Botnet Analysis Using Command and Control Channels

Scott Durno
scottdurno@gmail.com
15 December 2011

Carleton University

Abstract. Motivated by the goal of understanding the current state-of-the-art for analysis, detection and mitigation of botnets on an Internet connected enterprise network, I have surveyed recent research that specifically targets the necessary command and control communication channels that are used to manage the lifecycle and campaigns conducted by botmasters utilizing large, distributed botnets. The specifics of botnet command and control feature space is explained, along with research that targets the subsets of these features.

1 Introduction

One of the most significant current problems for network security experts is the proliferation of botnets; large networks of exploited computers that are under the control of a remote master (botmaster, bot herder) who can manage the lifecycle and activities of the exploited computers in the *botnet* to conduct a potentially large variety of malicious activities using a variety of effects [11]. Botnets are used by malicious actors for many purposes such as spam campaigns, key logging, click fraud, scareware schemes, spyware, distributed denial of service (DDoS), fast-flux phishing support, and other criminal endeavours. The number of current and abandoned botnets is not known however there are a few that are better known such as Rustock, Mega-D, and Storm. These botnets and others have provided insight into command and control (C2 or C&C) methods that have been used by security researchers to help build detection algorithms, however it is only natural that as researchers and network security analysts become more proficient at detecting and disabling botnets, botmasters have become more skilled and creative at hiding their malware and communication channels.

The bot malware that has been used to infect and control the computers in the botnet can be deployed with a number of built-in capabilities and can be updated, refocused, or even deleted by the botmaster. The key factor that all botnets have in common is the requirement for a C2 infrastructure and protocol for the botmaster to direct the activities of the bots through the Internet; see *Figure 1.* Early bots typically used Internet Relay Chat (IRC) as their communications channel as bots initially grew out of the IRC community. Over time botmasters have expanded their capabilities to use hypertext transfer protocol

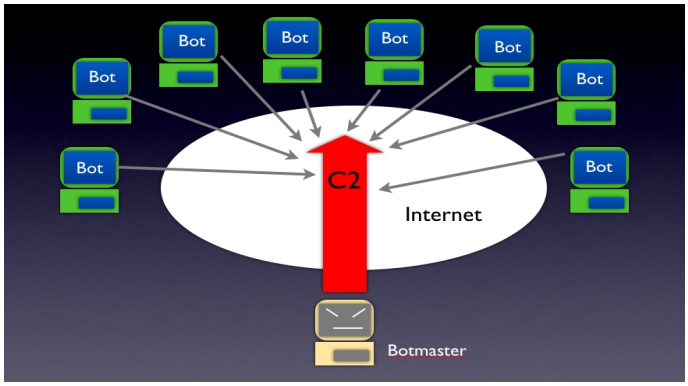

Fig. 1. Simplified botnet C2 environment

(HTTP) and peer-to-peer (P2P) for botnet command and control; some more skilled developers have programmed custom C2 protocols for their botnets to help obfuscate their activities.

Network Security is aimed at protecting the confidentiality, integrity and availability (CIA) of network capabilities and data [21]. Network security experts work to make a network more resilient to exploit by many methods that include the deployment of antivirus/anti-spyware, deployment of firewalls, Intrusion Prevention Systems (IPS), updating operating systems and applications [19], developing and updating security-related system policies in order to limit the perceived threats and risks to the CIA of the target network. The network security posture must be responsive to the evolution of threats and therefore must be informed by ongoing security research and the analysis of network defence incidents. **Network Defence** in this context can be considered the actions that must be taken by network security and/or network defence staff when network security actions have failed to prevent a network infection [20]. Defensive actions in this scenario include finding, containing and cleaning infected hosts; and any analysis that follows these actions that can be used to create a more robust network security posture to help mitigate the vulnerabilities that allowed the attacker to be successful. In studying botnets and their C2 traffic the ultimate goal is to develop the tools and strategies that will strengthen the security and defence of networks against the threat of bots being deployed on network hosts or to provide methods to find and clear successful infections as soon as possible once discovered.

This remainder of this paper is organized as follows. Section 2 describes the basics of botnet command and control channels with a focus on the motivations and challenges of researching them. Section 3 describes specific areas of research focused on botnet command and control analysis. Section 4 describes recent research targeting the areas of analysis discussed in Section 3 including the strengths, weaknesses and overlap between the various research streams. Section

5 presents conclusions and suggestions for possible future work with respect to enhancing botnet analysis and the goal of proactive defence.

