Still Beheading Hydras: Botnet Takedowns Then and Now

Yacin Nadji, Roberto Perdisci, and Manos Antonakakis

Abstract—Devices infected with malicious software typically form botnet armies under the influence of one or more command and control (C&C) servers. The botnet problem reached such levels where federal law enforcement agencies have to step in and take actions against botnets by disrupting (or "taking down") their C&Cs, and thus their illicit operations. Lately, more and more private companies have started to independently take action against botnet armies, primarily focusing on their DNS-based C&Cs. While well-intentioned, their C&C takedown methodology is in most cases ad-hoc, and limited by the breadth of knowledge available around the malware that facilitates the botnet. With this paper, we aim to bring order, measure, and reason to the botnet takedown problem. We improve an existing takedown analysis system called *rza*. Specifically, we examine additional botnet takedowns, enhance the risk calculation to use botnet population counts, and include a detailed discussion of policy improvements that can be made to improve takedowns. As part of our system evaluation, we perform a postmortem analysis of the recent 3322.org, Citadel, and No-IP takedowns.

Index Terms—K.6.m, K.5.m

1 Introduction

Botnets represent a persistent threat to Internet security. To effectively counter botnets, security researchers and law enforcement organizations have been recently relying more and more on *botnet takedown* operations. Essentially, a botnet takedown consists of identifying and disrupting the botnet's command-and-control (C&C) infrastructure. For example, in 2009 law enforcement and security operators were able to takedown the Mariposa botnet, which at that time consisted of approximately 600,000 bots. The takedown operation was accomplished by first identifying the set of domain names through which bots would locate their C&C network infrastructure. By seizing this set of domains via a collaboration with domain registrars, security operators effectively "sinkholed" the botnet, thus shunting the C&C traffic away from the botmaster and avoiding any further commands to be issued to the bots.

While sophisticated botnet developers have attempted, in some cases successfully, to build peer-to-peer (P2P) botnets that avoid entirely the use of C&C domains [1], most modern botnets make frequent use of the domain name system (DNS) to support their C&C infrastructure. This is likely due to the fact that DNS-based botnets are much easier to develop and manage compared to their P2P-based counterparts, and yet provide a remarkable level of *agility* that makes a takedown challenging. For example, the Mariposa case required a coordinated effort involving law enforcement, security operators, and domain registrars across several different countries. In addition, some recent takedown efforts [2] have caused some level of collateral damage, thus raising both technical issues and policy-related questions regarding the efficacy of botnet takedowns.

In this paper, we propose a novel *takedown analysis system*, which we call *rza*. Our main goal is to provide a way to "go back in time" and *quantitatively analyze past takedown efforts* to highlight incomplete takedowns and identify what worked and what could have been done better. Specifically, *rza* identifies additional domains that are likely part of a botnet's C&C infrastructure by examining historical relationships in the DNS and analyzing the botnet's malware samples. This aids the takedown process by identifying domains that may have been missed by hand, both from the network-level and the malware-level, aggregating this information, and automatically labeling the domains with evidence of their maliciousness. While *rza* focuses on disrupting botnets that use DNS-based C&C infrastructure, it can also assist in cases where botnets are more advanced and use domain name generation algorithms (DGA) or communicate using a peer-to-peer structure. In particular, *rza* provides the first few steps for remediating advanced C&C infrastructure: (i) identifying DNS-based primary C&C infrastructure, if it exists; (ii) automatically identifying if the botnet has DGA or P2P capabilities; and (iii) automatically identifying the malware samples that exhibit these behaviors to triage binaries for reverse engineering. To successfully takedown DGA/P2P botnets we must fully understand their non-deterministic portions, such as the randomness seed for DGAs [3] or the peer enumeration and selection algorithms for P2P [1]. If we disable a botnet's primary infrastructure but do not account for the DGA-based backup mechanism, our efforts will be futile.

We show that in cases of past takedowns, likely malicious domain names were left unperturbed. Worse yet, in

- *Y. Nadji and M. Antonakakis are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. E-mail: {yacin, manos}@gatech.edu.*
- *R. Perdisci is with the Department of Computer Science, University of Georgia, Athens, GA. E-mail: perdisci@cs.uga.edu.*

Manuscript received 16 July 2015; revised 14 Oct. 2015; accepted 21 Oct. 2015. Date of publication 29 Oct. 2015; date of current version 1 Sept. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2015.2496176

some cases malicious domains were unintentionally given enterprise-level domain name resolution services. We show that *rza* can identify additional sets of domain names that ought to be considered in a future takedown, as well as automatically identify malware contingency plans when their primary C&C infrastructure is disabled.

In this paper, we improve our work described in [4] and present the following contributions:

- We perform two additional botnet takedown postmortem analyses to explain the current state of botnet takedowns. We show that takedowns are still largely ineffective and run the risk of drawbacks from collateral damage or friendly fire.
- We augment risk calculation to include population measurements to further improve the takedown recommendation engine and analyze more recent botnets.
- We expand the policy discussion based on what we have learned in the interim period.

The remainder of the paper is structured as follows: Section 2 provides the necessary background on the DNS, botnet takedowns, and our datasets. Section 3 describes *rza* in detail and Section 4 describes the process for interrogating malware samples. Section 5 presents our postmortem experiments and analyses of three recent, high-profile takedown attempts. In Section 6.2 we discuss non-technical difficulties associated with performing takedowns that, if alleviated, would likely improve the process with respect to coordination and preventing potential collateral damage.

2 Background

In this section, we first provide an historical explanation of some past takedowns and explain why they deserve to be studied in detail. Then, we describe the datasets used by *rza* to perform takedown analysis and to build the takedown recommendation system.

2.1 Botnets and Takedowns

Botnet takedowns are not uncommon, and may take many different forms. Considering the heterogeneous nature of client machines and the difficulty in keeping individual machines clean from infection, taking down the botnet C&C is an attractive alternative. A successful takedown eliminates most external negative impacts of the botnet, effectively foiling further attacks (e.g., spam, DDoS, etc.) by the infected hosts, which can number in the millions. In the past, takedowns have been performed by revoking sets of C&C IP addresses from hosting providers, de-peering entire Autonomous Systems (AS), or, more recently, sinkholing or revoking C&C domains.

Conficker is an Internet worm that infected millions of computers and remains one of the most nefarious threats seen on the Internet to date [3]. Conficker's latter variants employed a DGA that would generate 50,000 pseudo-random domain names every day to communicate with its C&C server. The takedown of Confiker required immense coordination across hundreds of countries and top-level domains (TLDs), and numerous domain registrars and registries. The takedown efforts were coordinated by the Conficker Working Group (CWG) [3]. The takedown required reverse-engineering the malware binaries, and reconstructing the DGA. Then, the CWG pre-registered all 50,000 domains per day that could potentially be used for C&C purposes, thus preventing the botmaster from regaining control of the bots. The success of CWG's efforts highlight the importance of participation and support from key governing and regulatory bodies, such as ICANN, and the need of cooperation between the private sector and governments around the world.

Mariposa, a 600,000-strong botnet of Spanish origin, provides another example of a takedown operation initiated by a working group that relied on sinkholing known malicious domains. Interestingly, Mariposa's botmasters were able to evade a full takedown by bribing a registrar to return domain control to the malicious operators [5], underscoring the fact that barriers to successful takedowns are not only technical ones.

The DNSChanger [6] "click-jacking" botnet was also taken down through a working group. DNSChanger altered upwards of 300,000 clients' DNS configurations to point to rogue DNS resolvers under the control of the attackers. This allowed the attackers to direct infected hosts to illegitimate websites, often replacing advertisements with their own to generate revenue. DNSChanger had to be taken down by physically seizing the botnet's rogue DNS servers. The takedown was accomplished in late 2011. Largely considered successful, the DNSChanger once again shows the importance of collaboration when performing comprehensive takedowns.

Not all takedowns are performed at the DNS-level, however, as shown in the takedowns of McColo [7], AS Troyak [8], and other "bulletproof hosting providers," or networks known to willingly support malicious activities. These are extreme cases where the networks in question essentially hosted only malicious content, and removing the entire network would disable large swaths of botnets and related malicious network infrastructure. The effect of these takedowns were indirectly measured by witnessing drops in spam levels, for example, upwards of two-thirds decrease after McColo's shutdown [9]. Unfortunately, if a particular botnet relied on the DNS to perform C&C resolutions into these bulletproof networks, once a new host was provisioned the threat would continue. Sure enough, we saw spam levels rise back to normal levels as botnets moved to other hosting providers [10].

2.2 Datasets

rza relies on two primary data sources: a large passive DNS (pDNS) database and a malware database that ties malicious binaries to the domain names the query during execution.

Passive DNS: A passive DNS database stores historic mappings between domain names and IP addresses based on successful resolutions seen on a live network over time. pDNS databases allow us to reconstruct the historical structure of DNS-based infrastructure based on how it was used by clients. Our pDNS is constructed from real-world DNS resolutions seen in a large North American ISP collected by Damballa, Inc. beginning on January 1, 2011. The data is approximately 27 terabytes compressed. This allows us to identify the *related historic domain names* (RHDN) for a given IP, namely all domains that resolved to that IP in the past.

