Hunting for Invisibility: Characterizing and Detecting Malicious Web Infrastructures through Server Visibility Analysis

Jialong Zhang*, Xin Hu†, Jiyong Jang†, Ting Wang‡, Guofei Gu*, and Marc Stoecklin†

*Texas A&M University, {jialong,guofei}@cse.tamu.edu
†IBM Research, {xinhu, jjang, mpstoeck}@us.ibm.com
‡Lehigh University, ting@cse.lehigh.edu

***Abstract*—Nowadays, cyber criminals often build web infrastructures rather than a single server to conduct their malicious activities. In order to continue their malevolent activities without being detected, cyber criminals make efforts to conceal the core servers (e.g., C&C servers, exploit servers, and drop-zone servers) in the malicious web infrastructure. Such deliberate *invisibility* of those concealed malicious servers, however, makes them particularly distinguishable from benign web servers that are usually promoted to be public.**

In this paper, we conduct the first large-scale measurement study to investigate the *visibility* of both malicious and benign servers. From our intensive analysis of over 100,000 benign servers, 45,000 malicious servers and 40,000 redirections, we identify a set of distinct features of malicious web infrastructures from their locations, structures, roles, and relationships perspectives, and propose a lightweight yet effective detection system called VISHUNTER. VISHUNTER identifies malicious redirections from visible servers to invisible servers at the entryway of malicious web infrastructures. We evaluate VISHUNTER on both online public data and large-scale enterprise network traffic, and demonstrate that VISHUNTER can achieve an average 96.2% detection rate with only 0.9% false positive rate on the real enterprise network traffic.

I. INTRODUCTION

Today's cyber crimes are no longer monotonous. Attackers set up a variety of malicious servers, e.g., exploit servers, C&C servers, and phishing servers, to effectively perpetrate their criminal activities. Based on a recent threat report from Websense [7], malicious websites have increased by nearly 600% worldwide since 2012. Different malicious servers, such as redirectors, exploit kits, C&C, and payment servers, often join forces to leverage their diverse functionalities, and to create more efficient and anonymous web infrastructures for malware distribution, control, and monetization.

Existing systems to identify malicious web infrastructures fall into three main categories. The first category focuses on analyzing web contents to determine their maliciousness. For example, web-based infections can be detected by analyzing changes between the base version and a modified version of web contents [11] or JavaScript libraries [19]. Others utilize instrumented browsers [26] or JavaScript engines [12] to automatically visit suspicious websites, and examine the runtime system or browser behaviors for the signs of drive-by download attacks. The second category investigates evasion techniques that attackers often use to hide their malicious activities. These work identifies characteristics unique to the evasion behaviors to detect malicious servers, e.g., web search cloaking [29, 30], fast fluxing [14, 16], and domain generation [9, 25]. The last category looks into the topology of malicious infrastructures. For example, redirection chains [23, 28] and server ranks (e.g., according to PageRank-based approach) [20] are studied to identify malicious websites.

While existing solutions demonstrate their effectiveness in detecting malicious servers or server infrastructures, we note that they still have their respective limitations. For example, content analysis often introduces non-trivial overhead, rendering it impractical for large-scale network analysis. Instrumentation may also be blocked due to fingerprinting techniques [13]. While topology-based approaches can be content-agnostic, they often either demand a large and diverse web user base in order to collect sufficient redirection data and construct redirection graphs [28], or require malicious hosts as seeds to bootstrap the system [20]. Thus, understanding the intrinsic properties of malicious infrastructures and interactions among the constituent compromised/malicious servers is often critical in building an effective detection system.

In this paper, we study malicious web infrastructures from a novel perspective, i.e., *visibility*. In a nutshell, we define the visibility[1] of a server as whether the server is visible to benign users, for instance, through search engines. To obtain a comprehensive understanding of whether invisible malicious web infrastructures are indeed popular, we investigate 100,000 benign servers and nearly 45,000 malicious servers collected from both *public blacklists* and *real-world enterprise networks*. Our key findings include: 1) Unlike legitimate servers, a large number of malicious servers tend to be invisible, especially for some categories such as C&C servers and exploit servers. C&C servers used to communicate with bot-infected machines are almost always invisible to benign users. This is because bot masters naturally try to minimize the exposure of their C&C servers to keep their malicious activities under the

[1] An in-depth discussion on visibility will be in Section II.

radar. 2) Server visibility alone is not sufficient to precisely pinpoint malicious web infrastructures because a small number of benign servers are also invisible. Therefore, we further examine 41,190 redirections collected from a large enterprise network and observe that the "entrance" into malicious web infrastructures, i.e., redirection from visible servers to invisible servers, is significantly different from that of benign web infrastructures.

Motivated by these findings, we design a lightweight yet effective system, VISHUNTER, to detect the malicious web infrastructure by detecting the "entrance" to it. VISHUNTER extracts 12 features including several novel visibility-related features, and uses a trained classifier to identify the "entrance" to the malicious web infrastructures. Our evaluation with one month traffic from a large enterprise network shows that VISHUNTER achieves an average true positive rate of 96.2% at a false positive rate of 0.9% for the malicious entrance detection.

Contribution. The main contributions of the paper are summarized as follows:

New findings. To the best of our knowledge, we are the first to conduct a large-scale and comprehensive study on the visibility of malicious web infrastructures, offering in-depth insights into the visibility trends of malicious servers as well as the significant differences between the entrance to benign and malicious invisible infrastructures. We characterize these differences using graph-, location-, role-, and relation-based features, and demonstrate that they can be leveraged to accurately detect entrances to malicious web infrastructures.

