Botnet Research Survey

Zhaosheng Zhu
Northwestern Univ.
zzh321@cs.northwestern.edu

Guohan Lu
Tsinghua Univ.
lguohan@gmail.com

Yan Chen
Northwestern Univ.
ychen@northwestern.edu

Zhi Judy Fu
Motorola Labs
Judy.fu@motorola.com

Phil Roberts
Internet Society
roberts@isoc.org

Keesook Han
AFRL
Keesook.Han@rl.af.mil

Abstract

Botnets are emerging threat with hundreds of millions of computers infected. A study shows that about 40% of all computers connected to the internet in the world are infected bots and controlled by attackers([2]). This article is a survey of recent advances in botnet research. The survey classifies the botnet research into three areas: understanding botnets, detecting and tracking botnets, and defending against botnets. While botnets are widespread, the research and solutions for botnets are still in their infancy. The paper also summarizes the existing research and proposes future directions for botnet research.

1 Introduction

According to the explanation in [1], botnet is a term for a collection of software robots, or bots, which run autonomously and automatically. They run on groups of zombie computers controlled remotely by attackers. A typical bot can be created and maintained in four phases.

1. *Initial Infection*: A computer can be infected in several different ways. For example, 1) Being actively exploited. The host has some vulnerability (e.g. DCE-RPC). A malicious program then exploits the vulnerability and runs on the host. 2) Malware was automatically downloaded while viewing web pages. 3) Malware was automatically downloaded and executed through opening an email attachment. 4) USB autorun.
2. *Secondary Injection*: In this phase, the infected hosts download and run the bot code, then become a real bot. The download can be via `ftp`, `http` and `P2P` (e.g., `Trojan.Peacomm`) as discussed in § 2.1.
3. *Malicious Activities*: The bot communicates to its controller to get commands/instructions for conducting activities such as spam, DDoS and scanning. Currently a more sophisticated technique called fast-flux service networks are gaining popularity(§ 2.1.4). The command communication can be IRC-based, HTTP-based, DNS-based or using P2P protocol to avoid single point of failure.
4. *Maintenance and Upgrade*: The bot continuously upgrades its binary in this phase.

Botnets are always classified according to their command and control architecture. For example, those who use the Internet Relay Chat (IRC) protocol are known as IRC based botnets.

We classify current botnet research into three areas: understanding botnets, detecting and tracking botnets, and countering against botnets. We will discuss them respectively in the subsequent sections.

2 Understanding Botnet

Most current research focuses on understanding botnets. There are mainly three types of papers in this area.

- *Bot Anatomy*: The papers in this category provide extensive analysis of a specific kind of bot for case study. The analysis mainly focuses on its network level behavior, usually involving the use of binary analysis tools.
- *Wide-area Measurement Study*: The second group of papers provides measurement studies through tracking botnets to reveal different aspects of botnets in the internet, such as botnet size, traffic generated, their usages and dynamics. Currently only IRC-based botnets have been studied.
- *Botnet Modeling and Future Botnet Prediction*: The third group of papers discusses the theoretical model-

0730-3157/08 $25.00 © 2008 IEEE
DOI 10.1109/COMPSAC.2008.205

ing of botnets, the possible future evolution of botnets and countermeasures against them.

We will describe each in the following subsections.

2.1 Bot Anatomy

2.1.1 IRC Bot

In [6], it analyzed the source code for four bots, Agobot, SDBot, SpyBot and GT bot, which are all IRC-based bots. Among these botnets, only Agobot is a fully-developed bot, and the other three are like toys. Agobot has provided the following five features.

- *Exploits*: It can exploit many well known OS vulnerabilities (e.g. buffer overflow) and back doors left by other viruses.
- *Delivery*: It separates exploits and delivery. Once the first step exploits succeed, it opens a shell on the remote host to download bot binary. The binary is encoded to avoid network-based signature detection.
- *Deception*: The bot has the module to test for debuggers (e.g. SoftIce) and VMWare once it is installed. If it detected VMWare it stopped running. So VMWare-based Honeypot cannot run Agobot.
- *Function*: It can steal system information and monitor local network traffic.
- *Recruiting*: It recruits using botmaster controlled horizontal and vertical scannings.

