2013 IEEE Symposium on Security and Privacy

Finding the Linchpins of the Dark Web: a Study on Topologically Dedicated Hosts on Malicious Web Infrastructures

Zhou Li, Sumayah Alrwais
Indiana University at Bloomington
{*lizho, salrwais*}*@indiana.edu*

Yinglian Xie, Fang Yu
MSR Silicon Valley
{*yxie, fangyu*}*@microsoft.com*

XiaoFeng Wang
Indiana University at Bloomington
xw7@indiana.edu

***Abstract*—Malicious Web activities continue to be a major threat to the safety of online Web users. Despite the plethora forms of attacks and the diversity of their delivery channels, in the back end, they are all orchestrated through *malicious Web infrastructures*, which enable miscreants to do business with each other and utilize others' resources. Identifying the linchpins of the dark infrastructures and distinguishing those valuable to the adversaries from those disposable are critical for gaining an upper hand in the battle against them.**

In this paper, using nearly 4 million malicious URL paths crawled from different attack channels, we perform a large-scale study on the topological relations among hosts in the malicious Web infrastructure. Our study reveals the existence of a set of topologically dedicated malicious hosts that play orchestrating roles in malicious activities. They are well connected to other malicious hosts and do not receive traffic from legitimate sites. Motivated by their distinctive features in topology, we develop a graph-based approach that relies on a small set of known malicious hosts as seeds to detect dedicate malicious hosts in a large scale. Our method is general across the use of different types of seed data, and results in an expansion rate of over 12 times in detection with a low false detection rate of 2%. Many of the detected hosts operate as redirectors, in particular Traffic Distribution Systems (TDSes) that are long-lived and receive traffic from new attack campaigns over time. These TDSes play critical roles in managing malicious traffic flows. Detecting and taking down these dedicated malicious hosts can therefore have more impact on the malicious Web infrastructures than aiming at short-lived doorways or exploit sites.

I. Introduction

Technological progress often comes with side effects. Look at today's Web: not only does it foster a booming Web industry, but it also provides new opportunities to criminals who are rapidly industrializing their dark business over the Web. Today once you unfortunately click a malicious URL, chances are that those who victimize you are no longer individual, small-time crooks but an underground syndicate: some luring you to visit malicious websites through various channels (Spam, tweets, malicious advertising, etc.), some buying and selling your traffic through redirection, and the receiving ends of the traffic performing different exploits (drive-by downloads, scams, phishing etc.) on your system on behalf of their customers. Such a complicated attack is orchestrated through *malicious Web infrastructures*, which enable those miscreants to do business with each other and utilize others' resources to make money from their misdeeds. Indeed, such infrastructures become the backbone of the crimes in today's cyberspace, delivering malicious web content world wide and causing hundreds of millions in damage every year.

Malicious Web infrastructures. Given the central role those infrastructures play, an in-depth understanding of their structures and the ways they work becomes critical for counteracting cybercrimes. To this end, prior research investigated the infrastructures associated with some types of channels (e.g., Spam [2], black-hat Search-Engine Optimization (SEO) [11]) and exploits (e.g., drive-by downloads [23]). What have been learnt includes the parties involved in a malicious campaign (e.g., affiliates, bot operators [28]), the underground economy behind such campaigns and how these parties work together [15], [12]. Of particular interest is the discovery of extensive sharing of resources (e.g., compromised systems, redirection servers, exploit servers) within some categories of malicious activities.

With progress being made in this domain, still our knowledge about the malicious Web Infrastructures is rather limited. Particularly, all these prior studies stay at individual redirection chains that deliver malicious content through a specific channel (e.g., spam [2], twitter [14], malvertising [16]) or lead to a specific type of illicit activities (e.g., drive-by downloads, underground pharmaceutical business). What is missing here is an in-depth understanding of the big picture: what is the topological view of today's malicious Web infrastructures, and how are malicious entities related to each other and the legitimate part of the Web, across different redirection chains, different channels, and exploits? The answer to these questions could help us identify the linchpins of the dark infrastructures, differentiating those valuable to the adversary from those expendable. As a result, we will be able to build more effective and robust techniques that disrupt malicious activities at their common weak spots, without knowing their semantics and relying on any channel/attack specific features such as URL patterns that often can be easily evaded by the adversary. Also, knowing the topological relations among malicious entities, one can make better use of what has already been detected) to discover other malicious parties.

Topologically dedicated malicious hosts. To gain further understanding of malicious Web infrastructures, we study

1081-6011/13 $26.00 © 2013 IEEE
DOI 10.1109/SP.2013.18

nearly 4 million malicious URL paths crawled from different feeds, particularly the topological relations among the hosts involved and their connectivity with legitimate Web entities. Our key finding is the existence of a set of *topologically dedicated malicious hosts* that play orchestrating roles in the infrastructures. From the data we have, all URL paths going through them are confirmed to be malicious. These dedicated malicious hosts have a set of distinctive topological features. First, they seem to have strong connections with each other by forming tight *Host-IP Clusters* (HICs) that share IP addresses and Whois information. Second, they are extensively connected to other malicious parties, hosting over 70% of the malicious paths in our dataset. Finally, they are not found to receive any legitimate inputs, though they may redirect traffic to legitimate parties, e.g., when they cloak.

Our work. Since these topologically dedicated hosts and their HICs play a central role in linking different malicious paths together, it becomes important to detect them for breaking the malicious infrastructures. In our research, we come up with a new topology-based technique designed to catch these hosts without relying on the semantics of the attacks they are involved in. Intuitively, these dedicated hosts are rather easy to reach from the dark side of the Web while extremely hard to reach from the bright side. This observation fits perfectly with the concept of PageRank [3]: that is, they should be popular in the dark world but unpopular outside. Our approach starts with a relatively small set of known malicious HICs as seeds and a large number of known legitimate HICs as references, and propagates their initial scores across a Web topology using the PageRank algorithm to compute legitimate and malicious scores for other unknown HICs. In the end, those highly endorsed by the malicious hosts but way less so by the legitimate ones are identified as dedicated malicious HICs.

Our approach works surprisingly well: in our evaluation based upon 7-month data crawled from Alexa top websites [1], our approach detects about 5,000 new topologically dedicated malicious hosts and over 20,000 malicious host paths that are not captured by existing solutions, at a false detection rate as low as 2%. Our study further reveals the roles, the operation models, and the monetization strategies of these dedicated malicious hosts, particularly those that work as Traffic Distribution Systems (TDSes), which are professional traffic buying and selling systems that manage and keep record of traffic-exchange transactions. Our major detection results and interesting findings include:

• Our algorithm achieves a high detection rate. Even with a small set of seed malicious HICs (5% of the labeled ones), we can discover a large number of other malicious HICs, with an expansion rate of 12 times.

• Our detection algorithm is general across the use of different malicious seeds, including drive-by downloads and Twitter spam in our experiments. It can also detect malicious hosts set up through different attack channels, such as drive-by downloads and scam in our data.

• For the set of dedicated malicious hosts that serve as TDSes, they are much more long-lived than doorways or exploit sites (65 days vs. 2.5 hours). They receive malicious traffic from new attack campaigns over time. Disrupting their operations has more long-lasting effects than taking down doorways or exploit sites.

• Our study shows that even after TDSes are taken down, they continue to receive a large amount of traffic, 10 times more than legitimate parked domains. Such traffic is leveraged by domain owners through parking services to continue to gain revenues from ad networks.

Contributions. The contributions of the paper are summarized as follows:

• *New findings*. We conduct the first study on topologically dedicated malicious hosts, and discover their pervasiveness in malicious Web infrastructures and the critical roles they play. Our study reveals their topological features and the way they are utilized. We show that TDSes play an important role in managing and exchanging traffic flows the adversary uses to deliver malicious content and bring to the light how these malicious dedicated hosts evolve with time and how they are monetized by domain owners through parking services even after their domain names are taken down.

• *New techniques*. We develop a new technique that expands from known malicious seeds to detect other malicious dedicated hosts, based upon their unique features. Our approach works effectively on large-scale real data, capturing a large number of new malicious hosts at a low false detection rate.

Roadmap. The rest of the paper is organized as follows: Section II presents the background information of our research, including how data was collected; Section III discusses a measurement study over the data, which reveals the important role played by dedicated malicious hosts and their prominent features; Section IV describes the design and implementation of our detection technique; Section V evaluates our detection system on its efficacy; Section VI reports a study on all the malicious dedicated hosts we found; Section VII discusses a few issues of our study and potential future work; Section VIII reviews the related research and Section IX concludes the paper.

II. Data Collection

In this section, we explain the data collection process and the methodology we use to label data for our study. This process serves two purposes: (1) It helps us prepare data for building the Web topology graph (Section III); (2) It labels known malicious and legitimate portions of the Web using existing techniques, so that we can study their distinctive topological features for detection. Later in Section IV, we show how we can leverage the topological features learned during this process to detect malicious URLs and hosts not identified before.