New techniques. We develop lightweight yet effective techniques to detect malicious web infrastructures by exploring the nature properties of cyber criminals, which cannot be easily circumvented. In addition, VISHUNTER dramatically reduces the amount of network traffic required for analysis because it focuses on the visibility transition between visible and invisible servers, instead of the entire redirection chains. Furthermore, compared to existing solutions that require a large and diverse user base [28], VISHUNTER is more lightweight and able to detect entrances to malicious web infrastructures even when there are only a few clients accessing them.

II. SERVER VISIBILITY

The concept of visibility used in this work hinges on how a normal user locates a server. For popular or frequently visiting websites, a user may directly type the domain names into the address bar or follow bookmarks to access the web servers. We consider these servers visible to normal users. For other (less popular) websites, a user may locate them using search engines, and access the websites through search results. We also consider these web servers visible if the server itself (not only the domain name) is indexed by search engines. Intuitively, benign servers are more likely to be *visible* to normal users because their owners are often motivated to promote their websites in search engines to increase their user base. On the contrary, malicious servers, especially the ones in the core malicious infrastructure (e.g., exploit servers), are less likely to be indexed by search engines because attackers try to minimize their exposure and avoid being detected. Therefore, we determine server visibility as follows (cf. Fig. 1).

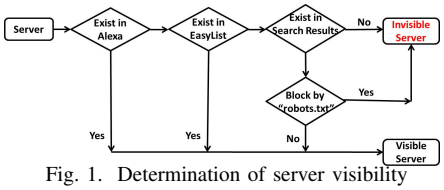

Fig. 1. Determination of server visibility

1) For a domain name, we extract its second-level domain (SLD) based on [6]. Since a SLD is commonly associated with the organization that registers the domain, we assume that a domain has the same visibility as its SLD. In the remaining of the paper, the term domain denotes its SLD unless stated otherwise.
2) If a domain is a well-known (e.g., `google.com`), we consider it visible. We utilize two public whitelists to determine well-known domains: top 1 million domains from Alexa [1], and domains from EasyList [3]. Alexa provides a global popularity ranking of domains based on their collected web traffic. EasyList provides a list of well-known Ad-network domains and trackers. Notes that domains in whitelists only reflect the popularity of them, it does not means that they are benign domains.
3) For other domains, we query them in search engines [2] and obtain top 100 [3] search results. If the queried domain appears in the search results, meaning that it was crawled and indexed by the search engines, we further examine the indexed content of the domain. If the site owner blocks the access to the content (e.g., Google displays a message under the domain name "A description for this result is not available because of this site's robots.txt"), we consider the domain invisible; otherwise, it is considered visible. Note that since multiple search results may be returned for a domain, for the domain to be classified as invisible, none of them should have available content. This is to avoid misclassifying legitimate cases where a site owner may use robots.txt to prevent search engines from crawling their sensitive webpages (e.g., admin interface).
4) If the queried domain does not appear in the top 100 search results, we consider it invisible.
5) All IP addresses are considered as invisible.

Noting that the definition of server visibility depends on search engine results that may change dynamically along with time, we evaluate the server visibility over time in section V-B.

III. MEASUREMENT STUDY OF SERVER VISIBILITY

Our hypothesis is that legitimate servers are more likely to be visible because their owners have the incentives to promote their products or services. However, certain categories of malicious servers (e.g., exploit servers), tend to remain

[2]We use Google search engine in current implementation. However, other search engines could be used to reduce possible bias of search engine results.

[3]100 is the maximum number of search results per page. Since we directly search the domain, the pages on that domain are usually returned in the first.

invisible for several reasons. First, from the cyber criminals' perspective, they may only want to allure their targeted victims to reach the core malicious servers in order to minimize the exposure to security analysts because previous work has shown that it is easy to pinpoint malicious servers using search engines [15]. Second, from the search engines' perspective, it may not be straightforward to crawl and index malicious servers. Some malicious servers are intentionally isolated from the World Wide Web without any hyper-links pointing to them. Other malicious servers may be ephemeral and dynamically changing such as domains generated by domain generation algorithms (DGAs). Third, search engines employ their own algorithms (e.g., PageRank) to index and rank servers. In general, they prefer to return to their users websites with high reputation. Hence, malicious websites may not be indexed by search engines or not be returned to the users.

To validate our hypothesis, we conduct a comprehensive measurement study on the visibility of both benign and malicious servers using real-world datasets.

A. Datasets

Public Blacklists. This dataset consists of domains from two blacklists: Malware Domain List [4] and DNS-BH Malware Domain Blocklist [2], both of which have been widely used as ground truth in existing work [10]. We collected 43,768 malicious SLDs from 2009 to 2014 from [4] and 10,967 SLDs from 2011 to 2014 from [2], and performed visibility check on Oct 2014. Results show that around 80% of them were invisible. One caveat here is that malicious domains may have already expired leading to search engines may not index them anymore. To avoid such bias, we selected only the domains that were blacklisted in 2013 and 2014, and verified their existence by checking if the domains could be successfully resolved to IP addresses. As a result, this dataset contains 875 malicious SLDs from Malware Domain List (`M_PB1`) and 5,739 malicious SLDs from DNS-BH (`M_PB2`).

Enterprise Traffic. To avoid possible bias caused by public blacklists, we also captured real network traffic from a large enterprise network from June 16 to June 20, 2014, from which we extracted 462,226 unique servers. Among them, 1,782 servers (`M_Enter`) were detected as malicious by an internal intrusion detection system (IDS). For those that were not detected by the IDS, we randomly chose 100,000 benign servers that did not share any clients with malicious servers in `M_Enter` and labeled them as `B_Enter`. We performed the viability check on these servers at the same time we captured them. We also collected one month traffic of an institute in June 2013. Due to privacy and storage constraints, we stored only the metadata and the header information of all the web requests and responses, which included request URL paths, HTTP response codes, host names, user agents, referers, cookies, and so on. Notice that without the content of the web pages, we were not able to directly determine certain types of redirections, e.g., JavaScript or iFrame based redirections. Therefore, we focused on HTTP header redirections in this dataset. In total, we extracted 41,190 redirections(`R_Enter`) and performed the viability check on this data on June 2014.

