Self-Destructible Concentrated P2P Botnet

Mukesh Kumar, Pothula Sujatha, P. Manikandan, Madarapu Naresh Kumar, Chetana Sidige and Sunil Kumar Verma*

School of Engineering and Technology, Department of Computer Science
Pondicherry University-605014
Puducherry, INDIA
Indian Institute of Information Technology Allahabad, INDIA*

Abstract

Small botnets are tough to detect and easy to control by the botmaster. Having a small botnet with high speed internet connectivity than large but slow connection is more effective and dangerous in nature. According to diurnal dynamics studies only about 20 percent of computers are always online, to maximize a botnet attack power, botmaster should know diurnal dynamics of her botnet. In our project we are designing a peer-to-peer bot. This bot after infecting any of the system first check the internet connection speed of the interface, if it is not up to the desired speed i.e. 2 Mbps the bot will kill itself because slow speed bots are not desired. In another scenario bot will sense is it in a honeypot trap? If so it will kill itself so that the whole botnet could not be exposed to the defender. We will suggest the mitigation techniques to defend bots with these types of properties.

Keywords- *Peer-to-Peer, Botnet, Honey pot, Firepower*

1. Introduction

As technology for internet security matured Internet malware, and Ransom ware domination also increased. Users and organizations are suffering a lot by these attack emerging trend[1]. These hackers became equipped with more advanced technologies and planning their attack in better-organized manner which is more dangerous than earlier years. The botnet crime results E-mail spam, extortion through denial-of-service attacks, identity theft, data theft and click fraud resource consumption etc. A “botnet” is a network of systems affected by malwares known as “bots”. These bots has one specific property that distinguish them with other malwares, they can be remotely operated and controlled. This specific property of bots makes them weapons for various denials of service attacks. These bots are distributed over the internet having enormous cumulative bandwidth if controlled by the Botmaster, to attack any target on the internet. The concept of botnets is evolved from just the last decade, due to open source communities day by day new variants of bots with new stealthy protocols and infection capability are attacking and affecting the victim.

Botnet-based attacks are becoming more powerful and dangerous in such case security professionals needs to understand the newly developed bots. For understanding and study of the bots various works has been done by the researchers across the world [4],[7],[8],[9],[10],[11]. Internet Relay Chat(IRC) based botnets are the first kind of bots using C&C (Command & Control) architecture as a centralized systems. Recent years are more prominent with new technology based bots for their Command & Control. A new type of the bot using Peer-to-Peer topology for the spreading of command and control by the botmaster is more prominent. Various works have been done to understand and create detection frameworks and systems to detected and dismantle the botnets. Various detection mechanism for IRC based botnets are proposed[12],[15],[16]. As now a days Peer-to-Peer botnets are more dangerous in nature detection framework is proposed for them [12],[13],[14]. As per our understanding new kind of bots can be generated easily for creating and developing a mitigation system for the botnets we have to understand their capability and activity. For this purpose development framework for new bots should be created.

2. Related Works

Bots and Botnets are very hot topics for last few years [10], [1]. The first ever Peer-to-Peer bot Storm Bot had control over a million systems. In 2003 first ever bots and botnets properties and overview is discussed by Puri and McCarty. Today the main concentration of bots researcher are on Peer-to-peer bots because of their sustainability and robust network topology formation makes tough to detect and dismantle. Various authors proposed different types of Peer-to-Peer bots. [3] developed a Stochastic Model of Peer-to-Peer botnet to understand different factors and impact of the growth of the botnet. The botnet stochastic model was constructed in the Mobius software tool, which was designed to perform discrete event simulation and compute analytical/numerical solution of models by inputting various input parameters. This kind of research helps to understand the behavior of botnets and it became easy to create mitigation systems and framework for these bots. In botnet technology various works are going on for the detection and mitigation of the Peer-to-Peer botnets.

Authors has proposed an advanced hybrid peer-to-peer botnet [19] which concentrated on the problem of are using the liability constraint of the security professional to detect installed honeypot, because honeypots are not allowed to participate in the real attack scenario. But still some probability remains for the capture of the bots and reverse engineered to understand their strength. This lack of security in bots capture by the defender make the whole botnet susceptible to get exposed.

