upper DNS Hierarchy

Manos Antonakakis[‡*], Roberto Perdisci[†], Wenke Lee[*],
Nikolaos Vasiloglou II[‡], and David Dagon[*]

[‡]Damballa Inc.
{manos,nvasil}@damballa.com
[*]Georgia Institute of Technology, School of Computer Science
wenke@cc.gatech.edu,dagon@sudo.sh
[†]University of Georgia, Department of Computer Science
perdisci@cs.uga.edu

Abstract

In recent years Internet miscreants have been leveraging the DNS to build malicious network infrastructures for malware command and control. In this paper we propose a novel detection system called Kopis for detecting malware-related domain names. Kopis passively monitors DNS traffic at the upper levels of the DNS hierarchy, and is able to accurately detect malware domains by analyzing *global* DNS query resolution patterns.

Compared to previous DNS reputation systems such as Notos [3] and Exposure [4], which rely on monitoring traffic from *local* recursive DNS servers, Kopis offers a new vantage point and introduces new traffic features specifically chosen to leverage the *global* visibility obtained by monitoring network traffic at the upper DNS hierarchy. Unlike previous work Kopis enables DNS operators to *independently* (i.e., without the need of data from other networks) detect malware domains within their authority, so that action can be taken to stop the abuse. Moreover, unlike previous work, Kopis can detect malware domains even when *no* IP reputation information is available.

We developed a proof-of-concept version of Kopis, and experimented with eight months of real-world data. Our experimental results show that Kopis can achieve high detection rates (e.g., 98.4%) and low false positive rates (e.g., 0.3% or 0.5%). In addition Kopis is able to detect new malware domains days or even weeks before they appear in public blacklists and security forums, and allowed us to discover the rise of a previously unknown DDoS botnet based in China.

1 Introduction

The Domain Name System (DNS) [17, 18] is a fundamental component of the Internet. Over the years Internet miscreants have used the DNS to build malicious network infrastructures. For example, botnets [1, 21, 27] and other types of malicious software make use of domain names to locate their command and control (C&C) servers and communicate with attackers, e.g., to exfiltrate stolen private information, wait for commands to perform attacks on other victim machines, etc. In response to this malicious use of DNS, *static* domain blacklists containing known malware domains have been used by network operators to detect DNS queries originating from malware-infected machines and block their communications with the attackers [16, 19].

Unfortunately, the effectiveness of static domain blacklists are increasingly limited because there are now an overwhelming number of new domain names appearing on the Internet every day and attackers frequently switch to different domains to run their malicious activities, thus making it difficult to keep blacklists up-to-date.

To overcome the limitations of static domain blacklists, we need a detection system that can *dynamically* detect new malware-related domains. This detection system should:

(1) Have *global visibility* into DNS request and response messages related to large DNS zones. This enables "early warning", whereby malware domains can be detected before the corresponding malware infections reach our local networks.
(2) Enable DNS operators to *independently* deploy the system and detect malware-related domains from within their authority zones without the need for data from other networks or other inter-organizational coordination. This enables practical, low-cost, and time-efficient detection and response.
(3) Accurately detect malware-related domains even in the absence of reputation data for the IP address space pointed to by the domains. IP reputation data is often difficult to accumulate and is fragile. This issue may become particularly important as IPv6 is deployed in the near future, due to the more expansive address space.

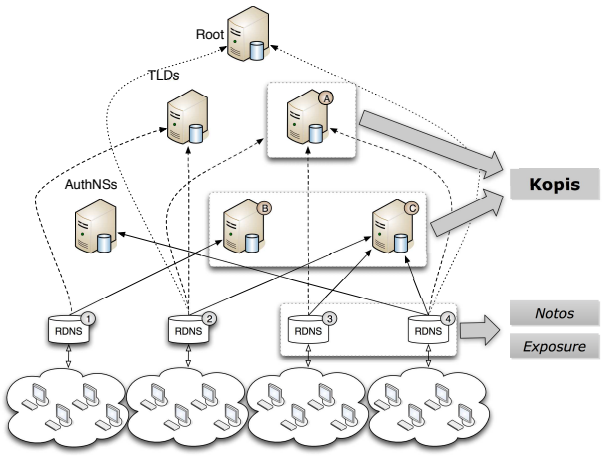


Figure 1: Overview of the levels at which Kopis, Notos, and Exposure perform DNS monitoring.

Recently researchers have proposed two dynamic domain reputation systems, Notos [3] and Exposure [4]. Unfortunately, while the results reported in [3, 4] are promising, neither Notos nor Exposure can meet all the requirements stated above, as Notos and Exposure rely on passive monitoring of recursive DNS (RDNS) traffic. As shown in Figure 1, they monitor the DNS queries from a (limited) number of RDNS servers (e.g., RDNS `3` and `4`), and have only partial visibility on DNS messages related to large DNS zones. To obtain truly global visibility into DNS traffic related to a given DNS zone, these systems need access to a very large number of RDNS sensors in many diverse locations. This is not easy to achieve in practice in part due to operational costs, privacy concerns related to sharing data across organizational boundaries, and difficulties in establishing and maintaining trust relationships between network operators located in different countries, for example. For the same reasons, Notos and Exposure have not been designed to be independently deployed and run by single DNS operators, because they rely on data sharing among several networks to obtain a meaningful level of visibility into DNS traffic.

On the other hand, monitoring DNS traffic from the upper DNS hierarchy, e.g., at top-level domain (TLD) server `A`, and authoritative name servers (AuthNSs) `B` and `C`, offers visibility on *all* DNS messages related to domains on which `A`, `B`, and `C` have authority or are a point of delegation. For example, assuming `B` is the AuthNS for the `example.com` zone, monitoring the DNS traffic at `B` provides visibility on all DNS messages from all RDNS servers around the Internet that query a domain name under the `example.com` zone.

Following this intuition, in this paper we propose a novel detection system called Kopis, which takes advantage of the global visibility available at the upper levels of the DNS hierarchy to detect malware-related domains. In order for Kopis to satisfy the three requirements outlined above, it needs to deal with a number of new challenges. Most significantly, the higher up we move in the DNS hierarchy, the stronger the effects of DNS caching [15]. As a consequence, moving up in the hierarchy restricts us to monitoring DNS traffic with a coarser granularity. For example, at the TLD level we will only be able to see a small subset of queries to domains under a certain delegation point due to the effects of the DNS cache.

Kopis works as follows. It analyzes the streams of DNS queries and responses at AuthNS or TLD servers (see Figure 1) from which are extracted statistical features such as the diversity in the network locations of the RDNS servers that query a domain name, the level of "popularity" of the querying RDNS servers (defined in detail in Section 4), and the reputation of the IP space into which the domain name resolves. Given a set of known legitimate and known malware-related domains as training data, Kopis builds a statistical classification model that can then predict whether a new domain is malware-related based on observed query resolution patterns.

Our choice of Kopis' statistical features, which we discuss in detail in Section 4, is determined by the nature of the information accessible at the upper DNS hierarchy. As a result these features are significantly different from those used by RDNS-based systems such as Notos [3] and Exposure [4]. In particular, we were pleasantly surprised to find that, while Notos and Exposure rely heavily on features based on IP reputation, Kopis' features enabled it to accurately detect malware-related domains even in the absence of IP reputation information. This may become a significant advantage in the near future because the deployment of IPv6 may severely impact the effectiveness of current IP reputation systems due to the substantially larger IP address space that would need to be monitored.

To summarize, we make the following contributions:

- We developed a novel approach to detect malware-related domain names. Our system leverages the global visibility obtained by monitoring DNS traffic at the upper levels of the DNS hierarchy, and can detect malware-related domains based on DNS resolution patterns.

- Kopis enables DNS operators to *independently* (i.e., without the need of data from other networks) detect malware-domains within their scope of authority, so that action can be taken to stop the abuse.

- We systematically examined real-world DNS traces from two large AuthNSs and a country-code level TLD server. We performed a rigorous evaluation of our statistical features and identified two new feature families that, unlike previous work, enable Kopis to detect malware domains even when no IP reputation information is available.

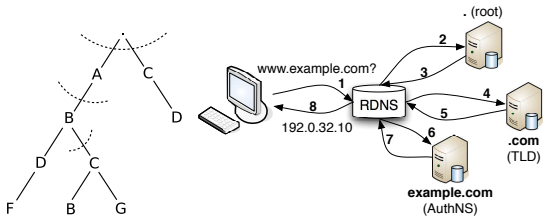
- We developed a proof-of-concept version of Kopis, and experimented with eight months of real-world data. Our experimental results show that Kopis can achieve high detection rates (e.g., 98.4%) and low false positive rates (e.g., 0.3% or 0.5%). More significantly, Kopis was able to identify previously unknown malware domain names several weeks before they appeared in blacklists or in security forums. In addition, using Kopis we detected the rise of a previously unknown DDoS botnet based in China.