Although using direct source analysis can give us a clear insight about a bot, this approach is quite limited. The biggest problem is that most bots do not have source code available. Therefore, more sophisticated methods, for example, system-level analysis and networking-level analysis for the botnet behavior are needed.

2.1.2 HTTP Bot

It analyzed the binary of an HTTP-based spam bot module in Rustock rootkit([8]). The command and control (C&C) is http based. To ensure the anonymity, the communication channel is encrypted. In this paper a binary analysis tool IDA Pro is used to analyze the binary and find the encryption key. The paper summarizes that a typical process for the spam bot to send a spam is as following.

1. The bot asks the controller for local processes/files to kill and delete.
2. The controller sends back system information.
3. The bot asks for SMTP servers.
4. The bot gets failure responses from the SMTP servers.
5. The bot gets spam message
6. The bot gets target email addresses.

In [21], the author described an HTTP-based DDoS bot, `BlackEnergy`. The bot is only used for DDoS attacks. However, the bot does not have any exploit activities, so it cannot be captured by Honeynet. The paper only described the commands used in C&C, but did not describe their method to obtain samples. Once a sample is captured, the botmaster can be tracked.

In [12], it discussed `Clickbot.A`, a low-noise click fraud bot. The client is propagated via email attachment. The botnet also uses HTTP protocol for their command and control. The paper provided the source code of its botmaster, written in PHP. The paper discussed the details of its click fraud process. It is also shown that the client participating in click fraud also sends spams, implying that the client performs multi-tasks.

2.1.3 P2P Bot

The author claims that centralized control of botnets offers a single point of failure for the botnet([15]). So more stable architectures, like P2P based architecture, will be used by botnet operators. And it analyzes one case study: Trojan.Peacomm with binary analysis. The author captures this binary using Honeypot. The analysis is mainly based on blackbox techniques. They only discuss the network activities of an infected host, but did not perform binary analysis of the code. In their paper they found the P2P technology (Kademlia algorithm) is used to get the URL to download real bot binary in the secondary injections discussed in § 1.

In [19], the author analyzes one of the most widespread P2P botnets by analyzing the binary and networking traces. Also it proposes some techniques, for example, Eclipsing Content and Polluting the file, to disrupt the communication of botnet P2P networks.

2.1.4 Fast-flux Networks

We have mentioned that fast-flux networks are increasingly used as botnets Command and control networks. There are many servers in the blackhat circles, such as the phishing websites. These websites are valuable assets of them, so they really want to hide their IP addresses from outsiders. In order to achieve such a goal, they let a user first connect to a compromised computer, which serves as a proxy, to forward the user requests to a real server and the response from the server to the user.

In [4] it introduces a new type of techniques called Fast-flux service networks for this purpose. The DNS records

of a real website point to the computers of Fast-flux networks. The network uses a combination of round-robin IP addresses and a very short Time-To-Live (TTL) for any given particular DNS Resource Record (RR) to distribute a user's request to a large number of compromised computers.

The Fast-flux motherships are the controlling parts of the fast-flux service networks. It is very similar to the command and control (C&C) systems found in conventional botnets but provides more features. It is observed that these nodes are always hosting both DNS and HTTP services, for being able to manage the content availability for thousands of domains simultaneously on a single host.

This paper also presents a case-study for one specific fast-flux network. They collected information on the IP addresses assigned to the domain name and how those IP addresses (A and NS records) changed over time. They then did some statistical analysis, for example, the distribution of AS Breakdown for DNS Flux Networks. They found lots of compromised computers were involved. There are altogether 3,241 unique IP addresses. 1,516 were advertised as NS records, while 2,844 were short lived TTL used for HTTP proxy. The above result is only one example, altogether they monitored 80,000 flux IPs with over 1.2million unique mappings.

2.2 Wide-area Measurement Study

It presents a honeynet-based botnet detection system as well as some findings on botnets across the Internet([24]). The systems are composed of three module.

1. *malware collection*: use a lightweight responder `nepenthes` and unpatched WindowsXP in a virtualized environment.
2. *Graybox testing*: to learn botnet "dialect".
3. *Botnets tracking*: an IRC tracker (*drone*) to lurk in IRC channel and record commands. DNS tracking, a novel method to estimate botnet size using DNS cache.

For data collection, it deploys a modified version of the nepenthes platform in darknet to collect malware. To complement the role of nepenthes, it also uses Honeynet in which the honeypots are running unpatched instances of Windows XP in a virtualized environment.

