Analyzing DNS Activities of Bot Processes

Jose Andre Morales
Institute for Cyber Security
University of Texas at San Antonio
jose.morales@utsa.edu

Areej Al-Bataineh
Department of Computer Science
University of Texas at San Antonio
aalbata@cs.utsa.edu

Shouhuai Xu
Department of Computer Science
University of Texas at San Antonio
shxu@cs.utsa.edu

Ravi Sandhu
Institute for Cyber Security
University of Texas at San Antonio
ravi.sandhu@utsa.edu

Abstract

Detecting bots is becoming increasingly challenging with the sophistication of current bot technology. Most research has focused on identifying infected host machines but is unable to identify the specific bot processes *on the host. This research analyzes* active *bot processes with emphasis on a newly identified vector of detection based on DNS activities occurring throughout the bot life cycle with a primary focus on the early stage of the cycle (i.e., when bots first join a botnet). Specifically, we propose criteria for detecting bot processes based on their reaction-to-DNS-response behavior (RD behavior). Our experimental results confirm that the newly identified vector of detection can, in most cases, accurately identify bot processes during the early stage in their life cycle and can improve detection results of current commercial bot detection software.*

1 Introduction

Bots have emerged as one of the premiere tools for malicious activities in the cyberspace because they can generate high profits for malware authors and deployers. A bot can be defined as a malware which infects host machines and in a stealthy manner joins a centralized or peer-to-peer (P2P) botnet. These bots establish communication channels to receive instructions from a bot master. Most published botnet research focuses on the network layer so as to investigate botnet characteristics and structures. There is much less research on analyzing bots from the host perspective, let alone focusing on the specific bot process running on a host machine.

This research adds to host-based behavior analysis of botnets with a focus on a bot process's DNS activity (DNS and Reverse DNS(rDNS) queries). More specifically, it focuses on the process's reaction-to-DNS-response behavior (RD behavior) occurring in the initial join phase during the early stages of a bot process's life cycle. We stress that our detection approach of RD behavior can be implemented at any point in a bot's life cycle, we choose to focus on the early stage attempting to prevent damage and distribution in the host machine and network. During the initial join phase, bots may frequently use DNS activity to assist in locating their Command & Control (C&C) servers or other peers. We organize different paths of RD behavior that can occur in the join phase as a directed tree, classifying expected versus anomalous, and thereby suspicious, RD behavior. We analyzed five currently active centralized and P2P bots, benign network applications and non-bot malware. During analysis, we identified suspicious RD behavior. We compared our analysis of two commercial bot detectors and combined the results to improve detection accuracy. The contributions of this research are:

- Identify a novel vector of suspicious process behavior based on the process's reaction-to-DNS-response (RD behavior). We further represent suspicious behavior via paths on a directed tree of DNS activity combined with RD behavior.
- Enhance host-based detection methods with a new vector of detection, namely suspicious RD behavior. We target bot processes rather than just bot machines.
- We show that this suspicious behavior often occurs in the early stages of bot execution, thus detection at this point in time can prevent the bot from executing received commands.

The rest of this paper is organized as follows: Section 2 presents related work, Section 3 describes anomalous RD behaviors and paths, Section 4 describes our experimentation with data collection, results, analysis and limitations, Section 5 gives our conclusions and future work and Section 6 is acknowledgments.

2 Related Work

Analysis research employs different techniques to illustrate botnet size and scope and to estimate the number of bots in a given botnet, network structures, types of communication channels and stealth methods [3, 7, 9, 1, 11]. Detection research has successfully detected botnet presence mostly using captured network layer data [4, 5, 2, 9, 8]. Using IP addresses, server names, spatial-temporal correlations and sequences of events, these techniques not just identify the botnet but also its members, servers, periods of peak activity and possible location of bot masters. Host-based research [14] has been successful in tracing the execution cycle of known bot samples and their usage of tainted network data in system calls. Our research is complementary to the aforementioned related work either in approach (network-end vs. host-end) or in detection granularity (bot machines vs. bot processes). As such, these various techniques should be integrated into a single comprehensive solution framework.

