Beheading Hydras: Performing Effective Botnet Takedowns

Yacin Nadji
College of Computing
Georgia Institute of Technology
Atlanta, GA
yacin.nadji@cc.gatech.edu

Manos Antonakakis
Damballa, Inc.
Atlanta, GA
manos@damballa.com

Roberto Perdisci
Department of Computer Science
University of Georgia
Athens, GA
perdisci@cs.uga.edu

David Dagon
College of Computing
Georgia Institute of Technology
Atlanta, GA
dagon@sudo.sh

Wenke Lee
College of Computing
Georgia Institute of Technology
Atlanta, GA
wenke@cc.gatech.edu

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 propose a takedown analysis and recommendation system, called *rza*, that allows researchers to perform two tasks: 1) a postmortem analysis of past botnet takedowns, and 2) provide recommendations on how to successfully execute future botnet takedowns. As part of our system evaluation, we perform a postmortem analysis of the recent Kelihos, Zeus and 3322.org takedowns. We show that while some of these takedowns were effective, others did not appear to have a significant long-term impact on the targeted botnet. In addition to the postmortem analyses, we provide takedown recommendation metrics for 45 currently active botnets, where we find that 42 of them can likely be disabled entirely by using a DNS-based takedown strategy only.

CCS'13, November 04 - 08 2013, Berlin, Germany

ACM 978-1-4503-2477-9/13/11
http://dx.doi.org/10.1145/2508859.2516749.

Categories and Subject Descriptors

K.6.m [**Management of Computing and Information Systems**]: Security; K.5.m [**Legal Aspects of Computing**]: Contracts

General Terms

Botnets

Keywords

botnet takedowns; takedown analysis; takedown policy

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 [18], 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 [13] 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 and recommendation system*, which we call *rza*. Our main goals are: (1) 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; (2) to build a *takedown recommendation engine* that aims to automatically enumerate a botnet's C&C infrastructure, and suggest appropriate domain sets to disable to maximize the chance of success. 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 (P2P). 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 [18]. 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. Worst yet, in 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 summary, we make the following contributions:

- We propose *rza*, a takedown analysis and recommendation system that allows us to measure and reason about the success of past and future takedown efforts. To the best of our knowledge, we are the first to propose such a botnet takedown analysis system.

- We apply *rza* to analyze three recent botnet takedown operations. We show that while some takedowns were effective, others did not appear to actually disrupt the entire targeted botnet.

- We use *rza*'s recommendation engine to analyze 45 live botnets, and discuss in which cases a DNS-based takedown operation is likely to succeed and what steps would be necessary to accomplish the takedown in practice. Of these, 42 could be straightforwardly eliminated using only DNS sinkholing.

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. Section 4 presents our postmortem experiments and analyses of three recent, high-profile takedown attempts. Section 5 presents the output of *rza* when applied to 45 recently identified, distinct botnet C&C infrastructures. In Appendix A we discuss non-technical difficulties associated with performing takedowns that would make takedowns more complete if alleviated.

2. BACKGROUND

In this section, we first provide an historical explanation of some past takedowns and explain why takedowns 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 [10], underscoring the fact that barriers to successful takedowns are not only technical ones.

The DNSChanger [19] "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 [8], AS Troyak [11], 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 [5].

2.2 Datasets

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

Passive DNS.

A passive DNS (pDNS) 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. 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. 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.

To enable our takedown analysis we define the following functions over the pDNS database:

- 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.

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 ISC-SIE passive DNS database [4], 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 as well as the output from a commercial malware feed.