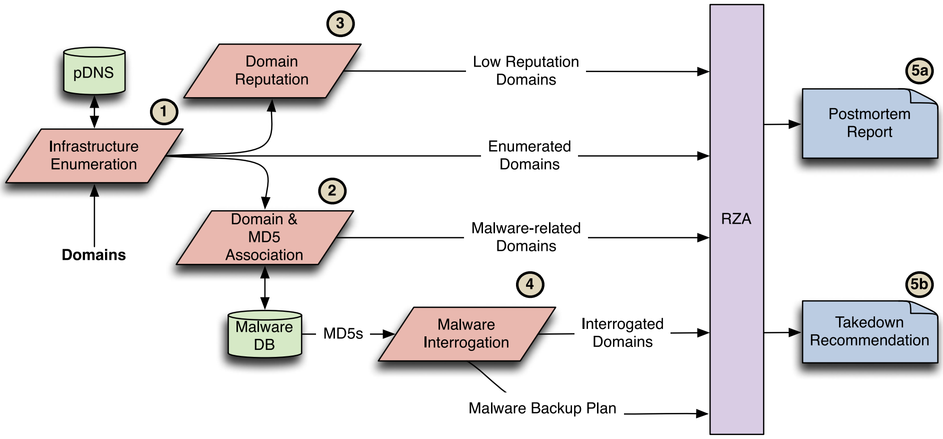

Fig. 1. Overview of *rza*.

Also, pDNS allows us to find the *related historic IP addresses* (RHIP) for a given domain name, i.e., all the IPs to which the domain resolved to in the past. Furthermore, the RHIP/RHDNs can be limited to domain-to-IP mappings that occurred during a particular time frame of interest, thus allowing us to focus on the crucial days before and after a takedown took place.

We provide the following abstract functions to more clearly explain how the data are accessed and processed in the context of *rza*:

- RHIP(`domain`, `start_date`, `end_date`): returns all domains historically related to the `domain` argument over the period between the desired start and end dates. For example, RHIP(`foo.com`, 2012/01/01, 2012/01/05) would return the set of all IP addresses `foo.com` successfully resolved to between January 1st, 2012 and January 5th, 2012, inclusive.
- RHDN(`IP`, `start_date`, `end_date`): similarly, RHDN returns all domains historically related to the IP argument over the period between the start and end dates.
- Volume(`domain` and/or `IP`, `date`): the total successful lookup volume to the argument domain, IP, or domain and IP tuple on the argument date.
- Population(`domain`, `date`): the total number of distinct clients that queried the argument domain, or domain and IP tuple on the argument date. Population can only be computed for domains that are queried after November 14, 2012 due to data availability.

It is important to note that our use of private pDNS data was dictated mainly by convenience and cost issues. To demonstrate that *rza* can properly function using different sources of passive DNS data, we obtained temporary access to the Farsight passive DNS database [11], which is available to other researchers and offers an arguably more global perspective.

Malware domains We also make use of a separate malware database that contains mappings between a malware sample's MD5 sum and binary and the domain names and IP addresses it has queried during dynamic malware analysis. Each entry in the database is a 4-tuple that includes the MD5 of the malware sample, the queried domain name, the resolved IP address, and the date and time of the analysis. These data are collected from a combination of internal malware analysis output from Damballa as well as the output from a commercial malware feed. The commercial feed goes back until January 2007 and when combined with Damballa's internal data contains roughly 505 million malware execution runs and is approximately 200 gigabytes compressed.

3 RZA SYSTEM

In this section, we detail the internals of *rza*, our takedown analysis and recommendation system.

3.1 Overview

Fig. 1 shows the overall process implemented by *rza*. Given a set of known seed botnet domains D_S, *rza* can be asked to generate either a "Postmortem Report" or a "Takedown Recommendation".

In the "Postmortem Report" mode, the input domains represent the domains known to have been targeted by an historic takedown. This produces a report that shows the effectiveness of the takedown of the domain names (Fig. 1, step 5a) with respect to the expanded infrastructure *rza* identifies.

In the "Takedown Recommendation" mode, the input domains represent the currently known malicious domains used for C&C infrastructure. Furthermore, the takedown recommendation engine explores possible network resources that may be used by the botnet as a C&C backup mechanism, and suggests any additional measures that must be taken after the primary C&C is disabled to fully eliminate the threat (Fig. 1, step 5b).

At a high level, the processing steps executed by *rza* are similar when producing both the "Postmortem Report" and "Takedown Recommendation", despite the difference in inputs and the meaning of the results. The steps are:

1. Expand the initial domain seed set D_S using the pDNS database to identify other domains that are likely related to the botnet's C&C infrastructure. Intuitively, domains are cheap but IP addresses are relatively more expensive. By identifying additional domains that resolve to the same hosts as malicious

domains, we can identify other potentially malicious domains related to the botnet.

2. Identify the subset of the expanded domains that are queried by known malware samples. If a domain both points to a host known to facilitate a C&C and is also used by known malware, it increases the likelihood of that domain itself being malicious as well.
3. Identify the subset of the expanded domains with low domain name reputation. Similar to the intuition of Step 2, a domain that points to a known malicious host and also has low domain reputation is more likely to itself be malicious.
4. Analyze the malware samples identified in Step 2. In addition to straightforward dynamic malware analysis, we trick executing malware samples into believing that their primary C&C infrastructure is unavailable using a custom malware analysis system [12] to extract additional C&C domain names. Intuitively, domains used by malware related to the infrastructure we are studying are likely to be related and malicious. Furthermore, we use the results of the analysis to identify malware contingency plans that would allow the botnet to continue to function after its primary C&C infrastructure has been disabled (e.g., a DGA-based or P2P C&C).
5. Output either the "Postmortem Report" or "Takedown Recommendation" depending on the mode of operation selected at the beginning.

The guiding principle we follow with *rza* is to push our understanding of malicious C&C infrastructure towards completeness. Only once we have fully enumerated the C&C infrastructure can we successfully disable it. We can begin to enumerate C&Cs from the network-level by identifying historic relationships between domain names and hosts using pDNS evidence, and from the host-level by interrogating malware samples. Since the pDNS may contain additional domains not necessarily related to the botnet in question, we identify subsets of domains so we can focus our investigative efforts on those that are most likely to be malicious and not inundate ourselves with information. Each subset serves a different purpose: the low reputation subset holds the domain names from the network-level that are most likely to be malicious. The subset of domains queried by malware represents a reasonable baseline to expect from prior takedowns, as much of this information is readily available to the security community. The subset gleaned from malware analysis contains the domains from the host-level that are the most likely to be malicious. We can use these sets to measure the effectiveness of past takedowns and recommend domains for future takedowns.

In the remainder of this section we describe each of these high-level tasks in detail, and discuss how they work together to suggest a takedown response.

3.2 Infrastructure Enumeration

Botnets often make use of the DNS to increase the reliability of their C&C infrastructure, for example using domain name fluxing or simply replacing retired or blacklisted domains with new domains. This cycling of domains, however, leaves a trail in the pDNS database and can be used to enumerate the infrastructure. For example, consider a malware sample m that on day t_1 uses domain d_1 as its primary C&C domain, but on day t_2 switches to domain d_2 to evade the blacklisting of d_1. Assume d_1 and d_2 resolve to the same IP address. Analysis during either t_1 or t_2 yields only one of the possible domains, but the relationship between d_1 and d_2 can be identified in a pDNS database because both resolved to the same IP address.

Using the passive DNS database and the seed domain set D_S, we compute the enumerated infrastructure domain set D_e using Algorithm 1. First, the related-historic IPs of D_S are retrieved and known sinkhole, parking, and private IP addresses are removed. The related-historic domain names for the remaining IPs are retrieved, and any benign domain names are removed, yielding the enumerated infrastructure of D_S: D_e. The relationships retrieved from the pDNS database are within a range of dates to ignore historic relationships that are no longer relevant. This constant is customizable and was empirically chosen.

Algorithm 1. Infrastructure Enumeration Procedure

```
Input: $D_S$, startdate, enddate: seed domain set, and bounding dates
Output: $D_e$: enumerated domain set
  $I_b \leftarrow$ set of known sinkhole, private, parking IPs
  $W_d \leftarrow$ set of Alexa top 10,000 domain names
  $I \leftarrow RHIP(D_S, \text{startdate}, \text{enddate})$
  $I \leftarrow I \setminus I_b$
  $D_e \leftarrow RHDN(I, \text{startdate}, \text{enddate})$
  $D_e \leftarrow D_e \setminus W_d$
  return$D_e$
```

To understand why we filter out benign domains consider an attacker that, in an attempt to mislead our analysis, temporarily has their malicious domains resolve into benign IP space (e.g., Google's) or uses a popular hosting provider (e.g., Amazon AWS). If either of these occur, the D_e domain set may include unrelated, benign domain names. To handle this, we filter domains if they are a member, or are a subdomain of a member, of the set of the Alexa top 10,000 domain names. These domains are unlikely to be persistently malicious and should not be considered for takedown. IP addresses that are non-informative (private, sinkhole, etc.) are also removed, as the domains that resolve to them are unlikely to be related. For example, malware domains sometimes point to private IP addresses (e.g., `127.0.0.1`) when they are not in use, which if not removed would link otherwise unrelated domain names. We use the Alexa top 10,000 when performing malware interrogation (see Section 4), and for consistency we use it here as well. In future work we intend to explore the effect of using smaller and larger whitelists on the generated sets and their accuracy.

3.3 Categorizing the Expanded Infrastructure

Not all domains identified during the infrastructure enumeration process are guaranteed to be malicious, but we can identify subsets that are more likely to be malicious. For example, a domain that resolves to an IP address in a virtual web hosting provider is likely to have many benign and unrelated domains that resolve to the same infrastructure as well. To account for this, we focus on domains with known

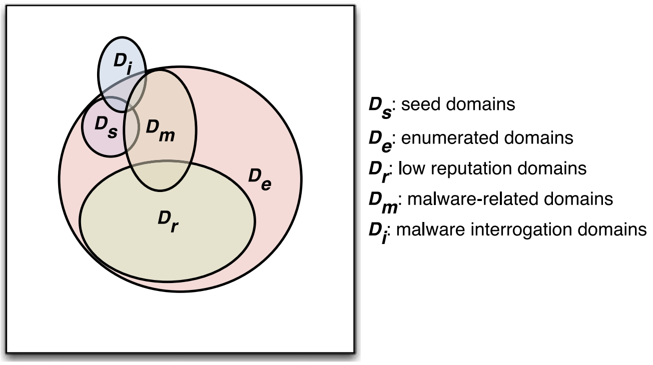
D_i
D_s
D_m
D_e
D_r
D_s: seed domains
D_e: enumerated domains
D_r: low reputation domains
D_m: malware-related domains
D_i: malware interrogation domains

Fig. 2. Venn diagram of identified infrastructure sets.