A. Data Sources

Feed	Start	End	# Doorway URLs
Drive-by-download	3/2012	8/2012	1,558,690
Warningbird	3/2012	5/2012	358,232
Twitter	3/2012	8/2012	1,613,924
Top sites	2/2012	8/2012	2,040,720

Table I
DATA FEEDS USED IN THE STUDY.

We use four different data feeds to bootstrap data collection. Each data feed includes a set of doorway URLs that we leverage to crawl and analyze the redirection topology. Our data feeds include:

- *Drive-by-download feed:* Microsoft provides us with around 30 million doorway URLs that were found to deliver malicious contents (mostly drive-by downloads) and we sample 5% (1.5 million) from them for study.
- *Warningbird feed:* We download the malicious URL set posted by the WarningBird project [14]. This set includes over 300k suspicious URLs in Twitter spam.
- *Twitter feed:* We run Twitter Search APIs [29] to pick top 10 trending terms every day. We use these terms to collect related tweets and extract all the URLs they contain. This process gives us 1.6 million URLs.
- *Top-site feed:* We gather Alexa top 1 million web sites and update them every week. We obtain 2 million URLs in total, most of which are legitimate.

All together, we have gathered about 5.5 million initial URLs, which serve as inputs to a set of crawlers described below during a 7-month period to collect data.

B. Redirection Chain Crawling

We deploy 20 crawlers, each hosted on a separate Linux virtual machine with a distinctive IP address, to explore the URL redirection paths of these 5.5 million doorway URLs. Each crawler is built as a Firefox add-on, which keeps track of all the network requests, responses, browser events and page content it encounters in a visit. Based on such information, our approach automatically reconstructs a redirection URL path for the visit, which links all related URLs together.

More specifically, such URL paths are built in a mostly standard way, similar to Google's approach [23] except the part for analyzing Javascript. Our approach detects redirections from HTTP status code (e.g. 302), Meta refresh tag, HTML code (e.g., iframe tag) and JavaScript. The dots (URLs) here are connected using different techniques under these different types of redirections. Actually, Firefox gives away the source and destination URLs through browser events when the redirection has HTTP 3xx status code or is triggered by Meta refresh, which allows us to link the source to the destination. For those caused by HTML, we can find out the URL relation according to the Referral field of the destination URL. What gives us trouble is the redirection triggered by Javascript code, which is not specified upfront by any HTTP and HTML fields. This problem is tackled in prior research [23] by simply searching the script code to look for the string similar to the URLs involved in the HTTP requests produced after running the code: if the edit distance between the URL in the request and part of the content the script carries is sufficiently small, a causal relation is thought to be found and the URL of the document hosting the script is connected to the request.

A problem for the approach is that it cannot capture a redirection when the adversary obfuscates the JavaScript code, which is common on today's Web: what has been found is that increasingly sophisticated obfuscation techniques have been employed to evade the detection that inspects redirections [5]. To mitigate this threat, we resort to dynamic analysis, instrumenting all JavaScript DOM APIs that can be used to generate redirections, e.g., `document.write`. When such an API is invoked, our crawler inspects the caller and callee to connect the URLs of their origins.

To increase our chance of hitting malicious websites, we set the crawler's user-agent to IE-6, which is known to contain multiple security-critical vulnerabilities. In addition, to avoid some malicious servers from cloaking to the requests with an empty Referral field [7], we set the Referral field of the initial request for each URL to `http://www.google.com/`. After visiting a web page, the crawler also cleans cookies to avoid tracking.

C. Data Labeling

For each visit, our crawler generates a set of URLs and connects them according to their causal relations, which gives us a set of *URL paths* in terms of URL redirection chains. From URL paths, we further derive a set of *host paths* that keep only host names along the redirection chains. We proceed to label all the crawled URLs, URL paths, and host paths as malicious or legitimate using a combination of existing tools and methods.

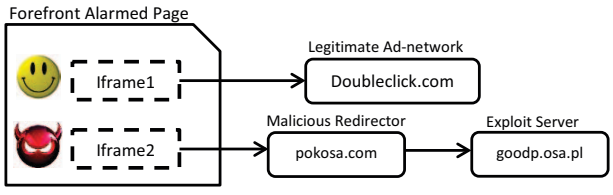


Figure 1. Redirections from a Forefront alarmed Page. The first redirection path is not marked as malicious since it leads to only a legitimate party. The second redirection path is a malicious path as the redirection is generated from an iframe injected by an attacker.

Labeling of malicious URLs and paths. Specifically, we first use Microsoft Forefront, an Anti-Virus scanner, to scan the contents associated with individual URLs encountered

during the crawling.[1] Once a node (i.e., a URL) is flagged by the scanner as containing malicious contents (typically code), the URL is labeled as *malicious*. The data crawled from Alexa top sites and Twitter feeds yield mostly legitimate URLs. The data crawled from the drive-by download and the Warningbird feeds, however, do not always yield malicious URLs for each visit. The reason is that drive-by download doorway URLs are sometimes hosted on compromised hosts, which may have already been cleaned up when we visit them. Therefore, the scan we perform helps avoid falsely marking them as malicious.

Once we label a URL as malicious, we treat all the URL paths going through it as *suspicious paths*. However, not all suspicious paths are malicious. For example, a malicious doorway page may lead to multiple paths, and only one of them leads to exploits. This happens when a malicious doorway page contains multiple URLs, some of which are legitimate and redirect to other legitimate sites (e.g., doubleclick), as illustrated in Figure 1. To avoid marking them as malicious, we further inspect whether there exists another non-doorway URL on a suspicious path also marked as malicious. If so, we label the corresponding path as a *malicious path*. For the paths whose doorway pages directly contain exploit code, we label these paths as malicious without the need of examining other URLs. If all the URL paths corresponding to a host path are labeled as malicious, we label the host path as malicious as well.

	paths	malicious paths	malicious URLs	legitimate URLs
Drive-by-download	17,228,137	3,789,640	238,596	1,079,903
WarningBird	19,858	19,858	5,587	6,871
Twitter	10,429	10,429	464	3,100
Top Sites	339,877	105,428	6,121	23,219
Total	17,598,301	3,925,321	250,627	1,111,104

Table II
DATA STATISTICS AFTER LABELING.

Labeling of legitimate URLs. We also label the remaining URLs that correspond to reputable domains or known ad services as legitimate URLs. To do so, we first cluster the non-malicious URLs based on their domains and manually examine the URL clusters with over 1,000 URLs each. Among these clusters, we identify 19 reputable ones, such as `google.com` and `facebook.com`, and we use them to label legitimate URLs. In addition, we use EasyList [21] and EasyPrivacy [22] to identify ad-networks and trackers. These two lists are also utilized by the popular browser plugin Adblock plus [20] to block ads and tracking scripts. Finally, since URL shorteners (e.g., `t.co`) are extensively used by Twitter users to embed URLs in Tweets, we also identify them using a known list compiled for this purpose [18].

Of course, this labeling process is not exhaustive. All it does is to provide a set of URLs and paths that are confirmed malicious or legitimate based on existing tools (e.g., Forefront, whitelists). The rest of the URLs (78.51%) are treated as *unknown* and our goal is to come up with a methodology for automatically detecting malicious parties from them.

[1]We do not use Google Safebrowsing because a reported malicious URL may be hosted on a compromised site and already be cleaned by the time of our crawl.

III. TOPOLOGY-BASED MEASUREMENTS

In this section, we study the properties of malicious URLs and host paths. We focus on examining the topologies and the connections of malicious and legitimate entities. Our measurements reveal the existence of a set of topologically dedicated hosts that play critical roles in malicious activities. The unique properties of these hosts inspire us to develop a graph-based detection approach, which can capture these hosts without any information about their semantics, e.g., the content they accommodate or the code they run.

A. Hostname-IP Cluster (HIC) Construction

To study Web entity topologies, one natural way is to examine individual URLs or hostnames. However, prior research shows that attackers often register hundreds of malicious hostnames, all pointing to a small set of IP addresses under one domain registrar [19]. Once a hostname is detected, attackers can quickly switch to another one in the pool. From the topology perspective, the signals of such short-lived individual URLs or hostnames may not be strong enough to distinguish them.

Instead, we explore the groups of URLs or hostnames that are controlled by the same attackers. For this purpose, we construct *Hostname-IP Clusters (HICs)* that capture the intrinsic sharing relations between hostnames and IP addresses. The concept of HICs has been used in prior research [33] to detect servers that play central roles in drive-by download campaigns. A problem of their definition is that it is solely based upon the relations between IPs and hostnames, which does not work well on today's Web, where attackers increasingly utilize hosting or cloud services. When this happens, all the hosts running on a cloud server will be clustered together.

Our solution is to use the Whois information [31] to guide this clustering process: two hosts sharing IPs are considered to be related only if their domain names are from the same registrar. Since malicious hosts strongly prefer low-cost, less well known registrars (see Section III-C), this treatment turns out to be very effective. More precisely, our HIC construction process is as follows:

I We assign a unique HIC instance to every hostname.