Table I summarize the data collection results.

TABLE I
DATA COLLECTION

	M_PB1	M_PB2	M_Enter	B_Enter	R_Enter
# of servers	875	5,739	1,782	100,000	165,957
# of redirections	N/A	N/A	N/A	N/A	41,190

B. Server Visibility Study

1) Visibility of Malicious Servers: We first determined the visibility of malicious servers in `M_PB1`, `M_PB2`, and `M_Enter` using the process outlined in Section II. To better understand how visibility correlates with specific malicious types, we measured the visibility distribution of malicious servers within each attack category. Based on the description of malicious functionalities in `M_PB1` and `M_PB2`, we classified them into 63 and 109 categories respectively.

We then calculated *invisibility ratio*, defined as the number of invisible servers over the total number of servers in a category. The CDF of the invisibility ratio distribution across all the categories is illustrated Figure 2. Only around 10% of categories from `M_PB1` had the ratios lower than 30%, meaning that the servers in those categories were likely to be visible. For `M_PB2`, around 35% of categories had low ratios.

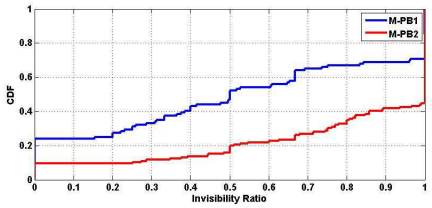

Fig. 2. Invisibility ratio distribution across all categories

Table II lists the top 5 largest *visible* malicious categories in terms of their sizes. Most visible servers in `M_PB1` were compromised servers and had labels, such as "leads to", "iFrame", and "JavaScript". `M_PB2`, on the other hand, had different distribution of malicious servers with more social-engineering type of attacks like phishing and rogue software. These servers, by their nature, were designed to be easily accessible to unsuspecting users, and therefore were more likely to be visible. Overall, less than half of servers, 356 (44.49%) in `M_PB1` and 363 (31.33%) in `M_PB2` belonged to these categories.

TABLE II
TOP 5 LARGEST VISIBLE MALICIOUS CATEGORIES

M_PB1			M_PB2		
Category	# of servers	Invisibility ratio	Category	# of servers	Invisibility ratio
leads	246	25.20%	unsafe	159	22.01%
spyware	21	28.57%	highrisk	65	23.07%
iFrame	17	29.41%	fake	27	25.93%
JavaScript	14	21.43%	phishing	13	15.38%
compromised	5	0%	rogue	53	0.25%

TABLE III
TOP 5 LARGEST INVISIBLE MALICIOUS CATEGORIES

M_PB1			M_PB2		
Category	# of servers	Invisibility ratio	Category	# of servers	Invisibility ratio
blackhole	31	90.32%	malware	652	65.18%
-	21	52.386%	malspam	215	64.65%
drive-by	5	66.66%	zeus	81	60.49%
java	5	100%	fake_flash	34	73.53%
fake	4	100%	putter_panda	27	96.29%

As a comparison, Table III lists the top 5 largest *invisible* malicious categories with their sizes. We can see that malware

and exploit servers, such as blackhole, Zeus and drive-by-download, were among the most common invisible malicious server types. For example, more than 90% of blackhole servers and more than 96% of "putter panda" servers, which were C&C servers used in a cyber espionage campaign, were hidden from search engine crawlers.

Noting that the invisibility ratio for largest invisible malicious category is not 100%. This is because that some of them are actually compromised servers rather than malicious servers. For example, `northerningredients.com` was visible according to our visibility definition, and it was labeled as malicious by both `Malciousdomain.com` and VirusTotal. Further investigation on its Whois information and its web contents revealed that the food company's domain was registered in 2006 with an expiration date in 2019. Such a long history made us believe that it was a compromised legitimate domain instead of a malicious server set up by cyber criminals. In addition, other visible malicious servers usually shared certain patterns in their contents or URLs, which could be leveraged to efficiently detect a group of similar servers using existing work such as EvilSeed [15] and PoisonAmplifier [33].

We further checked the visibility of `M_Enter` from the enterprise network. 67.51% of them were invisible, and 13.99% of them belonged to the top 1 million Alexa web list, indicating that they were likely compromised or abused.

2) Visibility of Benign Servers: Next, we checked the visibility of benign servers in `B_Enter`. As expected, only very small portion of them, i.e., 6,626 (6.63%) were invisible. To better understand the underlying causes for their invisibility, we manually analyzed a set of randomly selected 100 servers. We found that most of the benign invisible servers were: 1) new servers which had not been indexed; 2) service providers which actively blocked crawlers or chose not to be indexed by search engines. As a result, we can leverage these characteristics to distinguish them from their malicious invisible servers, which we will elaborate in Section IV-B.

Lessons learned: As demonstrated above, there exist significant and consistent differences between certain malicious servers and legitimate servers in terms of their visibility status. These findings suggest that visibility could be an effective feature for malicious server identification. However, we also note that visibility alone is not sufficient, as many legitimate servers, although few percentages, may not be directly accessible to users through search engines for various reasons. Therefore, we explore several new redirection-based features to augment visibility and minimize false positives.

C. Visibility Study on Redirections

Based on the visibility of each server, we clustered the redirections in `R_Enter` into four categories: visible to invisible (1,063 (2.58%)), visible to visible (36,727 (89.17%)), invisible to invisible (1,559 (3.78%)) and invisible to visible (1,841 (4.47%)). Unsurprisingly, the majority of redirections were among visible servers, which were mostly benign redirections with a few from one compromised server to another.