(often public) malware associations, and domains that have low domain name reputation.

Using the passive DNS, we expand the initial seed domain set, D_S, into the expanded set D_e. Next, we identify $D_m \subseteq D_e$ and $D_r \subseteq D_e$, the subset of domain names in D_e with known malware associations and low domain name reputation, respectively. Malware associations are retrieved from our domain name to malware MD5 database and are commonly available in the security community [13]. To determine if a domain name has low reputation, we use a system similar in spirit to [14], [15] which scores domain reputation between 0.0 and 1.0, where 1.0 denotes a low reputation (i.e., likely malicious) domain name. Any domains with > 0.5 reputation are considered malicious and are added to D_r. Unlike D_r and D_m, the set D_i is not necessarily a subset of D_e. Any domains that are used by malware during malware interrogation are added to D_i. These domains expand our coverage as they may unearth domain names that were not previously included in D_e. During our postmortem analysis, we compare these sets to the domains that were actually involved in the takedown (D_S).

Fig. 2 shows a Venn diagram representation of a possible configuration of enumerated infrastructure sets. All sets, excluding D_i, are subsets of D_e. D_i is the most likely to include domains outside of the scope of D_e, but suffers the most from the problem of completeness as it relies on dynamic malware analysis.

For the postmortems presented in Section 5, the set D_i is not computed due to time limitations. This set is most important to compute when performing a takedown recommendation. Since the focus of this paper is on additional postmortem studies, computing the D_i set is not performed. For completeness, however, the process of computing it and how it is used to perform a takedown recommendation remains.

3.4 Takedown Recommendation Engine

Using the four aforementioned techniques, we can run our takedown protocol as shown by the decision tree in Fig. 3. Suppose we are interested in taking down a hypothetical botnet where the current known infrastructure is $D_S = \{$`01.hans.gruber.com`$\}$. After enumerating the infrastructure, we identify the additional domain name `02.hans.gruber.com` that resolves to the same IP as the `01` child domain. We identify and retrieve the malware samples that have queried the `01...` and `02...` domain names and interrogate them. We identify an additional domain name, `03.hans.gruber.com`, when the first two domain names fail to resolve. Since we identified a finite number of new domain names, we re-run the process with the expanded set of three domain names and this time the malware analysis yields no behavioral changes from what we have already identified. In the event a DGA or a P2P backup scheme is present, the DGA must be reverse-engineered or the P2P network must be subverted as described in [1] after disabling the main C&C infrastructure, respectively.

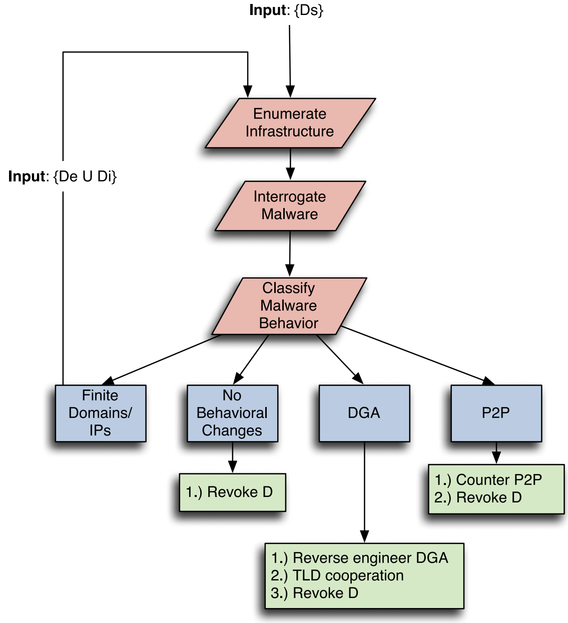

Fig. 3. Takedown recommendation engine shown as a decision tree. D in this case represents either $D_r \cup D_i$, which only targets C&C domains that are very likely to be malicious or $D_e \cup D_i$, or the "nuclear" option that should only be used when the threat of the botnet outweighs potential collateral damage.

The question remains which sets of domains should be revoked or sinkholed in order to terminate the botnet's C&C infrastructure, which ultimately must be decided by human operators. In the case where eliminating the botnet is more important than any possible collateral damage that may be incurred, the set of domains in $D_e \cup D_i$ should be targeted, which we consider to be the "nuclear" option. This contains any domain name associated with the C&C infrastructure as well as domains queried by the related malware. In other scenarios, however, this may incur too much collateral damage. We recommend revoking $D_r \cup D_i$ instead in these cases, as these domains are very likely to be malicious. These decisions should be made by threat researchers based on the potential risks associated with deactivating these domain names. Another, less extreme option is to simply block these domains at the network's egress point. This allows enterprise-sized networks to protect themselves while lessening the negative impact incurred by collateral damage.

Ground truth for C&C infrastructure is difficult to come by, which makes evaluating true positives and false positives exceedingly difficult. To roughly estimate this, we present the precision and recall of each set against the "correct" set of $D_r \cup D_i$. If we assume that domains flagged

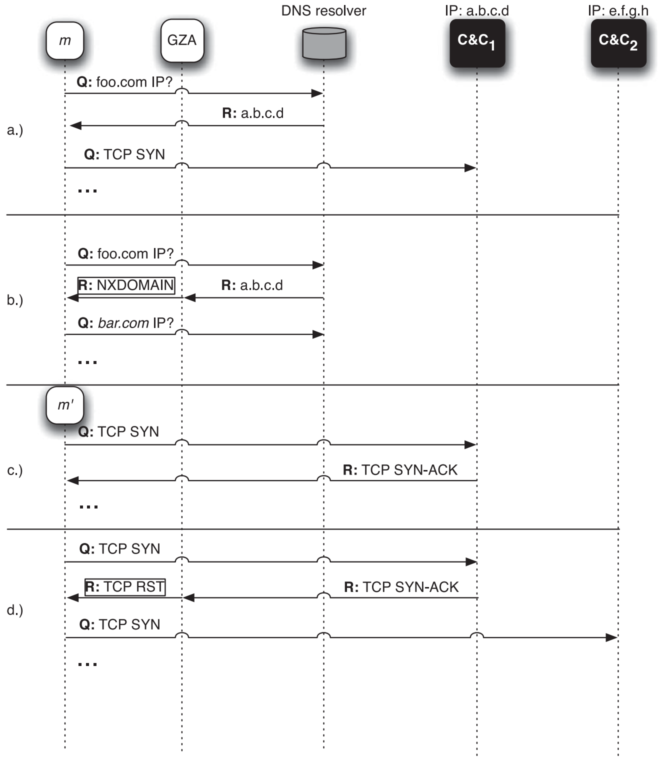

Fig. 4. Malware samples m (a-b) and m' (c-d) initiating a connection with the C&C server. m connects by first performing a DNS query to determine the IP address of its C&C server followed by initiating a TCP connection. Sample m' connects directly to the C&C using a hard-coded IP address. Examples (a) and (c) connect without intervention by a game, while (b) and (d) have false information (denoted by boxes) injected.

as low reputation or used by malware known to be affiliated with a given botnet are malicious, we can use this union to roughly correspond to ground truth. In our case, the precision of a set D is the fraction of the number of domain names d that are $d \in D \wedge d \in D_r \cup D_i$ over the size of D or $|D|$ and the recall is the fraction between the same number of domain names as in the precision but over the size of the "correct" set, or $|D_r \cup D_i|$.

3.5 Use of Other Sources of pDNS Data

Out of both financial and analysis convenience, we ran our experiments using Damballa's internal passive DNS database. To show that our results are not tied to private data and can be replicated by other researchers, we run a subset of our experiments using Farsight's passive DNS database [11]. While the database is not exactly public, it is generally available to practicing researchers and professionals in the security community (possibly for a fee). As pDNS data becomes more popular, we expect the number of these databases to increase and become more easily accessible by researchers.

Using Farsight's pDNS database and *rza*'s process outlined in this section, we generate the D_e, D_m, and D_r domain sets of one postmortem takedown and five current botnets. As before, we compute the respective TIR values of each set. We chose the five botnets with the largest C&C infrastructure that Damballa began tracking in April, 2013. Our results from the Farsight dataset are presented in Section 5.6.

4 Malware Interrogation

Understanding how malware behaves when its primary infrastructure is disabled is critical to performing a successful takedown. Not only does this reveal additional network assets used in the botnet's infrastructure, it can also reveal sophisticated back up plans, such as DGA- or P2P-based C&C schemes, that must be carefully handled to ensure a successful takedown. In this section, we describe how we play games with malware and how we design and evaluate our heuristics for interrogating malware samples for contingency plans.

4.1 Playing Games with Malware

Malware uses the same network protocols that benign software uses when performing malicious activity. Despite the fact that many network protocols exist, nearly all communication on the Internet follows one of two patterns:

1. A transport layer (e.g., TCP and UDP) connection is made to an IP address directly, **or**
2. A DNS query is made for a domain name (e.g., `google.com`) and a connection to the returned IP address is made as in #1.

Higher-level protocols leverage these two use cases for nearly all communication. If we can assume malware relies on these two patterns for contacting its C&C servers and performing its malicious activities, these are the patterns we must target during analysis.