II We start to merge these HICs in a similar way to that in prior work [33]. The construction process iteratively inspects every pair of HICs. We first compute the overlapping of their IPs. Let IPS_1 be the IP set for HIC H_1, and IPS_2 be that of HIC H_2. H_1 and H_2 are considered to be merged if the Jaccard distance $\frac{IPS_1 \cap IPS_2}{IPS_1 \cup IPS_2}$ is larger than a threshold T_{IPS}. Similar

to [33], we set this threshold to 0.5, to accommodate the IP variations caused by content-distribution networks (CDN) and fast-fluxing [10]. Besides this criterion, we take an additional step to check their Whois information. Only if their registrars are also identical can we merge them together. The above process iterates until no HIC pairs can further merge.

Figure 2 illustrates this process. HIC_1 and HIC_2 can be merged since their IP address overlapping is 60% and they have the same registrar. HIC_3 is not merged with any other HICs because its registrar is different from others.

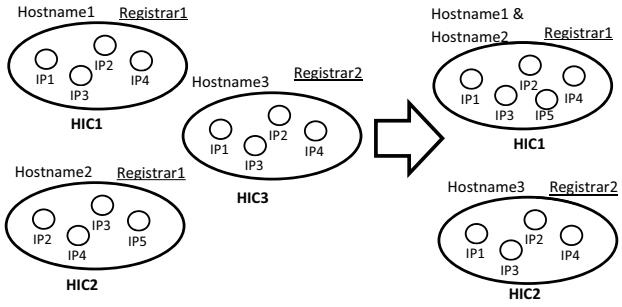


Figure 2. HIC generation process.

B. Topologically Dedicated Malicious HICs

All together, we obtain 1,951,313 HICs using the above method from our data. Among them, 15,273 are found to *only* host confirmed malicious URL paths (and the corresponding host paths) in our datasets (collected over a 7-month period). This topological property differentiates them from other HICs, which contain at least one URL path that we cannot confirm. We call the former *dedicated malicious HICs* and the latter *non-dedicated malicious HICs*.

These dedicated HICs apparently play a critical role in the malicious activities: they are attached to 76.2% of the malicious paths across all the data sources in Table II. Although we have no ground truth about whether the dedicated malicious HICs are indeed set up by malicious parties, we find that their hostnames usually exhibit patterns of domain rotations and that they are often registered under unpopular domain registrars[2]. Table III lists the top 10 (ranked by the number of paths going through) dedicated malicious HICs in our datasets. Such observations suggest that these HICs may correspond to dedicated hosts that are set up for just malicious uses, e.g., "central servers" for drive-by download campaigns [33].

C. Graph Properties of Dedicated Malicious HICs

When we examine the inter-connections among HICs, we find that these dedicated HICs are not isolated. Instead, they tend to connect to each other. To understand their connectivity, we build an *HIC graph* by linking two HICs with a directed edge if there is an URL redirection between their hosts. In total, we have 1,951,313 HIC nodes and 9,058,597 edges on the HIC graph.

Closely examining these dedicated malicious HICs, we find that they are highly intertwined: among 15,273 dedicated malicious HICs, 12,942 (84.74%) are located on a fully connected subgraph. The dedicated malicious HICs are also intensely connected with other non-dedicated malicious HICs: 80.40% of non-dedicated malicious HICs are directly or indirectly connected to at least one dedicated HIC. This observation indicates that the dedicated malicious HICs are quite easy to reach from the "dark" world. Starting from a few malicious URLs and following their redirect chains, you may easily reach some dedicated malicious HICs.

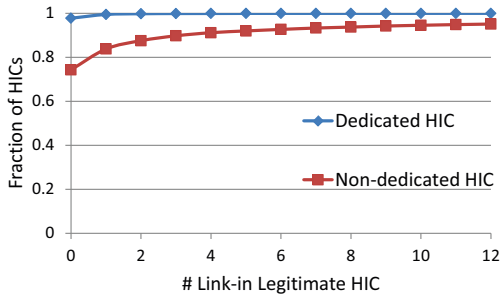


Figure 3. CDF of the number of Legitimate Link-in HIC between Dedicated HICs and Non-dedicated HICs

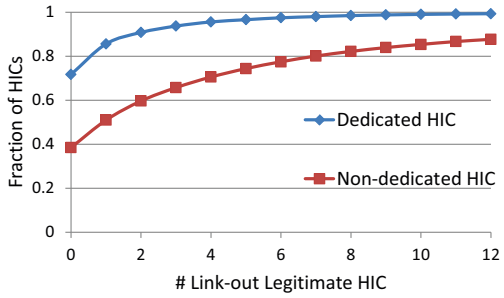


Figure 4. CDF of the number of Legitimate Link-out HIC between Dedicated HICs and Non-dedicated HICs

In contrast, these dedicated malicious HICs rarely receive traffic from legitimate or unknown parties (labeled by the methodology in Section II-C), even when these legitimate parties do appear on malicious paths. In terms of such a "link-in" relation, the dedicated malicious HICs are more remote to legitimate parties than non-dedicated malicious HICs. Figure 3 shows that 97.75% of the dedicated malicious HICs do not receive any traffic redirections from legitimate HICs. For the rest 2.25% of dedicated malicious HICs that do, they mostly correspond to malicious entities that have

[2]According to [6], the five best domain providers are NameCheap, 1&1, Go Daddy, Name and Gandi.

Rank	Hostnames	Registrar
1	lsbppxhgckolsnap.ru, vznrahwzgntmfcqk.ru, ...	NAUNET-REG-RIPN
2	viagrabuytoday.com, buycialistodoors.com, ...	INTERNET.BS CORP
3	searchstr.com, ssrsearch.com	INTERNET.BS CORP
4	sqwlonyduvpowdgy.ru, qlihxnncwioxkdls.ru, ...	NAUNET-REG-RIPN
5	tadalafil-mastercard.ordercialisonlineus.com, viagra-brand-viking.cialisshopsale.com, ...	INTERNET.BS CORP
6	soxfurspwauosdis.ru, iqsxbaoyzweerppq.ru, ...	NAUNET-REG-RIPN
7	freshtds.eu	PDR Ltd.
8	puvbgoizrqsxsxzq.ru, fkffqgkqfqdxekvq.ru, ...	NAUNET-REG-RIPN
10	michaelmazur.net	TUCOWS.COM CO.

Table III
TOP RANKED HICS

infiltrated legitimate ad networks (e.g., Doubleclick) and receive traffic from them [16]. By comparison, 25.70% of non-dedicated malicious HICs receive traffic redirections from other legitimate HICs. This observation shows that compared to legitimate or non-dedicated malicious HICs, the topologically dedicated malicious HICs are much harder to reach from the bright side of the Web.

In terms of the "link-out" relations, dedicated malicious HICs are less likely to redirect traffic to legitimate HICs. This usually happens when those malicious parties cloak. Figure 4 shows that 28.30% of the dedicated malicious HICs redirect their visitors to legitimate hosts, compared with 61.53% of non-dedicated malicious HICs that do the same.

The graph properties of these dedicated malicious HICs show that they are well connected and easy to reach from known malicious URLs, but they are much harder to get to from legitimate ones. This observation provides strong implications for developing the right technique to detect them. Particularly, the well-known PageRank algorithm fits well with such topological properties, and therefore we adopt it in our research to detect those hosts without relying on their semantic information. Note that what we focus on here is *dedicated* malicious HICs. Those non-dedicated, particularly compromised hosts, may not have such graph properties. As a result, the PageRank approach may not be applicable to find them. In the next section, we explain the this detection method in detail.

IV. DETECTING DEDICATED MALICIOUS HICS

Our measurement study shows that there exist a set of topologically dedicated malicious HICs. These dedicated HICs are important because they appear to be the linchpins of malicious Wed infrastructures, linking to 76.2% malicious host paths across all the datasets we have crawled over a 7-month period. Since all the paths going through the corresponding hosts are malicious, detecting such dedicated malicious HICs can help us discover many other malicious hosts including doorways, redirectors, and others.

To detect such dedicated hosts, we explore the unique topological features of these HICs. Of most interest are their strong connections with other malicious hosts, and their tenuous relations with legitimate hosts (Section III-C).

Compared with prior approaches [5], [11], [33] that rely on the contents (e.g., URL patterns) or semantics (e.g., drive-by downloads) of specific types of attacks or specific data sources for detection, our approach utilizes *only the topological information of malicious Web infrastructures*. An important advantage of this approach is that it works on different types of attacks and different sources of data, regardless whether the attack is drive-by download, scam, or is carried through spam tweets [14] or malvertising [16], as long as it exhibits the same topological properties used for detection, which in our case is the connectivity of dedicated malicious HICs. Moreover, such an approach can be more difficult to evade by active adversaries: the dedicated malicious HICs could cloak to the crawlers, redirecting traffic to `google.com`, but they cannot easily change their connection structures to receive more traffic from legitimate hosts or less traffic from other malicious hosts.