3. RZA SYSTEM

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

3.1 Overview

Figure 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 (Figure 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 (Figure 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

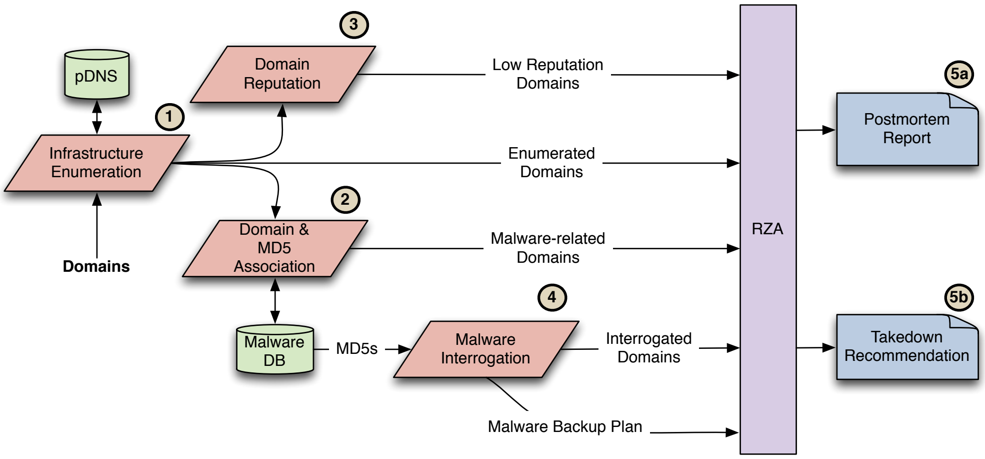

Figure 1: Overview of *rza*.

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 [17] 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 (RHIP) of D_S are retrieved and known sinkhole, parking, and private IP addresses are removed. The related-historic domain names (RHDN) 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 but we empirically chose seven days based on the trend in domain name activations and deactivations to the domain names contained in D_e as described in Appendix C.

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

```
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 $D_e$
```

Algorithm 1: Infrastructure enumeration procedure.

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 in Section 3.3, 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 Malware Interrogation

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 [17] 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.

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. Our heuristics successfully classified 97% of the samples' contingency plans correctly.

3.4 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 (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 [20]. To determine if a domain name has low reputation, we use a system similar in spirit to [1, 2] 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).

Figure 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.

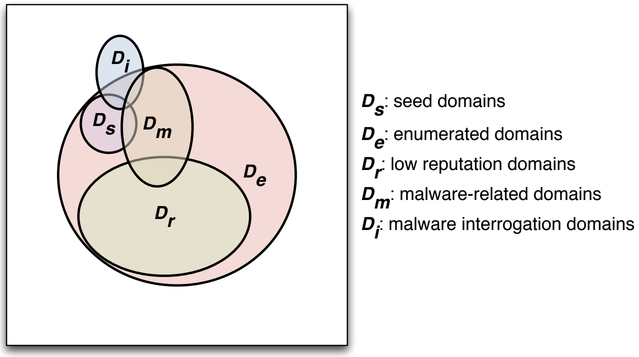
D_i
D_s
D_m
D_e
D_r
D_s: seed domains
D_e: enumerated domains
D_r: low reputation domains
D_m: malware-related domains
D_i: malware interrogation domains

Figure 2: Venn diagram of identified infrastructure sets.

3.5 Takedown Recommendation Engine

Using the four aforementioned techniques, we can run our takedown protocol as shown by the decision tree in Figure 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 [18] after disabling the main C&C infrastructure, respectively.

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 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 $d \in D \land d \in D_r \cup D_i$ 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 $|D_r \cup D_i|$.

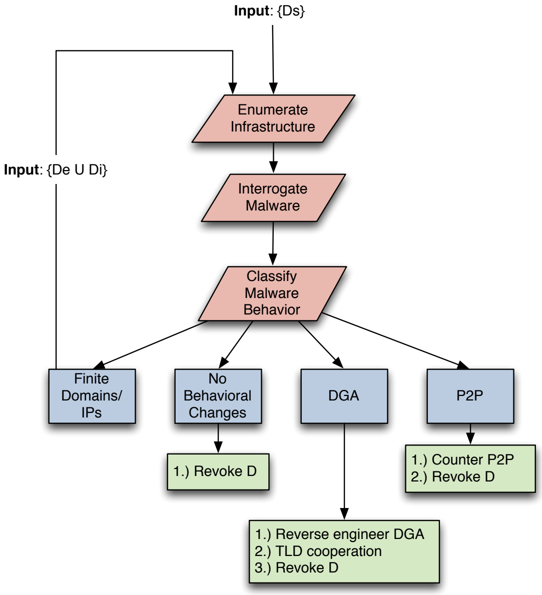

Figure 3: Takedown recommendation engine shown as a decision tree. D in this case represents either $D_r \cup D_i$, which only targets C&C domains that are very likely to be malicious or $D_e \cup D_i$, or the "nuclear" option that should only be used when the threat of the botnet outweighs potential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

3.6 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 the Internet System Consortium/Security Information Exchange's (ISC/SIE) passive DNS database [4][1]. 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.

