Botnet Construction and Monitoring Using Honeypot, Based on Network Security

[1]Rajeswari.M, [2]Bhavani.A, [3]Kamaleswari.S.P
[1,2,3]Dept. of IT, Velammal Institute of Technology, Anna University, Chennai, Tamilnadu, India

Abstract
A "botnet" consists of a network of compromised computers controlled by an attacker ("botmaster"). Recently botnets have become the root cause of many Internet attacks. Especially machines with broadband connection that are always on are a valuable target for attackers. Crackers use it for their own advantage.
To be well prepared for future attacks, it is not enough to study how to detect and defend against the botnets that have appeared in the past. More importantly, we should study advanced botnet designs that could be developed by botmasters in the near future. In this paper, we present the design of an advanced hybrid peer-to-peer botnet. Compared with current botnets, the proposed botnet is harder to be shut down, monitored, and hijacked. It provides robust network connectivity, individualized encryption and control traffic dispersion, limited botnet exposure by each bot, and easy monitoring and recovery by its botmaster. Possible defenses against this advanced botnet are suggested.
In this paper we detect the attacker by using honeypot and also without using honeypot technique. "Honeypots" have been successfully deployed in many defense systems. A group of bots, referred to as a botnet, is remotely controllable by a server and can be used for sending spam mails, stealing personal information, and launching DDoS attacks.

Keywords
Botnet, Peer-to-Peer, Robustness, Honeypot

I. Introduction
Explosive growth of the Internet provides much improved accessibility to huge amount of valuable data. However, numerous vulnerabilities are exposed and the number of incidents is increasing over time. Especially, recent malicious attempts are different from old-fashioned threats, intended to get financial benefits through a large pool of compromised hosts. For example, they steals personal information which can lead significant financial losses and simultaneously, used for delivering spam mails, and launching DDoS (Distributed Denial of Service) attack.
A large pool of compromised hosts, called bots, communicate with a bot controller to coordinate the network of bots. Such a network is commonly referred to as a botnet. An attacker, called a botmaster, controls a botnet to perform various malicious activities.
One prominent characteristic of botnets is the use of command and control (C&C) channels. The main purpose of the channels is to deliver the commands of a botmaster. And today's botnets use the Internet Relay Chat (IRC) protocol [1], which is mainly designed for group communication in discussion forum called channels. But the channels are now used for the communication of a botnet among distributed bots and their controller.
Defending against botnets is a pressing problem that is still not well comprehended, though botnets first appeared several years ago. Former defense mechanisms focused on a particular symptom of bots or a signature of bot programs. Even though the studies were meaningful to develop better defense mechanisms, their approaches have intrinsic limits such as the ineffectiveness for detecting unknown bot programs which are a slight modification of an existing Bot program or newly generated bot programs. Recent studies such as [2] on Botnet measurements and their detection also have the same weakness for the variants of bot programs .E-mail spam, extortion through denial-of-service attacks [1], and click fraud [2] represent a few examples of this emerging trend. "Botnets" are a root cause of these problems [3-5].
Attackers even go a step further and bring different bots together. Such a structure, consisting of many compromised machines which can be managed from an IRC channel, is called a botnet. IRC is not the best solution since the communication between bots and their controllers is rather bloated, a simpler communication protocol would suffice.

II. Related Work
Researches have proved that in the last ten years, Internet users have been attacked unremittingly by widespread email viruses and Internet scanning worms. Some of the more devastating email viruses include Melissa in 1999, Love Letter in 2000, W32/Sircam in 2001, and MyDoom, Netsky and Bagle in2004, etc. Similarly damaging internet-scanning worms include CodeRed in 2001, Slammer, Blaster in 2003, Witty and Sassar in 2004 [6].
Strangely, we have not seen a major virus or worm outbreak since the Sassar worm incident in May 2004.. Instead, their attention has shifted to compromising and controlling victim computers, an attack scheme which provides more potential for personal profit. Turning our focus now to recent trends in computer security, honeypot is a general and effective attack detection technique.
A "honeypot" is a special constructed computer or network trap designed to attract and detect malicious attacks. In recent years, honeypots have become popular, and security researchers have generated many successful honeypot-based attack analysis and detection systems (such as [3, 9, 13, 17, 22, 29].) honeypot owners have liability constraints such that they cannot allow their honeypots to send out real attacks (or send out too many real attacks.) Botnet is a new trend in Internet attacks.
In 2003, Puri from SANS Institute [23] presented an overview of bots and botnets; McCarty [18] discussed how to use a honeynet to monitor botnets. Currently, there are two techniques to monitor botnet activities. The first technique is to allow honeypots or honeynets to be compromised and join in a botnet [7, 12, 21]. Behaving as normal "bots" in the botnet, these honeypot spies provide valuable information of the monitored botnet activities.