3 RD Behavior

A bot process's life cycle starts with a successful host infection. After successful infection, the bot attempts to join a botnet. We call this the *initial join phase*. After a successful join, the bot starts receiving and executing commands. Since bots obtain instructions from peer bots or a central server, the bot cannot participate in botnet activities until it joins an active bot network. The join phase can be expressed in three general steps:

1. Obtain IP addresses of peer bots or a central server.

2. Attempt connecting to obtained IP addresses.

3. Join the botnet and obtain instructions.

A bot can attempt to join a botnet several times throughout its life cycle. The steps of the initial join phase can be generalized to represent any join attempt throughout the bot's life cycle. DNS activity mostly occurs in step one. The reaction-to-DNS-response behavior mostly occurs in step two. IP addresses and/or domain names are either hardwired or dynamically generated by bots. These internally obtained addresses and names are used by bots to acquire active peer bots or central server IP addresses. Within this acquisition period DNS activity plays a critical role in one of three forms: First, DNS query of internally obtained domain names harvest a set of possibly active IP addresses. Second, successful rDNS query may convince the bot that the IP address is active and can return new domain names for future DNS queries harvesting more active IP addresses. Third, a bot may conduct a DNS query on an internally obtained domain and, if successful, conduct a rDNS query on the returned IP address(es). The third approach gives the strongest confidence level of an active IP address assuming both queries are successful.

We define a process's reaction-to-DNS-response behavior (abbreviated as RD behavior) as the sequence of events a process executes in reaction to the returned results of a requested DNS or Reverse DNS query. An event is the series of program statements executed to accomplish one specific task such as a TCP `SYN` for initial connection attempt or `ACK` for successful connection. When a process performs a DNS or rDNS request we classify the process's RD behavior to be expected or anomalous. Expected RD behavior is defined by two criteria:

1. An IP address that fails to resolve to a domain name in a rDNS query is not used in any connection attempt.

2. The returned IP address of a successful DNS query or the IP address used in a successful rDNS query is used in a successful connection attempt.

Based on these two criteria we define anomalous RD behavior as:

3. An IP address that fails to resolve in a rDNS query is used in a successful or failed connection attempt.

4. The returned IP address of a successful DNS query or the IP address used in a successful rDNS query is used in a unsuccessful connection attempt.

We equate anomalous to *suspicious RD behavior* (SRDB) which mostly occurs in bot (and some non-bot malware) processes but not in benign processes. Any process exhibiting SRDB is considered suspicious of being malware (bot or non-bot).

We represent the RD behavior with the directed tree shown in figure 1 based on the RD behavior rules 1 - 4 and steps 1 and 2 of the join phase. Step 3 of the join phase is excluded based on an assumption that the sequence of events needed to accomplish this step will occur only after a successful connection with a peer bot or command center. We consider both successful and failed