Significant exposure of the network topology when one of the bot is captured, making easy for the botmaster for the overall control of the botnet. They also included some concept of Honey Pot awareness in their bot system. But still few problems with communication channel and the capture and re engineering of the bot is remains.[7] Predicting a new botnet from the framework and comparing its performance with known ones. Loosely Coupled peer-to-Peer botnet lcbot, which is stealthy and can be considered as a combination of existing P2P botnet structure. Their botnet architecture still follows the idea of "Buddy list" or routing information of the infected host or friend bots. Which keep the whole botnet easy to exposed if one of the bot got captured by the defender. Peer list construction is the main concept behind any P2P botnet which also leave the complete bot exposed any time to the Defender. [5],[6],[17] Authors giving an idea of Honey pot aware bots, and botnets. Honeypots are the only way to observe and understand the activities of a bot. That also makes botnet prone to be exposed to the defender and help them to create a mitigation system for the botnet. Bot masters.

3. Proposed P2P Botnet Architecture

3.1 Classification of Our Bots

We classified our bots very extensively so that it becomes easy to control and operate the botnet by the botmaster.This classification is mainly to refine the bots used in the attack for an effective firepower and less prone for the exposure to the defender. First of all we will group our bots on the basis of their bandwidth if the infected system has a internet connectivity to the outside world equal or greater than our specified bandwidth then only we will consider them to build our botnet these kind of bots we call as Live bots, otherwise we will discard the further infection and these kind of bots will be called as Dead bots and they will not participate in further creation of the botnet. Further we will classify Live bots in two groups one Peer bots which will have global IP addresses without firewall or proxy servers in between, and rest all bots including 1) bots with global IP addresses with firewall or proxy 2) bots with dynamically allocated global IP addresses 3) bots with private IP addresses. We will call second group of bots as Non-peer bots. Further bots are dedicated for the purpose of either infecting other victims or only for attack purpose. If the bot is dedicated for infection of other victims then the code module will send the existing peer list to newly infected bot. In case of attack bots the code module will be spam emails, DDoS command and control handling.

We will mainly concentrate to prevent detection of the Peer bots because they are security bottle neck for our botnet to get exposed to the defender as they only contain peer list or seed list information of other Peer bots. The Peer bots will be able to act as a server for other Peer and Non-Peer bots and client for other Peer bots. Non-Peer bots will be able to act only as a client, they will have entry of other Peer bots only in their peer list.

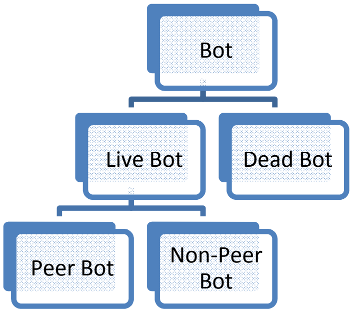


Fig:1 Classification of Proposed Bot

According to the following properties suicide module will delete the peer list information.

1- Bot master intentionally wants to kill the bot, sends command through the communication channel.
2- It will delete peer list in the peer bot to save whole botnet.
3- If average availability of bot in a week is less than desired bot will kill itself.
4- In case of any threat or danger like honey pot trap it will execute suicide module to delete peer list information.

3.2 Botnet Infection

The infection and propagation of the bot is a very important for the purpose of robust bot network and control . The bot will use the backdoor created by other worms for infection. It will first infect the system with first stage small infection code, after wards it will execute various commands and modules to check the bandwidth of the infected system interface, if it is upto desired speed then it will become as live bot and next step of the infection will proceed otherwise it will be declared as a dead bot which further not participate in the bot formation. After the bandwidth check it will perform the IP type checking to confirm weather that infected system can work as a Peer bot or Non-Peer bot. These bots will further propagate to infect other systems.

Stage 1: Initial Infection (Compromising the system)

I. Install the initial Infection files
II. Check the connection speed of victim
III. Decide whether the compromised host is Live or Dead Bot

Stage 2: Participating in Peer network (Creation of Botnet)

I. Connect to the Peers
II. Update Peers list
III. Search the network for encrypted URL

Stage 3: Secondary Injection (Code for attack purpose)

I. Connect to the encrypted URL
II. Download the secondary injection code
III. Execute the code to enhance Bots Power

3.3 Peer Network Creation

Newly infected bots will communicate with the existing bots to update their peer list and other information's. The bot cannot participate in the attack until it connects to the other peers in to the existing botnet, and become ready to share the command and control given by the Botmaster. For every newly connected bot in the botnet we'll use hashing of IP addresses, and a key for the identification by the botmaster. A Bot's identifier is chosen by hashing the bot's IP address, while a key identifier is produced by hashing the key. The identifier length must be large enough to make the probability of two Bots or keys hashing to the same identifier negligible. Identifiers are ordered in an *identifier circle.* Key is assigned to the first node whose identifier is equal to or follows (the identifier of) in the identifier space. This node is called the *successor node* of key , denoted by *successor(k).* Consistent hashing is designed to let bots enter and leave the network with minimal disruption. To maintain the consistent hashing mapping when a bot n joins the network, certain keys previously assigned to n's successor now become assigned to n. When bot n leaves the network, all of its assigned keys are reassigned to n's successor. No other changes in assignment of keys to nodes need occur.