2 Background and Related Work

DNS Concepts and Terminology The domain name space is structured like a tree. A domain name identifies a node in the tree. For example, the domain name `F.D.B.A.` identifies the path from the root "." to a node `F` in the tree (see Figure 2(a)). The set of resource information associated with a particular name is composed of resource records (RRs) [17, 18]. The depth of a node in the tree is sometimes referred to as *domain level*. For example, `A.` is a top-level domain (TLD), `B.A.` is a second-level domain (2LD), `D.B.A.` is a third-level domain (3LD), and so on.

The information related to the domain name space is stored in a distributed *domain name database*. The domain name database is partitioned by "cuts" made in the name space between adjacent nodes. After all cuts are made, each group of connected nodes represent a separate *zone* [17]. Each zone has at least one node, and hence a domain name, for which it is *authoritative*. For each zone, a node which is closer to the root than any other node in the zone can be identified. The name of this node is often used to identify the zone. The RRs of the nodes in a given zone are served by one or more *authoritative* name servers (AuthNSs). AuthNSs that have complete knowledge about a zone (i.e., they store the RRs for all the nodes related to the zone in question in its zone files) are said to have *authority* over that zone [17, 18]. AuthNSs will typically support one or more zones, and can delegate the authority over part of a (sub-)zone to other AuthNSs.

DNS queries are usually initiated by a *stub resolver* on a user's machine, which relies on a *recursive DNS resolver* (RDNS) for obtaining a set of RRs owned by a given domain name. The RDNS is responsible for directly contacting the AuthNSs on behalf of the stub resolver to obtain the requested information, and return it to the stub resolver. The RDNS is also responsible for caching the obtained information up to a certain period of time, called the *Time To Live* (TTL), so that if the same or another stub resolver queries again for the same information within the TTL time window, the RDNS will not need to contact the authoritative name servers (thus improving efficiency). Figure 2(b) enumerates the steps involved in a typical query resolution process, assuming an empty cache.



(a) DNS Tree (b) Domain Resolution

Figure 2: Example of DNS tree and domain resolution process.

Related Work To the best of our knowledge, Wessels et al. [30] were the first to analyze DNS query data as seen from the upper DNS hierarchy. The authors focused on examining the DNS caching behavior of recursive DNS servers from the point of view of AuthNS and TLD servers, and how different implementations of caching systems may affect the performance of the DNS.

Recently, Hao et al. [13] released a report on DNS lookup patterns measured from the `.com` TLD servers. Their preliminary analysis shows that the resolution patterns for malicious domain names are sometimes different from those observed for legitimate domains. While [13] only reports some preliminary measurement results and does not discuss how the findings may be leveraged for detection purposes, it does hint that a malware detection system may be built around TLD-level DNS queries. We designed Kopis to do just that, namely monitor query streams at the upper DNS hierarchy and be able to detect previously unknown malware domains.

Several studies provide deep understanding behind the properties of malware propagation and botnet's lifetime [7, 25, 29]. An interesting observation among all these research efforts is the inherent diversity of the botnet's infected population. Collins et al. [6] introduced and quantified the notion of "network uncleanliness"

from the temporal and spatial network point of view, showing that it is very probable to have a large number of infected bots in the same network over an epoch. They also discuss that this could be a direct effect of the network policy enforced at the edge. Kopis directly uses the intuition behind these past research efforts in the *requester diversity* and *requester profile* statistical feature families.

A number of research efforts can be found in the area of DNS blacklisting and reputation. Felegyhazi et al. [11] recently proposed a DNS reputation blacklisting methodology based on `WHOIS` information, while Antonakakis et al. [3] and Bilge et al. [4] propose dynamic reputation systems based on passive RDNS monitoring. Our system is complementary to the above mentioned works. To the best of our knowledge, we are the first to analyze DNS query patterns at the AuthNS and TLD server level for the purpose of detecting domain names related to malware.

3 System Overview

Kopis monitors streams of DNS queries to and responses from the upper DNS hierarchy, and detects malware domain names based on the observed query/response patterns. An overview of Kopis is shown in Figure 3.

Our system divides the monitored data streams into epochs $\{E_i\}_{i=1..m}$ (currently, an epoch is one day long). At the end of each epoch Kopis summarizes the DNS traffic related to a given domain name d by computing a number of statistical features, such as the diversity of the IP addresses associated with the RDNS servers that queried d, the relative volume of queries from the set of querying RDNS servers, historic information related to the IP space pointed to by d, etc. We defer a detailed description and motivations regarding the features we measure to Section 4. For now, it suffices to consider the *feature computation* module in Figure 3 as a function $\mathcal{F}(d, E_i) = v_d^i$ that maps the DNS traffic in epoch E_i related to d into a feature vector v_d^i.

Kopis operates in two modes: a *training* mode and an *operation* mode. In training mode, Kopis makes use of a *knowledge base* $\mathbf{KB}$, which consists of a set of known malware-related and known legitimate domain names (and related resolved IPs) for which the monitored AuthNS and TLD servers are authoritative or a point of delegation. Kopis' *learning module* takes as input the set of feature vectors $\mathbf{V}_{train} = \{v_d^i\}_{i=1..m}, \forall d \in \mathbf{KB}$, which summarizes the query/response *behavior* of each domain in the knowledge base across m days. Each domain in $\mathbf{KB}$, and in turn each feature vector in $\mathbf{V}_{train}$, is associated with a label, namely *legitimate* or *malware*. We can therefore use supervised learning techniques [5] to learn a statistical classification model $\mathcal{S}$ of DNS query patterns

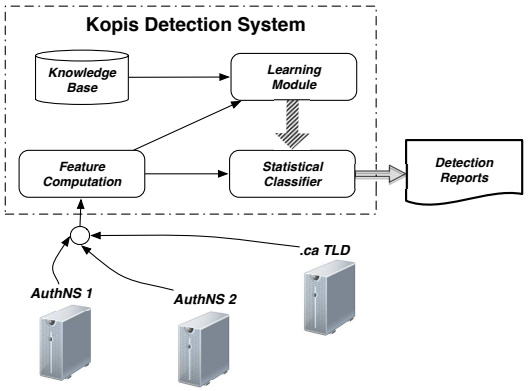


Figure 3: A high-level overview of Kopis.

related to legitimate and malware domains as seen from the upper DNS hierarchy.

In operation mode, Kopis monitors the streams of DNS traffic and, at the end of each epoch E_j, maps each domain $d' \notin \mathbf{KB}$ (i.e., all *unknown* domains) extracted from the query/response streams into a feature vector $v_{d'}^j$. At this point, given a domain d' the statistical classifier $\mathcal{S}$ (see Figure 3) assigns a label $l_{d',j}$ and a confidence score $c(l_{d',j})$, which express whether the query/response patterns observed for d' during epoch E_j resemble either known legitimate or malware behavior, and with what probability. In order to make a final decision about d', Kopis first gathers a series of labels and confidence scores $\mathcal{S}(v_{d'}^j) = \{l_{d',j}, c(l_{d',j})\}, j = t, .., (t+m)$ for m consecutive epochs, where t refers to a given starting epoch E_t. Finally, Kopis computes the average confidence scores $\overline{C}_M = avg_j\{c(l_{d',j})\}$ for the *malware* labels assigned to d' by $\mathcal{S}$ across the m epochs, and an alarm is raised if $\overline{C}_M$ is greater than a threshold θ.

4 Statistical Features

In this section we describe the statistical features that Kopis *extracts* from the monitored DNS traffic. For each DNS query q_j regarding a domain name d and the related DNS response r_j, we first translate it into a tuple $\mathcal{Q}_j(d) = (T_j, R_j, d, IPs_j)$, where T_j identifies the epoch in which the query/response was observed, R_j is the IP address of the machine that initiated the query q_j, d is the queried domain, and IPs_j is the set of resolved IP addresses as reported in the response r_j. It is worth noting that since we are monitoring DNS queries and responses from the upper DNS hierarchy, in some cases the response may be *delegated* to a name server which Kopis does not currently monitor. This is particularly relevant to our TLD-level data feed, since most TLD servers are

delegation-only[1]. In all those cases in which the response does not carry the resolved IP addresses, we can derive the *IPs* set by leveraging a passive DNS database [24], or by directly querying the delegated name server.