We define a *network game* to be a set of rules that determine when to inject "false network information" into the communication between a running malware executable and the Internet. More specifically, false information is a forged network packet. Consider the running malware sample m in Fig. 4a. Sample m first performs a DNS query to determine the IP address of its C&C server located at `foo.com`. The returned IP address, `a.b.c.d`, is then used to connect to the C&C and the malware has successfully "phoned home". Sample m could also bypass DNS entirely if it were to hardcode the IP address of its C&C and communicate with it directly, as we see in Fig. 4c. This gives us two opportunities to play games with sample m as shown in Figs. 4b, and 4d: we can say the domain name resolution of `foo.com` was unsuccessful (b) or the direct connection to IP address `a.b.c.d` was unsuccessful (d). At this point, sample m has four possible courses of action:

1. Retry the same domain name or IP address,
2. Remain dormant to evade dynamic analysis and try again later,
3. Give up, or
4. Try a previously unused domain name or IP address.

In (b) and (d), we see the malware samples taking action #4 and querying a *previously unseen* domain name (`bar.com`) and IP address (`e.f.g.c`), respectively. Action #2 is a common problem in dynamic malware analysis systems in general and is further discussed in Section 4.4.

4.1.1 Notation

Stated more formally, let h be a machine infected with a malware sample m that is currently executing in our analysis system running game G_{name}. G_{name} is a packet

TABLE 1
Notation for Describing Games and the Sets They Generate

G_{name}	A game called *name*.
$G_{\mathrm{name}}(p)$	The result of G_{name}'s transformation on packet p.
$G_{\mathrm{name,m}}$	The set of network information, i.e., unique IP addresses and domain names contacted, generated when malware sample m is gamed by G_{name}.
G^{M}_{name}	The subset of malware samples from M that were successfully gamed by G_{name}.
$D(s)$, $I(s)$	Given a sample set, s, return the subset of unique domain names or IP addresses in s, respectively.

transformation function called *name*. Given a packet p, $G_{\mathrm{name}}(p)$ represents G_{name} *gaming* p and its value is either the original packet, or some altered packet p' that changes the intent of p. The implementation details of G_{name} determine when to return p or p'. For example, p could contain the resolved IP address of a queried domain name d, whereas p' says d does not exist. In all other ways, such as type of packet and source and destination IP addresses, p and p' are identical. As h communicates with the outside world, it sends question packets, q_i, in the form of domain name queries and requests to initiate a TCP connection and receives response packets, r_j, in the form of domain name resolutions and initiated TCP connections.[1] False information is provided to the host h by delivering $G_{\mathrm{name}}(r_j)$ in lieu of r_j. A *sample set* for m, $G_{\mathrm{name,m}}$ represents the set of unique domain names and IP addresses queried by m while running under G_{name}. The functions D and I operate on sample sets and return the subset of unique domain names **or** the subset of unique IP addresses, respectively. Given a set of malware sample MD5s, M, a *game set*, G^{M}_{name} represents the subset of samples that were "successfully gamed" by G_{name}. A game is considered successful if it forces a malware sample to query more network information than under a run without a game present. We formally define this in the following section. The described notation is summarized in Table 1 and will be used throughout the remainder of the paper.

4.1.2 Designing Games for Interrogating Malware

Crafting games without a priori knowledge of malware network behavior is difficult. Furthermore, a successful game for sample m may be unsuccessful for sample m'. By using generic games "en masse", we improve our chances of successfully gaming malware during analysis. We design a suite of games to coerce a given malware sample into showing its alternative plan during analysis. We apply *all* games to a malware sample to improve the likelihood of success. Each game focuses either on DNS or TCP response packets in an attempt to harvest additional C&C domain names or IP addresses, respectively. For a DNS response packet p_d, p'_d is a modified response packet that declares the queried domain name does not exist, i.e., a DNS `rcode` of `NXDOMAIN`. For a TCP response packet p_t, p'_t is a modified response packet that terminates the three-way TCP handshake, i.e., a TCP-RESET packet. In this paper, we choose to focus on DNS/TCP packets as they are the predominant protocols used to establish and sustain C&C communication; however, our approach is general and can be adapted to other protocols used less commonly in C&C communication. The design of an individual game is based on anecdotal evidence of how malware samples, in general, communicate. We design seven games to perform our analysis of alternative plan behavior in malware:

G_{null}: To provide a baseline to compare the effectiveness of future games, this game allows response packets to reach its host without modification. In other words, G_{null} is the identity function.

Note that this does not mean malware communication is allowed to run completely unfettered. We perform standard precautionary measures to prevent malicious activity from harming external systems. However, these measures are not considered part of our network games, but simply good practice when analyzing potentially malicious binaries.

G_{dnsw}: A popular domain name, like `google.com`, is unlikely to operate as a C&C server for a botnet. Therefore, DNS queries on popular domain names are unlikely to be concealing additional malicious network information. For a DNS response packet p_d, $G_{\mathrm{dnsw}}(p_d)$ returns p_d if the domain being queried is whitelisted and p'_d otherwise. Our whitelist is comprised of the top 1,000 Alexa domain names [16]. G_{dnsw} is successful for m iff $|G_{\mathrm{dnsw,m}}| > |D(G_{\mathrm{null,m}})|$.

G_{tcpw}: An IP address that resides in a known benign network is also unlikely to function as a C&C, much like a popular domain name. For a TCP response packet p_t, $G_{\mathrm{tcpw}}(p_t)$ returns p_t if the IP being queried is whitelisted and p'_t otherwise. Our whitelist is the dnswl IP-based whitelist [17]. G_{tcpw} is successful for m iff $|G_{\mathrm{tcpw,m}}| > |I(G_{\mathrm{null,m}})|$.

4.2 Methodology

We can interrogate a single malware sample under different environmental conditions to learn additional domains it may use to reach its C&C, as well as any contingency plans for C&C infrastructure failure. We identify the set of malware samples M that communicate with domains in D_e for interrogation. To accomplish this, we can use our existing system that studies malware's behavior under primary C&C failure [12] to automatically determine malware backup plans. We run an individual malware sample under five execution scenarios, extract the network endpoints the malware sample used to "phoned home", and based on the differences observed during executions, we identify likely backup plans.

Behaviorally, most malware when presented with unavailable centralized infrastructure resort to one of the following backup plans:

1. The malware simply retries connecting to hardcoded domains and/or IP addresses.
2. The malware attempts to connect to a *finite* set of additional domains and/or additional IP addresses.
3. The malware attempts to connect to an "*infinite*" set of domains and/or IPs. This occurs when a malware uses a DGA- or P2P-based backup system.

[1] More accurately, a TCP response packet is a TCP SYN-ACK packet as part of the TCP connection handshake.

TABLE 2
Malware Family Training Set Breakdown for Malware Interrogation

Malware Family	**Count**
Expiro.Z	100
Conficker	100
Murofet	97
TDSS/TDL4	92
ZeroAccess	82
zbot	25
Vilsel	25
Onlinegames	25
Fakealert	25
Boonana	20

We can isolate and detect these behaviors by running each sample and applying various packet manipulation scenarios to simulate infrastructure takedown. As a control, we manipulate *none* of the packets during execution. To show that a domain name has been revoked, we rewrite all DNS response packets that resolve non-whitelisted domain names to say the domain no longer exists (NXDomain). We run a sample under this scenario twice for durations t and $2t$. To feign IP address takedowns, we interrupt TCP streams with TCP reset (RST) packets when the destination is to a non-whitelisted IP address. We also run this scenario for durations t and $2t$. Intuitively, if the number of endpoints (domains or IPs) remains consistent across all runs, the malware sample does not include a contingency plan for C&C failure. If the number of endpoints is greater when the DNS or TCP rewriting is enabled, but remains similar between the two runs with different durations, we expect the malware contains a finite set of additional endpoints as a backup mechanism. However, if we see many more endpoints in the $2t$ duration run than in the t run, this suggests the malware is capable of constantly generating additional candidate domains or IPs to connect to, which indicates DGA or P2P behavior, respectively. In the event that the primary C&C infrastructure is already disabled as we would expect in the postmortem studies, the interrogation results still hold. If the botnet employs a backup DGA/P2P mechanism, we will still detect this as the t and $2t$ duration runs will still differ. The system may misclassify a sample as having no backup plan if its infrastructure is already disabled, but this is unlikely to effect *rza* from functioning properly. Consider a sample m that has a finite number of backup domains, but all of the primary domains have already expired and return NXDomain. The control run and DNS rewriting run will be identical and the sample will be misclassified as having no backup behavior, however, we will still identify all the backup domains so the results will still hold.

We empirically design heuristics using the above intuition and by analyzing 595 malware samples from 10 malware families with known contingency plans and catering our rules to perform the identification. Of the samples analyzed, 433 had no contingency plan, 55 used a DGA, 81 used P2P communications, and 22 employed a finite set of backup domains. None of the analyzed malware used a finite number of additional IP addresses.

TABLE 3
Confusion Matrix for Malware Interrogation

	dga	**finitedomain**	**finiteip**	**none**	**p2p**
dga	53	1	0	1	0
finitedomain	0	21	0	1	0
finiteip	0	0	0	0	0
none	4	2	1	426	0
p2p	1	1	1	1	77

4.3 Results

We interrogated 591 malware samples from 10 malware families shown in Table 2. The families have known contingency plans with which we can use to tune our heuristic rules to perform the identification. Of the samples analyzed, 433 had no contingency plan, 55 used a DGA, 81 used P2P communications, and 22 employed a finite set of backup domains. None of the analyzed malware used a finite number of additional IP addresses. Our heuristics successfully classified 97 percent of the samples' contingency plans correctly. A confusion matrix of the results is shown in Table 3.

This shows that with very simple heuristics one can correctly identify backup behaviors that may spoil an otherwise perfect takedown.