A. PageRank-based Detection

The connectivity features of the dedicated malicious HICs are well captured by the concept of *PageRank* [3], a technique widely used to evaluate the importance of web pages. In the web site ranking scenario, a web page is considered to be important and therefore has a high rank if it is well connected, easy to reach from other (randomly-selected) pages. This rank is computed by propagating the initial score of each web page across a directed hyperlink graph and iteratively updating the page's score based on the ranks of the pages that link to it. This idea has also been applied to detect web spam pages [9], comment spams [32] and spammers on social graphs [4].

In our case, what makes the dedicated malicious HICs unique is their unbalanced connections from (dedicated or non-dedicated) malicious HICs v.s. those from legitimate ones. Using PageRank as the yardstick, malicious HICs get high ranks from the dark Web and low ranks from the bright side of the Web. Therefore, our idea is to compute two different ranks and use them together for detection.

Specifically, each HIC on the HIC graph maintains a pair of scores, the good one that models its popularity among legitimate hosts, and the bad one that describes its rank among malicious hosts. The use of both scores help balance the good traffic that malicious hosts receive, for example, when DoubleClick is used to forward traffic to a malicious ad network [16], as well as the bad traffic that legitimate hosts gets, for example, when a malicious

host cloaks, redirecting a visitor's traffic to `google.com`. Given the fact that the overwhelming majority of the HICs are legitimate and tend to connect to each other and to even non-dedicated malicious HICs (that may correspond to compromised hosts), not only truly legitimate HICs but also those non-dedicated malicious ones tend to have much higher good scores than their bad scores. On the other hand, those whose bad scores are high but good scores are low are very likely to have played key roles connecting different malicious parties, while being separated from the legitimate world. In other words, they are likely dedicated malicious HICs. Thus if the bad score of an HIC is above a preset threshold α and the ratio of the good score to the bad score is below a threshold β, we consider this HIC as a dedicated malicious HIC. We discuss the settings for α and β in Section V-A.

Specifically, our approach runs the PageRank algorithm on the HIC graph described in Section III-C. The PageRank scores are computed by iteratively applying the following operations on each HIC on the graph, starting from a set of initial scores assigned to these HICs. The operation updates the score (bad or good) $PR_{i+1}(A)$ of an HIC A at the $i+1$ iteration using the score of another HIC X that has an directed edge originating from X to A at the ith step:

$$PR_{i+1}(A) = 1 - d + d \sum_{X \in M(A)} \frac{PR_i(X)}{L(X)} \qquad (1)$$

where d is a damping factor, $M(A)$ is the set of HICs pointing to A, and $L(X)$ is the number of outgoing edges from X to A.

Prior research [16] shows that malicious hosts on a path tend to stay together and those further away from them are less likely to be malicious. To model this observation and further control the level of the malicious rank (score) a non-dedicated host (e.g., a compromised website) receives, we adjust the scores of individual HICs, after the PageRank iterations, as follows. Consider a node A, which stands i hops away from its closest known bad node (see Section IV-B), its PageRank score s (good or bad) is adjusted to $s \times \theta^{i-1}$, where θ is a constant value. In our research, we set $\theta = 0.8$ when computing a bad score, which exponentially weakens as a host is further away from a malicious node. Therefore, only those very close to the dark world can receive a high bad score, as such a reputation does not propagate too far. In contrast, we use $\theta = 1$ for computing a good score, allowing the influence of a good host to propagate undamped throughout the HIC graph. In this way, any host (legitimate or not) with substantial connections to the legitimate world tends to get a high good score.

B. PageRank Score Settings and Propagation

To bootstrap the initial scores, we utilize Alexa top 60,000 sites and EasyList sites to assign initial good scores and Microsoft Forefront to find those that need to be given non-zero initial bad scores. Both known good and known bad hosts receive 1 as their initial good and bad scores respectively. Others just get 0 to start with. On the HIC graph, an HIC's good/bad scores are simply the aggregate scores of their corresponding hosts. For example, an HIC with n known legitimate hosts (on the whitelist) and m known malicious hosts (detected by the scanner) get an initial good score of n and a bad score of m.

These initial scores are propagated across the HIC graph through iterated updates of each HIC's scores using Equation 1, except that only part of the score $PR_i(X)$ is used to update $PR_{i+1}(A)$, based upon the *weight* of X's outbound link to A. This weight is determined by the ratio between the number of hosts A has and the total number of the hosts within all the HICs receiving inputs from X. In other words, if there are S hosts within the HICs getting traffic from X, and S_A of them are in A, we use $\frac{S_A}{S}$ to update $PR_{i+1}(A)$. Figure 5 illustrates how this update works.

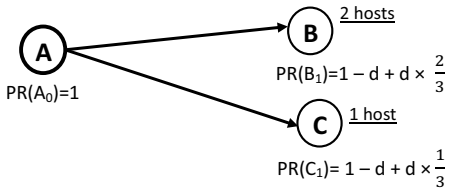


Figure 5. Weight distribution. Assuming A has an initial score 1, child B will receive a score $1 - d + d \times \frac{2}{3}$ and child C will receive a score $1 - d + d \times \frac{1}{3}$, as the number of host names within B is two times that of C.

C. Dedicated malicious HIC identification

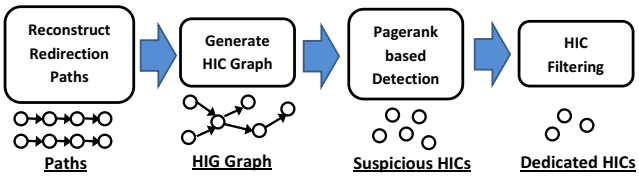


Figure 6. Detection Framework

After rounds of iterations, the scores of individual HICs converge. At that point, we can pick up a set of possibly dedicated malicious HICs whose bad scores are high and whose good to bad score ratios are low according to the thresholds α and β. To mitigate false positives[3], we conservatively remove from the detection results all the HICs that involve either a host name on the lists used for bootstraping good scores or a host name with a doorway URL discovered by our crawler. The doorway URLs are used here as a heuristic because they often correspond to compromised web sites as

[3]Note that false positive here refers to the situation that a non-dedicated malicious HIC or a legitimate one is labeled as dedicated malicious.

opposed to dedicate malicious sites. Figure 6 summarizes the entire processing flow of our detection.

V. Evaluation and Result Analysis

In this section, we report our evaluation of the topology-based detection method. We first describe our experiment setup, and then elaborate our experiment results, including a comparison study between our technique and the existing approaches that utilize simple topological features, in-degree, for ranking malicious websites. Finally we analyze detected HICs to understand their roles (e.g., exploit servers, redirectors) in malicious activities.

A. Evaluation

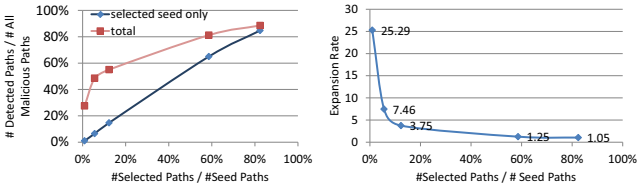
Experiment settings. We run the PageRank algorithm on the constructed HIC graph as specified in Section III-C with a threshold for bad score $\alpha = 0.9$. Since malicious hosts could redirect visitors to legitimate services, e.g., when they cloak, this lower-bound threshold (which is pretty high for a legitimate host) conservatively ensures that these legitimate parties will not be misclassified as malicious.

For the threshold β that records the ratio between good and bad scores, we select it according to the number of HICs that have non-zero initial scores. Suppose S_G HICs have non-zero good scores and S_B HICs have non-zero bad scores during bootstrap, the threshold β will be selected as $\beta = \frac{S_G}{S_B}\gamma$, where γ is a parameter and we set it to 10. This definition reduces the impact of the particular input dataset on the detection results.

Our HIC graph contains in total 60,856 HICs (91,464 host names) with non-zero initial good scores, using the Alexa top 60,000 site list and EasyList described in Section IV-B. We also have in total 52,847 HICs (106,872 host names) marked as malicious by Forefront. In our experiments, we randomly select a varying subsect (1%, 5%, 10%, 50%, and 90%) of known malicious HICs as seeds for setting the initial bad scores, simulating scenarios where we have knowledge about different numbers of confirmed malicious HICs for detection. In each case, β will be set differently based on the number of the bad seeds. For all experiments, we run 20 PageRank iterations to propagate scores. Note that the labeled seed sets may not be clean, as many malicious hosts cloak or have parked. We consider such cases common in practice, as it is in general hard to obtain clean, noise-free seed data.