Given a domain name d and a series of tuples $\mathcal{Q}_j(d), j = 1, .., m$, measured during a certain epoch E_t (i.e., $T_j = E_t, \forall j = 1, .., m$), Kopis extracts the following groups of statistical features:

Requester Diversity (RD) This group of features aims to characterize if the machines (e.g., RDNS servers) that query a given domain name are localized or are globally distributed. In practice, given a domain d and a series of tuples $\{\mathcal{Q}_j(d)\}_{j=1..m}$, we first map the series of requester IP addresses $\{R_j\}_{j=1..m}$ to the BGP prefix, autonomous system (AS) numbers, and country codes (CC) the IP addresses belong to. Then, we compute the distribution of occurrence frequencies of the obtained BGP prefixes (sometimes referred to as classless inter-domain routing (CIDR) prefixes), the AS numbers and CCs.

For each of these three distributions we compute the mean (three features) , standard deviation (three features) and variance (three features). Also, we consider the absolute number of distinct IP addresses (i.e., distinct values of $\{R_j\}_{j=1..m}$), the number of distinct BGP prefixes, AS numbers and CCs (four features in total). Overall, we obtain thirteen statistical features that summarize the diversity of the machines that query a particular domain name, as seen from an AuthNS or TLD server.

The choice of the *RD* features is motivated by the observation that the distribution of the machines on the Internet that query malicious domain names is on average different from the distribution of IP addresses that query legitimate domains. Semi-popular legitimate domain names (i.e., small business or personal sites) will not have a stable diverse population of recursive DNS servers or stubs that will try to systematically contact them. On the other hand popular legitimate domain names (i.e., zone cuts, authoritative name servers, news/blog forums, etc.) will demonstrate a very consistent and very diverse pool of IP addresses looking them up on a daily basis.

Malware-related domain names will have a diverse pool of IP addresses looking them up in a systematic way (i.e., multiple contiguous days). These IP addresses are very likely to have a significant network and geographical diversity simply because with the exception of targeted attacks adversaries will not try to control or restrain the geographical and network distribution of the machines getting compromised by drive-by sites and other social networking techniques. Intuitively, the diversity of the infected population will be different over a given time period, in comparison to that of benign domain names.

[1]Delegation-only DNS servers are effectively limited to containing `NS` resource records for sub-domains, but no actual data beyond its own `SOA` and `NS` records.

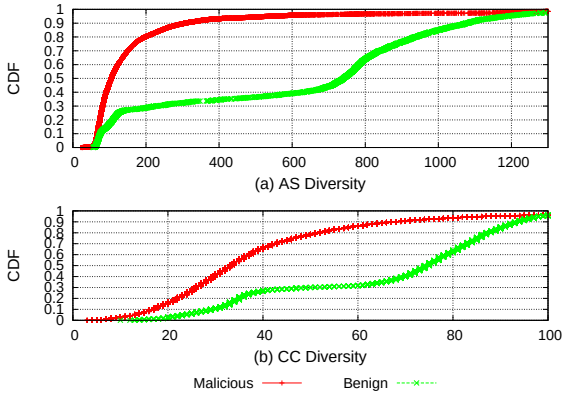


Figure 4: Distribution of AS-diversity (a) and CC-diversity (b) for malware-related and benign domains.

For example, Figure 4(a), which is derived from the dataset described in Section 5.3, reports the cumulative distribution functions (CDF) of the AS diversity of benign and malware-related domain names. In Figure 4(b) we can see the CDFs from the CC diversity for both classes in our dataset. We note that in both cases the benign domain names have a bimodal distribution. They either have low or very high diversity. On the other hand, the malware-related domain names cover a larger spectrum of diversities based on the success of the malware distribution mechanisms they use.

Requester Profile (RP) Not all query sources have similar characteristics. Given a query tuple $\mathcal{Q}_j(d) = (T_j, R_j, d, IPs_j)$, the requester's IP address R_j may represent the RDNS server of a large ISP that queries domains on behalf of millions of clients, the RDNS of a smaller organization (e.g., an academic network), or a single end-user machine. We would like to distinguish between such cases, and assign a higher weight to RDNS servers that serve a large client population because a larger network would typically have a larger number of infected machines. While it is not possible to precisely estimate the population behind an RDNS server, because of the effects of caching [15], we approximate the population measure as follows. Without loss of generality, assume we monitor the DNS query/response stream for a large AuthNS that has authority over a set of domains $\mathbf{D}$. Given an epoch E_t, we consider all query tuples $\{\mathcal{Q}_j(d)\}, \forall j, d$ seen during E_t. Let $\mathbf{R}$ be the set of all distinct requester IP addresses in the query tuples. For each IP address $R_k \in \mathbf{R}$, we count the number $c_{t,k}$ of different domain names in $\mathbf{D}$ queried by R_k during E_t.

We then define the weight associated to a requester's IP address R_k as $w_{t,k} = \frac{c_{t,k}}{\max_{l=1}^{|\mathbf{R}|} c_{t,l}}$. In practice, we assign a higher weight to requesters that query a large number of domains in $\mathbf{D}$.

Now that we have defined the weights $w_{t,j}$, given a domain name d' we measure its *RP* features as follows:

- Let $\{\mathcal{Q}_i(d')\}_{i=1..h}$ be the set of query tuples related to d' observed during an epoch E_t. Also, let $\mathbf{R}(d')$ be the set of all distinct requester IP addresses in $\{\mathcal{Q}_i(d')\}_{i=1..h}$. For each $R_k \in \mathbf{R}(d')$ we compute the count $c_{t,k}$ as previously described. Then, given the set $C_t(d') = \{c_{t,k}\}_k$, we compute the average, the biased and unbiased standard deviation[2], and the biased and unbiased variance of the values in $C_t(d')$. It is worth noting that the biased and unbiased estimators of the standard deviation and variance have different values when the cardinality $|C_t(d')|$ is small.

- Similar to the above, for each $R_k \in \mathbf{R}(d')$ we compute the count $c_{t,k}$. Afterwards, we multiply each count by the weight $w_{t-n,k}$ to obtain the set $WC_t(d') = \{c_{t,k} * w_{t-n,k}\}_k$ of weighted counts. It is worth noting that the weights $w_{t-n,k}$ are computed based on historical data about the resolver's IP address collected n epochs (seven days in our experiments) before the epoch E_t. We then compute the average, the biased and unbiased standard deviation, and the biased and unbiased variance of the values in $WC_t(d')$.

The *RD* and *RP* features described above aim to capture the fact that malware-related domains tend to be queried from a diverse set of requesters with a higher weight more often than legitimate domains. An explanation for this expected difference in the requester characteristics is that malware-related domains tend to be queried from a large number of ISP networks, which usually are assigned a high weight. The reason is that ISP networks often offer little or no protection against malware-related software propagation. In addition, the population of machines in ISP networks is usually very large, and therefore the probability that a machine in the ISP network becomes infected by malware is very high. On the other hand, legitimate domains are often queried from both ISP networks and smaller organization networks (having a smaller weight), such as enterprise networks, which are usually better protected against malware and tend to query fewer malware-related domains. As shown in Section 5 both set of features can successfully model benign and malware-related domain names.

[2]The biased estimator for the standard deviation of a random variable X is defined as $\hat{\sigma} = \sqrt{\sum_{i=1}^{N} \frac{1}{N}(\bar{X}_i - \mu)^2}$, while the unbiased estimator is defined as $\tilde{\sigma} = \sqrt{\sum_{i=1}^{N} \frac{1}{N-1}(\bar{X}_i - \mu)^2}$

Resolved-IPs Reputation (IPR) This group of features aims to describe whether, and to what extent, the IP address space pointed to by a given domain has been historically linked with known malicious activities, or known legitimate services. We compute a total of nine features as follows. Given a domain name d and the set of query tuples $\{\mathcal{Q}_j(d)\}_{j=1..h}$ obtained during an epoch E_t, we first consider the overall set of resolved IP addresses $\mathbf{IPs}(d,t) = \cup_{j=1}^{h} \mathit{IPs}_j$ (where IPs_j is an element of the tuple $\mathcal{Q}_j(d)$, as explained above). Let $\mathbf{BGP}(d,t)$ and $\mathbf{AS}(d,t)$ be the set of distinct BGP prefixes and autonomous system numbers to which the IP addresses in $\mathbf{IPs}(d,t)$ belong, respectively. We compute the following groups of features.

- *Malware Evidence*: includes the average number of known malware-related domain names that in the past month (with respect to the epoch E_t) have pointed to each of the IP addresses in $\mathbf{IPs}(d,t)$. Similarly, we compute the average number of known malware-related domains that have pointed to each of the BGP prefixes and AS numbers in $\mathbf{BGP}(d,t)$ and $\mathbf{AS}(d,t)$.