4.4 Evasion

Attackers are always attempting to evade newly created defenses. The most obvious ways to evade malware interrogation are through timing attacks, peer-to-peer validation of network resource connectivity, communicating with different protocols, or by evading dynamic analysis entirely with excessive timeouts prior to performing malicious behavior. We discuss these evasion techniques and present methods to address these shortcomings. Since our games use RFC-compliant network responses, malware is unable to determine if it is being gamed at the host-level and subsequently must use the network in clever ways to determine its execution environment.

Dynamic malware analysis systems generally execute malware for a fixed period of time, usually around 5 minutes per sample. Malware can remain dormant until this time passes to evade detection and analysis. Prior work addresses this limitation by finding these *trigger-based behaviors* and generating inputs to satisfy the triggers at runtime. This limitation applies to all dynamic analysis systems in general and is orthogonal to the problem we are trying to solve of increasing the network information an executing sample attempts to connect to.

Overhead incurred during usermode packet generation could enable a clever malware author to determine if they are being gamed or not. As a performance improvement, malware games will only route packets relevant to the game in question. For example, when $dnsw_{\mathrm{is}}$eing played, `iptables` will only route UDP packets with a port of 53 destined for a VM. If DNS packets take abnormally long, while packets of other types are unaffected this could alert a malware sample that it is being analyzed. Simply routing all packets through its game would apply this overhead uniformly across all packets, removing the signal.

Peer-to-peer evasion is when a malware sample verifies the results of a DNS or TCP request by asking another infected machine to perform an identical request. If a sample, m, cannot resolve a domain name d, but fellow infected hosts can resolve d successfully, m has reason to believe it is being run under our system. Communicating this information, however, requires the network. This forces m to succumb to gameplay one way or another; gaming of its initial C&C communication or gaming of verification queries to its peers. By focusing on the building blocks of network communication, we force all network activity to be gamed.

To perform a DNS query, a malware sample could query an HTTP-based DNS tool,[2] bypassing the DNS protocol entirely. Furthermore, it could directly connect to a C&C using a non-gamed protocol, such as UDP. These problems are easily addressed by running aggregate games and adding additional protocols. Querying an HTTP-based DNS lookup tool still requires *some* network activity so running DNS and TCP games *simultaneously* would prevent this lookup from succeeding. If an attacker uses another protocol, such as UDP, it is easy to write a new game that targets this new behavior. As malware adapts to the presence of network games, malware analysts can keep pace with malware authors without too much effort.

5 Postmortem Studies

In this section, we describe how we use *rza* to evaluate historical takedowns. We introduce the takedowns we study and describe the measurements we use to understand the effectiveness of the takedown. We end the section with our experimental results on the postmortem studies.

5.1 Postmortem Analysis

For our postmortems, we study the 3322.org NS takedown that targeted the Nitol botnet [2] (aka Operation b70), the Citadel botnet takedown [18], and the No-IP takedown [19]. We chose these takedowns because they are both recent and high profile. For each takedown, we collect the domains described in the temporary restraining orders (TRO) and use these as our seed domains (D_S).

Measuring takedown improvement Prior studies of botnet takedowns relied on secondary measurements, such as global spam volumes, to determine the success of a takedown. Instead, we directly measure the successful domain name resolutions to the identified infrastructure to proxy for the victim population. By comparing the lookup volume to the seed domains (D_S) with the lookup volume to the sets of domains identified by *rza*, we can determine if a takedown was successful and what domains it missed. For example, if all domain sets are equivalent, their lookup volumes will be identical and the takedown would be considered successful.

More formally, for each takedown, t, and its collected seed domains, D^t_S, we generate the enumerated infrastructure sets D^t_e, D^t_m, D^t_r and D^t_i using *rza*. D^t_e is generated using *only* successful DNS resolutions that were issued during the seven days *before* the takedown of t was performed according to the court documents.[3] This allows us to compare what was actually disabled and/or sinkholed during the takedown with what *rza* would have recommended.

For a period of 14 days surrounding the takedown, we plot the successful aggregate daily lookup volume to each of the previously identified sets. To quantify the gains in takedown effectiveness, we calculate the *takedown improvement ratio* as defined by Equation (1):

$$TIR(D_1, D_2) = \frac{MDLV(D_1)}{MDLV(D_2)}. \quad (1)$$

Where D_1 and D_2 are two domain name sets and $MDLV$ is a function on domain name sets that computes the median daily successful lookup volume. We use the median, rather than the mean, since we are interested in preserving long-term lookup volume trends, which are not captured by outliers. If $TIR(D^t_m, D^t_S) > 1$, this means the subset of D_e of malware-related domain names D^t_m had a stronger lookup volume and accounts for domain names missed by the takedown domains D^t_S. Conversely, if the $TIR \leq 1$, the takedown deactivated related malware domains already and was successful. We also identify malware backup behaviors.

Estimating risk: To provide a different perspective, we also quantify the potential risk of *collateral damage*, or the negative effect of mistakenly taking down benign domains. Ideally, we would represent this by the number of distinct clients that would be denied access to benign services, however, we can once again turn to the lookup volumes to proxy for this.

If we assume all infected botnet hosts behave identically, the aggregate lookup volume on a given day is proportional to the number of infected clients. At most, a single lookup corresponds to a distinct client reaching that domain, however, due to DNS caching effects, differences in malware variant and human behaviors, and network address translation (NAT), this is likely an overestimation of the actual client population. We assume that these behaviors are consistent with respect to queries towards a given botnet.

We quantify the potential risk of collateral damage for a takedown as the difference in the median lookup volume between an enumerated set and the initial seed domain set as defined by Equation (2):

$$Risk(D_1, D_2) = MDLV(D_1) - MDLV(D_2). \quad (2)$$

Using similar notation as seen in Equation (1). Intuitively, the difference between these two quantities is proportional to the number of individuals that would be inconvenienced by this takedown if *all* the domains in D_1 that are not in D_2 *are not malicious*. This provides an upper bound on the potential risk involved. The "nuclear option" of taking down all the domains in D_e, or sinkholing all domains that resolve to hosts known to provide C&C for a botnet, is the only way to ensure the C&C communication line is severed, however, this should be weighed against the potential risks.

[2] http://www.kloth.net/services/nslookup.php

[3] These are September 11th, 2012; June 5th, 2013 and June 30th, 2014 for the 3322.org, Citadel and No-IP takedowns, respectively.

An analyst wishing to perform a takedown can use the risk values to weigh whether to employ the “nuclear” option or the more reserved options as described in Section 3.4. In future work, we hope to improve the risk measure in two ways. First, we can correlate the risk value with the identified true and false positive rates during a real, or simulated, takedown. Furthermore, we wish to more accurately estimate the true population of visitors to infrastructure, malicious or otherwise. This can further help analysts by allowing them to weigh the likelihood of maliciousness against the population that would be affected by a takedown.

5.2 Improved Postmortem Analysis

In our previous takedown study [4], we were limited by using daily lookup volumes as a proxy for victims communicating with active malicious infrastructure. Clearly this is a limitation, but was due to the nature of the datasets we had access to. Fortunately, new data has become available to us and we can now compute the population of distinct clients that query a given domain name or resource record (i.e., domain and IP tuple) on a specific date. We use this new value to compute improved postmortem takedown measures and compare them to the previously defined ones. We now define $TIR+$ and $Risk+$, the improved analogues for TIR and $Risk$, respectively, which use population counts rather than lookup volume counts.

$TIR+$: For a period of 14 days surrounding the takedown, we plot the successful aggregate daily population count to each of the previously identified sets. To quantify the gains in takedown effectiveness, we calculate the *improved takedown improvement ratio* as defined by Equation (3):

$$TIR{+}(D_1, D_2) = \frac{MDP(D_1)}{MDP(D_2)}. \tag{3}$$

Where D_1 and D_2 are two domain name sets and MDP is a function on domain name sets that computes the median daily client population. We use the median, rather than the mean, since we are interested in preserving long-term population trends, which are not captured by outliers. If $TIR(D^t_m, D^t_S) > 1$, this means the subset of D_e of malware-related domain names D^t_m had a stronger lookup volume and accounts for domain names missed by the takedown domains D^t_S. Conversely, if the $TIR \leq 1$, the takedown deactivated related malware domains already and was successful.

$Risk+$: We quantify the potential risk of collateral damage for a takedown as the difference in the median client population between an enumerated set and the initial seed domain set as defined by Equation (4):

$$Risk{+}(D_1, D_2) = MDP(D_1) - MDP(D_2). \tag{4}$$

Using similar notation as seen in Equation (3). Intuitively, the difference between these two quantities is proportional to the number of individuals that would be inconvenienced by this takedown if *all* the domains in D_1 that are not in D_2 *are not malicious*. This provides an upper bound on the potential risk involved. The “nuclear option” of taking down all the domains in D_e, or sinkholing all domains that resolve to hosts known to provide C&C for a botnet, is the only way to ensure the C&C communication line is severed, however, this should be weighed against the potential risks.

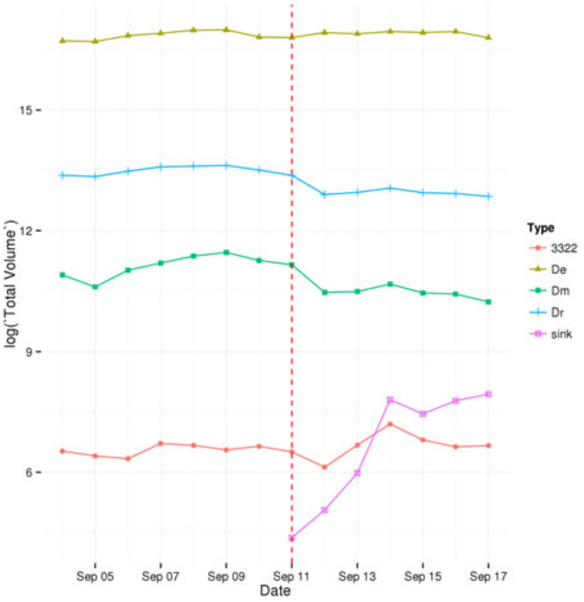

Fig. 5. 3322.org aggregate daily lookup volume (log-scale).