Many existing systems detect malicious redirections using the characteristics of the full redirection chains [22, 28], such as chain lengths, geolocations of landing and final servers, and so on. Unfortunately, such features can be easily manipulated when check them on the whole redirection chain. For example, attackers can change the length of the redirection chain by appending more/fewer compromised or malicious servers. Alternatively, attackers can also add arbitrary inner domain redirections since they can partially control compromised servers and fully control their own malicious servers.

In contrast, only the transition from visible to invisible servers is more resilient to manipulation whenever attackers want to lure unsuspecting users to their malicious infrastructures. Therefore, we only focus on such kind of redirection and inspect it from several different perspectives.

To collect ground truth, we used VirusTotal to label 1,063 redirections from visible servers to invisible servers. Specifically, if an invisible server was labeled as malicious by at least two anti-virus vendors, we considered the redirection as malicious. In this way, we finally collected 27 malicious redirections. To collect benign redirection cases, we checked the Whois history of invisible servers for the remaining redirections. We removed redirections whose invisible server had a lifetime (the expiration date minus the creation date) less than or equal to one year, which have a high chance to be malicious but not yet been labeled by VirusTotal. As a result, we collected 683 benign redirections.

Below we itemize our observations and lessons learned from these redirections

1) Location Attributes: Typically attackers can not control the place where the compromised benign servers are located so that visible compromised servers and invisible malicious servers would be located at different places. In VISHUNTER, we characterize the location difference using IP addresses, Whois information, and autonomous systems numbers (ASNs). Specifically, we consider a redirection to be made between different locations if its visible and invisible servers do not locate under the same IP subnet (/24), do not share the same Whois information, and do not have the same ASNs. Otherwise, the two servers are likely to be co-located. In our ground truth dataset, all but one malicious redirections had different locations. On the other hand, 188 (27.53%) of benign redirections had co-located visible and invisible servers. In fact, those visible servers usually hosted the home page of the company's website while those invisible servers provided internal or non-public services. These invisible servers may actively block search engine crawlers and/or there is no way for the crawlers to find them.

2) Structure Attributes: To capitalize on their resources, attackers often compromise a large number of servers and point them to a small set of their core malicious servers. Thus, we assume there should be an authority structure in the malicious redirections: multiple visible servers redirect to a few authority invisible servers. To characterize this hub/authority structure, we propose a metric "in-out ratio", defined as the degree of the invisible server over the degree of the visible server. A

ratio lower than 1.0 means that multiple visible servers redirect to only a few invisible servers, making the invisible servers authorities. On the other hand, a visible server that redirects to many invisible servers will have a high in-out ratio and become a hub in the redirection graph.

Looking at our ground truth dataset, 33.33% (9) of malicious redirections had invisible authorities, and 7.41% (2)had visible hubs. In comparison, for the benign redirections, only 3.22% (22) of them had invisible authorities while 80.23% (548) of them had visible hubs. For all the hubs, we further checked the number of IP addresses they redirected to. The intuition here is that malicious hubs may redirect to multiple domains hosted on the same IP address in order to minimize cost and maximize server utilization. On the other hand, benign hubs may redirect to multiple servers which belong to different organizations and thus have a large number of IP addresses. This intuition was also confirmed by our data: a malicious hub indeed redirected to two domains that shared the same IP address, whereas those benign hubs all redirected to multiple IP addresses.

3) Role Attributes: Some benign redirections are caused by advertisement networks. One notable characteristic of an advertiser is that it can redirect to a large number of both visible and invisible servers. In this work, we use the metric Num_{vis}, the number of redirections to visible servers, to define advertisers, and find 467 advertisers in the benign redirections with $Num_{vis} > 3$ (68.37%). We also define the reputation of an advertiser Rep as Num_{vis} / Num_{invis}, where Num_{invis} is the number of redirections to invisible servers. Essentially, an advertiser is considered to be more suspicious if it redirects to more invisible servers than visible servers. In fact, such a pattern has been used by certain cloaking servers, which redirect target users to either exploit servers or legitimate websites depending on users' operating systems and browser versions. 174 (37.26%) of advertisers redirected to more visible servers than invisible servers for benign redirections; therefore, they were more likely to be benign . In this dataset, there was no advertiser in malicious redirections.

4) Relation Attributes: Benign redirections often serve a purpose, for example, moving websites to another server, load balancing, delivering contents from local data centers, etc. We characterize such relationship from the following perspectives.

CDN: CDNs account for a large portion of the benign redirections. In addition to whitelisting well-known CDNs such as Akamai, CloudFront, etc., we apply a heuristic to attribute a redirection to a CDN if two servers involved have the same URL path and the path length[4] is longer than 2. 19.18% (131) of benign redirection in our dataset fell into this category whereas none of malicious redirections had a CDN relationship.

Other general partner relation: To identify other general partner relationships, we employ two heuristics leveraging both historical information and search engines results. First, if a specific redirection happens regularly over a long period of time T[5], we consider the two servers have a stable partner relationship. Second, we leverage search engines to reveal potential partnership between two servers. More specifically, we query two servers involved in the redirection in the search engines at the same time, for example, search "visible.com and invisible.com" in Google. If both of them appear in the same search results, we believe that there likely exist relationship between these two servers. In our dataset, most of such relationships were due to sharing contents and were also reported by websites like `siteslike.com` and `websitesalike.com`. To further eliminate potential false positives that may be caused by security websites analyzing a particular malicious server instance, we ignore such partner relationship if the search results contains any security related keywords such as "security", "virus", "malicious", and "malware", etc. As a result, only 5 (18.52%) of malicious redirections had partner relationships while 513 (75.11%) of benign redirections had such relationships.

Lessons learned: Detecting malicious redirections is a complicated task. Simply relying on features associated with compromised servers is immediately subject to evasion, given the attacker's freedom to use public services or multiple compromised servers. At the same time, detecting malicious terminal servers is hindered by their diversities and various cloaking techniques. However, observations from our measurement study convey a positive message: malicious redirections from visible servers to invisible servers exhibit distinguishable behaviors from their benign counterpart, which are more intrinsic to malicious infrastructures and difficult to evade.