- *SBL Evidence*: much like the malware evidence features, we compute the average number of domains from the Spamhaus Block List [22] that, in the past have pointed to each of the IP addresses, BGP prefixes, and AS numbers in $\mathbf{IPs}(d,t)$, $\mathbf{BGP}(d,t)$, and $\mathbf{AS}(d,t)$, respectively.

- *Whitelist Evidence*: We compute the number of IP addresses in $\mathbf{IPs}(d,t)$ that match IP addresses pointed to by domains in the DNSWL [9] [3] or the top 30 domains according to Alexa [2]. Similarly we compute the number of BGP prefixes in $\mathbf{BGP}(d,t)$ and AS numbers in $\mathbf{AS}(d,t)$ that include IP addresses pointed by domains in DNSWL or the top 30 Alexa domains.

The IPR features try to capture whether a certain domain d is related to domain names and IP addresses that have been historically recognized as either malicious or legitimate domains. The intuition is that if d points into IP address space that is known to host lots of malicious activities, it is more likely that d itself is also involved in malicious activities. On the other hand, if d points into a well known, professionally run legitimate network, it is somewhat less likely that d is actually involved in malicious activities.

[3]Domain names up to the *LOW* trustworthiness score, where *LOW* trustworthiness score follows the definition by DNSWL [9]. More details can be found at `http://www.dnswl.org/tech`.

Discussion While none of the features used alone may allow Kopis to accurately discriminate between malware-related and legitimate domain names, by combining the features described above we can achieve a high detection rate with low false positives, as shown in Section 5.

We would like to emphasize that the features computed by Kopis, particularly the *Requester Diversity* and *Requester Profile* features, are novel and very different from the statistical features proposed in Notos [3] and Exposure [4], which are heavily based on IP reputation information. Unlike Notos and Exposure, which leverage RDNS-level DNS traffic monitoring, Kopis extracts statistical features specifically chosen to harvest the "malware signal" as seen from the upper DNS hierarchy, and to cope with the coarser granularity of the DNS traffic observed at the AuthNS and TLD level. Furthermore, we show in Section 5 that, unlike previous work, Kopis is able to detect malware-related domains *even when no IP reputation information is available*.

The *Requester Diversity* and *Requester Profile* features can operate without any historical IP address reputation information. These two sets of features can be computed practically and *on-the-fly* at each authoritative or TLD server. The main reason why we identify the six *Resolved-IP Reputation* features is to harvest part of the already established IP reputation in IPv4. This will help the overall system to reduce the false positives (FPs) and at the same time maintain a very high true positives (TPs). We will elaborate more in Section 5 on the different operational modes of Kopis.

5 Evaluation

In this section, we report the results of our evaluation of Kopis. First, we describe how we collected our datasets and the related ground truth. We then present results regarding the detection accuracy of Kopis for authoritative NS- and TLD-level deployments. Finally, we present a case study regarding how Kopis was able to discover a previously unknown DDoS botnet based in China.

5.1 Datasets

Our datasets were composed of the DNS traffic obtained from two major domain name registrars between the dates of 01-01-2010 up until 08-31-2010 and a country code top level domain (`.ca`) between the dates of 08-26-2010 up until 10-18-2010. In the case of the two domain name registrars we were also able to observe the answers returned to the requester of each resolution. Therefore, it is easy for us to identify the IP addresses for the `A-type` of DNS query traffic. In the case of the TLD we obtained data only for 52 days and had to passively reconstruct the

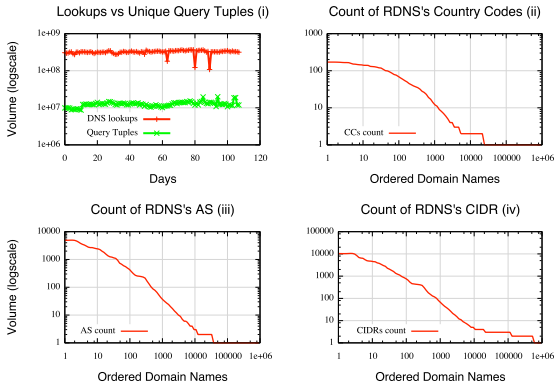


Figure 5: General observations from the datasets. Plot (i) shows the difference between the raw lookup volume vs. the query tuples that Kopis uses over a period of 107 days. Plots (ii), (iii) and (iv) show the number of unique CCs, ASs and CIDRs (in which the RDNSs resides) for each domain name that was looked up during one day.

IP addresses corresponding to the `A-type` of lookups observed.

An interesting problem arises when we work with the large data volume from major authorities and the `.ca` TLD servers. According to a sample monitoring period of 107 days we can see from Figure 5 (i) that the daily number of lookups to the authorities was on average 321 million. This was a significant problem since it would be hard to process such a volume of raw data, especially if the temporal information from these daily observations were important for the final detection process. On the same set of raw data we used a data reduction process that maintained only the query tuples (as defined in Section 4). This reduced the daily observations, as we can observe from Figure 5 (i), to a daily average of 12,583,723 **unique** query tuples. The signal that we missed with this reduction was the absolute lookup volume of each query tuple in the raw data. Additionally, we missed all time sensitive information regarding the periods within a day that each query tuple was looked up. As we will see in the following sections, this reduction does not affect Kopis' ability to model the profile of benign and malware-related domains.

Figures 5 (ii), (iii) and (iv) report the number of CIDR (i.e., BGP prefixes), Autonomous Systems (AS), Country Code (CC), respectively, for the RDNSs (or requesters) that looked up each domain name every day. The domains are sorted based on counts of ASs, CCs and CIDRs corresponding to the RDNSs that look them up (from left to right with the leftmost having the largest count). We observe that roughly the first 100,000 do-

main names were the only domains that exhibit any diversity among the requesters that looked them up. We can also observe that the first 10,000 domain names are those that have some significant diversity. In particular only the first 10,000 domain names were looked up by at least five CIDRs, or five ASs or two different CCs. In other words, the remaining domains were looked up from very few RDNSs, typically in small sets of networks and a small number of countries. Using this observation we created statistical vectors only for domain names in the sets of the 100,000 most diverse domains from the point of view of the RDNS's CC, AS and CIDR.

5.2 Obtaining the Ground Truth

We collected more than eight months of DNS traffic from two DNS authorities and the `.ca` TLD. All query tuples derived from these DNS authorities were stored daily and indexed in a relational database. Due to some monitoring problems we missed traffic from 3 days in January, 9 days in March and 6 days in June 2010.

Some of our statistical features require us to map each observed IP address to the related CIDR (or BGP prefix) AS number and country code (Section 4). To this end, we leveraged Team CYMRU's IP-to-ASN mapping [28].

Kopis' knowledge base contained malware information from two malware feeds collected since March 2009. We also collected public blacklisting information from various publicly available services (e.g., Malwaredomains [16], Zeus tracker [31]). Furthermore, we collected information regarding domain names residing in benign networks from DNSWL [9] but also the address space from the top 30 Alexa [2] domains verified using the assistance of the Dihe's IP address index browser [8]. Overall, we were able to label 225,429 unique RRs that correspond to 28,915 unique domain names. From those we had 1,598 domain names labeled as legitimate and 27,317 domain names labeled as malware-related. All collected information was placed in a table with first and last seen timestamps. This was important since we computed all IPR features for day n based only on data we had until day n. Finally, we should note that we labeled all the data based on black-listing and white-listing information collected until October 31^{st} 2010.

5.3 Model Selection

As described in Section 3, Kopis uses a machine learning algorithm to build a detector based on the statistical profiles of resolution patterns of legitimate and malware-related domains. As with any machine-learning task, it is important to select the appropriate model and important parameters. For Kopis, we need to identify the minimal observation window of historic data necessary for training. The observation window here is the number of epochs from which we assemble the training dataset.

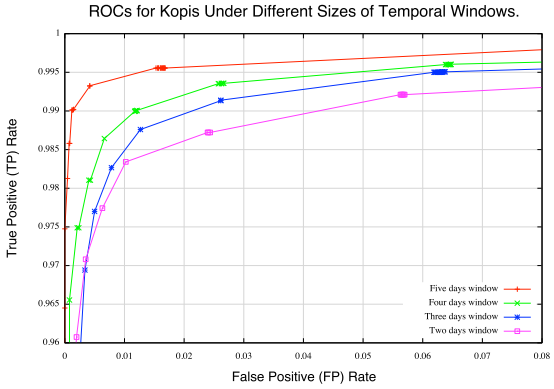


Figure 6: ROCs from datasets with different sizes assembled from different time windows.