$TIR+$ and $Risk+$ can only be computed for takedowns that occur after November 14, 2012 due to data availability.

For each of the following takedown postmortem analysis, the dashed red line on each plot indicates the date the takedown was performed according to the court proceedings. Each line plot represents either the aggregate daily lookup volume or aggregate daily population count to a subset of domains that are either directed to a sinkhole or contained within the enumerated infrastructure sets generated by *rza*. In all cases the D_e lookup volume represents an upper bound of malicious lookups.

5.3 3322.org

The 3322.org takedown represents an extreme case where *rza* would have improved a takedown’s effectiveness. This takedown was accomplished by transferring the entire 3322.org Name Server’s (NS) authority to Microsoft and domains deemed malicious resolved to a set of known sinkhole IP addresses. The daily volume plot for 3322.org is shown in Fig. 5. Unlike the Citadel takedown, domains were sunk on the day of the takedown and were limited to *.3322.org domain names. Unfortunately, this only accounted for a fraction of the lookups to domains with known malware associations, D_m, and domains with low reputation, D_r that resolved to hosts known to support malicious activity. We notice a drop in lookups to D_m and D_r when the takedown is performed, showing that most of the domains targeted by the takedown were likely malicious, however, the lookups to remaining infrastructure identified by *rza* are still frequent. All enumerated sets have TIR values greater than one. This agreement suggests that malicious domains were almost certainly missed during the 3322.org takedown effort. Of the 10,135 malware

TABLE 4
TIR and $Risk$ Values for 3322.org Takedown

Sets	TIR value	Risk
D_m, D_{mssink}	13.821	409,593.5
D_r, D_{mssink}	18.956	573,627.5
D_e, D_{mssink}	654.940	20,890,774

samples we analyzed, none of them had a P2P- or DGA-based contingency plan.

This case shows the importance of using multiple sources to determine related malicious infrastructure before performing a takedown. Simply identifying domains with known malware associations offers a substantial improvement on the effectiveness of the takedown. Further, the similarity between the D_m and D_r trends shows most of the domains overlap between the two, which only further bolsters the likelihood that they are indeed malicious. To make matters worse, all the domains that were not sinkholed were given enterprise-level domain name resolution services, despite the high probability they were involved in malicious activities. The computed TIR values for the 3322.org takedown are shown in Table 4. Unlike the previous two postmortems, *rza* identified numerous additional malicious domains that were left undisturbed by the takedown on 3322.org.

For the D_e and D_m sets, we have precision and recall of 0.06/0.95 and 0.38/0.03, respectively. These results further reinforce the need to include domain reputation as a measure in *rza*. Simply relying on passive DNS (for D_e) and malware associations (for D_m) overestimate and underestimate the malicious domain names, respectively.

5.4 Citadel

The Citadel takedown is an interesting case, where damage from "friendly fire" occurred and likely malicious domains remained active. First, we see in Fig. 6 that another sinkhole was present for some of the Citadel infrastructure (othersink), which has been confirmed by the sinkhole owners [20]. Confusingly, however, we do not see a decrease during the takedown, but eventually an increase in lookups. This is likely due to an increase in lookups as the sinkholes do not reply with proper commands, causing bots to reissue lookups. Next, we notice a drop in successful lookups for the Citadel domains claimed to have been taken down by Microsoft, but note there are still *many* successful lookups to non-sinkholed Citadel domains. Furthermore, we see very little impact to the D_m, D_r, and D_e sets after the takedown, indicating many malicious domains not targeted in the takedown are still successfully resolving. In Fig. 7 we show the aggregate active clients for each subset. Importantly, we see that they are strictly less than the lookup volume and all roughly follow the same trajectory, indicating that the daily lookup volume is an effective proxy for the unique client counts. We also see that the domain names claimed to have been taken down (citadel) are not all actually redirected to the sinkhole (citadel-mssink).

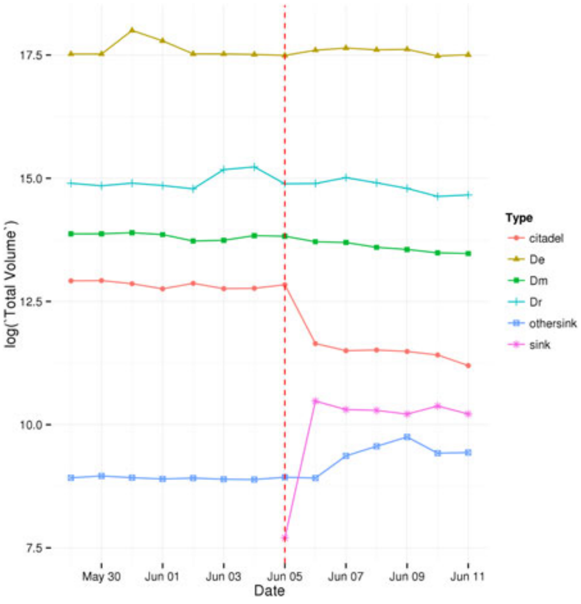

Fig. 6. Citadel aggregate daily lookup volume (log-scale).

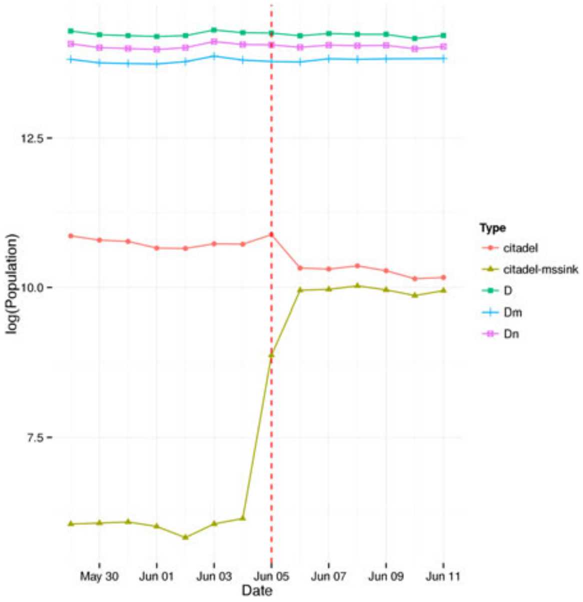

Fig. 7. Citadel aggregate daily active clients (log-scale).

TIR and $Risk$ values are shown in Table 5 and $TIR+$ and $Risk+$ are shown in Table 6. For D_m in both we see values similar to D_m in the 3322.org takedown case, but the TIR and $Risk$ values for D_r and D_e for Citadel are much higher. While we see a similar sharp increase for 3322.org, note the $TIR+$ and $Risk+$ values for Citadel increase much less rapidly. This shows that for small sets, TIR and $Risk$ values reasonably reflect reality, i.e., lookup volume remains a reasonable proxy. However, in large sets using the exact population as we do with $TIR+$ and $Risk+$ offers much more reasonable assessments for takedown improvement and risk evaluation.

TABLE 5
TIR and $Risk$ Values for Citadel Takedown

Sets	TIR value	Risk
D_m, D_{mssink}	25.821	892,349
D_r, D_{mssink}	81.703	2,900,934
D_e, D_{mssink}	1,134.503	40,744,905

TABLE 6
$TIR+$ and $Risk+$ Values for Citadel Takedown

Sets	TIR+ value	Risk+
D_m, D_{mssink}	23.354	948,610
D_r, D_{mssink}	29.482	1,208,680
D_e, D_{mssink}	35.780	1,475,908

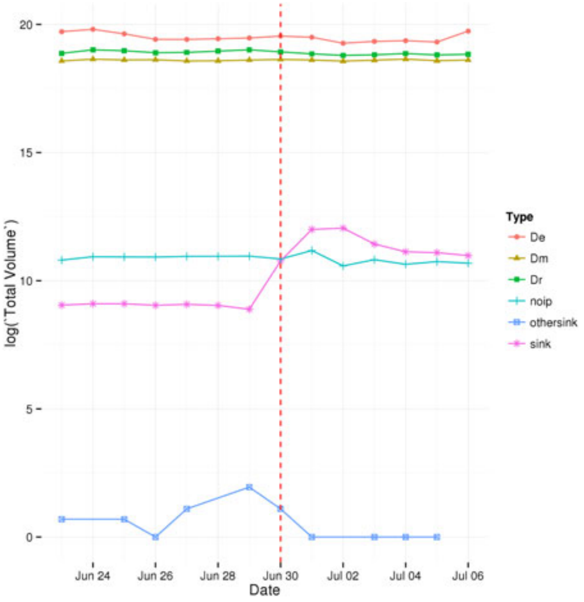

Fig. 8. No-IP aggregate daily lookup volume (log-scale).

This suggests that for small, malware based sets using the simple TIR value is sufficient, but with more complicatedly derived sets using the actual population is much more powerful. It should be noted that our population is biased (and smaller than the volume set), but still fairly representative of North America, which is the population the takedowns were meant to protect.

5.5 No-IP

The No-IP takedown generated a lot of press and was heavily condemned by the No-IP DNS provider [21]. Not only did Microsoft not contact No-IP about the malicious domains in question, but the takedown was performed in a similar manner to the aforementioned 3322.org takedown. Instead of taking down the No-IP nameservers, the zones used by the dynamic DNS names were taken over by Microsoft. Malicious domains were routed to the sinkhole, while the others were supposed to be routed faithfully. As in previous cases, there were many malicious domain names that were not sinkholed, as evidenced by the volume and client population counts in Figs. 8 and 9. One interesting point of note is the population counts are actually *higher* than the volumes. While this may seem counter-intuitive, it is an explainable artifact of our collection process. Volume counts are usually done above the recursive and the effects of caching cannot be easily measured. Population counts are done below the recursive and can register a lookup even if it is cache at the local recursive. This means that lookups to No-IP domains were more heavily concentrated and likely had longer TTLs than those in the Citadel case. This makes sense because No-IP is very popular even for benign hosting, which tend to have longer TTLs than malicious domains.