2 Discussion of Botnet Command and Control

Network computer hosts that have been infected with bot malware are often referred to as *bots* or *zombies* and the botnets they are part of present a growing network security challenge on any individual computer and particularly on large networks where the challenges of finding and disinfecting compromised hosts is especially difficult. Computer network security is often discribed as an arms race between attackers and network security experts where the advantages are generally in the favour of the attackers who only need to identify a single weakness in a network to be successful and the security analysts must find all of the vulnerabilities and successfully fix them which is an impossible challenge.

2.1 Botnet C2 Description

As described in SLINGbot [14] and by TrendMicro [16], the details and phases of bot and botnet lifecycle can be understood when broken into distinct features. A botnet's C2 will have a communication **topology**, for example a centralized topology could include a remote web server that all bots are programmed to contact for new commands, binary downloads, etc. Other topologies include hierarchical, peer-to-peer, broadcast each with its own specific message protocols. Each new host that becomes infected with the bot malware, either through a Trojan, drive-by download, or other method must then find and join the botnet; SLINGbot names this process the **rallying mechanism**. The **communication protocol** are often taken from existing communications protocols on the Internet such as IRC, AIM, HTTP and P2P however some botnet programmers create custom communication protocols that must be analyzed and described by researchers. The **control mechanism** describes the methods used by the botmaster to manage and direct the activities of his bot army; the botmaster will want to hide or obscure the control mechanism commands in order to prevent detection by security analysts. Finally, bots are often designed with a mechanism to authenticate new commands from the botmaster in order to prevent a third-party from commandeering the botnet; these **command authentication mechanisms** could include passwords, digital signatures, certificates.

2.2 Motivation

In order for network security and/or network defence staff to reduce or eliminate botnets that exist on large enterprise networks and protect their networks from reinfection they require strategies, algorithms and tools that are capable of providing good detection with low false positive rates without the need to make expensive network changes or noticeably impact network services provided to users of the network. In addition, the network security community is understandably

interested in finding solutions that can help develop pro-active network defence capabilities [14] that can be deployed to prevent a network infection rather than clean up an infection after it has occurred. These goals require capable and efficient solutions to this complex problem that is often constrained by limited network security infrastructure and minimal network security budgets.

2.3 Inference, Identification and Analysis Challenges

Command and control network traffic is present on any network that contains infected hosts that are in some active communications phase of their life and this C2 traffic provides an opportunity for network security/defence analysts to find, contain and eradicate any bot infected hosts in their enterprise. The challenge for network security analysis is that botmasters work hard to obfuscate their communications and they are difficult to discern from regular, legitimate network traffic especially in large, high traffic modern networks that may have terabytes of Internet traffic a day. In addition to this problem, bots may also be programmed to open large numbers of network connections to well-known, legitimate websites for several reasons including testing network availability, time/date checks and especially to help hide the few or one connection to its command and control by creating a large amount of uninteresting network traffic to hide the few malicious network packets from security analysts and their sensors.

Recently there has been a growing body of research aimed at understanding C2 traffic in an effort to detect, characterize and isolate botnet traffic in an automated way that can assist network defence without a large number of false positives that would create a lot of extra work for network support technicians who would be responsible for isolating, testing and cleaning/re-imaging potentially infected computers. The spread of bots and therefore botnets is providing a rich opportunity for research into the nature and methods of C2 communications.

3 Botnet Command and Control Communications Analysis

Research into the analysis and characterization of botnet C2 communications encompasses many varied approaches to this difficult problem space. The approaches surveyed in this paper cover the following analysis techniques: analysis of a known specific botnet's C2 traffic; offline analysis of network traffic containing botnet C2 traffic; studying known bot binaries and infected hosts computers in virtual environment (honeypot, honeynet or sandbox); interacting with active botnets via infiltration of a botnet; and developing simulations of known and potential botnet C2 traffic to facilitate pro-active botnet defences.

In most cases, the research in one area will benefit from advances made in one of the other lines of botnet C2 research; and overlaps between approaches is common and valuable. This is particularly true for the detection and analysis of bot traffic at the host level either in a honeynet or live traffic capture. The

confirmation of an actual bot infection at the host-level can be used to confirm findings made at the network traffic analysis level [1, 2].