[1] Since the additional experiments were run, the ISC/SIE's pDNS database has been acquired by a private company, Farsight Security Inc. The website suggests database access will continue to be made available to qualified security researchers and practitioners.

Using ISC/SIE'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 SIE dataset are presented in Appendix B.

4. 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.

4.1 Postmortem Analysis

For our postmortem analysis, we chose to study the takedowns of Kelihos [12] (aka Operation b79), a Zeus botnet instance [14] (aka Operation b71), and the 3322.org NS takedown that targeted the Nitol botnet [13] (aka Operation b70). 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_S^t, we generate the enumerated infrastructure sets D_e^t, D_m^t, D_r^t and D_i^t using *rza*. D_e^t 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[2]. 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.

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

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 $TIR(D_m^t, D_S^t) > 1$, this means the subset of D_e of malware-related domain names D_m^t had a stronger lookup volume and accounts for domain names missed by the takedown domains D_S^t. 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.

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

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.

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 3.5. 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.

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 the aggregate daily lookup volume 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.

4.2 Kelihos

The Kelihos botnet was a spam botnet that sent approx-

[2] These are September 11th, 2012; March 25th, 2012 and September 26th, 2011 for the 3322.org, Zeus and Kelihos takedowns, respectively.

Sets	TIR value	Risk
D_m, D_S	0.913	-399.5
D_r, D_S	5.690	21,555
D_i, D_S	0.022	-4,492.5
D_e, D_S	10.230	42,415.5

Table 1: TIR and $Risk$ values for Kelihos takedown. These values represent the improvement to a takedown based on *rza*'s output and the potential risk of collateral damage, respectively.

imately four billion spam messages[15] a day in its first iteration and was targeted for takedown in late 2011. We show the daily volumes for the sets D_S, D_e and D_m for Kelihos in Figure 4. The day Kelihos was taken down, we see lookups to the seed domains completely stop, showing that these domains were effectively remediated. The court order did not specify sinkholes to be used, which explains why the domains simply cease to resolve. The set of malware-related domains, D_m, and interrogated domains, D_i, also cut off sharply at this point, with a handful of successful resolutions occurring for D_m a few days after the takedown date and ceasing to resolve afterwards. This suggests the initial takedown missed some domains, but these were quickly remediated as well. D_r has a spike similar to D_e, and upon further investigation the spike was revealed to be a malicious domain that resolved into Kelihos' infrastructure but could not be confirmed to be a Kelihos C&C. This domain stopped resolving after the peak date (September 24$^{\text{th}}$).

The computed TIR values are shown in Table 1. Much like the daily volumes figure, the TIR values suggest this takedown was successful. We see large TIR values for D_e and D_r, which indicate additional malicious domains were left unperturbed that resolved into Kelihos' hosting infrastructure. The similar trend between D_e and D_r suggests that many of the extended infrastructure domains are in fact malicious and could have been removed during the Kelihos takedown effort.

For the D_e and D_m sets, we have precision and recall of 0.22/0.67 and 0.25/0.03, respectively. The recall for D_e is quite low as this means upwards of 30% of the domain names that are likely malicious were harvested from malware interrogation. This stresses the importance of labeling both from network information from pDNS, as well as information gathered from malware.

According to the analysis by *rza*, this takedown was largely a success, however, we know that new variants of Kelihos emerged soon after. Analyzing its 168 malware samples from before the takedown shows that a P2P C&C mechanism existed as a backup plan in the malware, which may have helped bootstrap its resurgence. This stresses the importance of being prepared to counter malware behavior after its primary infrastructure has been disabled.