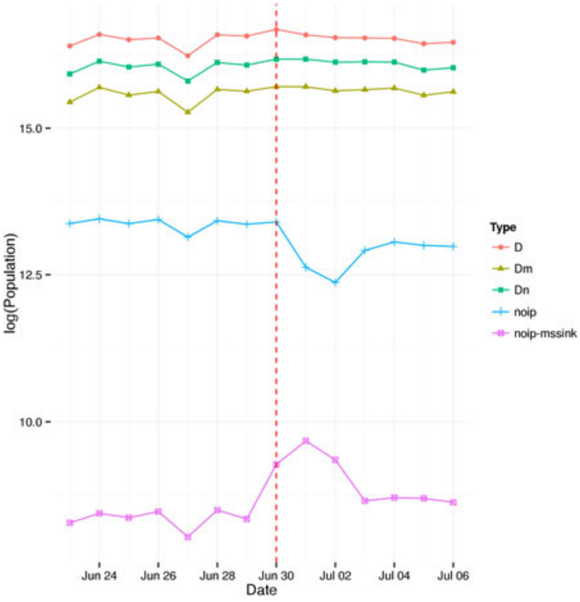

Fig. 9. No-IP aggregate daily active clients (log-scale).

TABLE 7
TIR and $Risk$ Values for No-IP Takedown

Sets	TIR value	Risk
D_m, D_{mssink}	2,537.85	120,847,924
D_r, D_{mssink}	3,379.05	160,920,164
D_e, D_{mssink}	5,994.491	285,511,928
D_m, D_{noip}	2,176.494	120,840,015
D_r, D_{noip}	2,897.919	160,912,255
D_e, D_{noip}	4,140.956	285,504,019
$D_{\mathrm{noip}}, D_{\mathrm{mssink}}$	1.166	7,909

TABLE 8
$TIR+$ and $Risk+$ Values for No-IP Takedown

Sets	TIR+ value	Risk+
D_m, D_{mssink}	1,146.945	6,188,102
D_r, D_{mssink}	1,305.379	7,043,646
D_e, D_{mssink}	3,021.707	16,311,820
D_m, D_{noip}	10.824	5,621,290
D_r, D_{noip}	12.319	6,476,834
D_e, D_{noip}	28.516	15,745,008
$D_{\mathrm{noip}}, D_{\mathrm{mssink}}$	105.965	566,812

TIR and $Risk$ values are shown in Table 7 and $TIR+$ and $Risk+$ are shown in Table 8. The results are presented differently because of the nature of the takedown. These values are computing with respect to D_{mssink} as well as D_{noip} which are No-IP domains sinkholed and No-IP domains affected (in any way) by the takedown. The first thing to notice is the $TIR/Risk$ values are *much* larger than the $TIR + /Risk+$ values. In the other takedowns, we were restricted to specific fully qualified domain names, but in this case all

TABLE 9
Farsight-Computed TIR Values

Takedown	D_e	D_m	D_r
Zeus	4.843	0.000	1.014
#1	1.108	1.012	1.082
#2	0.969	0.969	0.459
#3	0.787	0.787	0.718
#4	0.680	0.680	0.613
#5	1.944	1.451	1.122

domains under the zones were taken down. In these cases, volume is probably too relaxed of a proxy for population and exact population should be used if possible. Finally, we also include the risk in taking down all of No-IP rather than just the domains Microsoft meant to take down, i.e., those that appear in "mssink." In this case, we see at least 566 thousand people were negatively impacted by the takedown actions.

5.6 RZA with Farsight pDNS Data

We replicated part of our evaluation using only Farsight data. Specifically, we generated the D_e, D_m, and D_r domain sets and computed the respective TIR values of Zeus and five of the current botnets with the most domains we tracked in our paper. D_i sets were excluded due to time limitations. Our results from the Farsight dataset are shown in Table 9 and are largely consistent and show that the process employed by RZA can be done with other sources of pDNS data. The important detail to glean is that the process *rza* uses is independent of our private dataset and can be performed using public sources of passive DNS data. Due to regional variations, the TIR values are unlikely to be identical between the two datasets; however, the process and generated sets are the important factors.

5.7 Running Time

To show the benefit of *rza* even against clever attackers, we show the expected running time from start to finish in Equation (5). We show that the running time is short enough that pace can be kept with evading attackers. We show the time it took to perform the postmortem takedowns described earlier in this section. In each case, the running time is sufficiently short that attackers will not have substantial time to continue evading a takedown, which will likely be successful in the long run.

The total running time for performing a full takedown postmortem or a takedown recommendation analysis is T as defined by:

$$T = T_{IE} + T_{MI}, \tag{5}$$

where T_{MI} is the time to perform malware interrogation on the botnet and T_{IE} is the time it takes to perform infrastructure enumeration. Each sub-equation is defined below:

$$T_{IE} = 2\alpha P. \tag{6}$$

Equation (6) represents the time to fully enumerate the infrastructure of a botnet's criminal network. P represents the time to perform a full pass of the passive DNS database to identify related infrastructure and α represents the number passes that must be performed until the infrastructure converges. Two passes over the passive DNS database must be made in order to enumerate infrastructure; once to identify related historic IP addresses and a second time to identify related historic domain names. Recall Fig. 3 that shows the process to perform a takedown recommendation. If additional infrastructure is identified through malware interrogation, an additional pass over the passive DNS is needed to continue to expand our knowledge of the infrastructure.

Performing a pass over the passive DNS database can be done with either bulk or individual requests. In a bulk request, the running time is $\mathcal{O}(1)$ with respect to the number of domains but a single bulk request runs on the order of minutes, while with individual requests the running time is $\mathcal{O}(n)$ in the number of domain names that must be queried but a single request finishes in seconds. For ease of presentation, we only consider bulk requests but in an operational environment individual requests would be made where time would be saved. As we show, however, most infrastructure can be enumerated in a single pass

$$T_{MI} = \frac{m \times 7t}{\mathcal{M}}. \tag{7}$$

Equation (7) is the time to fully interrogate the malware of the botnet to extract any additional network endpoints that must be disabled, as well as any potential backup C&C behavior included in the botnet's infrastructure. This is limited by the number of machines we have available for performing malware analysis, $\mathcal{M}$, the duration of the control malware analysis run, t, and the number of malware samples related to the botnet's infrastructure m. Recall that each malware sample must be run five times, two of which for duration $2t$, in order to understand the malware's backup plan (see Section 4).

Bound and free variables: Throughout the evaluation of the systems, the following variables were bound:

- $M = 512$ virtual machines for malware interrogation.
- $t = 3$ minutes for the control malware execution timeout. Note malware may terminate earlier of its own accord, but malware will run for at most three minutes.
- $P = 21$ minutes to make a single, full pass over the passive DNS database.

This leaves two free variables to compute for each postmortem or takedown recommendation: α and m the number of runs of the system until convergence is achieved and the number of malware samples to be analyzed, respectively.

The timing information for the takedown postmortems is summarized in Table 10. In short, we see that full analyses can be done on the order of hours and can likely keep pace with agile botnet infrastructure.

6 Discussion

In this section we discuss the limitations of *rza*, as well as policy implications of our research and how botnet takedowns should be performed at a policy, rather than technical, level.

TABLE 10
Timings for End-to-End Run of Postmortem Studies

Takedown	α	m	T (in hours)
3322.org	1	12,745	9.41
Citadel	1	4,861	4.02
No-IP	1	18,913	13.63

6.1 Limitations

rza is a useful tool for understanding and assisting botnet takedowns, but it is not a panacea. In particular, *rza* can only assist in a limited sense against non-DNS-based botnet C&Cs or DNS C&Cs with non-overlapping infrastructure.

rza is focused on understanding and improving takedowns of traditional DNS-based C&C infrastructure so it is inherently limited in its ability to do so for non-DNS botnets and C&Cs that tunnel traffic over the DNS [22]. However, some of the techniques could still be used in other cases. First, we have shown that *rza* can help identify advanced C&Cs, such as DGA- or P2P-based protocols. Second, the techniques described—particularly those for malware interrogation—could be used to assist in analyzing botnets with direct IP C&C servers. *rza* can even help with advanced hybrid peer-to-peer botnets [23] by detecting the presence of P2P activity, and further help if operators choose to augment such infrastructure with more agility using domain names.

Furthermore, DNS botnet infrastructure could be organized such that it would evade identification during our infrastructure enumeration process. For example, if the C&C domain names never share IP infrastructure, the enumeration procedure described in Section 3.2 would fail to identify the additional network assets to disable. That being said, this defeats the purpose of using the DNS for agility and is unlikely to occur in the extreme case. This is why *rza* also performs malware analysis, allowing it to identify cases where passive analysis is insufficient.

6.2 Policy

Takedowns are currently performed in an ad-hoc manner with little oversight, which makes it difficult for the security community at large to assist by contributing intelligence. Furthermore, there is no standard policy for enacting a takedown at the DNS-level forcing companies to coordinate with multiple registrars, pay for expensive court proceedings, or both to disable botnets. Existing measures for handling domain name issues exist, however, in the form of handling trademark disputes.