In Figure 6, we see the detection results from four different observation windows. The ROCs in Figure 6 were computed using 10-fold cross validation. The classifier that produced these results was a random forest (RF) classifier under a two, three, four and five day training window. The selection of the RF classifier was made using a model selection process [10], a common method used in the machine learning community, which identified the most accurate classifier that could model our dataset. Besides the RF, during model selection we also experimented with Naive Bayes, k-nearest neighbors (IBK), Support Vector Machines, MLP Neural Network and random committee (RC) classifiers [10]. The best detection results reported during the model selection were from the RF classifier. Specifically, the RF classifier achieved a $TP_{rate} = 98.4\%$ and a $FP_{rate} = 0.3\%$ using a five day observation window. When we increased the observation window beyond the mark of five days we did not see a significant improvement in the detection results.

We should note that this parameter and model methodology should be used every time Kopis is being deployed in a new AuthNS or TLD server because the characteristics of the domains, and hence the resolution patterns, may vary in different AuthNS and TLD servers, and different patterns or profiles may best fit different parameter values and classifiers.

5.4 Overall Detection Performance

In order to evaluate the detection performance of Kopis and in particular the validity and strength of its statistical features and classification model, we conducted a *long-term* experiment with five months of data. We used 150

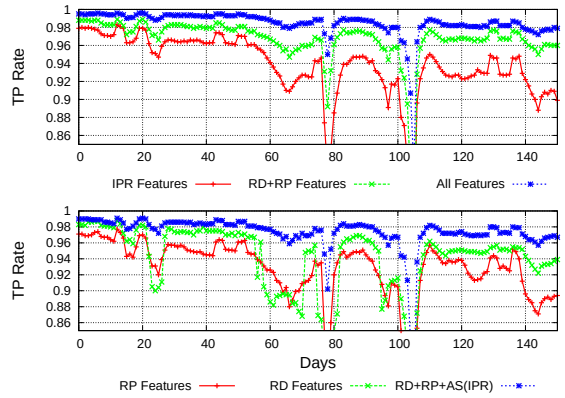


Figure 7: The distribution of TP_{rate} for combination of features and features families in comparison with Kopis observed detection accuracy.

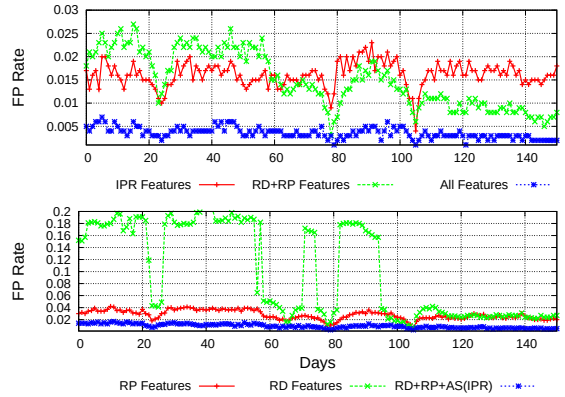


Figure 8: The distribution of FP_{rate} for combinations of features and features families in comparison with Kopis observed detection accuracy.

different datasets created over a period of 155 days (first 15 days for bootstrap). These datasets were composed by using a fifteen-day sliding window with a one-day step (i.e., two consecutive windows overlap by 14 days). We then used 10-fold cross validation[4] to obtain the FP_{rates} and TP_{rates} from every dataset. We picked three classification algorithms, namely, RF, RC, and IBK, which performed best in the model selection process (described in Section 5.3) because we wanted to use their detection rates during the long-term experiment.

In Figure 7 and Figure 8 we observe the distribution of the TP_{rates} and FP_{rates} for the RF classifier over the entire evaluation period. The average, minimum and maximum FP_{rates} for the RF were 0.5% (8 domains), 0.2% (3 domains) and 1.1% (18 domains), respectively, while the average, minimum and maximum TP_{rates} were 99.1% (27,072 domains), 98.1% (27.071 domains) and 99.8% (27,262 domains), respectively. The RF classifier's FP_{rates} were almost consistently around 0.6% or less. The TP_{rate} of the RF classifier, with the exception of six days, was above 96% and typically in the range of 98%. With the IBK classifier being the exception, the RF and RC classifiers had similar longterm detection accuracy. This experiment showed that Kopis overall has a very high TP_{rate} and very low FP_{rate} against all new and previously unclassified malware-related domains.

As described in Section 4, we define three main types of features. Next we show how Kopis would operate if trained on datasets assembled by features from each family, first separately and then combined. To derive the results from the experiments, we used as input the 150 datasets created in the previously described longterm evaluation mode. Then, for each one of these 150 datasets, we isolated the features from the RD, RP and IPR feature families into three additional types of datasets. In Figure 7 and Figure 8 we present the longterm detection rates obtained using 10-fold cross validation of these three different types of datasets. Additionally, we present the detection results from:

- The combination of RP and RD features (`RD+RP Features`).
- The combination of RD, RP and the features from the IPR feature family that describe the Autonomous System properties of the IP address that each domain name d points at (`RD+RP+IRP (AS) Features`).
- The detection results from the combination of all features combined (`All Features`).

The longterm FP_{rates} and TP_{rates} in Figure 7 and Figure 8 respectively, we show the detection accuracies from each different feature set. One may tend to think that the IPR (IP reputation) features hold a significantly stronger classification signal than the combination of RD and RP features, mainly because there are many resources that currently contribute to the quantification and improvement of IP reputation (i.e., spam block lists, malware analysis, dynamic DNS reputation etc.). However, Figure 7 and Figure 8 show that with respect to both the FP_{rates} and TP_{rates}, the combination of the RD and RP sets of features performs almost equally to the IPR features used in isolation from the remaining features. At the same time, using all features performs much better than using each single feature subset in isolation. This

[4]To avoid overfitting our dataset we report the evaluation results using 10-fold cross validation that implies that 90% of dataset is used for training and 10% for testing — in each of the 10 folds. This technique is known [14] to yield a fair estimation of classification performance over a dataset.

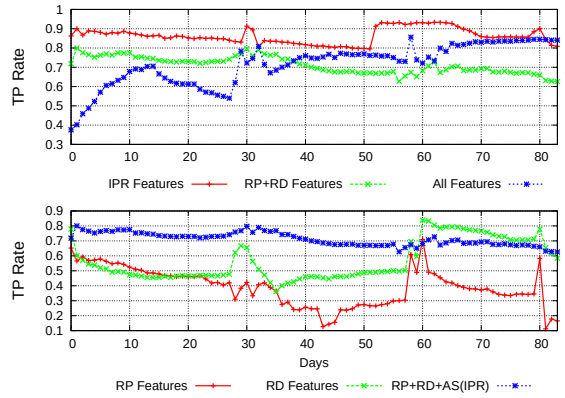


Figure 9: TP_{rates} for different observation periods using an 80/20 train/test dataset split.

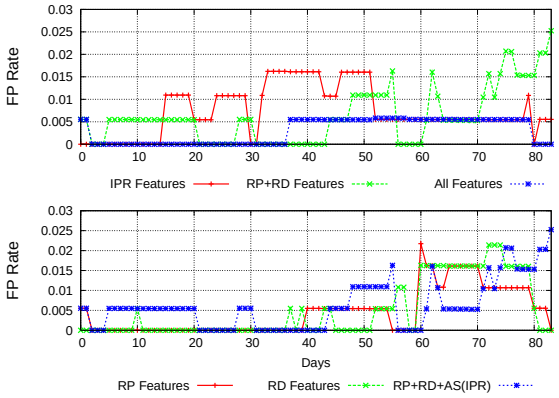


Figure 10: FP_{rates} for different observation periods using an 80/20 train/test dataset split.

shows that the combination of the RP and RD features contribute significantly to the overall classification accuracy and can enable the correct classification of domains in environments where IP reputation is absent or in cases where we cannot reliably compute IP reputation features "on-the-fly" (e.g., in some TLD-level deployments).

5.5 New and Previously Unclassified Domains

While the experiments described in Section 5.4 showed that Kopis can achieve very good overall detection accuracy, we also wanted to evaluate the "real-world value" of Kopis, and in particular its ability to detect new and previously unclassified malware domains. To this end, we conducted a set of experiments in which we trained Kopis based on one month of labeled data from which we randomly excluded 20% of both benign and malware-related domains (i.e., we assumed that we did not know anything about these domain names during training). This excluded 997 benign and 4,792 malware-related unique, deduplicated domain names from the training datasets. Then we used the next three weeks of data as an evaluation dataset, which contained the domains excluded from the training set mentioned above, as well as all other newly seen domain names. In other words, the classification model learned using the training data was not provided with any knowledge whatsoever about the domains in the evaluation dataset.