3.1 Specific Known Botnet Traffic Analysis

This section discusses research that focuses on the command and control communications of a previously identified botnet in an effort to create a fuller understanding of botnet management, strategies, C2 architectures, and protocols. The close study of a fully active, in-the-wild botnet is a valuable method in the effort to prevent future infections by that specific malware, but also related families of the same botnet and additionally allows researchers to test and tune tools and algorithms that can be used against other novel botnets in real-world applications.

3.2 Analysis of Network Traffic Containing Botnet C2 Traffic

The use of large databases of network traffic from enterprise networks or ISPs that contains botnet traffic can be used in various ways to enhance detection algorithms, develop inference models of C2 communications between bots and the botmaster's C2 channel, and possibly facilitate the work of researchers who are focused on future botnet command and control evolutions with an eye on the proactive security (defence) of computer networks. Anomalies in the high volume traffic such as certain DNS queries to known C2 domains, long-lived IRC connections, and communication graphs of peer-to-peer traffic can be used to narrow researchers' focus on network traffic that is more likely to be botnet C2 [2, 13]. The current reality in network defence is that human analysts attempt to identify suspicious network traffic based on network intrusion detection systems (IDS) that trigger alarms using a combination of vendor supplied and custom developed signatures to help provide clues that can assist with their analysis.

The volume of traffic on a large network in combination with the continuous evolution of botnet communication patterns makes the job of network security/defence analysts exceedingly difficult; therefore this is an area that is particularly ripe for developments in advanced analysis techniques.

3.3 Analysis of Botnet Binaries in a Virtual Environment

A well-established, yet continually evolving field of botnet analysis is focused on studying botnet activity in a safe, virtual network environment often referred to as a "honeynet". This type of analysis is commonly used as a basic underpinning of other types of botnet traffic analysis as it can provide a considerable amount of specific information about the activities of botnet binaries and/or infected network hosts. Newer bots are often designed with anti-detection or even anti-honeynet capabilities that can hamper this type of analysis, leading to research into newer methods of honeynet analysis to overcome these challenges [7].

3.4 Infiltration of Known Botnets

A growing number of researchers have directed their efforts at the active infiltration of a live botnet in order to gain a greater understanding of an active system of bots and the command and control traffic that botmasters use to manage the lifecycle of their herd of bots. Research in this area requires enough understanding of the specific botnet C2 to be able to mimic the expected communications with the botnet in order to prevent detection by the botmaster or trigger potentially damaging repercussions from the C2 infrastructure such as a DDoS attack [6].

3.5 Simulation of Known and Potential Future Botnet C2 Traffic

A newer and currently less active field of botnet research is aimed at the specific desire to create a proactive network defence against current and potential new botnets by developing botnet command and control frameworks. These frameworks can be used by analysts to construct new, unknown botnets and characterize their C2 traffic in order to develop detection or mitigation strategies before similar versions of these bots are designed by botmasters and deployed onto the Internet.

4 Survey of Botnet Command and Control Analysis Research

Sophisticated botnets are such large, widely distributed networks that are modular, implementation specific and polymorhpic that it is not possible to conduct focused research on all facets of botnets as a single topic. Thus, research teams generally focus on a subset of the botnet C2 feature space while leveraging advances shown other related research to help create an understanding of the "big picture" of the botnet problem [6]. As research advances, breakthroughs can be applied to real-world detection and mitigation tools and strategies aimed at reducing the impact of botnet proliferation on the Internet.

4.1 Indepth Analysis of a Known Botnet's C2 Traffic

Several recent papers have focused on in-depth analysis, reverse engineering and inference of a specific C2 architecture and protocols [3, 4, 9]. Mega-D is a well-known and widespread spam botnet that uses a HTTP-based command and control protocol that was initially reverse-engineered by Dispatcher [9]. Dispatcher is a tool for automatic reverse-engineering of C2 protocols used by botnets, and proved that the Mega-D botnet used an encrypted, non-SSL protocol on port 443; documenting 15 messages used to manage the botnet. Using the message protocol reverse-engineering from Dispatcher, researchers were able to infiltrate the Mega-D botnet, study its C2 architecture and observe the FireEye, Inc attempted takedown of the Mega-D botnet [4].