III. Existing Botnet
Our existing system used command – and – control server concepts. These C&C servers receive .Commands from their botmaster and forward them to the other bots in the network.

A. Flaws in Current Botnets
A weakness of traditional centralized botnet C&C is the single point of failure: an entire botnet can be dismantled by bringing down a single node (the C&C server). While using P2P for C&C provides some benefits in the form of resiliency to loss of single nodes , it also presents some (surrmountable/manageable) challenges such as latency and lack of reliability in message transmission. They

note the simultaneous though largely independent evolution of malicious bots and peer-to-peer protocols.

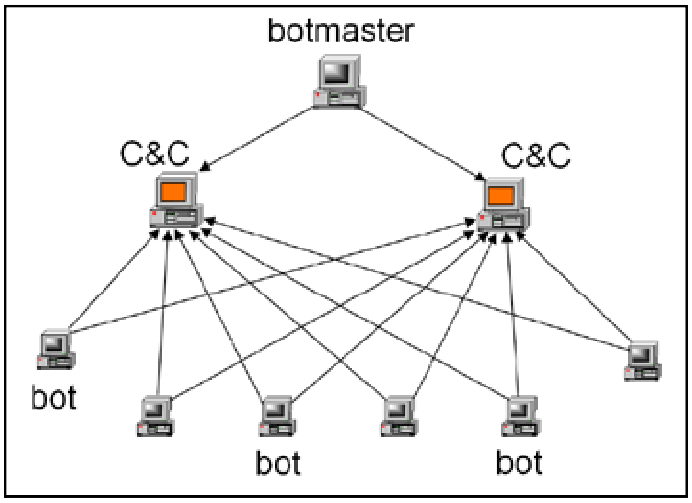

Fig. 1: Command and Control Architecture of a C&C Botnet

It is not expected that moving a botnet from a centralized to decentralized C&C topology will fundamentally alter how that botnet is used; it only affects the way in which messages -- commanding the bots to do X, Y, or Z -- are delivered to those bots.

A botmaster will lose control of her botnet once the limited number of C&C servers are shut down by defenders. Defenders could easily obtain the identities (IP addresses) of all C&C servers based on their service traffic to a large number of bots or simply from one single captured bot (which contains the list of C&C servers).

An entire botnet may be exposed once a C&C server in the botnet is hijacked or captured by defenders

IV. Hybrid P2P Botnets

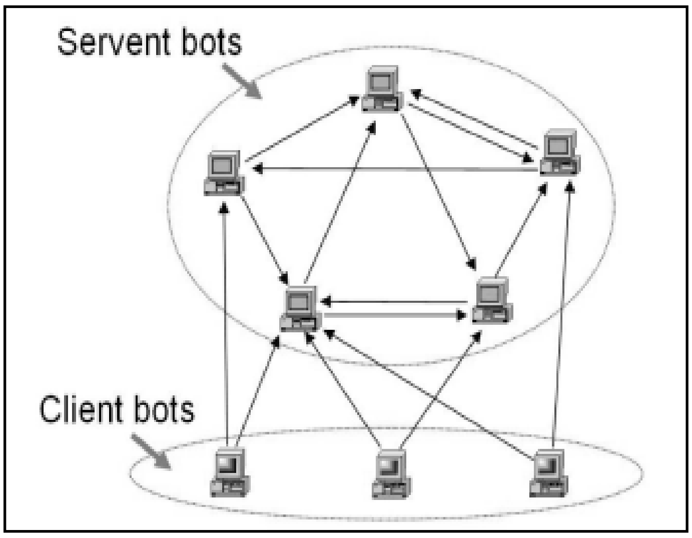

Fig. 2: Proposed Architecture

The lifetime of botnets is composed of three stages. Stage one - recruiting members, a botmaster needs to compromise many computers in the Internet, so that he/she can control them remotely. Stage two - forming the botnet, bots need to find a way to connect to each other and form a botnet. Stage three - standing by for instructions, after the botnet is built up, all bots are ready to communicate with their botmaster for further instructions.