We then classified the domains in the evaluation dataset, with the assistance of a Random Forest classifier, as we already discussed in Section 3. We used a training period of 30 consecutive days and a testing period of $m = 21$ days immediately following the training period. The detection threshold θ was set to 0.9 to obtain a good operational trade-off between false positives and detection rate. Our primary reasoning behind setting the threshold θ to 0.9 was to keep the FP_{rates} as low as possible so that an operator would only have to deal with a very small number of FPs on a daily basis. We repeated this evaluation four times during different months within our eight months of traffic monitoring.

In Figure 9 and Figure 10, we can see the results of these experiments. From left to right, we can see the evaluation on 21 days of traffic in February, March, May and June of 2010. We trained the system based on one month of traffic from January, February, March and May 2010, respectively. We chose these months because we had continuous daily observations (i.e., no data gaps) from both training and testing datasets. As in the longterm 10-fold evaluation, we performed the experiments using six different datasets obtained using different feature subsets.

We present the results in the same way as in Section 5.4. When we used all features we observed the average FP_{rates} was 0.53% ($\sim$ two domains), while the average TP_{rates} was 73.62% (3,528 domain names). For the `RP+RD Features` and `IPR Features` the average FP_{rates} were 0.54% ($\sim$ two domains) and 0.79% ($\sim$ two domains), respectively; while the average TP_{rates} were 69.19% (3,315 domain names) and 87.25% (4,181 domain names), respectively. The `RP+RD+AS(IPR) Features`, gave average $FP_{rates} = 0.66\%$ (or $\sim$ two domain names) and average $TP_{rates} = 65.05\%$ (or 3,117 domain names).

When we used the combination of all features we see that for the first 42 days of evaluation (February and March of 2010) Kopis had a virtually zero FP_{rates} and an average $TP_{rates} = 68\%$. In the following 42 days of evaluation, Kopis, had better TP_{rates} but with some ex-

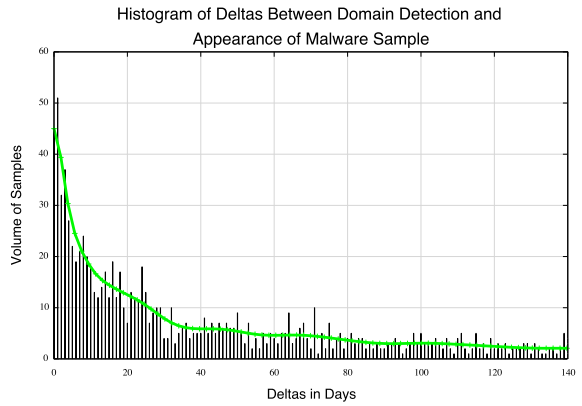


Figure 11: Kopis early detection results. The deltas in days between the Kopis classification dates and the date we've received a corresponding malware sample for the domain name.

tra false positives, always below 0.5%. Investigating the nature of the false positives, we observed that the domain names responsible are related to BitTorrent services, on-demand web-TV services and what appeared to be online gaming sites. We suspect that the main reason why these domains cause false positives is because the population of similar legitimate services was insufficiently represented during training, and therefore, the RF classifier failed to learn this behavior as being legitimate in training.

This experiment showed that Kopis — with all features used — can detect new and previously unclassified domains with an average TP_{rate} of 73.62% and average FP_{rate} of 0.53%. Although this is worse than the overall detection performance reported in Section 5.4, it is actually a good result considering that Kopis has no knowledge of the domains in the testing dataset. It implies that Kopis has good "real-world value" thanks to its ability to detect new, previously unseen attacks is at a premium.

Figure 11 shows the difference in days between the time that Kopis identifies a *true positive* domain as being malware-related, and the day we first obtained the malware sample associated with the malware-related domain from our malware feed. To perform this measurement, we used malware from a commercial malware feed with volume between 400 MB to 2 GB of malware samples every day. Additionally, we used malware captured from two corporate networks. As we can see, Kopis was able to identify domain names on the rise **even before** a corresponding malware sample is accessible by the security community. This result shows that Kopis can provide the ability to the registrars and TLD operators to preemptively block or take down malware related domains and remove botnets from the Internet before they become a large security threat.

5.6 Canadian TLD

Thus far, the experiments we have reported were all using data available at AuthNSs. A TLD server is one level above AuthNS servers in the DNS hierarchy, and as such, it has a greater global visibility but with less granular data on DNS resolution behaviors. In this section we report our experiments of Kopis at the TLD level.

We evaluated Kopis on query data obtained from the Canadian TLD. We used the same evaluation method introduced in Section 5.5 but with different training window sizes, testing epochs and classification thresholds. Before we describe the results, we should note that all TLD traffic needs passive reconstruction of the query data to identify the IPs addresses in the `A-type` resource records. We used a passive DNS database composed of data from four ISP sensors and the passive DNS database from SIE [24]. The Canadian TLD's traffic was harvested from SIE [24] (channel three).

Unfortunately, due to the fact that we obtained traffic from only 52 days (2010-08-26 until 2010-10-18) we had to use a smaller training epoch of 14 days (instead of one month). We evaluated Kopis using the RF classifier, 14 consecutive days as the training epoch, 14 days following the training epoch as the evaluation epoch, and setting the threshold $\theta = 0.9$. Two sequential training epochs had seven days in common. The exact training epochs were *08-27* to *09-11*, *09-04* to *09-18*, *09-11* to *09-25* and *09-18* to *10-02* while the corresponding evaluation epochs were *09-12* to *09-26*, *09-19* to *10-03*, *09-26* to *10-10* and *10-03* to *10-17*, respectively. Without changing the data labeling methodology, we assembled a dataset with 2,199 malware related and 1,018 benign unique deduplicated domain names.

In Figure 12 and Figure 13, we can see the results of this experiment. As with the experiments in Section 5.5, we evaluated Kopis in six modes, using as threshold $\theta = 0.5$. We should note here that the evaluation of the `RD+RP Features` reflects the evaluation mode with datasets that were composed only by the combination of RD and RP features. Such dataset can be extracted directly from data readily available at a TLD server (in other words, the `RD+RP Features` is the most "efficient" mode that Kopis can operate in and can be computed on the fly at a TLD server).

When we used all features we observed the average FP_{rates} was 0.52% ($\sim$ six domain names), while the average TP_{rates} was 94.68% (2,082 domain names). For the `RP+RD Features` and `IPR Features` the average FP_{rates} were 3.18% ($\sim$ 33 domain names) and 0.36% ($\sim$ four domain names), respec-

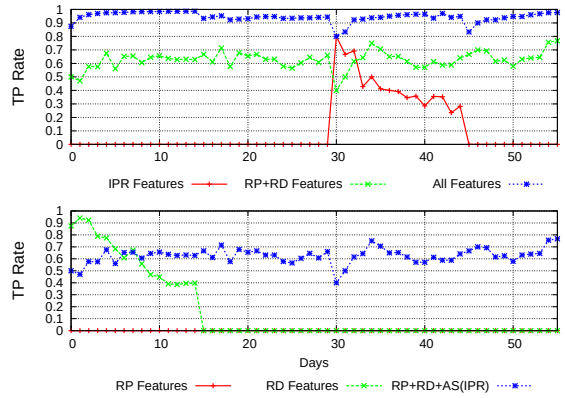


Figure 12: TP_{rates} achieved during evaluation of traffic obtained from `.ca` TLD.

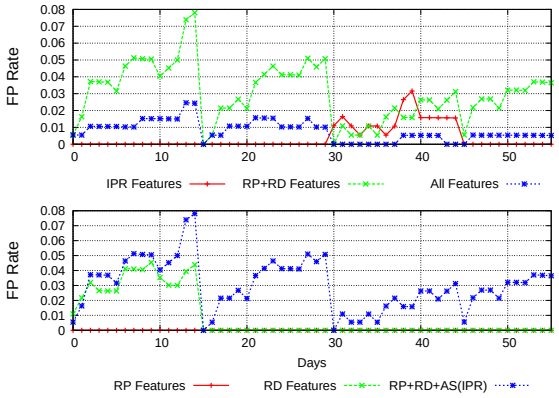


Figure 13: FP_{rates} achieved during evaluation of traffic obtained from `.ca` TLD.

tively; while the average TP_{rates} were 63.63% (1,399 domain names) and 10.84% (238 domain names), respectively. The `RP+RD+AS(IPR) Features`, gave the average $FP_{rates} = 1.03\%$ (or ten domain names) and average $TP_{rates} = 78.95\%$ (or 1,736 domain names).

