SinkMiner: Mining Botnet Sinkholes for Fun and Profit

(Extended Abstract)

Babak Rahbarinia[a], Roberto Perdisci[a,b], Manos Antonakakis[c], David Dagon[b]

[a]Dept. of Computer Science, University of Georgia

[b]College of Computing, Georgia Institute of Technology [c]Damballa Inc.

{babak,perdisci}@cs.uga.edu, manos@damballa.com, dagon@sudo.sh

1 Introduction

Botnets continue to pose a significant threat to Internet security, and their detection remains a focus of academic and industry research. Some of the most successful botnet measurement and remediation efforts rely heavily on *sinkholing* the botnet's command and control (C&C) domains [1]. Essentially, sinkholing consists of re-writing the DNS resource records of C&C domains to point to one or more *sinkhole IP addresses*, thus directing victim C&C communications to the *sinkhole operator* (e.g., law enforcement).

Sinkholes are typically managed in collaboration with domain registrars and/or registries, and the owner of the network range where the botnet C&C is sinkholed. Registrars often play a critical role in remediating abusive domains (e.g., by invoking rapid take-down terms commonly found in domain registration contracts, such as the "Uniform Rapid Suspension System" [3]). Collaboration with the sinkhole network range owners is needed to endure the possible IP reputation damage to their IP space, since sinkholes may appear as real C&Cs to others.

While some sinkhole IPs are publicly known or can be easily discovered (see Section 2.1), most are jealously kept as trade secrets by their operators, to protect proprietary black lists of remediated domains. Therefore, third-party researchers are often unable to distinguish between malicious C&C sites and remediated domains pointed to sinkholes.

In some cases, this stove-piping of sinkhole information can cause "friendly fire", whereby security operators or law enforcement may take down an already sinkholed C&C. This results in disrupting remediation efforts, and may in some cases bring more harm to the botnet victims (whose infected clients may turn to secondary or *backup C&C domains* not being remediated). It is therefore useful to build technologies capable of identifying whether or not a C&C domain and/or IP are part of a sinkholing effort.

In this paper, we present SinkMiner, a novel forensics system that enables the discovery of previously unknown sinkhole IPs and the related sinkholed domains by efficiently mining large passive DNS databases. Being able to discover "secretive" sinkhole operations has both benign and not-so-benign implications. On a purely benign side, labeling previously unknown sinkhole IPs may prevent "friendly fire," as mentioned above. Also, the discovery of sinkhole IPs may enable a much more precise measurement of the *effective lifetime* of C&C domains. On the other hand, the ability to identify sinkhole IPs may allow less-than-honest researchers to collect all related sinkholed domains, which could then be re-sold to third-parties as part of a domain blacklist, thus unfairly taking advantage of the often very meticulous and costly work done by the sinkhole operator.

Our system's ability to detect previously unknown sinkhole IPs is based on a somewhat surprising empirical observation: *sinkhole operators often relocate C&C domains from a sinkhole IP to another* (see Section 2.2). Therefore, given a small seed of known sinkhole IPs, we can leverage passive DNS databases to monitor the "behavior" or their sinkholed domains to track where they relocate — effectively discovering "by association" previously unknown sinkholes. This is in stark contrast with what common knowledge may suggest, namely that once a C&C domain falls into a sinkhole it will never escape until it expires or is "retired" by the sinkhole operator, making it “unresolvable”.

2 System Overview

The main goal of our system is to find new and previously unknown sinkhole IPs. We start with a list of few known sinkhole IPs, $\mathcal{S}$, which may be derived through manual investigation and/or personal communications with some sinkhole operators. Using a large passive DNS database (PDNS), we travel back in time and gather all the sinkholed domains $\mathcal{SD}$ historically related to IPs in $\mathcal{S}$. In other words, $\mathcal{SD}$ contains all domains that resolved to any of the IPs in $\mathcal{S}$ at least once during their lifetime (see Section 3 for more details). Next, we extract the full IP resolution history of the domains in $\mathcal{SD}$. One may expect that after a domain is sinkholed, it will continue to resolve to that sinkhole IP for the rest of its life. Nonetheless, we found numerous counterexamples. In practice, there exist many sinkholed domains that after pointing to an initial sinkhole IP later start to resolve to some other IPs, some of which are different known sinkholes whereas others are “unknown”. Our goal is to properly label this set of unknown IPs, which we call $\mathcal{S}_{pot}$ (potential sinkholes).

We empirically found that the IPs in the set $\mathcal{S}_{pot}$ fall in one of the following categories:

1. *New Sinkhole*: These are IP addresses owned by security operators and used for the purposes of taking over and/or studying botnets. A previously sinkholed domain name may move to a new sinkhole IP due to a deliberate relocation decision performed by the sinkhole operator.
2. *Parking*: Parking IPs are typically used as a “traffic vacuum” [2]. Often, when a domain name registration expires, a registrar (or third-party) may take ownership of the expired domain, and point it to a parking IP. Machines (e.g., infected machines) that still query the now expired domain are redirected to websites that serve advertisement, thus generating revenue. Therefore, as a sinkholed C&C domain registration expires, the domain may later start resolving to one or more parking IPs.
3. *NX-Domain Rewriting*: Some ISPs generate revenue from advertisement by redirecting machines that query for non-existent (NX) domains, including some expired C&C domains, to an ad-populated web page [4]. To this end, the DNS resolver owned by the ISP performs an on-the-fly rewriting of the DNS response, injecting a valid resource record into the answer section.