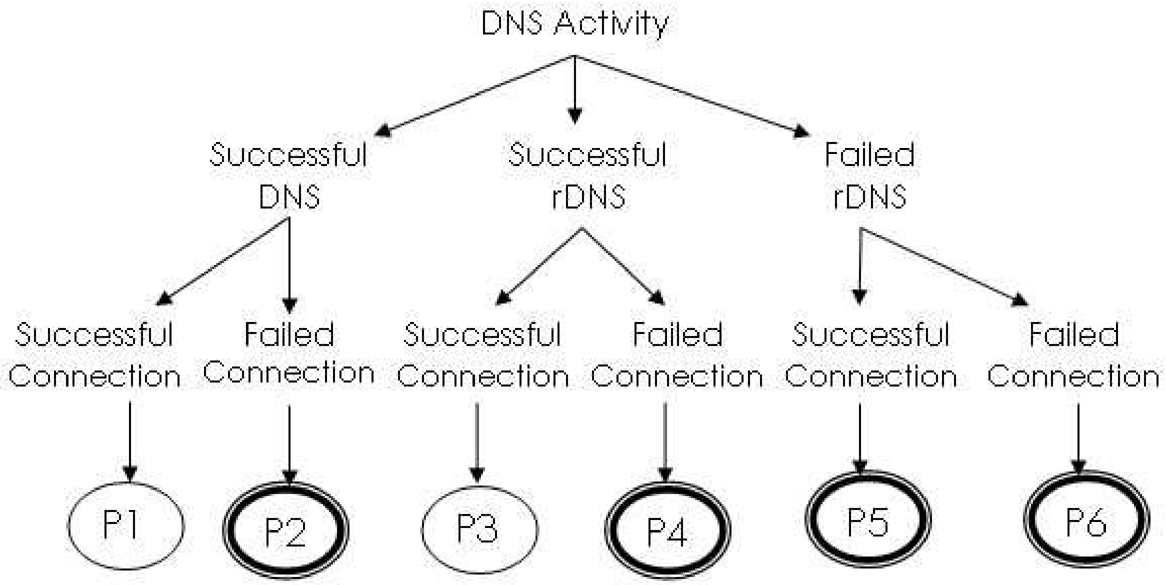

Figure 1. RD Behavior Tree: P_2, P_4, P_5 and P_6 are suspicious paths

TCP connection attempts associated with currently running processes on a host machine. We equate a successful connection with the completion of a TCP handshake [15] and a failed connection as a non-completed TCP handshake. We characterize a failed handshake as: an initial synchronization `SYN` sent to a remote host one or more times with no response or an initial `SYN` sent to a remote host is responded with an acknowledgment `ACK` and connection reset `RST`. We do not analyze further TCP activity beyond the handshake.

The tree is composed of paths, each representing a sequence of events that include a DNS activity and the process's reaction to the returned response. Paths P_1 - P_6 is the set of possible process RD behaviors. The paths represent expected behavior (single line circles) and SRDB (double line circles) that a process can follow. Paths P_1 and P_3 incorporate expected RD behavior rules 1 and 2, paths P_2, P_4, P_5 and P_6 are assumed SRDB due to their incorporation of the anomalous RD behavior rules 3 and 4. The tree does not consider paths with a failed DNS query since this response does not provide IP addresses usable in connection attempts. The SRDB paths that can be used to detect bot processes are defined as follows:

Path P_2. A failure to connect to an IP address obtained from a successful DNS query.

Path P_4. A failure to connect to an IP address used in a successful rDNS query.

Path P_5. A successful connection to an IP address used in a failed rDNS query.

Path P_6. A failure to connect with an IP address used in a failed rDNS query.

4 Experimentation

We present our analysis of active bot, benign and non-bot malware processes along with their manifestation of SRDB. Also presented is the evaluation of two commercially available host-based behavior bot detectors: `Trend Micro RuBotted` [17] and `Norton Anti-Bot` [13].

Data collection. Three test sets were analyzed: bots, non-bots, and benign.[1] The bots and their characteristics are listed in table 1. The non-bots: `Netsky.D`, `Bredolab.A`, `Ursnif.C`, `Brontok.Q` and `Lovgate.X` are a collection of malware samples circulating in the wild from January to May of 2009 [16, 10] with capabilities of Worms, backdoors, and Trojan downloaders. The benign set consisted of: `BitTorrent`, `avp`, `CuteFTP`, `LimeWire`, and `Skype`. We executed each sample on a VMWare Workstation virtual machine running Windows XP SP2 with no updates and no Antivirus. We collected network traffic using Windows Network Monitor [12] for a one hour collection period.

Bot	Purpose	C&C Architecture	C&C protocol	Uses Encryption	Stealth Mechanism
Bobax.O	Spamming	Centralized	UDP/TCP port 447	Yes	Dynamic DNS
Ozdok.A	Spamming	Centralized	HTTP port 80, port 443	Yes	
Waledac.A	Spamming	P2P	P2P HTTP port 80	Yes	Fast-flux & Double fast- flux
Wopla.AB	Spamming	Centralized	TCP port 8080	Yes	
Virut.A	Malware distribution	Centralized	IRC	No	

Table 1. Characteristics of Bots in Test Set

	DNS	rDNS	DNS &rDNS
Bot			
Ozdok	0	0	1
Bobax	0	0	2
Wopla	0	4	1
Waledac	0	40	2
Virut	0	2	0
Non-Bot Malware			
Netsky	1	1	11
Bredolab	0	1	0
Lovgate	0	0	1
Brontok	1	0	2
Ursnif	0	1	0
Benign			
BitTorrent	1	0	0
avp	1	0	0
cuteftp32	8	0	0
LimeWire	0	0	0
Skype	1	0	0

Table 2. Number of distinct IP addresses used in DNS and rDNS queries

4.1 Results and Analysis

DNS Activity. The number of distinct IP addresses involved in a DNS activity and a connection attempt (both successful and failed) by our test set is shown in table 2. The first column is our test set. The second column (DNS) is the total IP addresses acquired solely by a DNS query. The third column (rDNS) is total IPs used solely in a rDNS query. The last column (DNS & rDNS) is total IPs acquired through a DNS query and used in a rDNS query. The benign samples only used DNS queries to acquire IP addresses. The table shows our malware samples (bots and non-bots) used frequent rDNS queries alone or combined with DNS. In our results, several identified SRDB behaviors resulted from malware samples using DNS&rDNS combination where the DNS query acquires an IP address later used in an rDNS query; this led to several instances of SRDB P_4 - P_6.