During the `RP+RD Features` evaluation, we observed that the average TP_{rates} reached 63.63% while the average FP_{rates} were in the range of 3.18%. These were very promising results despite the relatively high FP_{rates} because we can operate Kopis using a sequential classification mode, starting with `RP+RD Features` followed by `All Features`. Kopis in this "in-series" classification mode can achieve a good balance of efficiency and accuracy.

More specifically, at the first step in the sequential process, Kopis is a "coarse filter" that operates in `RP+RD Features` with only the RP and RD statistical features and threshold $\theta = 0.5$. Any domain name that passes this filter (i.e., with a "malware-related" label) then requires additional feature computation, i.e., reconstructing the resolved IP address records, and further classification at the next step in the sequential process. On the other hand, domains that are dropped by this filter (i.e., with a "legitimate" label) are no longer analyzed by Kopis. Thus, the first step filter is essentially a data reduction tool, and the sequential classification process is a way to delay the expensive computation until the data volume is reduced. This technique is very important at the TLD level given the potentially huge volume of data.

In our experiments Kopis operating at the first step with `RP+RD Features` (and threshold $\theta = 0.5$) yielded an average data reduction rate[5] of 87.95% on the original dataset. After this reduction, at the second step, we evaluated Kopis on the (remaining) dataset using all features, and keeping the same threshold $\theta = 0.5$. The average FP_{rates} reported at this step by Kopis were zero while the average TP_{rates} were 94.44%. The overall FP_{rates} and TP_{rates} for this "in-series" mode were zero and 60.09% (1,321 domain names), respectively.

At this point we should note that the threshold θ was set again with the intention to have the FP_{rates} as close to 1.0% as possible but also not to sacrifice much of the TP_{rate} produced from the first classification process in the "in-series" mode. As we saw previously, even when we had some FPs created by the `RP+RD Features` (the first classification process in the "in-series" mode), the combination of statistical features in the second "in-series" mode was able to prune away these FPs. An operator may choose to lower the threshold θ even more and have as an immediate effect, the increase of domain names that will be forwarded to the second "in-series" classification process, with a potential increase in the overall TP_{rate} and FP_{rates}. The experiments in this section showed that by using an "in-series" classification process where different steps can use different (sub)sets of features and thresholds, Kopis can achieve a good balance of detection performance and operation efficiency at the TLD level.

5.7 DDos Botnet Originated in China

As discussed in Section 1, Kopis was designed to have global visibility so that it can detect domains associated with malware activities running in an uncooperative country or networks before the attacks propagate to net-

[5]We define the reduction rate as follows: $1 - \frac{TP_{malware}+FP_{malware}}{ALL}$, where $TP_{malware}$ is the true positives for the malware-related class, $FP_{malware}$ is the mis-classified as malware-related benign domain names and ALL all as the domain names in the evaluation dataset.

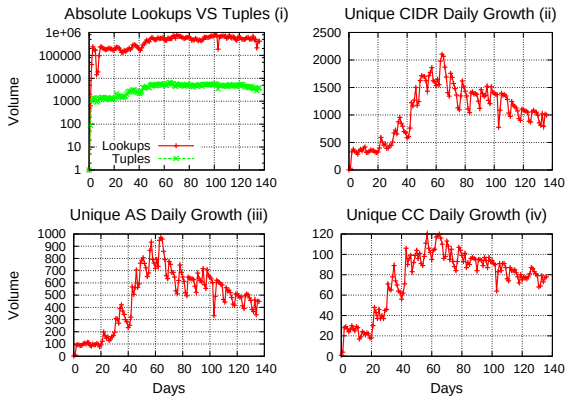


Figure 14: Various growth trends for the DDoS botnet. Day zero is *03-20-2010*.

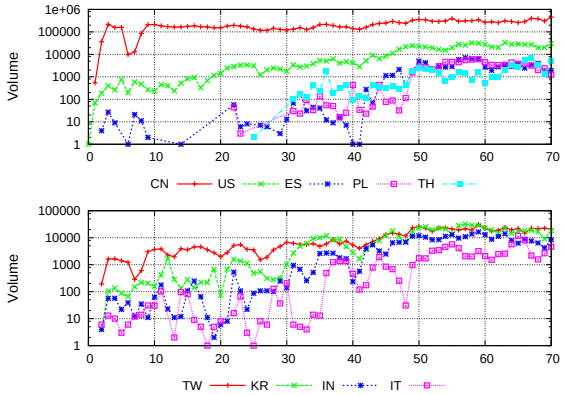


Figure 15: A snapshot from the first 70 days of the botnet's growth with respect to the country code-based resolution attempts for the DDoS botnet's domain names. Day zero is *03-20-2010*.

works that it protects. In this section, we report a case study to demonstrate Kopis's global detection capability.

Kopis was able to identify a commercial DDoS botnet in the first few weeks of its propagation in China and well before it began propagating within other countries, including the US. We alerted the security community, and the botnet was finally removed from the Internet in the middle of September 2010. Next we provide some intuition behind this discovery and why Kopis was able to detect this threat early.

This DDoS botnet was controlled through 18 domain names, all of which were registered by the attacker under the same authority (although with different 2LDs). Kopis was deployed at the AuthNS server and was able to observe resolution requests to these domains (even when the infected machines were initially not in the US) and classify them as malware-related because their resolution patterns fit the profiles of known malware domains in its knowledge base.

These domain names were linked with six IP addresses located in the following autonomous systems: 14745 (US), two in 4837 (CN), 37943 (CN) and two in 4134 (CN), throughout the lifetime of the botnet. We show the difference between the absolute DNS lookups versus the daily volume of unique query tuples in Figure 14 (i). The average lookup volume every day was 438,471 with average de-duplicated query tuples in the range of 3,883. Despite this significant data reduction, Kopis was still able to track and identify this emerging threat. In Figures 14 (ii), (iii) and (iv), we can see the daily growth of unique CIDRs, AS and CCs related to the RDNSs that queried the domain names used in the botnet.

An interesting observation can be made from Figure 15. In this figure we can see the daily lookup volume for the domain names of this botnet. Instantly we can see that the first big infection happened in Chinese networks in a relatively short period of time (in the first 2-3 days). After this initial infection, a number of machines from several other countries were also infected but nowhere close to the volume of the infected population in the Chinese networks. As an example we can see in Figure 15 that the first time more than 1,000 daily lookups were observed from the United States was more than 20 days after the botnet was launched. Also, other countries such as Poland and Thailand had the first infection 21 and 25 days after the botnet were lunched. Furthermore, large countries such as Italy, Spain and India reached the 100 daily lookup threshold 15 days later than the start of this botnet. Clearly, for countries like Poland and Thailand (and even Italy, Spain and India to a large extent) localized DNS reputation techniques could not have been able to observe a resolution request (or a strong enough signal) for any of the domain names related to this botnet, until the botnet had reached global scale, which was several weeks after it was launched. Figure 16 shows the volume of samples correlated with this botnet as they appeared in our malware feeds. We observe that the first malware sample related to this botnet appeared two months after the botnet became active.

To demonstrate the contribution of each feature family towards the identification of the domain names that were part of this botnet we conducted the following experiment. We trained Kopis with 30 days of data before the 5^{th} of May 2010. Then we computed vectors for all the domain names that were part of the botnet. We computed one vector every day for each domain name based on the information we had on the domain name and IP address up until that day. We classified each vector

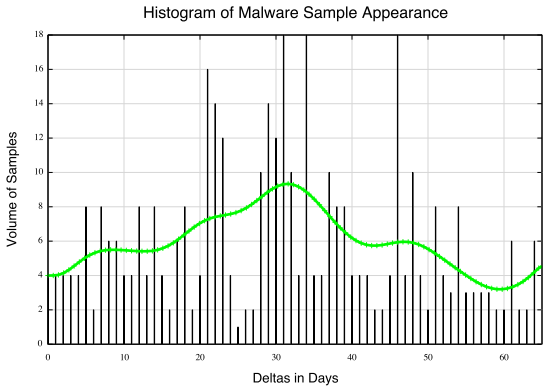


Figure 16: Volume of malware samples related with the DDoS botnet as they appeared in our feeds. Day zero is *05-20-2010*.

against four trained classifiers with the following set of features; `All Features`, `RD+RP Features`, `IPR Features`, and `RD+RP+AS(IPR) Features`. We then marked the first day that each classifier detected a domain name as malware-related, while setting the threshold $\theta = 0.9$. By doing so we identified the earliest day that the classifier would have detected the domain name without human forensic analysis on the results. The detection results from this experiment can be found in Table 1.