Note that we do not make any claims about the IPs that the C&C domains resolved to *before* they were sinkholed. That is, the set $\mathcal{S}_{pot}$ only includes IP addresses resolved by domains that previously pointed to a known sinkhole. In the following sections, we address the problem of distinguishing new sinkhole IPs from parking and NX-domain rewriting IPs.

2.1 Preliminary Labeling

In this section, we describe two methods we use to perform a preliminary labeling of the potential sinkhole IPs ($\mathcal{S}_{pot}$).

Popularity-based labeling One thing that we observed while studying the characteristics of known sinkholes, is that sinkhole IPs are pointed to (in time) by relatively large numbers of domains (e.g., several thousands). Therefore, given the set $\mathcal{S}_{pot}$, we query the PDNS database, and rank the IPs by “popularity”, and only consider IPs that in time were pointed to by more than θ_{pop} previously sinkholed domains.

Clearly, this subset of “popular” IPs may still include parking and NX-rewriting IPs. Therefore, we map the IPs to their autonomous system (AS) and consider as (highly likely) new sinkhole IPs only those addresses that are located within an IP space owned by well-known organization that are known to operate botnet sinkholes (e.g., Microsoft, Verisign, Google, ISC, etc.).

Name server-based labeling In addition, we consider the name server name associated with the remaining potential sinkhole IPs in $\mathcal{S}_{pot}$. This allows us to find additional sinkhole IPs, and to also label a large number of known parking IPs. For example, we label as sinkhole IPs those that are resolved by name servers such as `torpig-sinkhole.org`, `ns1.sinkhole.ch`, `dns3.sinkdns.net`, `sinkhole-00.shadowserver.org`, etc. In general, we search the

PDNS database for name server names that contain the keyword “sink”, and then perform a quick manual analysis to only select names that are clearly related to botnet sinkhole operations.

Similarly, we label as parking those IPs resolved by name servers such as `dns1.ns-park.net`, `park1.dns.ws`, `nx1.dnspark.net`, `one.parkingservice.com`, etc. Again, we leverage the PDNS database to find name server names containing the word “park”, and then perform a quick manual analysis to only select the most likely parking name servers.

Labeling popular NX-rewriting IPs is also feasible. For example, some ISP are very aggressive, and return an IP even for queries to invalid domain names, which should clearly return an NXDOMAIN error. Based on this and other empirical observations, we built a number of simple heuristics to automatically label the most likely NX-rewriting IP addresses.

2.2 Graph-based Labeling

While studying the “behavior” of botnet sinkholes, we noticed that in some cases sinkholed domains would be “relocated” from a known sinkhole IP to an uncategorized IP, and then back to another known sinkhole IP (not necessarily the original one). Other, more complicated patterns were also observed: some malware domains would relocate from a known sinkhole to an uncategorized IP, then to a different uncategorized IP, and so on, before moving back to a (possibly different) known sinkhole. While we are not entirely sure what drives this behavior, we believe sinkholes are sometimes relocated to enable some form of load balancing, or to isolate some botnets from each other, for the purpose of more precise measurements.

In other cases, sinkholed domains may “naturally” relocate to one or more parking or NX-rewriting IPs, as they expire without being reclaimed by the sinkhole operators. To efficiently distinguish among such behavioral patterns, we leverage the PDNS database to build a *graph database* around the set of known and potential sinkhole IPs, $\mathcal{S} \cup \mathcal{S}_{pot}$. Specifically, we build a weighted directed graph in which a node represents an IP address $p \in \mathcal{S} \cup \mathcal{S}_{pot}$. Given two nodes p_i and p_j, we draw an edge if there exists any domain name that, according to the PDNS database, first resolved to p_i and later started to resolve to p_j. The weight of the edge is equal to the number of such domains that transitioned from p_i to p_j during a given time window of interest.

Once the graph database is built, to discover new sinkholes we perform the following queries:

(1) $\mathcal{S} \rightarrow p_x \rightarrow \mathcal{S}$: We look for any node p_x “in between” known sinkhole IPs. In other words, we look for all cases in which there exist some domains that first pointed to a known sinkhole, then moved to p_x, and then relocated to another known sinkhole. Notice that there may be cases in which there are multiple domains that resolve to p_x, and these domains previously pointed to different sinkhole IPs. Similarly, domains that point to p_x may then relocate to different known sinkhole IPs.

