IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

A Comprehensive Analysis on Current Botnet Weaknesses and Improving the Security Performance on Botnet Monitoring and Detection in Peer-to-Peer Botnet

Sivakumar Kalimuthu[1], Kannan Ponkoodanlingam[2], Premylla Jeremiah[3], Umapathy Eaganathan[4], Anatte Schaffer Anak Juslen[5]

Research Scholar, Faculty of Computer Science and Engineering, Sathyabama University, India[1]

Sr. Lecturer, Faculty of Science Technology Engineering & Mathematics, INTI International University, Malaysia[2]

Research Scholar, University Technology Malaysia (UTM), Kualalumpur, Malaysia[3]

Research Scholar in Computer Science, Vels University, India[4]

IT Engineer, Singapore Telecommunications (Singtel) Limited, Kualalumpur, Malaysia[5]

Abstract: Programmed networks, generally known as botnets, such as a network connected by computers is known as botnet. In core, a bot is merely a sequence of either a series of commands or program, or scripts that is deliberately aimed to link to a server and accomplish a sequence of commands in order to develop a malicious or harmful attack. Botnets, basically it does several functions and it became the origin of many internet attacks. It is the main cause for the attack to take place. The attacker is the one who is going to attack the computers in the botnet. The attacker is known by the name as botmaster, in case of defending from attacks in future it is important to know how the attacks can be detected and prohibited from the past attacks. This paper is focus on, a comprehensive analysis on the current peer-to-peer (P2P) botnet [1, 2, 5] weaknesses, how to identify the threats and secure against the botnets, and how to increase the security performance level on botnet which is comparable on the past performance. It is considerably hard for an attacker to hijack or crash or hack the other system.

Keywords: Botmaster, Peer-to-Peer Botnet, Detection, Monitor, Malicious attack, Crash, Computer Security.

I. INTRODUCTION

In spite of the fact that botnets first appeared many years ago, still researchers are sparking the interest of the research community. Programs for intrusions attacks and malicious software have progressed a great deal over the past several years. Robot networks, popularly known as botnets, have a varied history. The term *botnets* is used to define networks of infected end-hosts, called *bots*, which are under the control of a human operator commonly known as a botmaster [1-3,5]. A “botnet” consists of a network of compromised computers controlled by an attacker [1, 2]. Fundamentally, a bot is just a progression of either program or commands or scripts that is intended to associate with usually server and execute a progression of command or commands. In a general sense it executes distinctive functions. It need not be malicious or unsafe. While botnets initiate vulnerable machines utilizing routines likewise used by different classes of malware, such as social engineering, remotely abusing software vulnerabilities [8], their characterizing trademark is the utilization of command and control [1,2,3] channels. The basic role of these channels is to spread the botmasters' charges to their bot armed forces. These channels can work over a mixture of network topologies and distinctive communication components, from built up Internet protocols to later Peer-to-Peer (P2P) [1,2,5] protocols. Botnets that have showed up to this point have had a typical integrated architecture, which means that, bots in the botnet associate straightforwardly to some uncommon hosts called "command-and-control” servers [1, 2, 3, 11, 12].

All computer users are at risk because we all surf the same Internet. There are just modest bunches of ways that cybercriminals can contaminate a host or system with their bots (or any type of malicious programming). These ways for the most part include some type of social engineering, which can be characterized as hacking the human mind. However, the malware attacks, email spam, DoS attacks are more common in internet because network of computers is the main area of attack to take place [1]. The other most important in the past is the common centralized architecture, each bots in the network is connected with some common host called “command and control servers“, this will forward the request or the information from attacker to all the bots. According to the attacker, the command and control servers are the weakest point for an attack to take place, can easily identify the properties of a computer. If a computer or a bot is hijacked or attacked,

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

the information about the entire botnet will be exposed to an attacker easily [7, 9, 11, 12]. The enormous issue is the discovery of newer threats and different forms of attack, there by compromising the security of the system [8]. So in order to prohibiting from the attacks we here are using the P2P botnet, to defend the attacks from the botmaster, we are using the honeypot, which will defend the Denial-of-Service (DoS) attacks [1,3]. It is quite safer way of networking using the Peer-to-Peer (P2P) [2] architecture.

II. BACKGROUND

All internet users are at risk because we all surf the same Internet. Each individual should be careful about person to person communication such as social networking assaults; amid the most recent couple of years a few vast botnets have been taken disconnected from the network. To keep these security triumphs and make their botnet infrastructure tougher and more robust, bot master have started to present new methods. However, organizations and governments experience the most damage from botnet assaults. Some of the threats to organizations include [4]:

Click scam: Visiting website pages and suddenly tapping on promotion flags to take gigantic totals of cash from web publicizing organizations.