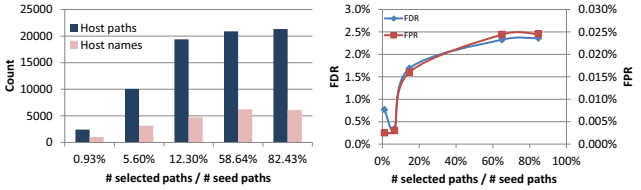
Metric	Definition
Recall	$N_{TP}/(N_{TP}+N_{FN})$
False Detection Rate (FDR)	$N_{FP}/(N_{FP}+N_{TP})$
False Positive Rate (FPR)	$N_{FP}/(N_{FP}+N_{TN})$

Table IV
Metrics Definition. N_{TP} is the number of true-positives. N_{FN} is the number of false-negatives. N_{FP} is the number of false-positives. N_{TN} is the number of true-negatives.



(a) (b)

Figure 7. (a) Recall. (b) Expansion rate.



(a) (b)

Figure 8. (a) New findings. (b) FDR/FPR.

Results. We use several metrics to evaluate our results (see Table IV). First, we evaluate the recall of malicious host paths by examining the percentage of all confirmed malicious paths being correctly detected by varying our seed data size. Note that once we detect an HIC as malicious, we treat all the host paths going through this HIC as malicious. Figure 7 shows that using 5% known (dedicated or non-dedicated) malicious HICs as seeds, which correspond to 33,547 (6%) malicious host paths, we can detect 242,776 (48.59%) other malicious host paths[4], resulting in over 7 times of expansion in detection. The recall and the expansion rate gradually converge as we increase the percentage of seed malicious HICs. This trend is expected as we approach the limit of the recall that we can achieve.

In addition to detecting already labeled malicious paths, our method can also detect malicious paths that are not identified by existing solutions. Figure 8 (a) shows that our detector discovers more than 20,000 new malicious host paths using 90% labeled malicious HICs as seeds. These newly detected paths are mostly crawled from the Top-site and Twitter feeds, and they go through 6,080 unique host names. Through manual analysis, we find that most of these cases are not detected by Forefront because they either use HTTP status code (e.g., 302) for redirection,without relying on the use of malicious code, or their script signatures are not included by Forefront.

Finally, we evaluate our false detection and false positive rates in Figure 8 (b). For newly detected malicious HICs, we use several methods in combination to validate them, including comparing against the Google Safebrowsing list, performing content clustering analysis and URL pattern analysis (Section V-B). For the small set of remaining cases that cannot be resolved using these methods, we manually

[4]This result already excludes false-positive paths.

go through each case. The validation process shows that our false positive rate (FPR) is very low, less than 0.025%. The false detection rate (FDR) is also as low as 0.34% when using 5% seeds. Since our seed data may not be clean, the false detection rate grows with using more seeds, and it reaches 2.36% in the worst case.

Detection with seed rolling. To further improve the detection coverage, we repeat the detection process by feeding the set of detected results as new seeds to the system and re-calculate the PageRank scores. This "seed rolling" process can iterate a few times. To demonstrate the value of seed rolling, we use 5% known malicious HICs as seeds and iterate our detection for 3 times. For each new round of detection, the seed data are appended with doorway hosts (and HICs) that link to the detected HICs in the last round. We use only doorway hosts to pick new seeds. This is a conservative option because in a majority of cases, a malicious path is always associated with a malicious doorway (in addition to other malicious hosts along the redirection chain).

Figure 9 shows that after 3 iterations, the detection coverage can be significantly increased: the number of detected host paths is increased from 242,776 (48.59%) to 361,675 (72.38%), resulting in over 12 times of expansion. More prominently, it helps us discover 30,358 new host paths. In the meantime, the false detection rate (FDR) is bound to 2.63%, still low for practical use.

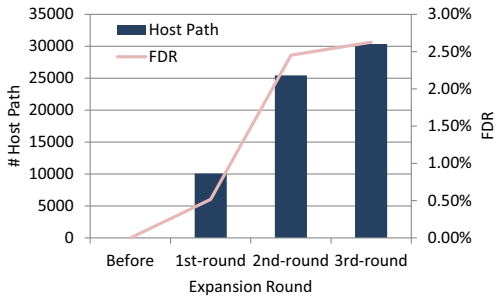


Figure 9. New Findings and FDR with Seed Rolling

Comparison with the in-degree based approach. Previous studies (e.g., [30], [27]) have proposed simple topological features such as "in-degrees" for ranking malicious sites. Intuitively, if a site receives traffic from many other malicious sites, it is also suspicious. For comparison, we also implement an in-degree based approach for detection based on the HIC graph. For each HIC node, we measure its malicious in-degree from other known malicious HICs. If the malicious in-degree is above a threshold, we detect this HIC as also malicious.

Similarly, we use 5%, 10%, 50% and 90% of the known malicious HICs as seeds and examine the false detection rate under different requirements of recall. As comparison,

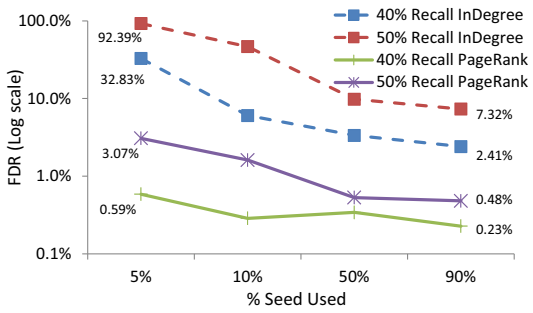


Figure 10. Comparison between in-degree based approach and our PageRank based approach. Both approaches are evaluated under the requirement of 40% and 50% recall of malicious host paths.

we change the threshold of the good/bad score ratio for PageRank based detection to adjust the recall rates accordingly. Figure 10 shows that the in-degree based approach causes much larger FDR than our approach. The reason is that many legitimate sites, such as `google.com`, may also frequently receive redirected traffic from malicious sites. Simply picking HICs by in-degree will mistakenly identify them as malicious.

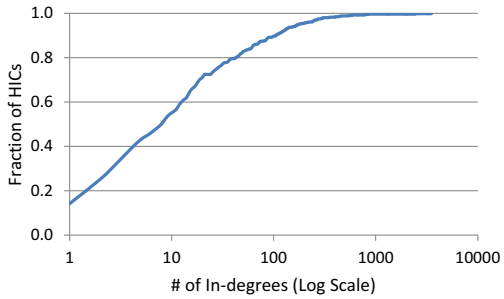


Figure 11. CDF of the in-degree distribution of the HICs that are detected by our approach (using 5% seed).

In addition to causing high false positive rates, the in-degree based approach will also miss malicious HICs that have small in-degrees. Figure 11 shows the in-degree distribution of the set of HICs detected by PageRank. With 5% seeds, our detector identifies 508 malicious HICs and their in-degree distribution is quite diverse, ranging from 1 to 3,514. Therefore, using our approach, we can detect not only big central servers, but also dedicated smaller servers in malicious activities.

Detection with different seeds and result sharing. One practical question for our approach is the sensitivity of its detection results to the use of different types of seeds. Whoever using our tools may wish to bootstrap detection based on any malicious data feeds that they may obtain. To answer this question, we compare the detection results using two different types of feeds. The first type is the drive-

by download feed, and the second type is the combined Top-site and Twitter feed. For both types of feeds, we use Forefront to scan the crawled data and identify malicious HICs as initial seeds. We then compare the results using 5% seeds derived from the drive-by-download feed and all the seeds derived from the Top-site and Twitter feed, so that the number of initial seeds are roughly similar. Table V presents the results. These two different seed sets result in similar numbers of detected host names, and the FDRs from both sets are low. This result shows that our approach is general across different types of seed data.

When we compare the detection results, we find that although we obtain these two seed sets through different channels, they have large overlaps in the set of detected results, among which 29.91% host names and 37.09% host paths are detected in both cases.

	N_D	N_T	$\frac{N_D \cap N_T}{N_D \cup N_T}$
Host names	5,458	5,157	29.91%
Host paths	236,763	118,544	37.09%
FDR (host paths)	0.34%	0.75%	-

Table V
COMPARISON OF THE RESULTS DETECTED USING 5% SEEDS DERIVED FROM THE DRIVE-BY-DOWNLOAD FEED AGAINST THAT FROM THE TOP-SITE AND TWITTER FEED. N_D IS THE NUMBER USING THE DRIVE-BY-DOWNLOAD FEED AND N_T IS THE NUMBER USING THE TOP SITE AND TWITTER FEED.

We also find that using the seeds from the drive-by-download feed, we could detect scam attacks crawled from the WarningBird feed. Using all of the bad seeds from the drive-by-download feed, we identify 6 overlapping malicious host names and 4,125 (56.21%) overlapping malicious host paths crawled from the WarningBird feed (See Table VI). These malicious hosts are not directly flagged by Forefront but are detected through PageRank. Most of the detected paths go through (`188.72.233.144`), which is powered by an open-source tracker kit. Many scam pages from the WarningBird feed redirect traffic to this host. This observation indicates that attackers are already leveraging dedicated services from different channels and using them for different purposes.