A. Stage One: Recruiting Bot Members

P2P networks are gaining popularity of distributed applications, such as file-sharing, web caching, network storage [5]. In these content trading P2P networks, without a centralized authority it is not easy to guarantee that the file exchanged is not malicious. For this reason, these networks become the ideal venue for malicious software to spread. It is straightforward for attackers to target vulnerable hosts in existing P2P networks as bot candidates and build their zombie army.

Many P2P malware have been reported, such as Gnuman [6], VBS. Gnutella [6] and SdDrop [7]. On the contrary, P2P botnets we have witnessed recently [1-2, 8] do not confine themselves to existing P2P networks.

B. Stage Two: Forming the Botnet

Upon infection, the next important thing is to let newly compromised computers join the botnet network and connect to other bots. Otherwise, they are just isolated individual computers without much use for botmasters. Now for the convenience of further discussion, we first introduce three terms: “parasite”, “leeching” and “bot-only” P2P botnets. Each of them represents a class of P2P botnets. In a parasite P2P botnet, all the bots are selected from vulnerable hosts within an existing P2P network, and it uses this available P2P network for command and control. For example, the early version of Storm botnet [2] belongs to this class of botnet. A bot-only P2P botnet builds its own network, in which all the members are bots, such as Stormnet [2] and Nugache [8].

C. Stage Three: Standing by for Instructions

Once a botnet is built up, all the bots are standing by for instructions from their botmaster to perform illicit activities or updates. Therefore C&C mechanism is very important and is the major part of a botnet design. The C&C mechanisms can be categorized as either pull or push mechanism. Pull mechanism, i.e., “command publishing subscribing”, refers to the manner that bots retrieve commands actively from a place where botmasters publish commands.

On the contrary, push mechanism, i.e., “command forwarding”, means bots passively wait for commands to come and will forward received commands to others. For centralized botnets, pull mechanism is commonly used.

V. Honeypot

In computer terminology, a honeypot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. A honeypot is valuable as a surveillance and early-warning tool. While it is often a computer, a honeypot can take other forms, such as files or data records, or even unused IP address space. Honeypots should have no production value, and hence should not see any legitimate traffic or activity. Whatever they capture is therefore malicious or unauthorized.

One practical application of this is the spamtrap - a honeypot that thwarts spam by masquerading as a type of system abused by spammers.

Honeypots can help prevent attacks in several ways. The first is against automated attacks, such as worms or auto-rooters.

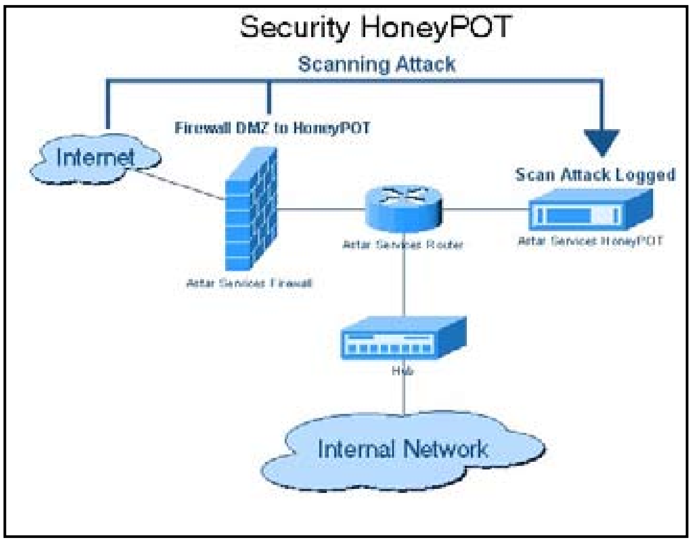

Fig. 3: Security Using Honeypot

These attacks are based on tools that randomly scan entire networks looking for vulnerable systems. If vulnerable systems are found, these automated tools will then attack and take over the system (with worms self-replicating, copying themselves to the victim).