In a dynamic bot network, bots can join (and leave) at any time. The main challenge in implementing these operations is preserving the ability to locate every key in the bot network. In order for lookups to be fast, it is also desirable for the finger tables to be correct. We'll maintain a table in each bot for the storing the peer information and the identifier key generated by the hash function using the bots IP address. This finger table is important to maintain the robust connectivity of the bot to the network. Our emphasis will be here to maintain the correct finger table as accurate as possible.

3.4 Botnet Communication Channel Architecture

Each Peer Bot will contain list of its next two peer bots and other two non-peer bot information in seed list. The non-peer bot will have only two entries of peer bot information with the condition that they both peer bot will contain the information of each other. The bot master will pass the command to any one of the Peer bot depending upon the diuranal dynamics that particular bot will be selected for the first command passing to the whole botnetwork. After getting the command by the botmaster the peer bot will share this command to its next neighbor peer bot as will connected non-peer bot that will ensure the effective communication, for the purpose of command passing priority will be given to the peer bot. Because peer bot can work as client as well as server too, and connected to other peer bots . On the basis of this topology the communication will be handled.

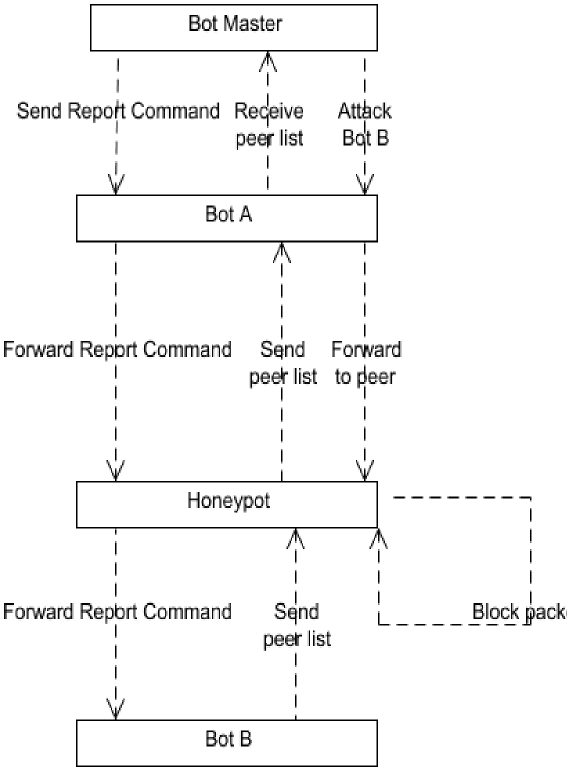


Fig 2: Bot Communication

4. Simulation and Experimental Results

We are presenting simulation results and snapshots of our proposed peer to peer suicide botnet. Our experiment is still in its inception stage, in its current scenario we became successful to implement and execute few properties of our proposed model of botnet. In the simulation model implemented using java technology the botmaster is able to command bots through listing their IP addresses. If necessity arises botmaster sends kill command to the bots to destruct itself.

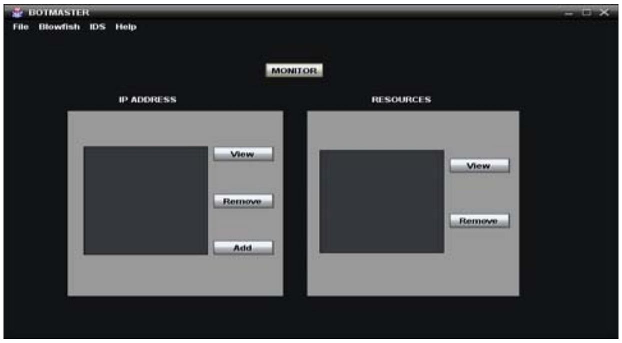
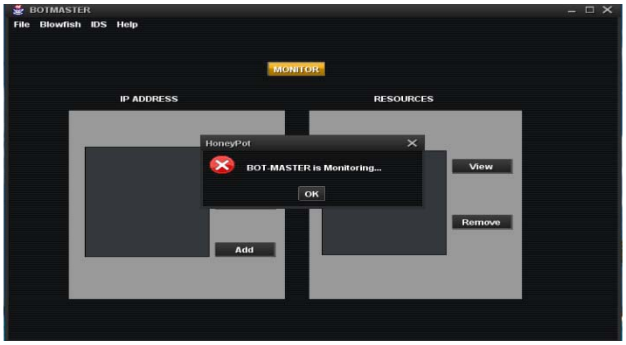
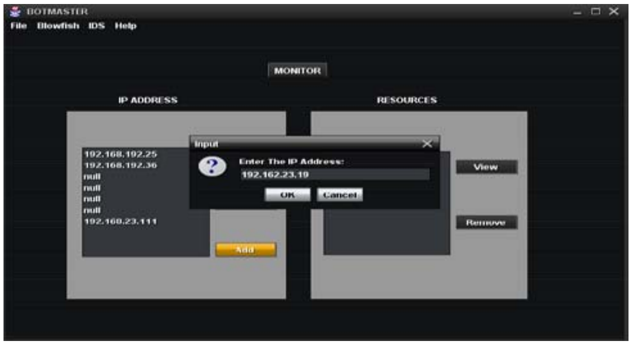