Distributed DoS attacks: Saturating transmission capacity is to forestall honest to goodness activity. These assaults are regularly done by contenders, disappointed clients, or those with a political plan.

File system penetration: Accessing basic frameworks to take client information, representative protection data, competitive advantages, corporate financials, and so on.

Deactivating present-security: **A**verting clean-up endeavours or taking by adversary bot proprietors.

Spam: Using the assets and transfer speed of different frameworks to send tremendous volumes of spam.

Source code infection: Poisoning the whole source code tree by embedding unapproved and imperceptible changes or finding extra vulnerabilities to abuse.

The consequences of these assaults can be entirely extreme, costing organizations noteworthy labour and time to tidy up. Furthermore, organizations can have their administrative or industry compliances repudiated. Lawful liabilities are likewise likely from clients, representatives, or other people who experience the ill effects of an organization's lacking efforts to estab and undetectable changes or discovering additional vulnerabilities to exploit.

A botnet is a gathering of contaminated end-host under the command of a botmaster. In figure 1 is outlines the different stages in a typical botnet life-cycle. Botnets as a rule seize new victims by remotely misusing a powerlessness of the software product running on the victim. Botnets acquire infection techniques from a few classes of malware, including self-imitating worms, email infections, and so forth. Contaminations can likewise be spread by persuading casualties to run some type of malignant code on their machines (e.g., by executing an email connection). Once infected, the victims ordinarily executes a script that brings the picture of the real bot binary from a predetermined endless supply of the download; the bot paired introduces itself to the objective machine with the goal that it begins naturally every time the victim is rebooted.

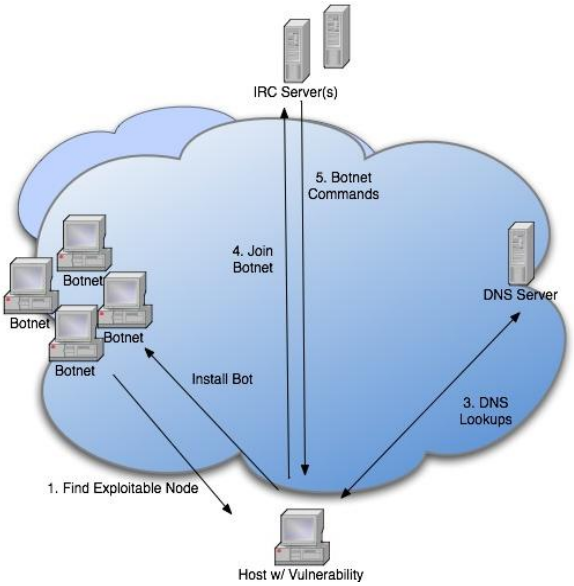

Fig. 1: The life-cycle of a typical botnet infection [2,20]

III. LITERATURE REVIEW

The peer-to-peer architectural design also has some disadvantages. Botnets such as slapper [2], sinit [4], phatbot[5] and nugade[6] also have some kinds of implementation in peer to peer architecture. Some has removed "bootstrap process" which is used in peer to peer protocols. Sinit used the "public key encryptography "for authentication [7].

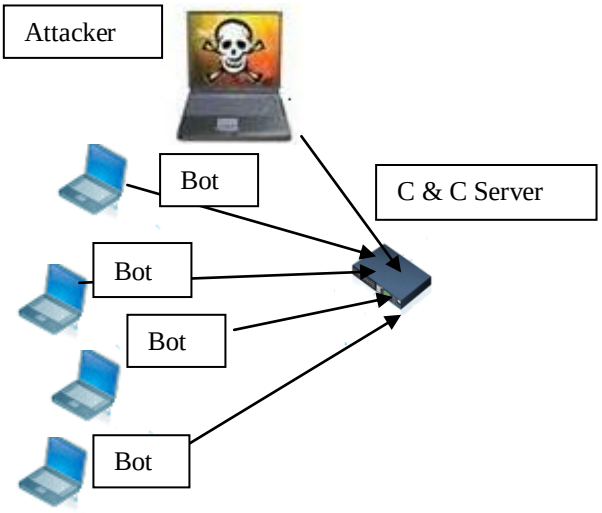

Fig 2: A command-and-control (c&c) architecture [2]

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

Which are the results will lead to poor construction of the architecture. There is a possibility of bot getting shutdown if security filter on the gnuetla caches server. These are the major drawbacks in the existing system. Botnet is always a hot research topic til many years in 2003, puri [8] has published a paper on a summary of bots and botnets. McCarty[9], has argued about how to use a honeypot to monitor botnet. Barford and yegnesaran (2006) gave the detailed & systematic dissection on well-known botnets in past. Now the current research is on the monitoring and detecting part [10]. Dagon et al. [11] has presented the botnet monitoring system by forwarding the DNS mapping of command and control servers for monitoring the botnet with the help of dynamic DNS. But this paper, takes place using the honeypot.