A research team from the University of California was able to use the C2 protocol reverse-engineering from Dispatcher to analyze and infer the complete communications state-machine for Mega-D C2 including back-channel communications between C2 servers that are not observable by network security analysts by querying the botnet infrastructure to build the communications model [3].

Future work in this area of research is focused on providing a tool and set of inference algorithms that would allow network security/defence analysts to rewrite or disrupt the normal flow of botnet C2 in order to provide greater protection of network and data assets against a wide array of botnet exploits.

4.2 Analyzing Large Volumes of Network Traffic Containing Botnet C2 Connections

BotGrep [2] is an algorithm that is designed to pinpoint efficient peer-to-peer (P2P) traffic in order to focus in on P2P botnet command and control network traffic that is part of the exteremely high volume of traffic that crosses an ISP or enterprise network backbone. BotGrep relies on applying filtering and clustering algorithms on large sets of communications graphs $G = (V, E)$ where V represents the set of hosts and E represents the communications between hosts. The premise behind BotGrep is that many recent botnets use efficient P2P protocols such as Kademlia [24] to implement C2 communications to rapidly disseminate information to all bots in the botnet ($fast\text{-}mixing$). This $fast\text{-}mixing$ characteristic, where *random walks* of a communications graph display very short convergence times, allows the BotGrep algorithm to filter out likely botnet hosts from huge amounts of Internet traffic and provide indicators to network security analysts of likely candidate bots in their networks. One weakness of BotGrep is that it cannot itself completely verify an infected host and analysts must further check the host with other types of investigative tools; however BotGrep can shorten the analysis of a network by providing a target list of potential bots. The authors claimed a 93-99% detection rate for bots in their tested scenarios.

Future work on the BotGrep advances are likely aimed at using the inferred topologies to sever the bot from the botnet by interfering/dropping critical communication links with the C2 structure [2].

4.3 Virtual Environment Bot Analysis

There are many types of malware analysis environments that are called *honeypots* and a well-known paper on Honeynet Project [18] describes a distributed system of honeypots that are used for related analysis. The basic concept of these types of systems is to trap malware in an isolated virtual network and observe the actions and communications produced by the bot malware in an attempt to connect with its expected C2 channel and conduct the operations for which it was designed. A considerable amount of information about specific bots has been obtained using virtual environments, however newer bots are often designed to detect the honeypot and may alter their actions to appear harmless or conduct no activity at all which hampers the value of the analysis in many cases.

JACKSTRAWS [1] uses machine learning techniques to profile behaviour graphs that detail information gained from host-based information that matches a host's system calls, network connections and data transactions. The authors used samples of pre-classified, network-active botnet malware, proven command and control detection signatures, and signatures that detect known harmless traffic as the basis for their research. The approach for JACKSTRAWS is to deploy malware binaries in a dynamic malware testing environment such as Anubis [22] or BitBlaze [23] and allow the malware to execute while the testing environment captures related system calls, data flow and network connections. Using signatures for known benign and malicious C2 traffic the researchers are able to categorize the behaviour graphs into botnet and *non*-botnet related connections. There is one behaviour graph for each network connection where nodes are system calls and directed edges capture data dependencies between nodes (system calls), e.g. a malware binary calls a system function that returns the operating system's ID number and then sends that ID number across a network connection to a command and control node on the Internet.

The sets of known botnet behaviour graphs are mined using machine learning graph mining algorithms to identify frequently appearing subgraphs in the the botnet C2 behaviour graphs. These subgraphs are then compared against the graphs of known harmless connections; retaining only those botnet C2 subgraphs that are not isomorphic (subgraphs do not appear in the set of benign connection graphs) [1]. The retained botnet C2 subgraphs are then clustered based on similar activities such as sending system information to a C2 host. The final step involves generating command and control detection templates by calculating a *weighted minimal common supergraph* for each cluster of botnet C2 behaviour subgraphs; essentially these are like behaviour graphs with system call nodes and data dependency edges where the template's edges and nodes are present in every behavour subgraph in a given cluster. The detection templates can then be used to match against databases of unknown behaviour graphs and identify connections related to previously known and also not yet identified bots with a very low false positive rate. The IPs and domains associated with the detected C2 connections can then be used by network security/defence analysts to deploy countermeasures (or network blocks) to prevent further botnet command and control communications on these channels.