Fig:3 Bot Master Control Interface



BOT-MASTER is Monitoring...

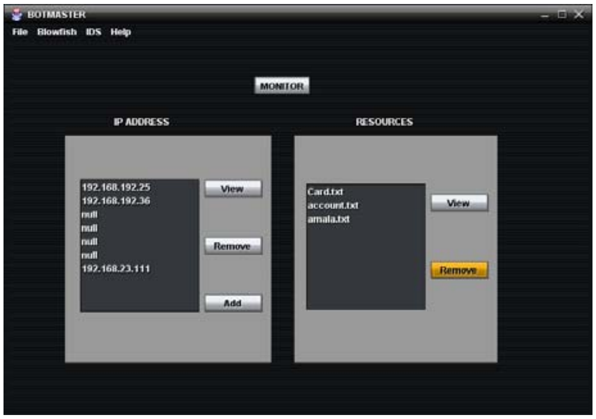
Fig:4 Bot Monitoring by Bot Master



192.168.192.25
192.168.192.36
null
null
null
null
192.168.23.111

Enter The IP Address:
192.162.23.19

Fig:5 Selecting Bot IP Address by Bot Master to send Kill Command



```
192.168.192.25
192.168.192.36
null
null
null
null
192.168.23.111
```

```
Card.txt
account.txt
amala.txt
```

Fig:6 IP address listing of Bot by Bot Master

5. Conclusion

Implementation of new types of bots will facilitates to understand the future bots which can be created by the attackers. Study and simulation results of our bot provide framework to understand the bot working and there communication channel architecture. This bot is tough to control because of the peer network topology but harder to reverse engineered or trapped by the honey pots. It provide small but high fire power bot network to the bot master which is tough to shut down.

References

[1] B. McCarty, “Botnets: Big and Bigger,” IEEE Security & Privacy Magazine, vol. 1, no. 4, pp. 87-90, July-Aug. 2003.

[2] Elizabeth Van Ruitenbeek and William H. Sanders, “Modeling Peer-to-Peer Botnets”, Quantitative Evaluation of Systems 2008 IEEEComputerSociety,DOI 10.1109/QEST.2008.43, 5th International Conference on Quantitative Evaluation of SysTem Palais du Grand Large at Saint Malo, France 14th-17th September, 2008

[3] Julian B. Grizzard, Vikram Sharma, Chris Nunnery, and Brent ByungHoon Kang, “Peer-to-Peer Botnets: Overview and Case Study”, In USENIX Workshop on Hot Topics in Understanding Botnets (HotBots’07) April 10 2007, Cambridge, MA, USA

[4] Justin Leonard, Shouhuai Xu and Ravi Sandhu, “A Framework for Understanding Botnets”, 2009 International Conference on Availability, Reliability and Security Fukuoka Institute of Technology, Fukuoka, Japan March 16-March 19.

[5] Ping Wang, Lei Wu, Ryan Cunningham,Cliff C. Zou, “Honeypot Detection in Advanced Botnet Attacks”, Int. J. Information and Computer Security, Vol. 4, Issue 1, 2010, DOI: 10.1504/IJICS.2010.031858

[6] Simon Innes, Craig Valli, “Honeypots: How do you know when you are inside one?”, the 4th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 4th 2006.

[7] A Framework for P2P Botnet, Su Chang, Linfeng Zhang, Yong Guan, Thomas E. Daniel,2009 International Conference on Communications and Mobile Computing

[8] The Zombie Roundup: Understanding, Detecting and Disrupting Botnets, Evan Cooke, Rarnam Jahanian, Danny McPherson Electrical Engineering and Computer Science Department Arbor Networks.