	Total	PageRank	Scanner
Host names	4,456	6	0
Host paths	7,338	4,125	0

Table VI
DETECTED HOST NAMES AND HOST PATHS THAT OVERLAP WITH THE WARNINGBIRD FEED, USING THE DRIVE-BY-DOWNLOAD SEEDS.

B. Detection Result Analysis

The low false positive rates of our detection suggest that those captured HICs are likely dedicated malicious HICs. One important question then is "what are the roles played by these hosts in the malicious Web infrastructures?" To answer this question, we focus on the set of dedicated malicious HICs and categorize their URLs based the roles that they play in an attack. We use several methods to perform such categorization, including:

Role	URLs	URL paths
exploit	13,216	89,019
click-fraud	5,955	36,761
scam	29,411	632,644
fakeav	1,604	1,805
other	1,031	90,962
redirector	286,275	2,479,695
unknown	69,062	526,952
total	406,553	3,088,741

Table VII
ROLES OF URLS.

- *Forefront reporting:* When Forefront detects a piece of malicious code, it also provides a type code such as `Exploit:JS/Blacole.BK`, `Rogue:JS/FakePAV`, which can be used to infer its role.
- *Content and URL clustering:* We cluster page contents based on their DOM structures as well as URLs based on the URL patterns. We then manually examine large clusters to determine their categories.
- *Safebrowsing reporting:* We check if an URL is also reported by Safebrowsing, which sometimes provides hint about its role, e.g., malware, or phishing.

Table VII shows the role breakdown of the set of malicious URLs associated with the dedicated malicious HICs. We find that the dedicated malicious HICs are tied to a variety of roles including exploit servers, scam hosts, redirectors, etc.

Among these categories, redirectors are of a dominant fraction (70.4%) and they play active roles on 80% of the malicious paths. Among these redirectors, 31.98% of them are hosts that run Traffic Distribution Systems (TDS), a suite of traffic buying and selling tool kits that are extensively used in underground ecosystems. We discover these systems using the URL patterns of known TDS tool kits [7]. Several large TDSes that carry obvious URL patterns (hence not including less famous TDS services) alone count for 56.25% of the malicious paths. Compared with exploit servers, redirectors are less well studied and little has been known about their operations. Given the important roles that they play in malicious activities, in the next section, we report an in-depth study on these TDSes to understand their characteristics and monetization strategies.

VI. IN-DEPTH STUDY ON TDS

As discussed in Section V-B, over 50% of the malicious paths turn out to go through Traffic Direction Systems [7], which are underground traffic brokers who buy from traffic generators (e.g., malicious doorways) and sell to traffic consumers (e.g., exploit servers). Such services facilitate traffic exchanges between malicious parties, allowing attack executors to dedicate their resources to running and managing their attacks rather than wasting their time and resources

on procuring traffic. Although such systems have been there for years, relatively little is known about how they operate in the wild, compared with other types of dedicated hosts such as exploit services [8].

In our research, we focus on those TDSes as the representative of topologically-dedicated malicious hosts. This section reports the most interesting observations we made, particularly, our discovery of their important roles in malicious activities (connecting to over 52.67% of doorway URLs, Section VI-A), their surprisingly long life span (65.21 days of median life time, Section VI-C) and the monetization activities involving them even after they are parked (receiving possibly 10 times as much traffic as legitimate parking domain does, Section VI-D).

A. Landscape

To understand how these TDSes work, we first need to find out what tool kits they use, how popular they are, where they get traffic from and where they send traffic to.

Feed	Doorway URLs(%)	Malicious Paths(%)
Drive-by-download	53.85	58.06
Warningbird	0.93	0.34
Top sites	34.51	10.39
Twitter	26.25	1.40
All	52.67	56.25

Table VIII
TDS PREVALENCE PER FEED

As described in Section V-B, we identify TDSes from their URLs, which bear unique patterns of the tool kits they are built upon. A recent report [7] shows that just like the kits extensively used by exploit services, there are a whole set of off-the-shelf TDS kits that can be conveniently utilized by adversaries to manage, administrate and log traffic coming in and out of their systems. Among them, the most popular ones are Sutra TDS, Simple TDS, and Advanced TDS. Using known URL patterns, we find that the Sutra TDS kit is the most popular one, covering 71.02% of the TDS URLs in our set. Sutra is not a free kit, whose price ranges from $200 to $270, but it has a wide range of supported features [13]. The second most popular kit is the Simple TDS, an open source kit that covers 10.19% of the TDS URLs.

Prevalence. In Section V-B, we show that TDSes have taken a lion's share among all the detected dedicated HICs: 52.67% of doorway URLs are found to send web traffic to these TDSs. Table VIII further illustrates the important roles they play in funneling traffic from different data sources. Except Warningbird, all data sources have significant numbers of URLs that lead to TDSes. We also find TDSes to be prevalent in paths not alarmed by ForeFront.

Inbound Traffic. Over 97.1% of TDSes receive web traffic directly from doorways while only 6.37% of them get traffic from non-doorway redirectors. For the doorways that bring traffic to TDSes, some of them are intentional, e.g, adult sites. Many others are compromised sites.

Figure 12 shows the cumulative percentage of new doorway domains and IP addresses that bring traffic to TDSes during our crawling period. We can clearly see a step function, which shows that doorway domains are often compromised and set up by attackers in batches and correspond to different attack campaigns. Thus, studying the incoming traffic to TDSes can also be used to detect attack campaigns. It is also worth noting that there is sharing of IP addresses among compromised doorway domains. In total there are 18,369 new doorway pages and 12,711 unique IP addresses.

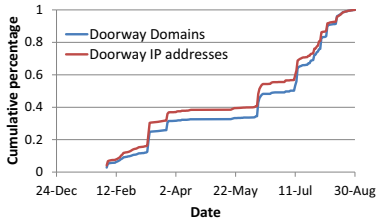


Figure 12. Cumulative percentage of new doorway domains and IPs that redirect traffic to TDSes during our 7 months of crawling.

TDS Status		TDS URLs(%)	TDS hosts(%)	TDS Paths(%)
Inactive TDSes	Parked	69.66	23.9	51.07
	Suspended	12.69	12.25	4.91
	Appear to be down	8.58	55.65	2.58
Active TDSes	Redirecting to search engines	0.03	1.14	0.004
	Redirecting to None search engines	15.50	21	41.43
All TDSes		126,180	3,168	2,211,291

Table IX
LANDSCAPE OF TDS OPERATION. PERCENTAGES ARE CALCULATED TO THE TOTAL OF ALL TDSS

Outbound Traffic. During our crawling, we find that some TDSes would not redirect traffic further to other websites. We call them *inactive* TDSes. These inactive TDSes can be suspended for resell, parked, or appears to be down (not resolving to an IP address or send back error response code). Table IX shows the breakdown among the inactive TDSes. A majority (69.66%) of the TDS URLs are parked, 12.69% are suspended and 8.58% appear to be down by either giving us error codes or not resolving.

Note that the TDSes giving us error response codes may not be truly inactive. Actually, it is reported that TDSes tool kits can perform IP filtering [13] [7]. This finding has also been confirmed by our analysis of the Simple TDS kit. Indeed, we have observed that among the TDSes that have been crawled multiple times, some of them lead to exploit servers when we first crawl them, but later give 404 responses or forward the crawler to `google.com` on subsequent visits. A more detailed look into the intersection between the live TDSes and the TDSes of other categories is provided in Table X where we find that 23.08% of the live TDSes are taken down in subsequent visits. Note that our

subsequent visits happen only when the same TDS appears in another path for a later crawl.

In parked hosts	21 (3.11%)
In suspended hosts	0
In hosts not resolving	156 (23.08%)
In hosts responding with error codes	205 (30.33%)

Table X
ACTIVE TDSES APPEARING IN OTHER CATEGORIES

We further study attack types associated with the active TDS paths, except those that cloak by leading to search engines. 49.11% of them are found to connect to exploit servers, 3.40% go to scam sites, and 60.80% of them redirect to the places whose attack types cannot be confirmed. The percentages here do not add up to 100%, as some TDS hosts lead to multiple types of attacks.

B. TDS hosting infrastructure

As discussed before, we study TDSes as a representative of topologically dedicated hosts. For this type of services, a question we hope to answer is how these services, at least their domains, are hosted. In our research, we find that these TDS hosts extensively utilize free web services like free domain providers and dynamic DNS (DDNS) providers. DDNS providers such as `freetcp.com` let users register sub-domains (e.g.,aaa.freetcp.com) and resolve them to the users' own IPs. Similarly, free domain providers such as `uni.me` also give away sub-domains for free, but unlike DDNS services, they offer free hosting on their IP addresses.

To quantify the TDSes hosted by different infrastructures, we utilize a few known lists to identify free domain providers, DDNS providers, and URL shorteners. The lists are downloaded from `malwaredomains.com`, which are updated on a daily base. Using these lists, we find that 26.44% of the TDSes use DDNS, 14.39% use free domain providers and 0.7% use URL shortners. Please note that these figures are lower bounds because the lists could be incomplete. The actual number can be higher.