Suspicious RD behavior. The number of instances of SRDB paths for each sample is shown in table 3. The values represent the total number of distinct IP addresses used with the specific SRDB path. For example, the bot Waledac has a value of 9 for path P_5 indicating Waledac used 9 distinct IP addresses that failed an rDNS query and also failed to establish a connection. None of the benign processes followed paths P_4, P_5, and P_6 implying these processes follow expected behavior rule 1 of ignoring IP addresses associated with a failed DNS activity. BitTorrent and Cuteftp32 has one instance each of P_2.

	P_2	P_4	P_5	P_6
Bot				
Ozdok	0	0	0	1
Bobax	2	1	0	1
Wopla	0	0	0	1
Waledac	0	25	9	7
Virut	0	0	0	1
Non-Bot Malware				
Netsky	12	10	2	0
Bredolab	0	1	0	0
Lovgate	1	0	1	0
Brontok	0	0	0	1
Ursnif	0	0	1	0
Benign				
BitTorrent	1	0	0	0
avp	0	0	0	0
cuteftp32	1	0	0	0
LimeWire	0	0	0	0
Skype	0	0	0	0

Table 3. Test Results: number of distinct IP addresses with the respective identified suspicious RD behavior paths

BitTorrent issued a DNS query for

[1] Malware Identified with VirusTotal [18].

tracker.torrentbox.com and failed to connect possibly due to the server being down given that several SYN packets were sent with no response. Cuteftp, on the other hand, was tested purposefully against inactive ftp servers. We could consider these cases as false positives, but pruning P_2 from table 3 eliminates the two false positives while not producing any false negatives. P_2 is the only path dealing with successful DNS queries, thus we consider P_2 to be an anomalous but not suspicious RD behavior and exclude it from further evaluations.

In comparison with benign samples, the results of malware samples, bots and non-bots, differs significantly. All of these samples had at least one IP address in an SRDB path. This leads us to the conclusion that SRDB could be used to detect other types of malware beyond bots. The collection period was one hour but all the needed information occurred within the first 7 minutes of each malware's (bot and non-bot) life cycle indicating SRDB can be detected at a very early stage.

Amongst the bot samples, P_6 was dominant; each bot had at least one instance of it. The most interesting bot was Waledac, with the highest count of SRDB paths totaling 41. rDNS queries was used for all IPs except one. The IP address 220.66.255.89 used in one instance of P_6 was returned from a DNS query of besthandycap.com. This bot is P2P and known to have around 30 hardwired IP addresses of peers [19]. It also uses fast-flux [6, 11] domains hosted on peers to keep them connected. In our analysis it attempted to connect to these peers but with low success rate, explaining P_4 to be 25. On the other hand, the IPs were used in rDNS queries to get domains that might lead to discovery of new peers. The results suggest fast-fluxing can produce significant SRDB.

Commercial bot detectors. The detection accuracy test of the host-based behavior bot detectors Trend Micro RuBotted [17] and Norton Anti-Bot [13] along with our SRDB results and enhanced evaluations are summarized in table 4 with X = not detected, $\surd$ = detected. The first column is the test set, the next two columns show the detection results of the two commercial bot detectors and the fourth one shows if the sample exhibited SRDB or not. In the last two columns we combine the results of SRDB with each detector in a logical OR ($\vee$) operation.

Table 4 shows that SRDB yielded higher detection accuracy for bots and non-bot malware than the two bot detectors with no false negatives and no false positives. This supports our initial claim that SRDB only occurs in malware and especially in bots, but it does not occur in benign processes. This makes SRDB a critical feature to consider in any behavioral anti-bot or anti-malware solution. Rubotted and Anti-Bot combined with our results yielded perfect results. This indicates the addition of our detection vector to other bot detection solutions can generate a more robust approach with greatly improved detection accuracy.

Impact of results. Our results indicate benign processes tend to mostly follow expected RD behavior, bots (and non-bot malware) follow expected RD behavior and SRDB. The presence of SRDB in the non-bot malware samples is encouraging, showing the detection applicability of this vector to malware categories beyond bots. IP addresses of failed rDNS queries were ignored by the benign samples but often used by the bot and non-bot malware samples. We conclude the most highly occurring SRDB are failed connection attempts associated with successful rDNS queries, path P_4, and all connection attempts with IP addresses associated with failed rDNS queries, paths P_5 and P_6.