4.3 Zeus

The Zeus takedown targeted a large botnet that used the popular malware kit Zeus to create its malware. This takedown relied on sinkholing the seed domains. We show the daily volumes for the sets D_S, D_e, D_m, D_r, as well the volumes for *domains in* D_e that resolve into sinkholes operated by Microsoft and the other sinkholing party, in Figure 5. Of the 2,825 malware samples analyzed, none of them included a P2P- or DGA-based contingency plan.

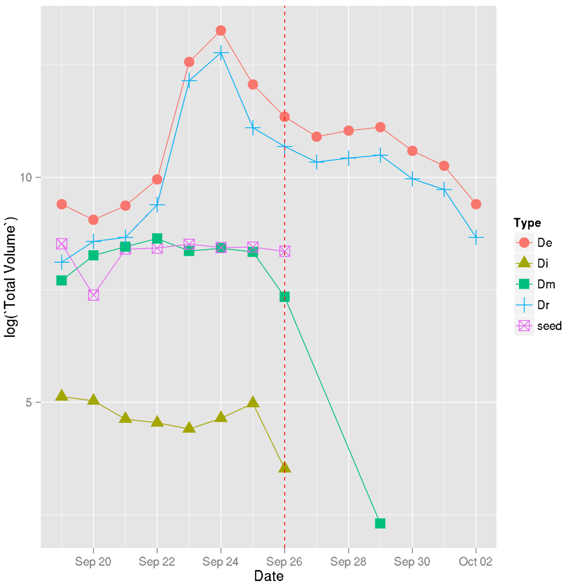

Figure 4: Kelihos aggregate daily lookup volume (log-scale).

The first observation is that unlike in the case of Kelihos, Microsoft began sinkholing domains *before* the date specified in the court order as evidenced by the non-zero query volume resolving into Microsoft's sinkholes before the takedown date. To reiterate, domains that resolved only to the sinkhole before the takedown date were not included to prevent prior uses of the sinkhole from interfering with our results. Furthermore, the volume of lookups that resolve into the sinkhole are orders of magnitude larger than the lookups only to the seed domains, suggesting that domains *not* specified in the court order were also sinkholed. We see a spike in lookup traffic directed towards the seed domains and domains that resolve to Microsoft's sinkhole, indicating increased sinkholing action at the time of takedown. *rza*'s D_m set captured fewer domains than those sunk by Microsoft's sinkhole, however, there is a large discrepancy in lookups to domains flagged as malicious by our reputation system, i.e., lookups to the domains in set D_r. We see a drop in lookups to D_r that corresponds to the Microsoft sunk domains, which indicates D_r subsumes the set of sunk domains. The other sinkhole operation experienced a similar drop after the Microsoft takedown, which suggests there was contention over which domains belonged to which sinkhole.

In this takedown, the ad-hoc nature of takedowns made coordination between companies difficult and the lack of oversight allowed the court order to not be followed exactly. While Microsoft was clearly sinkholing more domain names, the takedown interfered with an existing takedown. Without a centralized method of communicating who is sinkholing what, this pattern of stepping on other researchers' toes is likely to continue.

The computed TIR values are shown in Table 2. We compare against both the seed domain set, and the set of domains resolving into Microsoft's sinkhole. With respect to the seed domain set, we nearly tie considering malware-related domains and capture many more lookups to potentially malicious domains when considering the dataset de-

Sets	TIR value	Risk
D_m, D_S	0.979	-11,357.5
D_r, D_S	3.921	1,641,580
D_i, D_S	0.148	-478,874
D_e, D_S	14.321	7,486,221
D_m, D_{mssink}	0.553	-444,265.5
D_r, D_{mssink}	2.215	1,208,672
D_i, D_{mssink}	0.084	-911,782
D_e, D_{mssink}	8.091	7,053,313

Table 2: TIR and $Risk$ values for Zeus takedown.

rived from reputation, D_r. The story is similar when compared to domains that resolve into the Microsoft sinkhole, but to a lesser extent. Recall the volumes for Microsoft sinkhole resolutions *only* include domains we identified in D_e. This suggests that not only were these deemed malicious by a 3$^{\text{rd}}$ party, but they were added by Microsoft independent of the domains listed in the court order.