A. Detecting Using Honeypot

One way that honeypots can help defend against such attacks is slowing their scanning down, potentially even stopping them. Called sticky honeypots, these solutions monitor unused IP space

1. Bot Master Node Implementation

We implement the server node. This node may send instruction to any other node. A bot master can monitor the other node. Bot Master maintain the detail about the bot. a botmaster issues a special command, called a report command, to the botnet, thereby instructing every bot to send its information to a specified machine that is compromised and controlled by the botmaster. This data collection machine is called a sensor.

2. Servent Bot Implementation

we implement the Servent Bot. Servent Bot contains bots that have static, non private IP addresses and are accessible from the global Internet. Bots in the first group are called servent bots since they behave as both clients and servers.

VI. Botnet Robustness Study

Let C(p) denote the connected ratio and D(p) denote the degree ratio after removing top p fraction of mostly-connected bots among those peer-list updating servent bots-this is the most efficient and aggressive defense that could be done when defenders have the complete knowledge (topology, bot IP addresses ...) of the botnet. C(p) and D(p) are defined as:

$$C(p) = \frac{\text{\# of bots in the largest connected graph}}{\text{\# of remaining bots}}$$

$$D(p) = \frac{\text{Average degree of the largest connected graph}}{\text{Average degree of the original botnet}}$$

The metric C(p) shows how well a botnet survives a defense action; the metric D(p) exhibits how densely the remaining botnet is connected together.

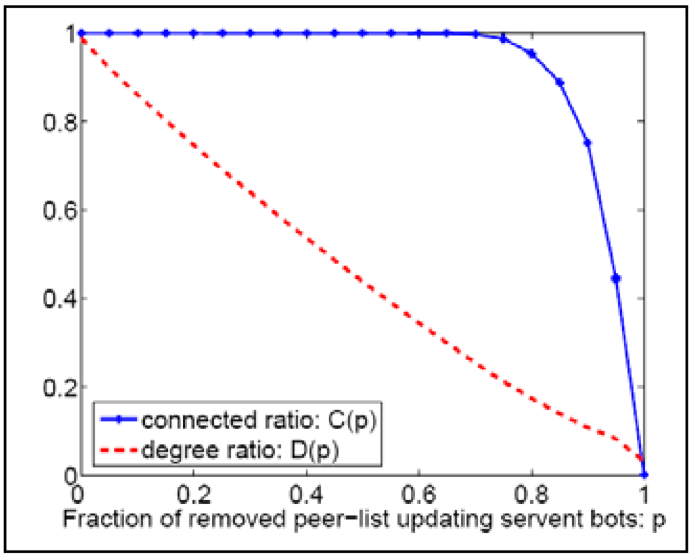

Fig. 4: Botnet Robustness Study

VII. Monitoring Via a Dynamically Changeable Sensor

To monitor the proposed hybrid P2P botnet, a botmaster issues a special command, called a report command, to the botnet thereby instructing every bot to send its information to a specified machine that is compromised and controlled by the botmaster. This data collection machine is called a sensor. The IP address (or domain name) of the centralized sensor host is specified in the report command. After a report command has been sent out by a botmaster, it is possible that defenders could quickly know the identity of the sensor host (e.g., through honeypot joining the botnet [3,6]), and then either shut it down or monitor the sensor host. To deal with this threat, a botmaster may implement any of the following procedures:

- Use several sensor machines instead of a single sensor.
- Select sensor hosts that are harder to be shut down or monitored, for example, compromised machines in other countries with minimum Internet security and International collaboration.
- Manually verify the selected sensor machines are not honeypots (see further discussion in Section 9).
- Wipe out the hard drive on a sensor host immediately after retrieving the report data.
- Specify expiration time in report command to prevent any bot exposing itself after that time.
- Issue another command to the botnet to cancel the previous report command once the botmaster knows that the sensor host has been captured by defenders.