5 Conclusion and Future Work

This research analyzed a process's expected and suspicious reactions-to-DNS-response (RD behavior) between bot, non-bot malware and benign processes. Bots may frequently depend on DNS activity to initially join a botnet though expected behavior may not always be followed. Combining suspicious RD behavior analysis with two commercial bot detectors improved detection accuracy. We conclude the most highly suspicious process RD behavior are failed connection attempts linked to successful reverse DNS queries and all connection attempts using IP addresses associated with failed reverse DNS queries. Future work includes evaluating various samples belonging to the same bot family, developing a formal definition of RD behavior and combined evaluation with several other host-based solutions to further increase bot process detection accuracy.

6 Acknowledgments

This work is partially supported by grants from AFOSR, ONR, AFOSR MURI, and the State of Texas Emerging Technology Fund.

References

[1] D. Dagon, G. Gu, C. P. Lee, and W. Lee. A taxonomy of botnet structures. *Computer Security Applications Conference, Annual*, pages 325–339, 2007.

[2] J. Goebel and T. Holz. Rishi: identify bot contaminated hosts by irc nickname evaluation. In *HotBots'07: Proceedings of the first conference on First Workshop on*

	Rubotted	Anti-Bot	SRDB	SRDB $\vee$ Rubotted	SRDB $\vee$ Anti-Bot
Bot					
Ozdok	X	X	√	√	√
Bobax	X	√	√	√	√
Wopla	X	√	√	√	√
Waledac	X	X	√	√	√
Virut	√	√	√	√	√
Non-Bot Malware					
Netsky	X	√	√	√	√
Bredolab	X	X	√	√	√
Lovgate	X	√	√	√	√
Brontok	X	√	√	√	√
Ursnif	X	X	√	√	√
Benign					
BitTorrent	X	X	X	X	X
avp	X	X	X	X	X
cuteftp32	X	X	X	X	X
LimeWire	X	X	X	X	X
Skype	X	X	X	X	X

Table 4. Detection Analysis: RuBottd, Anti-Bot and Suspicious RD Behavior

Hot Topics in Understanding Botnets, pages 8–8, Berkeley, CA, USA, 2007. USENIX Association.

[3] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: overview and case study. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 1–1, Berkeley, CA, USA, 2007. USENIX Association.

[4] G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium (Security'08)*, 2008.

[5] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. BotHunter: Detecting malware infection through ids-driven dialog correlation. In *Proceedings of the 16th USENIX Security Symposium (Security'07)*, August 2007.

[6] T. Holz, C. Gorecki, Y. Rieck, and F. Freiling. Measuring and detecting fast-flux service networks. *Proceedings of the Network & Distributed System Security Symposium (NDSS '08)*, 2008.

[7] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9, Berkeley, CA, USA, 2008. USENIX Association.

[8] X. Hu, M. Knysz, and K. G. Shin. Rb-seeker: Autodetection of redirection botnets. *16th Annual Network and Distributed System Security Symposium*, 2009.

[9] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 7–7, Berkeley, CA, USA, 2007. USENIX Association.

[10] Kaspersky lab viruslist. `http://www.viruslist.com`.

[11] J. Nazario and T. Holz. As the net churns: Fast-flux botnet observations. *3rd International Conference on Malicious and Unwanted Software, MALWARE 2008*, pages 24–31, 2008.

[12] Network monitor 3.2. `www.microsoft.com/Downloads`.

[13] Norton anit-bot. `http://en.wikipedia.org/wiki/Norton\_AntiBot`.

[14] E. Stinson and J. C. Mitchell. Characterizing bots' remote control behavior. In *DIMVA '07: Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 89–108, Berlin, Heidelberg, 2007. Springer-Verlag.

[15] Tcp handshake. `http://en.wikipedia.org/wiki/TCP\_handshake`.

[16] The wildlist organization international. `http://www.wildlist.org`.

[17] Trend micro rubotted. `http://www.trendsecure.com/portal/en-US/tools/security\_tools/rubotted`.

[18] Virustotal, free online virus and malware scan. `http://www.virustotal.com/`.

[19] Waledac questions answered. `http://www.lavasoft.com/mylavasoft/company/blog/waledac-questions-answered`.