For the D_e and D_m sets, we have precision and recall of 0.03/0.98 and 0.30/0.01, respectively. Most of these values are quite low, with the exception of D_e's recall, which is unsurprising. This indicates most of the malicious domains could be identified through passive DNS. The low precision value for D_e indicates that many of these domains should probably not be targeted in a takedown and the low precision for D_m suggests that while many have low reputation and are likely malicious there are no known malware associations, reinforcing the motivation for using domain name reputation.

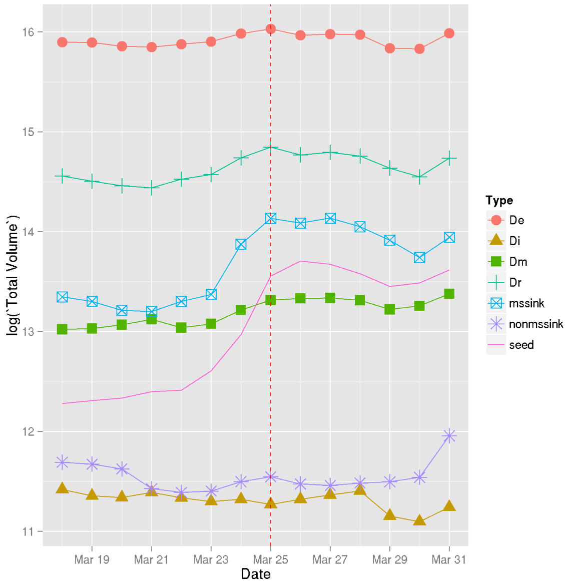

Figure 5: Zeus aggregate daily lookup volume (log-scale).

4.4 3322.org

The 3322.org takedown represents the most 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 Figure 6. Unlike the Zeus 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. We see D_i closely matches the sinkholed domain names, suggesting this is the primary method that was used to identify the takedown domains. Unlike the previous two cases, 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 samples we analyzed, none of them had a P2P- or DGA-based contingency plan.

Sets	TIR value	Risk
D_m, D_{mssink}	13.821	409,593.5
D_r, D_{mssink}	18.956	573,627.5
D_i, D_{mssink}	1.049	1,560
D_e, D_{mssink}	654.940	20,890,774

Table 3: TIR and $Risk$ values for 3322.org takedown.

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 3. 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 are similar to those for Zeus and 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. TAKEDOWN RECOMMENDATION ANALYSIS

In this section, we run *rza*'s takedown protocol on 45 botnet C&Cs being tracked by Damballa, Inc. during the month of April, 2013 and present the results. We chose to use the C&Cs already tracked by Damballa out of convenience and it is important to stress that they could be substituted by any set of domain names known to correspond to a botnet's C&C infrastructure. There are many publicly available sources of this information that allow similar experiments to be repeated. The calculated TIR values and predicated backup plans for the 45 botnets are shown in Ta-

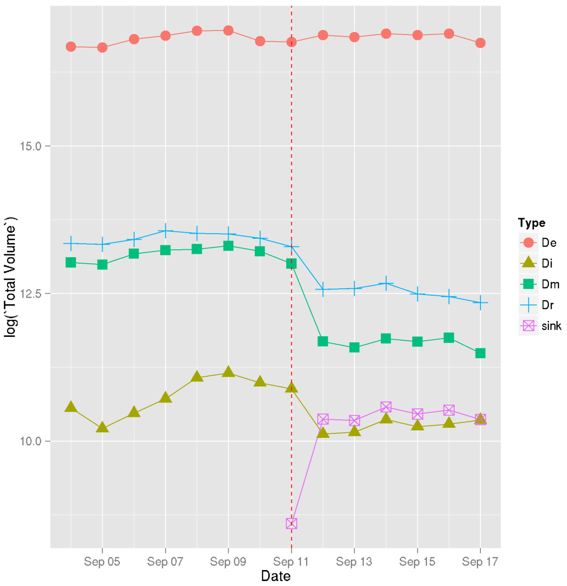

Figure 6: 3322.org aggregate daily lookup volume (log-scale).

ble 4 and the associated Risk values are shown in Table 5.