There are also several interesting findings in this paper.

- Botnet scanning traffic containing large percentage of Internet background radiation.
- Most of botnet scanning behavior is well-controlled by its commander.
- 90% bots stay in IRC channel for less than 50 minutes.
- Over 80% bots are generally detected by Anti-virus software, e.g. `Norton`.
- Small botnets receive a larger portion of control and mining commands. Large botnets have a larger percentage of cloning and downloading commands (DDoS).

In [26] it mainly characterizes the network-level behavior of spammers. For example, (1) IP address, AS and country of spammers. (2) The characteristics of spamming botnets. To identify a set of hosts that are sending email from botnets, they used a trace of hosts infected by the W32/Bobax (.Bobax.) worm from April 28-29, 2005. And based on the findings, it suggests developing algorithms to identify botnet membership based on network-level properties.

Several papers are developing methods to reveal more properties of botnets, for example, to estimate botnet sizes, either its *footprint* or *live population*. The following are some existing work.

1. *Botnet Infiltration*: In [13] it lets a *drone* to join the botnet and record joining bot information on the channel. However only 52% of the botnets they tracked make bot join information available. A well-developed botnet will surely not make such information available. Moreover, this method is solely IRC-based.
2. *DNS Redirection*: In [10] it counts infected bots by manipulating the DNS entry associated with a botnet's IRC server and redirecting connections to a local sinkhole. However, it can only count bots which issue DNS requests to this DNS server.
3. *DNSBL* In [27] it monitors lookups to a DNS-based blackhole list to expose botnet membership. However, it does not reveal which botnet the bot belongs to, and only applies when the bots are used to send spam.
4. *DNS Cache*: In [24] it uses DNS cache snooping to uncover a botnet's footprint. However, the result is just a lower bound on its true DNS footprint and subjected to three problems. For example, a cache hit was recorded only if a bot made a lookup query to its local DNS server.

However, to estimate the botnet size is still a problem. In [25], the author points out that several issues may make counting botnet memberships more complicate. For example, temporary bot mitigation and bot cloning. It suggests synthesizing the results from multiple independent views of a botnet's behavior.

2.3 Botnet Modeling and Future Botnet Prediction

There are also several papers on modeling botnets. It creates a diurnal propagation model based on the fact that computers that are offline are not infectious, and any regional bias in infections will affect the overall growth of the botnet([11]). Realizing that the trend to small botnets may be more dangerous than big botnets, in [28] it proposes a superbot model that the botnets are designed to be coordinated into a network of botnets. [29] discusses an advanced botnet which considers the following challenges:

1. How to generate a robust botnet even though some bots are removed?
2. How to prevent significant exposure of the network topology even though some bots are detected?
3. How to easily monitor and obtain the complete information of a botnet by its botmaster?
4. How to prevent (or make it harder) defenders from detecting bots via their communication traffic patterns?

So it proposes to use a hybrid P2P botnet instead of pure P2P structure to improve the stability of a botnet. Traditional C&C botnet uses one or two hosts as central controllers. Because the controllers of a botnet can be easily identified and shut down once one of the bots has been identified, the paper suggests using some bots as botnet controllers (*servant bots*), which resembles the super node in current P2P network.

In [9] it mainly discusses the botnet structures based on their utilities to botmasters. One conclusion shows that random graph botnets (e.g., those using P2P formations) are highly resistant to both random and targeted responses.

Although there are several papers on the modeling of botnet, we still have no idea how close these models are to the botnets in the real world. More accurate models may help us get more knowledge about botnet and give a better prediction to the development of botnet.

3 Detecting and Tracking Botnet

There are mainly two approaches of botnet detection and tracking methods. One is honeynet based method and the other is based on passive traffic monitoring.

3.1 Honeynet

There are many papers [23, 24] discussed how to track botnet using Honeynet, and how to use tools to collect malware [5]. In [22], Jose Nazario from Arbor Networks discusses several challenges in developing a botnet tracking tool. In summary, first, there are several tools available to collect malware, but no tool for tracking the botnet. Secondly, the tracking tool needs to understand the botnet's "jargon" in order to be accepted by the botmaster. Moreover, the increasing use of anti-analysis techniques used by the blackhat circle makes the development of the tool even more challenging.