If a botmaster simply wants to know the current size of a botnet, a probabilistic report would be preferred: each bot uses a small probability p specified in a report command to decide whether to report. Then the botnet has roughly X/p bots if X bots report. Such a probabilistic report could minimize the telltale traffic to the report sensor.

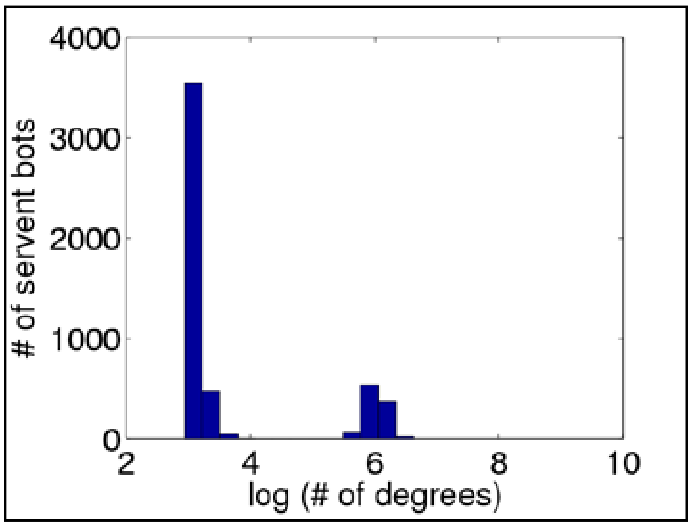

Fig. 5: Servent Bot Degree Distribution

Fig. 5 shows the degree distribution for servent bots (client bots always have a degree of M) when a botnet uses all three construction procedures. We assume the peer-list updating procedure is executed just once when 1,000 (25% of) servent bots have been infected. This figure shows that although those 4000 servent bots infected after the peer-list updating procedure still have small degrees, the first 1000 servent bots used in peer-list updating have large and balanced connection degrees, ranging from 300 to 500. They form the robust backbone, connecting the hybrid P2P botnet tightly together.

VIII. Conclusion

To be well prepared for future botnet attacks, we should study advanced botnet attack techniques that could be developed by botmasters in the near future. In this project, we present the design of an advanced hybrid P2P botnet. Compared with current botnets, the proposed one is harder to be monitored, and much harder to be shut down.

It provides robust network connectivity, individualized encryption and control traffic dispersion, limited botnet exposure by each captured bot, and easy monitoring and recovery by its botmaster. To defend against such an advanced botnet, we point out that honeypots may play an important role. We should, therefore, invest more research into determining how to deploy honeypots efficiently and avoid their exposure to botnets and botmasters.

References

[1] Honeyed security advisory 2004-001: Remote detection via simple probe packet, [Online] Available: http://www.honeyd.org/adv.2004-01.asc, 2004.

[2] X. Jiang, D. Xu. Collapsar,"A vm-based architecture for network attack detention center", In Proceedings of 13thUSENIX Security Symposium, August 2004.

[3] G. Kesidis, I. Hamadeh, S. Jiwasurat,"Coupled kermackm-ckendrick models for randomly scanning and bandwidth saturating internet worms", In Proceedings of 3rd InternationalWorkshop on QoS in Multiservice IP Networks (QoSIP, pp. 101–109, February 2005.

[4] C. C. Zou, L. Gao, W.Gong, D. Towsley,"Monitoring and early warning for Internet worms", In Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)", pp. 190–199, October 2003.

[5] CERT. CERT/CC advisories, [Online] Available: http://www.cert.org/advisories/.

[6] E. Cooke, F. Jahanian, D. McPherson,"The zombie roundup: Understanding, detecting, and disrupting botnets", In Proceedings of SRUTI: Steps to Reducing Unwanted Traffic on the Internet, July 2005.

[7] J. Corey.,"Advanced honey pot identification and exploitation", [Online] Available: http://www.phrack.org/fakes/p63/p63-0x09.txt, 2004