Overall, the TIR values indicate we are gaining additional infrastructure information as we did in the postmortem cases. Similarly, we see very large TIRs for the expanded infrastructure set, D_e, further evidence that infrastructure related by the passive DNS includes additional domains that need to be narrowed down by categorizing the domain sets. These TIR values were smaller, as we saw with the postmortems, with the malware-related and reputation-based sets—D_m and D_r, respectively—contributing the bulk of the newly observed lookup volumes.

In addition to describing the enumerated infrastructure sets, we also identify the backup mechanisms, if any, present in the botnet. If a botnet's malware has no backup plan, it is a prime candidate for a smooth, DNS-only takedown, otherwise we have identified the necessary conditions for performing an effective takedown. The most important finding, however, is that of the 45 botnets we studied 42 of them had no contingency plan for central C&C failure, suggesting the bulk of these botnets can be successfully taken down *without* requiring additional measures, such as reverse engineering a DGA or combating a P2P-based C&C. This suggests that while performing a takedown is difficult, we are likely to succeed in many cases.

6. CONCLUSION

We presented *rza*, a takedown analysis and recommendation system that performs postmortem analyses of past takedowns, as well as making recommendations for performing more effective takedowns in the future. *rza* would be useful in helping to both expedite the takedown process, as well as ensuring future takedowns are more complete.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments and insightful questions, and our shepherd Dr. Shiuhpyng Shieh for his guidance in improving our paper. We also thank RZA for reinventing rap production and

ID	$\mathbf{D_e}$	$\mathbf{D_m}$	$\mathbf{D_r}$	$\mathbf{D_i}$	Backup Plan
1	7.229	2.719	6.815	1.446	finite-domain
2	13.669	0.000	23.891	275.751	p2p
3	0.856	0.000	0.764	0.142	none
4	2.808	1.158	2.602	0.554	dga
5	12.005	10.117	11.612	0.023	none
6	20.632	1.448	15.665	0.044	none
7	2.130	0.015	0.798	11.917	none
8	289.387	154.932	233.521	0.000	none
9	42.570	0.000	23.522	1.395	finite-domain
10	0.746	0.000	0.597	0.241	finite-domain
11	3.783	1.068	3.208	0.255	none
12	13.115	3.809	11.896	0.246	none
13	10.139	1.698	8.726	0.697	none
14	8.266	0.000	8.259	3.190	none
15	2.028	0.189	0.131	0.094	finite-domain
16	471.226	76.176	392.788	1.045	finite-domain
17	87.004	33.807	72.759	4.052	none
18	27.036	1.810	14.952	1.021	none
19	8715.005	816.740	8696.197	8.520	none
20	170.752	0.743	10.210	0.405	finite-domain
21	7.260	2.012	5.828	0.056	none
22	13.492	1.364	12.011	0.000	none
23	14.146	0.000	12.891	7.449	none
24	52.593	0.000	52.174	0.967	finite-domain
25	223.201	6.869	21.504	0.000	none
26	1067.604	0.000	1062.696	0.001	finite-domain
27	293.466	46.070	232.437	0.005	none
28	1.251	0.311	0.923	0.082	dga
29	13.589	0.547	4.886	1.100	finite-domain
30	380.568	27.986	350.686	0.082	none
31	17.700	0.000	16.837	0.500	finite-domain
32	5.857	4.679	5.724	0.575	finite-domain
33	25.042	4.085	21.460	3.094	none
34	0.156	0.139	0.152	0.014	finite-domain
35	25.048	1.272	15.638	0.000	none
36	12.121	7.364	11.183	0.336	finite-domain
37	11.698	10.329	11.544	2.321	none
38	91.364	0.457	9.618	0.000	none
39	4.640	1.085	3.694	0.599	finite-domain
40	7.491	0.303	6.865	257.059	finite-domain
41	3.161	0.485	2.700	0.187	finite-domain
42	2.378	0.487	2.372	1.288	finite-domain
43	33.227	12.958	31.650	3.014	none
44	21.761	2.219	3.061	1.108	none
45	2.217	0.101	2.150	0.103	none

Table 4: Recent botnet TIR values (compared against D_S) and backup plan classification.