IV. System Design

A. System Overview

Based on the study in Section III, we observe that there exist certain categories of malicious servers that are always invisible, and there exist notable differences between entrances to malicious invisible infrastructures and benign invisible infrastructures. Therefore, VisHunter focuses on the redirections from visible servers into invisible servers, and leverages discriminative features to detect the entrances into malicious web infrastructures, especially for the entrances to the exploit infrastructure, where most traffic comes from compromised servers to exploit servers.

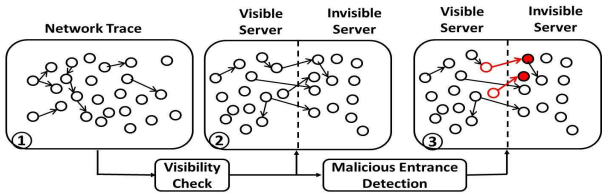

Fig. 3. System overview.

An overview of VisHunter is shown in Figure 3. VisHunter takes HTTP traffic as an input and extracts all observed servers (nodes in ①) as well as the redirections among them (edges between servers in ①). Then, VisHunter

[4]We measure the path length based on the number of slashes ("/") in URLs.

[5]In current implementation, we set T=1 month.

checks the visibility of each server and divides them into two categories: visible servers and invisible servers (②). Next, only the redirections from visible servers to invisible servers are submitted to the malicious entrance detection component, where a trained classifier is used to identify malicious entrances (red edges in ③).

B. Detecting Malicious Entrances

We define an "entrance" as a redirection from SLD of a visible server to SLD of an invisible server. An entrance is considered malicious if the destination invisible server is malicious, i.e., belonging to attackers' invisible malicious web infrastructures. Based on our intensive study in Section III-C, we propose 12 features that characterize the differences between benign entrances and malicious entrances, which are difficult for attackers to evade without a significant amount of cost. As summarized in Table IV, the features leveraged by VISHUNTER are divided into four groups.

TABLE IV
FEATURE SELECTION

Aspects	Features	Novelty
Location	IP location	[22]
	Whois location	New
	AS location	[22]
Graph	In-degree of invisible server	New
	Out-degree of visible server	New
	In-Out-ratio	[27]
	IP diversity of invisible server	[24]
Role	Advertiser	New
	Reputation of advertiser	New
Relation	CDN	New
	Partner based on history	New
	Partner based on search results	New

Location-based Features: This group aims to capture the location differences of the entrances. Since this group of features are derived from the physical locations of the entrances, it is difficult for attackers to forge.

Location-based features proposed in existing work are not as resilient as our proposed features. For example, the location differences between a landing server and a terminal server [22] can be easily changed by adding another redirection to the original server after an infection. An attacker can simply add an iFrame in the malicious server such that it sends users back to the landing server. In this way, it can evade location features in [22]. On the other hand, as long as attackers rely on compromised servers to redirect users to their malicious web infrastructures, VISHUNTER will detect the location difference.

Graph-based Features: This group aims to characterize the structure of entrances based on graph properties. Attackers would be required to change their fundamental operation structures for evasion. For example, to evade the invisible authority feature, malicious invisible servers would be allowed to use only a few compromised servers to redirect to them, which effectively limits the effectiveness of attackers' operations.

Role-based Features: This group aims to distinguish between benign and malicious entrances to advertisement infrastructures. To evade this group of features, attackers are required to either abuse public known advertisers, for example, `googleadservices.com` to redirect traffic, which is not trivial since those services usually have strict scrutiny processes, or to directly redirect users using compromised servers, which can be detected through other features (e.g., location-based features).

Relation-based Features: The group aims to characterize general entrances of benign infrastructures. CDN-based features are hard to evade since malicious servers usually have different paths with compromised servers. At first glance, partner based on search results feature seems easier to evade, e.g., attackers may circumvent search engine partner relationships by posting compromised servers and malicious servers together. However, such behavior could lead existing work (e.g., EvilSeed [15] and PoisonAmplifier [33]) to find more malicious servers easily.

As a result, 12 features are extracted for each entrance, and a classifier (J48 decision tree) is trained with known malicious and benign entrances to detect the entrances to malicious web infrastructures. We acknowledge that attackers may artificially manipulate some of our proposed features to evade VISHUNTER. However, it is not trivial for attackers to evade the detection based on the combined use of all features without putting a significant amount of investment.

V. EVALUATION

A. Data Trace and Ground Truth

Enterprise Traffic: We collected 6 months (from July, 2013 to Oct, 2013 and from Nov, 2014 to Dec, 2014) traffic from the same institute described in section III-A.

Online Public Malware Traffic: We also crawled malware traces from `malware-traffic-analysis.net` [5] which provided more than 402 pcap traces of malware collected from June, 2013 to Jan, 2015. Those traces showed detailed analysis on how malware was delivered, typically through compromised servers and drive-by-download exploit kits. Specifically, 213 cases used various redirection methods to deliver malware, including JavaScript redirection, iFrame redirection, and HTTP header redirection. Among them, 159 redirections were from visible servers to invisible servers. Others were either because the incomplete traces made it impossible to obtain the corresponding compromised servers or attackers leveraged public servers (e.g., dynamic DNS server `redirectme.net`) as their exploit servers.

B. Time impacts on visibility

As the server visibility depends on search engine results that may change dynamically, we measured the stability of visibility over time. Specifically, we recorded the visibility of each server in `M_Enter` and `B_Enter` when we collected the dataset (June 2014) and then re-checked their visibility status every two months. The results are summarized in Table V. We can see that server visibility was relatively stable. Only about 2% of the servers, regardless of their maliciousness, changed their visibility status after 6 months, though malicious servers seemed slightly more volatile. Further investigation showed that visible servers changed into invisible primarily because their domain names expired and hence were removed from

search engine indexes. On the other hand, for those invisible servers that changed into visible, it was mainly because those servers were newly registered and recently indexed by search engines or they completed website construction and unblocked the crawlers in "robots.txt".