3.2 Traffic Monitoring

In [20] it described a network-wide system to identify botmasters based on transport layer flow information. It gathers traffic flow information from many vantage points within the network. The core idea is based on the attack and control chain of the botnet. The major steps are listed as follows:

1. Identify bots based on their attack activities, such as scanning, emailing of spam and viruses, or DDoS traffic generation. The activities are reported by other security system.
2. Analyze the flows of these bots to find candidate controller connections (CCC).
3. Analyze the CCC to locate the botmaster.

This paper also gives us some interesting results. For example, based on the long-time observation, it estimates the bot stays 2-3 days on the same controller in average. In [14] it presented a passive monitoring system (*Rishi*) to track botnets based on the bots' IRC nicknames. The core idea is that the format of nicknames used by the bots is different from that of a normal user, e.g. `USA|016887436` is a typical nickname used by the bots. The author uses regular expression for the detection. The system is deployed on a border router of a campus network running two weeks, and here are their findings:

- Results are compared with their NIDS system (*Blast-o-Mat*). 82 bots were detected while only 34 were detected by *Blast-o-Mat*. *Blast-o-Mat* detected 20 hosts which were not picked up by *Rishi*.
- None of the botnets uses port traditional IRC port 6667 for C&C.

However, this approach is quite limited. For example, IRC Nickname can be changed to resemble normal user. And it can not detect HTTP botnet, or the botnet of which the communication is encrypted, e.g. Rustock mentioned in § 2.1.

The following are two more advanced detection tools. A BotHunter system is presented which consists of a correlation engine that is driven by three malware-focused network packet sensors, each charged with detecting specific

stages of the malware infection process([17]). It finds the suspicious flows which match BotHunter's infection dialog model. Based on the observation that bots within the same botnet will likely have spatial-temporal correlation and similarity, it proposes using network-based anomaly detection to identify botnet C&C channels([18]).

The most recent work appears in [16]. In this paper presents classifying networking traffic to detect botnet, which is independent of the botnet protocol and structure.

4 Defenses Against Botnet

Unfortunately, only a few papers proposed defense technologies against botnet. The most effective way is to shutdown the botmaster once we identify it. However, this task is far from trivial. The following discusses the defense and some practical issues with this approach.

4.1 Spam

In [7] it proposed a distributed, content independent spam classification system to defend from botnet generated spams. A little bit unexpected, the system does not utilize previous botnet detection results to ban emails generated by bots. The basic idea of the system is that "*A host that has recently sent large amounts of e-mails may be a spam-bot. Consequently, any e-mail coming from such hosts is potentially spam, and if the source has a dynamically allocated IP address (or simply a dynamic IP address) and the sender is not in the recipient's address book or list of past recipients or senders, then it is almost certain that the e-mail is spam.*"

The system consists of following parts:

1. Identifying the source of emails
2. Keeping track of how many emails were recently sent by a source
3. Disseminating this information for the purposes of classifying future emails.

The effectiveness of this system is unknown since it is still in the process of Implement.

4.2 Enterprise Solutions

Trend Micro provided Botnet Identification Service([3]). The company provide the customers the real-time botnet C&C botmaster address list via BGP peering between Trend Micro BIS router and the customers' BGP border router. This service charges 9 cents per user for 500,000 users. However, *Fast-Flux* networks can make Trend Micro's solution much less effective.

5 Conclusion and Possible Future Work

While botnets are widespread, the botnet research is still in its infancy. This paper surveys state-of-art botnet research that can be categorized into three areas, i.e. understanding botnet, detecting & tracking botnets, and countering against botnets. In understanding botnet research, it is proposed to learn botnet behaviors and characteristics through source code analysis, binary analysis or wide area measurement. Some formal models are also proposed to predict botnet advancement. In detecting & tracking botnet researches, honeynet and traffic monitoring approaches are proposed to detect botnets based on some of their unique behaviors. Finally, the research on defending against botnet proposes to simply shut down botmaster after they are identified.

Those current botnet study is still in a preliminary stage. Previous analysis shows that majority of botnet traditionally used IRC for their command and control. But we believe the botnets will advance to new communication architectures, for example, P2P-based botnet. And currently the defense against botnet is not very efficient, so much more work needs to be done in this field. Finally future botnet prediction may give us an advanced view of the botnet development. Good model can help people know the properties of botnet and thus control it. The following are some topics for possible future work.