Additionally, we find that many TDSes share IP addresses. The top 12% of the IP addresses cover 21.5% of the TDSes. More interestingly, many of these TDSes' IP addresses share IP prefixes: the top 5 (out of 131) autonomous system numbers (ASN) associated with these TDSes belong to a few small cloud and hosting service providers, as illustrated in Table XI.

#	ASN#	ASN Name	Country	Number of IPs
1	16265	LEASEWEB	NL	45
2	24940	HETZNER	DE	33
3	28753	LEASEWEB-DE	DE	19
4	44050	PIN-AS	RU	13
5	21788	NOC-Network	US	10

Table XI
TOP 5 ASNS HOSTING TDSS

C. TDS malicious life time

During our investigations of TDSes and their operations, we observe that, unlike exploit domains, they tend to live long before they're detected. In this section we attempt to estimate their life times.

Data source. We leverage the "PassiveDNS" data set, which contains DNS records collected by the Security Information Exchange (SIE) [26] since April 2010. This data source has also been used in prior research [8], where raw DNS records for two months were used. For our study, we have an aggregated list of records over a 2 year period through the SIE API [25]. Each record contains two time stamps to indicate the first and the last times the record has been observed to have the same value (i.e. the Rdata field in a DNS packet).

To identify the *malicious* life span of a given host, we find that it is not enough to only consider the time between the first and the last valid `A` record (i.e. IP address lookup) as the prior work does [8]. This is because that even after a malicious domain is taken down and has no valid `A` records for a while, it could be acquired by a domain registrar who wants to sell it. During the reselling period, the domain has a valid DNS record and is resolvable. More over, after a domain is repurchased, it may become legitimate. Therefore, the simple way of computing the duration between the first and last valid DNS records would just estimate the *up time* of a domain, rather than the *malicious* life time.

To avoid the overestimates incurred by the above approach, we take a more conservative approach to just look for a *lower* bound of a host's malicious life span. That lower bound is estimated based on the time interval between the first and the last observed `A` records that carry at least one of the IP address(s) of a given host as discovered by our crawler when it is associated with malicious activities. As a result, what we get is the malicious life span for the TDSes whose IPs are known to our crawlers. In total, there are 1,334 hostnames for such TDSes.

Observations. Querying the "PassiveDNS" dataset for those TDSes, we retrieve the DNS records of 1308 hosts. Table XII lists the malicious life times per hosting type (DDNS, Free Domain hosts and others). The standard deviations in all categories are quite high and thus we also consider their medians. The median malicious life time for the hosts running on possibly dedicated domains (in the "neither" category) is 65.21 days, which is much higher than most malicious domains reported in the literature, e.g. 2.5 hours [8] . We observed some TDS hosts live for years. For example, `tr-af.com` started resolving to the same IP address since *11th Jan, 2011* and it is still up. Also interestingly, we find that hosts using DDNS tend to be taken down sooner than those using free domain providers. We believe this is due to the difference in the ways they operate. As DDNSs just provide DNS services, when they are noticed of malicious domains, they can simply choose to not resolving IPs for

such domains. For free domain providers, they provide both DNS and hosting service. Therefore, when they are notified of malicious domains and they see these domains still have a lot of incoming traffic (details in the next sub section), they could choose to monetize such traffic by redirecting such traffic to ad networks.

	DDNS hosts	Free domain hosts	Neither
Mean	43.20 d	105.32 d	138.59 d
Standard Deviation	99.02 d	128.74 d	200.88 d
Median	5.75 d	61.76 d	65.21 d
Total # of TDSes	371	154	745

Table XII
TDS MALICIOUS LIFE TIMES IN DAYS. NEITHER INDICATES HOSTS RUNNING ON DEDICATED DOMAINS (I.E. NOT USING DDNS OR FREE DOMAIN PROVIDERS)

D. TDS Parking

In Section VI-A, we discovered that the TDS hosts in 51% of TDS paths were parked. Such a high presence of domain parking warrants a closer look into the motivation behind such behavior which we elaborate on in this section. Before we study how these malicious domains are parked, we first review how regular domains are parked.

Legitimate domain parking. Parking services offer a way for newly acquired, underdeveloped domains, or domains reserved for future use to monetize their traffic through advertising. Domains can be parked either by setting the authoritative name server `NS` record to point to that of the parking services or using a redirector to send traffic to the parking services. Upon arriving at the parking services, there are two ways for monetization. The traditional way is to navigate a visitor to a page filled with sponsored contextual ads. Recently, a new model called ZeroClick was introduced to redirect the visitor directly to an advertiser's Web page in which case the visitor never lands at the parked domain.

Per parking service agreements [24], parking services allow only real user type-in traffic. They do not allow third party sites to redirect to parked domains. As legitimate parked domains are not actively in use, they typically do not receive a lot of traffic, except those that share very similar spellings with well-known sites, who get traffic through users typos.

TDS parking. TDSes receive redirections from many doorway pages, so they naturally have a lot of incoming traffic. As doorway pages usually reside on compromised domains owned by different entities, it is hard to clean all of them quickly. Therefore, even if an attack is detected and the corresponding malicious domains are taken down, there could still be a lot of traffic leading towards TDSes.

Domainers (i.e. domain owners) smartly leverage such a rich source of traffic by purchasing those suspended domains and monetizing their traffic through domain parking. Indeed, our dataset shows that 51.07% of the paths lead to parked TDSes. Using the "PassiveDNS", we identify 642 parked TDSes by checking whether the NS (Name Server) record or the hosting IP belongs to a parking service. The top 10 parking services and the number of TDSes parked with them are shown in Table XIII. Among them, `Bodis` Parking is the most popular parking service targeted by the new domain owners of TDS hosts. It is used to park 263 different TDS hosts. Besides parking services, we find that there are also parking managers who offer a centralized approach to manage a portfolio of domains parked with a number of parking services. `Above.com` is one such parking manager who acts as a middle man.

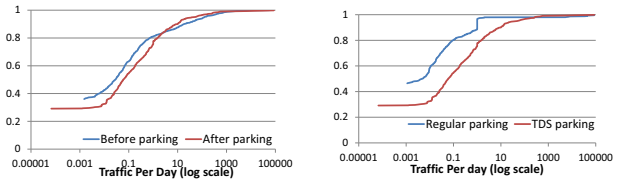
#	Parking Service	# TDSes
1	Bodis Parking	263
2	Dopa Parking	246
3	Oversee Parking	148
4	Above.com	86
5	Name-services.com	61
6	Parkpage.foundationapi.com	53
7	Sedo Parking	33
8	Name Drive Parking	18
9	Parking-page.net	17
10	Internet Traffic Corp	11

Table XIII
TOP 10 PARKING SERVICES

Traffic comparison before and after parking. We first compare the amount of traffic to these TDS domains before and after they are parked. Before parking, they can redirect traffic to exploit servers to monetize traffic while after parking, the new domain owners can monetize the incoming traffic through the advertising models offered by parking services. To compare the traffic volume, we utilize the previously introduced "PassiveDNS" dataset. As each aggregated DNS record contains the start time, end time, and lookup count, we calculate the number of lookups per day. Note that such numbers are lower bounds, as many lookups may be resolved through caching by local resolvers. Figure 13(a) displays the number of lookups per day before and after parking. As we can see from the figure, the amount of traffic does not change significantly after the domain is parked. Compromised door way pages still redirect traffic to the TDS domains, even after the TDSes are taken down and parked.

Traffic comparison between regular parked domains and parked TDSes. Next, we want to examine whether monetizing TDSes through parking is more profitable for domainers than regular domains. To achieve this, we obtain 664 regular parked domains on TrafficZ name servers on Nov15, one of the most reputable parking services, and obtain their lookup rates from the PassiveDNS dataset. Figure 13(b) shows traffic volume of regular vs TDS parked domains. It is surprising that parked TDSes have more than 10 times the the amount of lookups per day than the regular parked domains. This observation shows that even after the TDSes are taken down, they can still bring remarkable revenue to domainers.

Traffic monetization. We find that 61.66% of the parked TDSes use ad-networks and ad exchanges such as *DoubleClick* and *BidSystem*. While 56% go through tracker networks used for targeted advertising, 3.94% of the parked paths monetizing traffic directly through the ZeroClick model.



(a) Before and after TDS parking. After TDSes are taken down and parked, they continue to receive similar amount of traffic from compromised doorways.

(b) Regular vs TDS parking. The median lookup rate of TDSes is over 10 times higher than regular parked domains.

Figure 13. CDF of Traffic hits per day

VII. Discussion

Although some types of dedicated malicious services have long been known [8], topological studies on the hosts playing dedicated roles in malicious Web infrastructures, regardless of the specific types of malicious activities they are involved in, have never been done before. Such studies are important because they can bring us to new types of malicious services and help detect these linchpins of the dark Web even before knowing what they exactly do. Here we discuss what we have learnt from our first step on this direction and what needs to done in the follow-up research.