TABLE V
STABILITY OF VISIBILITY

time	visible → invisible		invisible → visible	
	M_Enter	B_Enter	M_Enter	B_Enter
Jun,2014	-	-	-	-
Aug,2014	20 (1.12%)	1,043 (1.04%)	22 (1.23%)	885 (0.86%)
Oct,2014	34(1.91%)	1,782(1.78%)	30(1.68%)	1,306(1.31%)
Dec,2014	47(2.64%)	2,615(2.61%)	36(2.02%)	1,583(1.58%)

C. Search Engines Comparison

To evaluate the possible bias of the visibility results for different search engines, we randomly select 100 servers from B_Enter and M_Enter respectively, and test their visibility on different search engines. We use visibility from Google search results as the baseline. "+" means other search engines find new visible servers, and "-" means other search engines mislabel some visible servers to invisible servers. Table VI shows the results. We can see that overall different search engines return similar results. Bing and Yahoo have very similar results since now Yahoo search engine is based on Bing. Google only mislabels few visible servers as invisible servers. The missed servers in M_Enter are probably because that search engines refrain from showing known malicious sites. In addition, the combination of different search engines can provide a better view of visibility, which could be leveraged in our future work.

TABLE VI
COMPARISON AMONG DIFFERENT SEARCH ENGINES

	Visible servers in B_Enter	Visible servers in M_Enter
Bing	+4,-4	+3,-14,
Yahoo	+4,-4	+4,-13,

D. Malicious Entrance Detection Results

To evaluate the performance of VISHUNTER, we first performed a 10-fold cross validation of VISHUNTER's classifier with S_{train}, the ground truth dataset we used for the measurement study on redirections in Section III-C. The J48 classifier achieved an average true positive rate of 96.2% at a 0.9% false positive rate.

We investigated the misclassified cases. The only missed malicious redirection (false negative) was because a visible server redirected to two invisible servers with different IP addresses, which was labeled as a benign advertisement behavior. For six false positive cases, they all redirected from a visible server to an invisible server. Two of the redirections, even though their domains were not detected by VirusTotal, included IP addresses labeled as malicious. For the redirection `deal4u.in` → `rggg.net`, the invisible server is now visible and for sale. The redirection `eoaclk.com` → `paragonhondaoffers.com` presented an interesting case. `paragonhondaoffers.com` was the website of a car company who blocked search engines' crawlers. This was not usual because car dealers would want to promote their offers. We further checked its Whois information and found that the domain was registered by a third party company that provided marketing services. The remaining two false positive redirections were essentially between the partner websites that provided similar products. This could be addressed by calculating topic similarity of two websites.

We further evaluated VISHUNTER with 6 months enterprise traffic and the public malware traces. Table VII presents the number of malicious entrances VISHUNTER detected. To get the ground truth of those detected entrances, we checked the invisible servers in the entrances against VirusTotal. If at least two anti-virus softwares detected an invisible server as malicious, we believed that the corresponding entrance was malicious, and marked it as "confirmed". If only one anti-virus software detected it as malicious, we marked the corresponding entrance as "suspicious". If a domain was expired, we marked it as "expired". For all the remaining servers, we manually verified them. If a server was reported by other resources (security blogs, analysis reports, and etc) as malicious, we marked the corresponding entrance as "manual"; otherwise, it was considered to be a false positive.

TABLE VII
MALICIOUS ENTRANCE DETECTION RESULTS

VISHUNTER	2013				2014		13-14
	Jul	Aug	Sep	Oct	Nov	Dec	Malware
	59	26	36	41	34	69	135
Confirmed	34	11	15	21	12	27	135
Suspicious	9	1	6	6	8	17	0
Manual	0	1	2	1	7	3	0
Expired	9	8	4	2	0	0	0
False Positive	7	5	9	11	7	22	0
False Negative	N/A	N/A	N/A	N/A	N/A	N/A	24

Note that the false positives here were not always benign entrances. Some of them did have suspicious behaviors. For example, the redirection `???ssdns.com/wp-content/favicon1.png` → `???cloudproxy.com/2devnulltracker`, was suspicious as it was from an image file `favico1.png` to a proxy server. However, since we were unable to procure an evidence to confirm its maliciousness, we conservatively labeled it as a false positive. Some other false positives were due to CDNs. For example, for the entrance `nv.ahcdn.com/axx/598132.flv` → `88.208.57.3/bxx/598132.flv`, the two servers shared similar path patterns and the same filename. However, since our CDN-related features required that two servers shared the same URI, VISHUNTER detected the entrance as a false positive. For the remaining false positives, we found that they were the entrances to benign partner web servers. One complementary feature to eliminate these false positives is to check topic similarity of the two servers.

We also note that VISHUNTER was capable of detecting new malicious invisible servers that were missed by VirusTotal, even though one would expect that for data from 2013, which were more than 2 year old, VirusTotal would have already captured the most, if not all, malicious servers. In fact, for the recent data from 2014, we have successfully submitted several new malicious cases to VirusTotal. Therefore, we believe that VISHUNTER, as a behavior-based approach, is complementary to the widely used blacklisting and signature-based methods (e.g., IDS), and has a potential to detect

targeted/stealthy attacks that elude public blacklists.

For the online malware traffic traces, VISHUNTER detected 84.9% of all malicious entrances. The missed cases were mainly the visible servers redirecting to multiple invisible servers with completely different IP addresses. Further investigation showed that this was because we aggregated all the redirections over one and half year together. For a shorter period of time, e.g. 1 month, the compromised servers or the public proxy servers abused by attackers only redirected to a limited number of invisible malicious servers.