Our postmortem studies illustrated several drawbacks to the current ad-hoc manner in which takedowns are performed, namely: a lack of coordination, little to no oversight, and an environment that discourages collaboration. Without an effective form of coordination, we will continue to see instances in where two or more security companies, with good intentions, will step on each others toes as we saw in the Citadel and No-IP takedown cases. We also saw oversight issues in the Citadel takedown where domains were clearly being sinkholed *before* the date presented in the court order. Yet another, but more subtle, oversight issue deals with the method of instigating these takedowns: court orders. Each of the court orders for the presented takedowns were filed under seal, meaning they are not open to the public and require either the claimant to release the record under their discretion, or other legal action to unseal the record.

Even more worrisome is language explicitly allowing further unverified action. In the 3322.org takedown temporary restraining order it was specified that "the authoritative name server ... [is] to respond to requests for the IP addresses of the sub-domains of 3322.org may respond to requests for the IP address of any domain listed in Appendix A[4] *or later determined to be associated with malware activity...*" [2] (emphasis ours). While an authoritative name server takeover technically grants this ability, if the purpose of the court order is to prevent collateral damage or unlawful takedown this clause effectively negates any future protection. It also suggests that the full scope of the threat was not clear at the time of the takedown by specifically permitting further cleanup actions.

Trademark and intellectual property interests were involved very early on in during the formation of the Internet Corporation for Assigned Names and Numbers (ICANN), which is responsible for coordinating, among other critical Internet infrastructure, the DNS. Through the World Intellectual Property Organization (WIPO), trademark interests were arguing for procedures to protect trademarks in the DNS as early as December 1998 [24], and successfully forced ICANN to require a dispute resolution procedure dubbed the "Uniform Dispute Resolution Policy" or URDP.

URDP is an ICANN policy that specifies independent arbitrators to oversee the process of dispute resolution. These "[i]ndependent arbitrators make a decision quickly and (relative to courts) inexpensively" [24] and are built in to the accreditation contracts to registrars. The UDRP [25] requires three conditions to be met to file a complaint:

i. Your domain name is identical or confusingly similar to a trademark or service mark in which the complainant has rights; and
ii. You have no rights or legitimate interests in respect of the domain name; and
iii. Your domain name has been registered and is being used in bad faith.

In its first year, UDRP successfully "handled over 2,500 cases involving nearly 4,000 names" [24] and has expanded since. In fact, ICANN is introducing The Uniform Rapid Suspension System (URS) [26] as a more expedited form of the URDP and is requiring new generic top-level domains (gTLDs) to follow URS in their contracts.

We suggest a similar procedure ought to be available to provide the security community a point of coordination and a formal process to follow when performing takedowns. It would reduce exorbitant fees paid to courts, would likely be faster, and would mandate oversight from arbitrators. The procedure could be applied to future TLDs as a test, much like URS. Automated systems like *rza* could serve an invaluable place in this process to reduce the burden on human operators and further expedite the takedown process.

[4] Appendix A of the cited restraining order.

7 Conclusion

We presented *rza*, a takedown analysis system that performs postmortem analyses of past takedowns. We have shown that *rza* would be useful in helping to both expedite the takedown process, as well as ensuring future takedowns are more complete. We presented a more accurate estimation of the takedown's improvement and risk, $TIR+$ and $Risk+$, that also illustrate our earlier intuition of using volume as a proxy for population was well founded. Finally, we perform postmortem analyses of the Citadel and No-IP takedowns. Both were unsuccessful and help show the continuing importance of having a tool like *rza* to understand historic takedowns.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments and insightful questions. This material is based in part upon work supported by US Department of Commerce (DOC) under grant no. 2106DEK. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DOC.

References

[1] C. Rossow, D. Andriesse, and T. Werner, "P2PWNED: Modeling and evaluating the resilience of peer-to-peer botnets," in *Proc. 34th IEEE Symp. Security Privacy*, 2013, pp. 97–111.

[2] Microsoft Corporation, "Microsoft Corporation v. Peng Yong et al.," virginia Eastern District Court, 2012.

[3] Conficker Working Group. (2011) Conficker Working Group: Lessons Learned [Online]. Available: http://www.confickerworkinggroup.org/wiki/uploads/Conficker_Working_Group_Lessons_Learned_17_June_2010_final.pdf

[4] Y. Nadji, M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee, "Beheading hydras: Performing effective botnet takedowns," in *Proc. 20th ACM Conf. Comput. Commun. Security*, 2013, pp. 121–132.

[5] B. Krebs. (2010). Mariposa Botnet Authors May Avoid Jail Time [Online]. Available: http://krebsonsecurity.com/2010/03/mariposa-botnet-authors-may-avoid-jail-time/

[6] U.S. Attorney's Office - Southern District of New York. (2011). Manhattan U.S. Attorney Charges Seven Individuals for Engineering Sophisticated Internet Fraud Scheme [Online]. Available: http://www.fbi.gov/newyork/press-releases/2011/manhattan-u.s.-attorney-charges-seven-individuals-for-engineering-sophisticated-internet-fraud-scheme-that-infected-millions-of-computers-worldwide-and-manipulated-internet-adverti sing-business

[7] B. Krebs. (2008). Major Source of Online Scams and Spams Knocked Offline [Online]. Available: http://voices.washingtonpost.com/securityfix/2008/11/major_source_of_online_scams_a.html

[8] R. McMillan. (2010). After takedown, botnet-linked ISP Troyak resurfaces [Online]. Available: http://www.computerworld.com/s/article/9169118/After_takedown_botnet_linked_ISP_Troyak_resurfaces

[9] B. Krebs. (2008). Spam Volumes Drop by Two-Thirds After Firm Goes Offline [Online]. Available: http://voices.washingtonpost.com/securityfix/2008/11/spam_volumes_drop_by_23_after.html

[10] M. Harris. (2009) Spammers recovering from McColo shutdown [Online]. Available: http://www.techradar.com/news/internet/spammers-recovering-from-mccolo- shutdown-591118

[11] Farsight Security, Inc. (2013). SIE/Farsight Security's DNSDB [Online]. Available: https://www.dnsdb.info/

[12] Y. Nadji, M. Antonakakis, R. Perdisci, and W. Lee, "Understanding the prevalence and use of alternative plans in malware with network games," in *Proc. 27th Annu. Comput. Security Appl. Conf.*, 2011, pp. 1–10.

[13] VirusTotal. VirusTotal Intelligence [Online]. Available: https://www.virustotal.com/en/documentation/private-api/, 2013.

[14] M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee, "Building a dynamic reputation system for DNS," in *Proc. 19th USENIX Conf. Security Symp.*, 2010, p. 18.

[15] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding malicious domains using passive DNS analysis," in *Proc. Netw. Distrib. Syst. Security Symp.*, 2011.

[16] Alexa. (2011, Mar.). Top sites (Retrieved) [Online]. Available: http://www.alexa.com/topsites

[17] dnswl. (2011, Mar.). DNS whitelist - protect against false positives. (Retrieved) [Online]. Available: http://www.dnswl.org

[18] Microsoft Corporation. (2013). Microsoft Corporation v. John Does 1-82. United States District Court for the Western District of North Carolina Charlotte Division [Online]. Available: http://botnetlegalnotice.com/citadel/

[19] Microsoft Corporatio. (2014). Microsoft Corporation v. Naser Al Mutairi, an individual; Mohamed Benabdellah, an individual; Vitalwerks Internet Solutions, LLC, d/b/a NO-IP.com; and Does 1-500. United States District Court District of Nevada [Online]. Available: http://www.noticeoflawsuit.com/

[20] Abuse.ch. (2013). Collateral Damage: Microsoft Hits Security Researchers along with Citadel [Online]. Available: https://www.abuse.ch/?p=5362

[21] N. Goguen. (2014). No-IP's Formal Statement on Microsoft Takedown [Online]. Available: http://www.noip.com/blog/2014/06/30/ips-formal-statement-microsoft-take down/

[22] K. Xu, P. Butler, S. Saha, and D. Yao, "DNS for massive-scale command and control," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 3, pp. 143–153, May/Jun. 2013.

[23] P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer Botnet," *IEEE Trans. Dependable Secure Comput.*, vol. 7, no. 2, pp. 113–127, Apr.–Jun. 2010.

[24] M. Mueller, *Ruling the Root*. Cambridge, MA, USA: MIT Press, 2004.

[25] Internet Corporation for Assigned Names and Numbers, "Uniform domain name dispute resolution policy," Tech. Rep. udrp-2012-02-25-en, 1999.

[26] Internet Corporation for Assigned Names and Numbers, "Uniform rapid suspension system," Tech. Rep. urs-2013-10-31-en, 2012.

Yacin Nadji received the PhD degree in computer science from the Georgia Institute of Technology as a member of the Georgia Tech Information Security Center and was co-advised by Drs. Manos Antonakakis and Wenke Lee. He is a postdoctoral researcher at the Georgia Institute of Technology under the supervision of Dr. Manos Antonakakis. He was a research contractor at Damballa, Inc.

Roberto Perdisci received the PhD degree from the University of Cagliari, Italy, with the Pattern Recognition and Applications Group. He is an associate professor in the Computer Science Department, University of Georgia, an adjunct assistant professor in the Georgia Tech School of Computer Science, and a faculty member of the UGA Institute for Artificial Intelligence. Before joining UGA, he was a postdoctoral fellow at the College of Computing, Georgia Institute of Technology, working under the supervision of Prof. Wenke Lee. He was also a principal scientist at Damballa, Inc.; and prior to joining Damballa, he was a research scholar at the Georgia Tech Information Security Center.

Manos Antonakakis is an assistant professor in the School of Electrical and Computer Engineering (ECE), and adjunct faculty in the College of Computing (CoC), at the Georgia Institute of Technology. Before joining the ECE faculty, he was the chief scientist at Damballa, where he was responsible for advanced research projects, university collaborations, and technology transfer efforts. He currently serves as the co-chair in the Academic Committee for the Messaging Anti-Abuse Working Group (MAAWG).