In the context of the query, we also set some constraints on the edge weights: we only consider an IP address p_x as a new sinkhole IP if the edge weights exceed a (tunable) threshold θ_w. We also require that the number of *distinct* “opening” and “terminal” $\mathcal{S}$ IPs that transit to/from p_x be above an adjustable threshold θ_n.

(2) $\mathcal{S} \rightarrow p_x \rightarrow p_y \rightarrow \mathcal{S}$: Similarly, we look for any pair of consecutive nodes p_x and p_y “in between” known sinkhole IPs. As for the previous query, we only consider p_x and p_y as new sinkhole IPs if the edge weights and the number of opening and terminal IPs exceed the mentioned thresholds.

Essentially, we currently use the graph database as a forensic analysis tool, to make investigating the behavior of sinkhole IPs easier, and to discover previously unknown sinkhole operations. In our future work, we plan to explore other types of queries and to fully automate the sinkhole detection process.

3 Preliminary Evaluation

To evaluate SinkMiner, we started from an initial list of 22 known sinkholes ($\mathcal{S}$) from 19 different Autonomous Systems (AS). Table 1 lists some of the ASes (we refrain from disclosing the initial sinkhole

Table 1: Examples of known sinkhole locations

ASN	Organization	Popularity	ASN	Organization	Popularity
14618	AMAZON-AES	46,959	1280	ISC	16,987
8069	MICROSOFT	16,522	2637	GEORGIATECH	15,390
30060	VERISIGN	11,168	15169	GOOGLE	630

Table 2: Examples of newly found sinkhole IPs

IP	ASN	Organization	Popularity
93.170.52.30	44557	DRAGONARA	817,563
216.239.32.21	15169	GOOGLE	535,638
69.25.27.173	10913	INTERNAP	347,902
208.91.197.101	40034	CONFLUENCE	337,539
174.129.212.2	14618	AMAZON	110,381
199.2.137.141	3598	MICROSOFT	1,367

IPs, because they were provided to us by collaborators and are not part of our new discoveries). By querying our PDNS database, which contains historic DNS information that dates back to the start of 2011, overall we extracted 2,945,483 sinkholed domains. However, many of these domains appeared to be related to DGA-based botnets[1]. To eliminate this "DGA noise", we filtered out domain names that appeared in the PDNS database for less than three days. This reduced our set of sinkholed domains to 130,901. The "popularity" column of Table 1 shows the number of domains pointing to sinkholes in the listed ASes. As mentioned before, many C&C domains change resolved IPs after being sinkholed. We observed such behavior in 51,371 domains (39%). Overall, we collected 5,576 distinct IPs that appear after a known sinkhole, which represent our set $\mathcal{S}_{pot}$.

Among the $\mathcal{S}_{pot}$ IPs, using the approach described in Section 2.1, we were able to identify 23 new (highly likely) sinkhole IPs based on popularity, and 15 based on name server names, thus expanding our initial set of sinkholes from 22 to 60. In the process, we were also able to label 475 IPs as related to parking services, and 7 IPs related to NX-rewriting.

Our graph database (Section 2.2) is built over the set of both known and potential sinkholes ($\mathcal{S} \cup \mathcal{S}_{pot}$). As mentioned above, using the preliminary labeling approach we were able to label some of the graph nodes in $\mathcal{S}_{pot}$ as either "popular" sinkhole, parking or NX-rewriting. Overall, the graph consisted of 5,613 nodes and 164,344 edges.

To set the detection thresholds θ_w and θ_n described in Section 2.2, we fine-tuned them so to obtain no false positives (FP). Here, we consider an IP p_x classified as sinkhole through our graph as a FP if it was previously labeled as either parking or NX-rewiring. By leveraging the graph database queries defined in Section 2.2, we were able to label 49 highly likely new sinkhole IPs. In particular, by manual inspection we verified that query (1) yielded 12 highly likely new sinkhole IPs, whereas query (2) yielded 37 new potential sinkholes. In our future work we plan to seek further confirmation through a more direct collaboration with sinkhole operators.

To summarize, SinkMiner allowed us to find 87 new (highly likely) sinkholes, thus expanding our initial list of 22 known sinkhole IPs to 109. Overall, these 109 IPs were resolved by 3,443,344 distinct domains. This demonstrates the potential impact of discovering new sinkhole IPs using C&C domain intelligence.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This material is based in part upon work supported by the National Science Foundation under Grant No. CNS-1149051. 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.

References

[1] Bruneau, G. DNS sinkhole, 2010. http://www.sans.org/reading_room/whitepapers/dns/dns-sinkhole_33523.

[2] Flanagan, E. M. No free parking: Obtaining relief from trademark-infringing domain name parking. *Minn. L. Rev. 92* (2007), 498–1966.

[3] ICANN. Request for Information - Uniform Rapid Suspension System. https://www.icann.org/en/news/rfps/urs-24sep12-en.pdf, September 2012.

[4] Weaver, N., Kreibich, C., and Paxson, V. Redirecting dns for ads and profit. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI), San Francisco, CA, USA (August 2011)* (2011).

[1] DGA = domain generation algorithm.