Based on the large amount of data crawled from the Web during a long period of time, we discover interesting topological features of these malicious hosts: they tend to have very close relations with malicious hosts but rather tenuous connections to even highly popular legitimate services. This finding leads us to the PageRank-based approach, which works effectively in detecting those dedicated services, without relying on their malicious semantics. On the other hand, we are far from fully exploiting the opportunities even our preliminary discoveries present to us: as an example, the paths associated with those dedicated hosts link to a large number of malicious nondedicated hosts, which need to be captured with right techniques. Also, progress can still be made on the detection of such dedicated hosts, particularly those also serving doorway pages.

Our preliminary analysis of those topologically dedicated hosts has already brought us to a type of understudied malicious services—TDSes. With all the findings we make, including their unexpectedly long life span and the way they are monetized even after taken down, more questions have been raised than solved. For example, to what extent do they monetize traffic? How can an ad network trace whether the traffic is redirected from malicious channels? Answers to these questions would be invaluable to the online industry.

While we are studying the attacker's infrastructures, attackers are actively tracing us as well. They now smartly record IP addresses of visitors and only deliver malicious content to each IP once. They can also employ various cloaking techniques. Moving forward, we feel the research community should unite to build a distributed crawling infrastructure and also leverage normal user inputs to better fight against attackers.

Finally, we call for the regulation on the usage and resell of takendown domains. Our study shows that even after TDSes are taken down, they continue to receive a large amount of traffic from compromised doorway pages. Such traffic is currently leveraged by domain owners to gain revenue from the ad networks. These actions should be prohibited as such traffic is not valid human generated ones.

VIII. Related Work

Study on malicious Web infrastructures. Malicious activities are becoming more organized and even turning into a big underground business. Prior research focuses on the malicious infrastructures associated with specific attack channels, such as Spam [2], black-hat Search-Engine Optimization (SEO) [11] and malvertising [16]. These studies analyze the different parties in the underground business and how the malicious campaigns operate. Our study provides a topological view of the malicious Web infrastructures and we study the dedicated malicious hosts and their relationships with other entities.

Detection of malicious entities. To detect malicious entities, prior research either relies on content analysis or redirection chain analysis. However, these approaches are not robust against attackers' ever-changing strategies. Code analysis [5] and URL patterns [11] can be evaded if attackers change signatures. Researchers have also explored the features of redirection chains including length, short sub-sequences, cross-site redirections [14], [16], [17]. These features are proved to be effective against specific attacks but they may not be fundamental to the malicious infrastructures. Instead, we utilize the topological features, such as the relationships between different entities, which are more difficult to evade. The detected hosts are dedicated for a variety of different malicious purposes, more than just the central servers studied in [33].

PageRank algorithm. Our approach adopts the PageRank algorithm [3] to differentiate the malicious and legitimate entities. This algorithm has been used to detect spammers in social networks [9], online comments [32] and web spam pages [4]. Our study shows that the malicious Web infrastructures have similar properties and PageRank is also effective on their topology graphs.

IX. Conclusion

In this paper, we report our study on a set of topologically dedicated hosts discovered from malicious Web infrastructures. Those hosts are found to play central roles in the dark

Web, serving over 70% of the nearly 4 million malicious redirection paths collected in our research and rarely being connected to by any legitimate hosts. Leveraging this unique feature, we develop a topology-based technique that detects these hosts without even knowing exactly what they do. This approach utilizes the PageRank algorithm to capture those with high status on the dark side of the Web but very much unknown on the bright side, and brings to the light thousands of dedicated hosts missed by the state-of-the-art malware scanner. Taking a close look at our findings, we learn that many of those hosts are actually TDSes, which play a key role in traffic exchange in malicious activities. Our further study on such services reveals their unusually long life span (65 days), compared with the exploit services studied before, and the way they are used to monetize traffic, even after their domains have been taken down.

Acknowledgements

We thank anonymous reviewers for their insightful comments. This work is supported in part by NSF CNS-1223477 and CNS-1117106. Alrwais also acknowledges the fund from the College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

References

[1] Alexa. Alexa top 500 global sites. http://www.alexa.com/topsites.

[2] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: characterizing internet scam hosting infrastructure. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 10:1–10:14, Berkeley, CA, USA, 2007. USENIX Association.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[4] P.-A. Chirita, J. Diederich, and W. Nejdl. Mailrank: using ranking for spam detection. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 373–380, New York, NY, USA, 2005. ACM.

[5] C. Curtsinger, B. Livshits, B. G. Zorn, and C. Seifert. Zozzle: Fast and precise in-browser javascript malware detection. In *USENIX Security Symposium*. USENIX Association, 2011.

[6] J. Fitzpatrick. Five best domain name registrars. http://lifehacker.com/5683682/five-best-domain-name-registrars, November 2012.

[7] M. Goncharov. Traffic direction systems as malware distribution tools. http://www.trendmicro.es/media/misc/malware-distribution-tools-research-paper-en.pdf, 2011.

[8] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, N. Provos, M. Z. Rafique, M. A. Rajab, C. Rossow, K. Thomas, V. Paxson, S. Savage, and G. M. Voelker. Manufacturing compromise: the emergence of exploit-as-a-service. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 821–832, New York, NY, USA, 2012. ACM.

[9] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004.

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

[11] J. P. John, F. Yu, Y. Xie, A. Krishnamurthy, and M. Abadi. deseo: combating search-result poisoning. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 20–20, Berkeley, CA, USA, 2011. USENIX Association.

[12] C. Kanich, N. Weavery, D. McCoy, T. Halvorson, C. Kreibichy, K. Levchenko, V. Paxson, G. M. Voelker, and S. Savage. Show me the money: characterizing spam-advertised revenue. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 15–15, Berkeley, CA, USA, 2011. USENIX Association.

[13] Kytoon. Sutra tds. http://kytoon.com/sutra-tds.html.

[14] S. Lee and J. Kim. WarningBird: Detecting suspicious URLs in Twitter stream. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, Feb. 2012.

[15] K. Levchenko, N. Chachra, B. Enright, M. Felegyhazi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, A. Pitsillidis, N. Weaver, V. Paxson, G. M. Voelker, and S. Savage. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *Proceedings of 32nd annual Symposium on Security and Privacy*. IEEE, May 2011.

[16] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 674–686, New York, NY, USA, 2012. ACM.

[17] L. Lu, R. Perdisci, and W. Lee. Surf: detecting and measuring search poisoning. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 467–476, New York, NY, USA, 2011. ACM.

[18] Malwaredomains. url_shorteners. http://mirror1.malwaredomains.com/files/url_shorteners.txt.

[19] U. Parasites. Runforestrun now encrypts legitimate js files. http://blog.unmaskparasites.com/2012/07/26/runforestrun-now-encrypts-legitimate-js-files/, 2012.

[20] R. Petnel. Adblock plus. http://adblockplus.org/en/.

[21] R. Petnel. Easylist. https://easylist-downloads.adblockplus.org/easylist.txt.

[22] R. Petnel. Easyprivacy. https://easylist-downloads.adblockplus.org/easyprivacy.txt.

[23] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All your iframes point to us. In *Proceedings of the 17th conference on Security symposium*, pages 1–15, Berkeley, CA, USA, 2008. USENIX Association.

[24] Sedo. Domain parking terms and conditions. https://sedo.com/us/about-us/policies/domain-parking-terms-and-conditions-sedocom/?tracked=1&partnerid=38758&language=us.

[25] I. SIE. Isc dnsdb api. https://dnsdb.isc.org/doc/isc-dnsdb-api.html.

[26] I. SIE. Security information exchange (sie) portal. https://sie.isc.org/.

[27] J. W. Stokes, R. Andersen, C. Seifert, and K. Chellapilla. Webcop: locating neighborhoods of malware on the web. In *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*, LEET'10, pages 5–5, Berkeley, CA, USA, 2010. USENIX Association.

[28] 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.

[29] Twitter. Using the twitter search api. https://dev.twitter.com/docs/using-search, 2012.

[30] Y. Wang, D. Beck, X. Jiang, and R. Roussev. Automated web patrol with strider honeymonkeys: Finding web sites that exploit browser vulnerabilities. In *In Proceeding of the Network and Distributed System Security Symposium (NDSS'06)*, 2006.

[31] Whois.net. Whois lookup - domain names search, registration, & availability. http://www.whois.net/, 2011.

[32] J. Zhang and G. Gu. Neighborwatcher: A content-agnostic comment spam inference system. In *In Proceeding of the Network and Distributed System Security Symposium (NDSS'13)*, 2013.

[33] J. Zhang, C. Seifert, J. W. Stokes, and W. Lee. Arrow: Generating signatures to detect drive-by downloads. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 187–196, New York, NY, USA, 2011. ACM.