For those confirmed malicious servers, we further extracted the earliest timestamp when they were detected by VirusTotal and compared it against our detection time. Figure 4 shows the CDF of the detection time difference distribution. We can see that when VISHUNTER detected those malicious servers, around 50% of them were still not detected by VirusTotal.

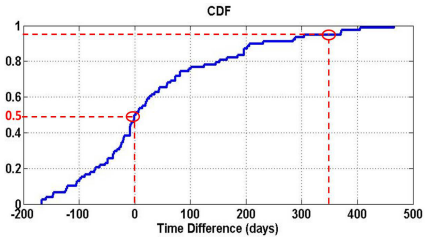

Fig. 4. Detection time difference distribution

E. Comparison with Existing Work

Unlike VISHUNTER, existing work on malicious redirection detection either target on specific attack channels or require a large and diverse user base, which limits their practicality. In this section, we quantitatively compare VISHUNTER with existing work SURF [22], and defer the qualitative discussion of the differences between VISHUNTER and other existing systems such as SpiderWeb [28] in Section VII.

SURF makes use of 9 features to detect malicious servers involved in a search poisoning attack. Three of them belong to poisoning resistance, which are only applicable for a search poisoning attack. Besides these attack-specific features, there remain six features. Since all the redirections in VISHUNTER from visible servers to invisible servers are already cross-site redirections, we ignore "total redirections hops" and "cross-site redirections hops". In fact, as discussed, these two features could be easily manipulated by attackers who have a control over the compromised and/or malicious servers. Since the information about "page to load/render errors" was not available in our dataset, we implemented a classifier based on the remaining three features for comparison.

Unfortunately, SURF missed all the malicious redirections because most benign and malicious cases shared similar features. 72.47% of benign redirections had different locations, and 39.97% benign redirections also redirected from domain names to IP addresses. Moreover, none of malicious redirections used cloaking techniques. We acknowledge that our comparison might not be comprehensive enough to draw a solid conclusion partially due to the fact that we were not able to completely reproduce SURF's classifier, and the main goal of SURF was to detect a search poisoning attack rather than general malicious redirections. Nevertheless, it is worth to note that redirection features alone are subject to circumvention by attackers, and leveraging visibility as a complementary feature allows VISHUNTER to achieve better detection accuracy and to be more robust against manipulation.

VI. DISCUSSION

Overhead: The most significant overhead in VISHUNTER is the visibility checking on the search engines. However, as shown in Section V-B, visibility of servers are not changed frequently. In other words, we do not need to check the visibility of all the servers everyday.

Limitation: For some visible malicious servers hosted on compromised servers, VISHUNTER may not guarantee to detect them. However, since those visible malicious servers are indexed by search engines, they become good targets for existing work, such as EvilSeed [15] and PoisonAmplifier [33], which explore the shared patterns among the malicious servers and use search engines to find them.

Evasion: An attacker who gains the knowledge of VISHUNTER may attempt to circumvent it by either manipulating the visibility of malicious servers or misleading the VISHUNTER classifier.

To manipulate the visibility of malicious servers, attackers can make their malicious domains to be public leading to malicious redirection from visible servers to visible servers, which will be filtered by VISHUNTER. One way to promote malicious domains to be public is to inject them into other compromised servers. However, this will make them easily to be detected by the administrators of those compromised servers. In addition, researchers can easily find all of the compromised servers by searching the malicious domain in search engines. In addition, if cyber criminals directly submit their domains to search engines, such malicious servers will not link with other benign servers. Therefore, we can use other features such as the number of search results, to assign some weights to the server visibility. Malicious servers will be visible with less weights.

To mislead the classifier, as discussed in Section IV-B, those features can not be easily evaded by attackers without causing a significant amount of cost. Therefore, even some adversaries may still find ways to bypass VISHUNTER, the resource constraints would limit the effectiveness of the adversaries' campaigns or raise higher cost for them.

VII. RELATED WORK

Studies on Malicious Web Infrastructures. Most of existing research on malicious web infrastructure study only focused on specific attack channels associated with malicious web infrastructures. Anderson et al. [8] studied the Internet infrastructure used to host and support scams in terms of its lifetime, stability, and so on. Li et al. [21] focused on malicious web advertising, and built a system to inspect advertisement delivery processes to detect malicious advertising activities. Zhang et al. [31] studied the infrastructure of comment spam

and built a detection system based on the spamming behavior. Recently, Li et al. [20] conducted a study on general malicious web infrastructures based on the redirection topology, and detected 12 times more malicious servers. However, their system required initial malicious seeds for bootstrapping and was not applicable to detect single malicious redirection. Zhang et al. [32] detected malicious infrastructure by grouping closely related servers from different perspectives, however, it required multiple infections and malicious servers involved in.

Studies on Malicious Redirections. Leontiadis et al. [18] conducted the first measurement study on a search poisoning attack and found that some high-ranking websites were compromised to dynamically redirect users to online pharmacies. Later, Lu et al.[22] detected malicious redirection chains in a search poisoning attack using a group of features (e.g., poisoning resistance) specific to search poisoning activities. Lee et al. [17] identified malicious redirections on Twitter using the tweet features, such as appearing frequencies and the correlation of redirection chains in tweets. Wang et al. [29] proposed an approach to indirectly detect malicious redirections based on the cloaking techniques used by attackers. More similar to our work is that of Stringhini et al. [28] which detected general malicious servers using the features extracted from interactions between a crowd of web users' browsers with websites. However, an immediate limitation of the system is its requirements for a large and diverse user base which may limit its applicability in practice. VISHUNTER differs from the previous work in that we designed 12 features (8 of them are newly proposed) from visibility perspective to characterize the differences between benign and malicious redirections, which are more robust against manipulation.