JACKSTRAWS has some specific advantages over many techniques in that it is not affected by the encryption of C2 traffic as it uses internal host information to create detection templates and conduct analysis, not the encrypted traffic itself; also JACKSTRAWS is able to data mine for only botnet C2 and is not unduly impacted by the many harmless connections the malware may create to obfuscate its purpose [1]. However, some weaknesses are also present in that bot binaries that do not show normal activity in a testing environment can prevent the capture of truly representative behaviour graphs and affect the analysis. The authors claim a 81.6% successful detection rate of real botnet C2 activity with a very low 0.2% false positive rate.

Malware that is inactive in a virtual testing environment may provide few clues as to its purpose, capabilities and C2 protocols. The ability to create a so called "smart honeynet" where the analysts could inject C2 traffic expected by the bot could overcome some of the challenges in this type of botnet analysis. No current work on this problem could be found at the time of writing and should be considered as a topic of future research for applications in practical network defence forensics.

4.4 Active Interaction with a Live Botnet

This field of research can provide excellent insights into the purpose, architectures, and protocols of botnet C2. Generally, infiltration of a live botnet requires a research team to allow an Internet connected host to become a member of an active botnet or become the *defacto* botmaster for an active botnet, such as [15, 4] and respectively [13]. However there are legal and ethical considerations for this type of research into botnet C2 that must be weighed along with the potential for the research network to become a target for attack by the actual botmaster if the research activity is discovered. Another recent paper [6] has approached this problem from a different perspective, where researchers deployed a known botnet, *Waldec*, in an isolated network environment in order to study in depth an entire botnet (C2 servers, P2P-based C2 traffic, spammer bots and repeater bots) during its lifecycle including under stress from takedown attempts.

A considerable amount of new information was gained about this particular botnet including the reason for design compromises that had to be made by the botmaster due to the complexities and challenges of managing a large, distributed bot army. One such new insight was the challenge of computing new session keys for every C2 transaction by the C2 server which appears to have led to the botmaster reusing the same session key for all bots for nearly a year in the actual Waldec botnet [10].

Another recent paper conducted a complete takeover of the Torpig botnet by registering domains that the Torpig command and control protocol would direct bots to contact the research team configured their server to mimic legitimate C2 commands to the botnet and were able to control the botnet for ten days and derive an extensive amount of analysis on the entire system [13].

The ethical and legal challenges of the research focus is likely to become more problematic as governments attempt to tighten rules surrounding the control/participation in botnet activities outside of foreign intelligence activities. Future work on live botnets may move in the direction of isolated, live malicious bot analysis.

4.5 Attempting to Predict the Future

The desire on the part of network security and defence analysts to deploy countermeasures against the problem of botnets before an infection occurs poses some interesting challenges about bots that have not yet been identified "in the wild". SLINGBot (**S**ystem for **L**ive **I**nvestigation of **N**ext **G**eneration **bot**s)

[14] proposes a framework that enables researchers to develop harmless botnets that generate realistic C2 communications while providing logging and monitoring of the deployed bots and their C2 traffic. These bots are deployed in a testing environment and their communications can be captured along with non-botnet related traffic for further analysis. A SLINGbot constructed botnet can be used to explore anomalies in botnet communications through a full lifecycle of infection from initial propagation, through consolidation, and control of a botnet campaign scenario. A configurable scenario driver controls the life stages of the botnet and supported C2 methods include HTTP, IRC, distributed P2P, TinyP2P, Kademlia distributed hash table P2P [24] and hierarchical variations on these.

A SLINGbot generated botnet, based on the TinyP2P C2 protocol, was the basis of research into the periodic nature of botnet communications [11] due to programming in bots that establishes the time period between checks for new commands or management control instructions. As the SLINGbot generated botnet can be configured by researchers to display new, unseen botnet C2 traffic, this research provided some interesting insights for future work that may advance botnet detection algorithms for real-world use by providing some clues (*heuristics*) before a previously undetected class of similar bots is deployed by a botmaster onto the Internet.