What the results show is that only the combination of all features can detect all the domain names until the end of August. On the other hand the IPR and the combination of RD+RP features detected more than half of the domain names by the middle of July, when the botnet was in its peak. We should also note that in the middle of July we saw the biggest volume of malware samples related to the botnet's domain names surfacing in the security community. Finally, we should also note that these 18 domain names appeared in public blacklists after the take-down of the botnet was publicly disclosed (September 2010). Obviously, this was not exactly how we detected the botnet. After the initial identification of the 7 domain names in the beginning of May and with some very basic forensic analysis, we managed to quickly discover the entire corpus of the related domains.

In an effort to place Kopis' early detection abilities in comparison with recursive-based reputation systems (like Notos and Exposure) we check in the passive DNS database at ISC when these 18 domain names first appeared. Fifteen of them never showed up in the RDNSs that supply ISC with DNS data. The remaining three domain appeared for the first time on the following dates: **2010-06-24 06:56:34**, **2010-07-01 14:06:47** and **2010-09-08 04:32:36**. This means that the first domain name related with this botnet appeared three months after the botnet was created and this would have been the earliest possible time that either Notos or Exposure could have detected these domain names assuming they were operating on passive DNS data from ISC — one of biggest passive DNS repositories worldwide. This clearly shows the need of detection systems like Kopis that can operate higher in the DNS hierarchy and provide Internet with an early global warning system for DNS.

Features/Dates	5/20	6/1	7/15	8/31
`All`	7	9	15	18
`RD+RP`	3	5	12	16
`IPR`	3	5	13	17
`RD+RP+AS(IPR)`	3	5	12	16

Table 1: Number of the botnet related domain names that each feature family would have detected up-until the specified date assuming that the system was operating unsupervised.

6 Discussion

In this section, we elaborate on possible evasion techniques and discuss some operational issues of Kopis.

6.1 Evasion techniques

Kopis relies significantly on the *Requester Diversity* (RD) and *Requester Profile* features. An attacker may attempt to dilute the information provided by the RP and RD features to evade Kopis. This could be achieved by resolving domain names from a diverse set of *open recursive* DNS servers or even from random IPs acting as stub resolver (e.g., using infected machines). This will not be as easy as it sounds, due to the RP feature family. This is because even if the adversary looks up domain names from various different IP addresses, the adversary will still have to look up a large number of domain names under the same authority to make the weight of each requester large enough to alter the RP features. Additionally, the adversary will have to repeatedly (for a long enough period of time) ask for different domain names served by the same authority in order to influence/dilute the RDNS weighing function.

In order to be able to artificially create the necessary signal that may dilute or even disturb the modeling of legitimate and malware-related domain names, the adversary would have to obtain access to traffic at the authority name or TLD servers. Furthermore, the adversary

would need a full list of statistical feature values used from Kopis. Such an attack would be similar in spirit to polymorphic blending attacks [12]. We note here that reliable and systematic access to DNS traffic at the authoritative or TLD level is extremely hard to obtain, since it would require the collaboration of the registrar that controls the AuthNS or the TLD servers.

Domain name generation algorithms (DGAs) have been used by malware families (i.e., Conficker [20], Zeus/Murofet [23], Bobax [26], Torpig [27] etc.) in the last few years. The new seed of these DGAs has typically the periodicity of a day. This implies that domain names generated by DGAs (and under the zones Kopis monitors) will be active only for a small period of time (e.g., a day). Due to the daily observation period mandatory for Kopis to provide detection results, such malware-related domain names will be potentially inactive by the time they are reported by our detection system. Operating Kopis with smaller epochs (i.e., hourly granularity) could potential solve this problem. We leave the verification of this operation mode to future work.

6.2 TLDs and Domain Registrars

As we have already discussed, just observing the DNS resolution requests at the TLD level will not provide sufficient information for the system to reconstruct the IP addresses mapped with the queried domain names. There are several ways to resolve this issue. The simplest way to reconstruct the IP addresses for a given domain name is to check a large passive DNS database. For the domains that are not replicated in the passive DNS database, we can use an *active probing* strategy to retrieve the resolved IP addresses with little overhead.

As a final classification heuristic, especially in the case of domain registrars, they can potentially combine Kopis with domain name registration information. Classification results from Kopis can be combined with domain name registration information (trivially accessible to domain registrars) in order to further reduce FPs but also provide an additional correlation between domain registration accounts that own domains with suspicious resolution behavior according to Kopis.

7 Conclusion

In this paper, we presented Kopis, a system that can operate at the upper DNS hierarchy and detect malware-related domains based on global DNS resolution patterns. To the best of our knowledge, Kopis is the first system that can operate at TLD servers and large authorities and provide DNS operators the ability of early detection of malware-related domains — even without information of the associated malware.

Kopis models three key signals at the DNS authorities: the daily domain name resolution patterns, the significance of each requester for an epoch, and the domain name's IP address reputation. Using more than half a year of real world data of known benign and malware-related domains from two major DNS authorities and the `.ca` TLD, our evaluation showed that Kopis can achieve high TP_{rates} (98.4% against all malware-related domains and 73.6% against new and previously unclassified malware-related domains) and low FP_{rates} (0.3% and 0.5%). Kopis was also able to detect newly created and previously unclassified malware-related domain names several weeks before they were listed in any blacklist and before information of the associated malware appeared in security forums. Finally, Kopis was used to identify the creation of a DDoS botnet in China. This ability to identify malware-related domains on the rise can provide the DNS operators the preemptive ability to remove rapidly growing botnets at the very early stage, thus minimizing their threats to Internet security.

References

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, 2006.

[2] Alexa. The web information company. `http://www.alexa.com/`, 2007.

[3] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a Dynamic Reputation System for DNS. In *the Proceedings of 19th USENIX Security Symposium (USENIX Security '10)*, 2010.

[4] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. Exposure: Finding malicious domains using passive dns analysis. In *Proceedings of NDSS*, 2011.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[6] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 93–104, New York, NY, USA, 2007. ACM.

[7] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *In Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*, 2006.

[8] dihe's IP-Index Browser. DIHE. `http://ipindex.homelinux.net/index.php`, 2008.

[9] DNS Whitelist Protect against false positives. DNSWL. `http://www.dnswl.org`, 2008.

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

[11] M. Felegyhazi, C. Keibich, and V. Paxson. On the poten-

tial of proactive domain blacklisting. In *Third USENIX LEET Workshop*, 2010.

[12] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee. Polymorphic blending attacks. In *In Proceedings of the 15 th USENIX Security Symposium*, pages 241–256, 2006.

[13] S. Hao, N. Feamster, and R. Pandrangi. An Internet Wide View into DNS Lookup Patterns. `http://labs.verisigninc.com/projects/malicious-domain-names.html`, 2010.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[15] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS performance and the effectiveness of caching. *IEEE/ACM Trans. Netw.*, 10:589–603, October 2002.

[16] MalwareDomains. DNS-BH malware domain blocklist. `http://www.malwaredomains.com`, 2007.

[17] P. Mockapetris. Domain Names - Concepts and Facilities. `http://www.ietf.org/rfc/rfc1034.txt`, 1987.

[18] P. Mockapetris. Domain Names - Implementation and Specification. `http://www.ietf.org/rfc/rfc1035.txt`, 1987.

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

[20] P. Porras. Inside risks: Reflections on conficker. *Commun. ACM*, 52:23–24, October 2009.

[21] 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.

[22] SBL. The Spamhaus Project Block List. `http://www.spamhaus.org/sbl/`, 2004.

[23] S. Shevchenko. Domain Name Generator for Murofet. `http://blog.threatexpert.com/2010/10/domain-name-generator-for-murofet.html`, 2010.

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

[25] S. Staniford, V. Paxson, and N. Weaver. How to own the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*, pages 149–167, Berkeley, CA, USA, 2002. USENIX Association.

[26] J. Stewart. Bobax trojan analysis. `http://www.secureworks.com/research/threats/bobax/`, 2004.

[27] 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 *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 635–647, New York, NY, USA, 2009. ACM.

[28] Team Cymru. IP to ASN mapping. `http://www.team-cymru.org/Services/ip-to-asn.html`, 2008.

[29] N. Weaver, S. Staniford, and V. Paxson. Very fast containment of scanning worms. In *In Proceedings of the 13th USENIX Security Symposium*, pages 29–44, 2004.

[30] D. Wessels, M. Fomenkov, N. Brownlee, and K. Claffy. Measurements and Laboratory Simulations of the Upper DNS Hierarchy. In *PAM*, 2004.

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