VIII. CONCLUSION

In this paper, we conducted the first visibility study of malicious web infrastructures. Our measurement study showed that most core malicious servers in malicious web infrastructure are not visible to benign users, and there exist significant differences between benign and malicious entrances to invisible web infrastructures. Leveraging our new findings, we designed a lightweight yet effective system, VISHUNTER, to detect the entrances to malicious web infrastructures. We believe that VISHUNTER greatly complements the previous work on detecting malicious web infrastructures in that VISHUNTER leverages several new features that are harder to be evaded. In the future, we will study the redirections from invisible servers to invisible servers to find more malicious servers under the malicious infrastructure.

IX. ACKNOWLEDGMENTS

This material is based upon work supported in part by the the National Science Foundation (NSF) under Grant no. CNS-1314823, CNS-1218929, and CNS-0954096, and the Air Force Office of Scientific Research (AFOSR) under Grant No. FA-9550-13-1-0077. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and AFOSR.

REFERENCES

[1] Alexa Internet. http://www.alexa.com/.
[2] DNS-BH-Malware Domain Blocklist. http://www.malwaredomains.com/.
[3] Easylist. https://easylist.adblockplus.org/en/.
[4] Malware domain list. http://www.malwaredomainlist.com/.
[5] Malware traffic analysis. http://malware-traffic-analysis.net/.
[6] TLD List. https://wiki.mozilla.org/TLD_List.
[7] Websense 2013 threat report. http://www.websense.com/assets/reports/websense-2013-threat-report.pdf.
[8] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: characterizing internet scam hosting infrastructure. In *USENIX Security Symposium'07*, 2007.
[9] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *USENIX Security Symposium'12*, 2012.
[10] M. Antonakakis, P. R, W. Lee, N. Vasiloglou, and D. Dagon. Detecting malware domains at the upper DNS hierarchy. In *USENIX Security Symposium'11*, 2011.
[11] K. Borgolte, C. Kruegel, and G. Vigna. Delta: automatic identification of unknown web-based infection campaigns. In *CCS*, 2013.
[12] M. Cova, C. Kruegel, and G. Vigna. Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code. In *WWW'10*, 2010.
[13] G. De Maio, A. Kapravelos, Y. Shoshitaishvili, C. Kruegel, and G. Vigna. PExy: The Other Side of Exploit Kits. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2014.
[14] C. Hsu, C. Huang, and K. Chen. Fast-flux bot detection in real time. In *RAID'10*, 2010.
[15] L. Invernizzi, S. Benvenuti, P. Comparetti, M. Cova, C. Kruegel, and G. Vigna. EVILSEED: A Guided Approach to Finding Malicious Web Pages. In *IEEE Symposium on Security and Privacy (Oakland'12)*, 2012.
[16] M. Konte, N. Feamster, and J. Jung. Fast flux service networks: Dynamics and roles in hosting online scams. *Retrieved August*, 13:2011, 2008.
[17] S. Lee and J. Kim. WarningBird: Detecting suspicious URLs in Twitter stream. In *NDSS'12*, 2012.
[18] N. Leontiadis, T. Moore, and N. Christin. Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade. In *USENIX Security Symposium'11*, 2011.
[19] Z. Li, S. Alrwais, X. Wang, and E. Alowaisheq. Hunting the Red Fox Online: Understanding and Detection of Mass Redirect-Script Injections. In *IEEE Symposium on Security and Privacy*, 2014.
[20] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. Wang. Finding the Linchpins of the Dark Web: a Study on Topologically Dedicated Hosts on Malicious Web Infrastructures. In *IEEE Symposium on Security and Privacy (Oakland'13)*, 2013.
[21] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang. Knowing your enemy: understanding and detecting malicious web advertising. In *CCS'12*, 2012.
[22] L. Lu, R. Perdisci, and W. Lee. SURF: Detecting and Measuring Search Poisoning. In *CCS'11*, 2011.
[23] N. Nikiforakis, F. Maggi, G. Stringhini, M. Z. Rafique, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, and S. Zanero. Stranger danger: exploring the ecosystem of ad-based URL shortening services. In *WWW*, 2014.
[24] R. Perdisci, I. Corona, and G. Giacinto. Early Detection of Malicious Flux Networks via Large-Scale Passive DNS Traffic Analysis. In *IEEE Transactions on Dependable and Secure Computing, 9(5), Sept.-Oct. 2012, pp. 714-726*, 2012.
[25] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero. Phoenix: DGA-based Botnet Tracking and Intelligence. In *DIMVA'14*, 2014.
[26] C. Seifert, I. Welch, and P. Komisarczuk. HoneyC - The Low-Interaction Client Honeypot. In *NZCSRCS'07*, 2007.
[27] J. W. Stokes, R. Andersen, C. Seifert, and K. Chellapilla. WebCop: Locating Neighborhoods of Malware on the Web. In *USENIX LEET*, 2010.
[28] G. Stringhini, C. Kruegel, and G. V. . Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. In *20th ACM Conference on CCS*, 2013.
[29] D. Y. Wang, S. Savage, and G. M. Voelker. Cloak and Dagger: Dynamics of Web Search Cloaking. In *CCS'11*, 2011.
[30] B. Wu and B. D. Davison. Cloaking and redirection: A preliminary study. In *AIRWeb'05*, 2005.
[31] J. Zhang and G. Gu. NeighborWatcher: A Content-Agnostic Comment Spam Inference System. In *NDSS'13*, 2013.
[32] J. Zhang, S. Saha, G. Gu, S. Lee, and M. Mellia. Systematic mining of associated server herds for malware campaign discovery. In *ICDCS*, 2015.
[33] J. Zhang, C. Yang, Z. Xu, and G. Gu. PoisonAmplifier: A Guided Approach of Discovering Compromised Websites through Reversing Search Poisoning Attacks. In *RAID'12*, 2012.