5 Conclusions and Possible Future Work

In this paper, I have provided an overview of bot infected network hosts that are under the control of a botmaster which have become a significant and growing security risk on the Internet. The discussion of the challenges of the "arms race" between botmasters and security researchers provided background for a survey of state-of-the-art research into several aspects of botnet analysis targeting botnet command and control communications. Based on the latest works reviewed in this paper, I can conclude that while considerable advancements have been made into our understanding of botnet control strategies and the C2 architectures that facilitate them, the overwhelming advantage currently resides with the botmasters. However, there are many descriptions of future work to advance the field as mentioned in the subsections of Section 4 of this paper.

My focus for future work may be related the topic of "smart honeynets" as described in Section 4.3 of this report.

References

1. G. Jacob, R. Hund, C. Kruegel, T. Holz: JACKSTRAWS: Picking Command and Control Connections from Bot Traffic. USENIX Security Symposium 2011.
2. S. Nagaraja, P. Mittal, C.Hong, M. Caesar, N. Borisov: BotGrep: Finding P2P Bots with Structured Graph Analysis. USENIX Security 2010.
3. C. Y. Cho, D. Babić, E. C. R. Shin, D. Song: Inference and Analysis of Formal Models of Botnet Command and Control Protocols. ACM CCS 2010.

4. C. Y. Cho, J. Caballero, C. Grier, V. Paxson, D. Song: Insights from the Inside: A View of Botnet Management from Infiltration. USENIX LEET 2010.
5. C. Nunnery, G. Sinclair, B. B. Kang: Tumbling Down the Rabbit Hole: Exploring the Idiosyncrasies of Botmaster Systems in a Multi-Tier Botnet Infrastructure. USENIX LEET 2010.
6. J. Calvet, C. R. Davis, J. M. Fernandez, J. Marion, P. St-Onge, W. Guizani, P. Bureau, A. Somayaji: The case for in-the-lab botnet experimentation: creating and taking down a 3000-node botnet. ACSAC 2010.
7. L. Liu, S. Chen: Malyzer: Defeating Anti-detection for Application-Level Malware Analysis. ACNS, 2009.
8. J. John, A. Moshchuk, S. D. Gribble, A. Krishnamurthy: Studying Spamming Botnets Using Botlab. USENIX NSDI 2009.
9. J. Caballero, P. Poosankam, C. Kreibich, D. Song: Dispatcher: Enabling Active Botnet Infiltration using Automatic Protocol Reverse-Engineering. ACM CCS 2009.
10. J. Calvet, P. Bureau: Malware Authors Dont Learn, and Thats Good! *IEEE MALWARE* 2009.
11. B. AaSadhan, J. M. F. Moura, D. Lapsley: Periodic Behavior in Botnet Command and Control Channels Traffic. IEEE GLOBECOM 2009.
12. G. Gu, V. Yegneswaran, P. Porras, J. Stoll, W. Lee: Active Botnet Probing to Identify Obscure Command and Control Channels. IEEE ASCAC 2009.
13. B. Stone-Gross, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, G. Vigna: Your Botnet is My Botnet: Analysis of a Botnet Takeover. ACM CCS 2009.
14. A. W. Jackson, D. Lapsley, C. Jones, M. Zatko, C. Golubitsky, W. T. Strayer: SLINGBot: A System for Live Investigation of Next Generation Botnets. IEEE CATCH 2009.
15. B. Stock, J. Goebel, M. Engelberth, F. Freiling, T. Holz: Walowdac Analysis of a Peer-to-Peer Botnet. *EC2ND*, 2009.
16. TrendMicro: Taxonomy of botnet threats. Technical Paper, 2006.
17. SANS Institute, InfoSec Reading Room: Bots & Botnet: An Overview available at *www.sans.org*. 2003.
18. L. Spitzner: The Honeynet Project: Trapping the Hackers. *IEEE Security and Privacy*. 2003.
19. Cisco Systems Inc., What is network security? available at *www.cisco.com*
20. Dept of Homeland Security: The National Strategy for Secure Cyberspace available at *http://www.dhs.gov/files/publications/cybersecurity.shtm* 2003.
21. Carnegie Mellon University, US-CERT: Introduction to Information Security. 2008.
22. Anubis: Analyzing Unknown Binaries available at *http://anubis.iseclab.org/*
23. BitBlaze: Binary Analysis for Computer Security available at *http://bitblaze.cs.berkeley.edu*
24. P. Maymounkov, D. Mazières: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. *Proceedings of Peer to Peer Systems Workshop*. 2002.