ID	$\mathbf{D_e}$	$\mathbf{D_m}$	$\mathbf{D_r}$	$\mathbf{D_i}$
1	376,591	103,904	351,529	26,974
2	6260	−494	11,312	135,766
3	−7425	−51,681	−12,205	−44,352
4	694,233	60,473	614,941	−171,099
5	485,427	402,158	468,070	−44,110
6	362,341	8262	270,663	−18,457
7	6103	−5320	−1088	58,936
8	1,249,414	666,902	1,007,383	−4332
9	255,237	−6140	138,285	2427
10	−5122	−20,195	−8131	−15,320
11	3,024,316	73,348	2,399,494	−1,086,629
12	89,727	20,801	80,695	−7406
13	9,150,880	698,974	7,735,834	−1,001,304
14	17,959	−2472	17,942	5412
15	585	−461	−494	−515
16	4,106,017	656,434	3,421,089	−8732
17	80,537	30,721	67,197	−936
18	3128	97	1676	−120
19	277,603	25,987	277,004	−32
20	5,719,294	−8674	310,317	−33,692
21	1,061,289	171,489	818,507	−169,516
22	5,586,161	162,678	4,923,961	−447,172
23	478,741	−36,417	433,040	234,848
24	204,816	−3970	203,153	−130
25	9,034,976	238,626	833,715	−40,652
26	31,892,669	−29,901	31,745,933	−29,860
27	2,023,910	311,893	1,601,580	−6889
28	692	−1897	−212	−2528
29	26,164,386	−942,098	8,075,579	−2,078,370
30	341,286	24,265	314,418	−825
31	23,614	−1414	22,393	−707
32	392,925	297,607	382,154	−34,345
33	3,798,382	487,342	3,232,483	−157,989
34	−6385	−6519	−6416	−7466
35	468,361	5289	285,099	−19,476
36	741,041	424,070	678,530	−44,278
37	452,686	394,741	446,180	−42,314
38	6,688,344	−40,216	637,882	−74,015
39	1,676,605	39,173	1,240,928	−184,676
40	5,515,527	−592,194	4,983,178	217,562,247
41	160,793	−38,321	126,529	−60,522
42	224,265	−83,469	223,419	46,946
43	243,983	90,534	232,045	−7571
44	3,648,885	214,216	362,154	−175,760
45	498,000	−367,626	470,334	−409,148

Table 5: Recent botnet *Risk* values (compared against D_S).

bringing us the Wu-Tang Clan.

This material is based upon work supported in part by the National Science Foundation under Grants No. CNS-1017265, CNS-0831300, and CNS-1149051, by the Office of Naval Research under Grant No. N000140911042, and by the Department of Homeland Security under contract No. N66001-12-C-0133. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, the Office of Naval Research, or the Department of Homeland Security.

APPENDIX

A. POLICY DISCUSSION

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 Zeus takedown case. We also saw oversight issues in the Zeus 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 *or later determined to be associated with malware activity...*” [13] (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 [16], 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” [16] and are built in to the accreditation contracts to registrars. The UDRP 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.

[6]. In its first year, UDRP successfully “handled over 2,500 cases involving nearly 4,000 names” [16] and has expanded since. In fact, ICANN is introducing The Uniform Rapid Suspension System (URS) [7] 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.

B. RZA WITH ISC/SIE PDNS DATA

We replicated part of our evaluation using only ISC-SIE 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 SIE dataset are shown in Table 6 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.

Takedown	$\mathbf{D_e}$	$\mathbf{D_m}$	$\mathbf{D_r}$
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

Table 6: SIE-computed TIR values.