[9] wide-scale Botnet Detection and Characterization Anestis Karasaridis, Brain Rexroud, David Hoeflin

[10] A Survey of Bots used for Distributed Denial of Service Attack, Vrizlynn L. L. Thing, Morris Solman, Naranker Dulay, http://www.doc.ic.ac.uk.

[11] Criminology of Botnets and their detection and Defense Methods, Jivesh Govil, Jivika Govil, IEEE EIT 2007 Proceedings, IEEE 2007

[12] Automatic Discovery of Botnet Communities of Large Scale Communication Network, Wei Lu, Mahbood Tavallaee and Ali A Ghorbani, ASIACCS’09, March 10-12, 2009, Sydney, NSW, Australia ACM 2009.

[13] P2P botnet detection using behavior clustering & Statistical Tests, Su Chang, Thomas E. Daniels, AISec’ 09, November 9, 2009 ACM 978-1-60558-781-3/09/11

[14] A Proposed Framework for P2P Botnet Detection, Hossein Rouhani Zeidanloo, Azizah Bt Abdul Manaf, Rabiah Bt Ahmad, Mazdak Zamani, Saman Shojae Chaeikar, IACSIT International Journal of Engineering And Technology, Vol, No. 2, April 2010,

[15] Detecting Botnets with Tight Command and Control, W. Timothy Strayer, Robert Walsh, Carl livadsa, David Lapsley,

[16] [16] A Novel Approach to Detect IRC-based Botnets, Wei WANG, Binxing FANG, Zhaoxin ZHANG, Chao LI, “ 2009 International Conference On Network Security, Wireless Communication and Trusted Computing, 2009 IEEE.

[17] Honeypot-Aware Advanced Botnet Construction and Maintenance Cliff C. Zou Ryan Cunningham , Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN’06) IEEE.

[18] VMM-Based Framework for P2P Botnets Tracking and Detection LingYun Zhou , 2009 International Conference on Information Technology and Computer Science

[19] Ping Wang, Sherri Sparks, and Cliff C. Zou Member IEEE, “An Advanced Hybrid Peer-to-Peer Botnet”, *IEEE Transactions On Dependable and Secure Computing, Vol. 7, No. 2, April-June 2010 page*

Mukesh Kumar received his Bachelor of Technology degree in Computer Science and Engineering from Uttar Pradesh Technical University Lucknow, India, in 2009. He is currently pursuing his master's degree in Network and Internet Engineering in the School of Engineering and Technology, Department of Computer Science, Pondicherry University, India. His research interests include Denial-of Service resilient protocol design, Cloud Computing and Peer to Peer Networks.

Pothula Sujatha is currently working as Assistant Professor and pursuing her PhD in Department of Computer Science from Pondicherry University, India. She completed her Master of Technology in Computer Science and Engineering from Pondicherry University and completed her Bachelor of Technology in Computer Science and Engineering from Pondicherry Engineering College, Pondicherry. Her research interest includes Information Security, Modern Operating Systems, Multimedia Databases, Software Metrics and Information Retrieval. Her PhD research is on performance Evaluation of MLIR systems.

Sunil Kumar Verma received his Bachelor of Technology degree in Computer Science and Engineering from Uttar Pradesh Technical University Lucknow, India in 2009. He is currently pursuing his master's degree in Cyber Law & Information Security from Indian Institute of Information Technology Allahabad. His research interests include Denial-of Service resilient protocol design, cryptography and network security.

.

P Manikandan is presently pursuing Master of Technology in Computer Science with specialization in Network and Internet Engineering from Pondicherry University, India. He has completed his Bachelor of Technology in Computer Science and Engineering from Bharathiyar College of Engineering and Technology affiliated to Pondicherry University. His research interest includes Wireless Communication, Network Security, distributed systems, Red Hat. Currently he is working on Thread Scheduling in Solaris.

Madarapu Naresh Kumar is presently pursuing Master of Technology in Computer Science with specialization in Network and Internet Engineering from Pondicherry University, India. He has completed his Bachelor of Technology in Computer Science and Engineering from JNTU Hyderabad. His research interest includes Cloud Computing, Web Services, Software Metrics, SOA and Information Retrieval. Currently the author is working on security issues in Cloud Computing

Chetana Sidige is presently pursuing M.Tech (Final year) in Computer Science of Engineering at Pondicherry University. She did her B.Tech in Computer Science and Information Technology from G. Pulla Reddy Engineering College, affiliated to Sri Krishnadevaraya University. Her research interest includes Network Security, Information retrieval Systems and Software metrics. Currently the author is working on Multilingual Information retrieval evaluation.