5.1 HTTP/P2P Botnet

IRC-based botnet has been studied extensively in recent years. The research on the other two kinds of botnet has just begun. The existing works are anatomy of some samples. Their network behaviors have been rarely studied, not to mention the number of botnet and the number of infected hosts. It's likely that more HTTP and P2P botnets will appear in the near future, so we need to pay more attention to them.

5.2 Fast-flux Network

There are still many unknown behaviors of *fast-flux* networks. How many of them are there? Who do them serve? What's the structure of its network? Is it the same as a typical IRC botnet or not? Is their botmaster also fast-fluxed? The binary analysis of its code will be extremely helpful.

References

[1] http://en.wikipedia.org/wiki/Botnet.

[2] Botnet scams are exploding. http://www.contentagenda.com/articleXml/LN760999245.html?industryid=45177.

[3] Trend micro botnet identification service. http://us.trendmicro.com/us/products/enterprise/botnet-identification-service/.

[4] T. H. P. R. Alliance. Know your enemy: Fast-flux service networks an ever changing enemy. http://www.honeynet.org/papers/ff/fast-flux.html, 2007.

[5] P. Baecher, T. Holz, M. Kotter, and G. Wicherski. The malware collection tool (mwcollect). http://www.mwcollect.org.

[6] P. Barford and V. Yegneswaran. An inside look at botnets. In *Special Workshop on Malware Detection Advances in Information Security*. Springer Verlag, 2006.

[7] A. Brodsky and D. Brodsky. A distributed content independent method for spam detection. In *First workshop on hot topics in understanding botnets*, 2007.

[8] K. Chiang and L. Lloyd. A case study of the rustock rootkit and spam bot. In *First workshop on hot topics in understanding botnets*. Sandia National Laboratories, 2007.

[9] D. Dagon, G. Gu, C. P. Lee, and W. Lee. A taxonomy of botnet structures. In *the 23 Annual Computer Security Applications Conference (ACSAC'07)*, 2007.

[10] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *13th Network and Distributed System Security Symposium (NDSS)*, Feb 2006.

[11] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *NDSS*, 2006.

[12] N. Daswani, M. Stoppelman, the Google Click Quality, and S. Teams. The anatomy of clickbot.a. In *First workshop on hot topics in understanding botnets*, 2007.

[13] F. Freiling, T. Holz, and G. Wicherski. Botnet tracking: Exploring a root cause methodology to preventing denial-of-service attacks. In *10th European Symposium on Research in Computer Security, ESORICS*, Sept 2005.

[14] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation. In *First Workshop on Hot Topics in Understanding Botnets*, 2007.

[15] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: Overview and case study. In *First Workshop on Hot Topics in Understanding Botnets*, 2007.

[16] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *USENIX Security*, 2008.

[17] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *16th USENIX Security Symposium*, 2007.

[18] G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic. In *NDSS*, 2008.

[19] T. Holz, M. Steiner, F. Dahl, E. W. Biersack, and F. Freiling. Measurement and mitigation of peer-to-peer-based botnets: A case study on storm worm. In *LEET*, 2008.

[20] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *First Workshop on Hot Topics in Understanding Botnets*, 2007.

[21] J. Nazario. Blackenergy ddos bot analysis. Technical report, Arbor Networks, Oct 2007.

[22] J. Nazario. Botnet tracking: Tools, techniques, and lesson learned. Technical report, Arbor Networks, 2007.

[23] H. Project and R. Alliance. Know your enemy: Tracking botnets. http://www.honeynet.org/papers/bots/, 2005.

[24] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *IMC*, 2006.

[25] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. In *First Workshop on Hot Topics in Understanding Botnets*, 2007.

[26] A. Ramachandran and N. Feamster. Understanding the network?-level behavior of spammers. In *Sigcomm*, 2006.

[27] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. In *2nd Workshop of Steps to Reducing Unwanted Traffic on The Internet (SRUTI)*, July 2006.

[28] R. Vogt, J. Aycock, M. J. Jacobson, and Jr. Army of botnets. In *NDSS*, 2007.

[29] P. Wang, S. Sparks, and C. C. Zou. An advanced hybrid peer-to-peer botnet. In *First Workshop on Hot Topics in Understanding Botnets*. University of Central Florida, 2007.