C. WINDOW JUSTIFICATION

We consider the activation of a domain name $d \in D_e$ to occur when d first begins to resolve to a global IP address in the observation period and a deactivation to be the day when $d \in D_e$ no longer resolves to a global IP address. Figure 7 shows the number of new domain activations and deactivations for the domains in D_e for one of the takedowns analyzed in Section 4, as well as the net change for each day. We see that around seven days prior to the takedown the number of activations/deactivations has achieved an equilibrium motivating the choice of a 14 day observation window around the time of the takedown. Other takedowns exhibited similar behavior, but the details are not reported due to space limitations.

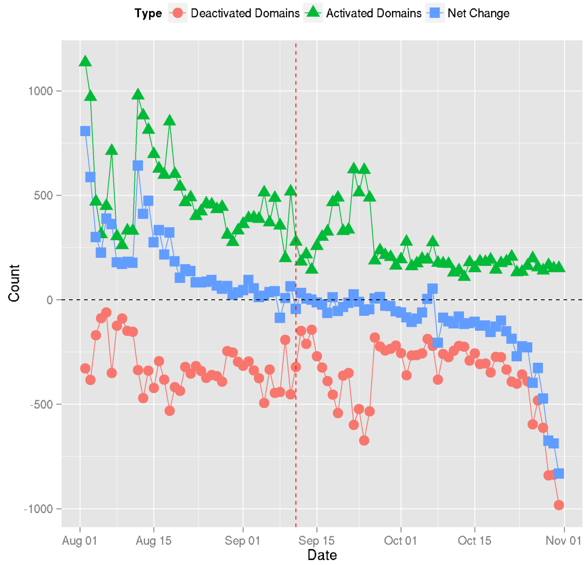

Figure 7: Domain name activations and deactivations in 3322.org takedown's D_e set.

D. REFERENCES

[1] M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee. Building a Dynamic Reputation System for DNS. In *Proceedings of the USENIX Security Symposium*, 2010.

[2] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Network and Distributed System Security Symposium (NDSS)*, 2011.

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

[4] Farsight Security, Inc. SIE/Farsight Security's DNSDB, 2013. https://www.dnsdb.info/.

[5] M. Harris. Spammers recovering from McColo shutdown, 2009. http://www.techradar.com/news/internet/spammers-recovering-from-mccolo-shutdown-591118.

[6] Internet Corporation for Assigned Names and Numbers. Uniform Domain Name Dispute Resolution Policy. Technical report, 1999.

[7] Internet Corporation for Assigned Names and Numbers. Uniform Rapid Suspension System. Technical report, 2012.

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

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

[10] B. Krebs. Mariposa Botnet Authors May Avoid Jail Time, 2010. http://krebsonsecurity.com/2010/03/mariposa-botnet-authors-may-avoid-jail-time/.

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

[12] Microsoft Corporation. Microsoft Corporation v. Dominique Alexander Piatti; Jone Does 1-22. 2011. Virginia Eastern District Court.

[13] Microsoft Corporation. Microsoft Corporation v. Peng Yong et. al. 2012. Virginia Eastern District Court.

[14] Microsoft Corporation. Microsoft v. John Does 1-39. 2012. New York Eastern District Court.

[15] E. Mills. Microsoft halts another botnet: Kelihos, 2011. http://news.cnet.com/8301-1009_3-20112289-83/microsoft-halts-another-botnet-kelihos/.

[16] M. Mueller. *Ruling the root*. The MIT Press, 2004.

[17] Y. Nadji, M. Antonakakis, R. Perdisci, and W. Lee. Understanding the prevalence and use of alternative plans in malware with network games. In *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC '11)*, 2011.

[18] C. Rossow, D. Andriesse, and T. Werner. P2PWNED: Modeling and Evaluating the Resilience of Peer-to-Peer Botnets. In *Proceedings of the 34th IEEE Symposium on Security and Privacy (S&P 2013)*, 2013.

[19] U.S. Attorney's Office - Southern District of New York. Manhattan U.S. Attorney Charges Seven Individuals for Engineering Sophisticated Internet Fraud Scheme, 2011. 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-advertising-business.

[20] VirusTotal. VirusTotal Intelligence. https://www.virustotal.com/en/documentation/private-api/.