IV. RELATED MODULES - ANALYSIS ON CURRENT P2P ARCHITECTURE WEAKNESSES

A. Bots classes

The bots in proposed work has two classifications. The first one is, the node has the static and non-private Internet Protocol (IP) address which is accessible globally; it is called as servent bots. The other is remaining bots, including dynamically allocated IP address; private IP address firewalls so that it cannot be connected to the internet that is, globally. This is called as the client bots, but both the client and the servent bots can be connected dynamically in the peer list. The bot classification will use the dynamic host configuration control protocol. The botmaster will collaborate along with bots to determine that bots.

B. Botnet architecture

Here we have three client bots and five servent bots two is the peer list size. Each bot consists of the identities.

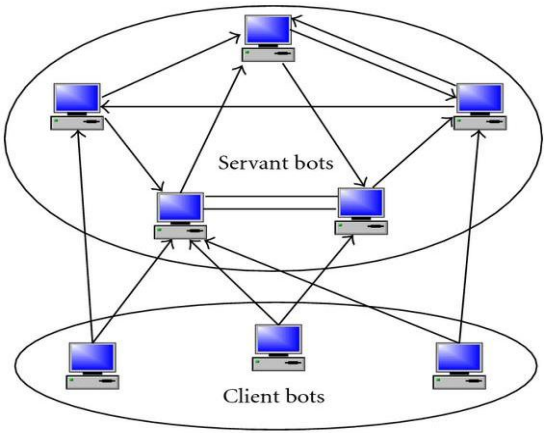

Fig 3: Hybrid P2P Botnet architecture [2]

The arrow represents here from bot “a to b” is that an initiation to connection to bot b. This shows the bots are interconnected botnet can send his command through any bot in the botnet. Both the groups are connected with each other. This will retrieve the peer list information when an attacker is issuing a command. When a bot has receiving a new command it will forward this command to all the other bots that is, the servent bots. This is meant by command forwarding and the current work has undirected topology.

C. Comparison between command-and-control botnet and p2p botnet

From the current botnets can easy see the extension of a C and C. The servent bots takes the role of command-and-control (C&C) servers of current botnet. These servent bots interconnect with each other bots in the network. Due to this large servent bots in the network it is difficult to be hijacked by the attacker (Bot Master). The most important among the botnet is communication between the bots. The peer to peer is equal to c&c botnets. The suggested architecture is having a more robust and complex communication architecture. This cannot be shut down easily.

Command authentication:

Comparing with the existing botnet the proposed work has command authentication. It has sufficient public key encryption. The botmaster will generate a pair of public or private key. The public key is hard coded here. As here we have strong authentication it is not possible to hijack other bots.

Individualized encryption key:

The peerlist architecture had made easy to implement the encryption. Here the servent bots is generated randomly. The servent bots ‘n’ has generated a random symmetric key encryption ‘En’. If suppose the bot A on the peerlist is denoted by ‘Ba’. It contains the IP address of N servent bots and the symmetric keys .so thus the peer list on the bot is defined as

$$En=\{(IPn1,En1),(IPn2,En2),\ldots,(IPnN,EnN)\}$$

Individualised Service port:

The proposed architecture enables the communication activity in the terms of the administration port. Despite the fact that the servent bot needs to acknowledge the associations from alternate bots it has likewise to run a server procedure that is running on the administration port.

Consider a peerlist on bot A is

$$En=\{(IPn1,En1,\ Pn),,\ldots,(IPnN,EnN,PnM)\}$$

Pn is the service port on the servent bot. the benefits of the individualized service port is having two benefits, even though the service port is a dangerous in analysing the traffic in network because of the structure of the network. Port makes the defenders to feel hard to detect a botnet based on the monitoring network traffic. P2P has a strong resistance against network traffic flow based detection, when it is joined with the customized encryption design [12]. Bonet always having a secrete backdoor to ensure the attacker cannot hijack the information. As it is mentioned before the service port can be choose randomly or selectively by a separate bot. Even though generating a service port is not worthy for botnet network traffic accepted to rarely used port is abnormal. The servent

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

bought selectively picks the service port by choosing an encryption port like port 22[SSH], 443 (HTTPS) for the encryption of bot communication traffic. It is not difficult if we already implemented the open source code honey [13]". It is really a tough job to detect a botnet while using individualized service port but this is not meant that a servent bought cannot be identified using botnet network traffic.

V. SECURITY PERFORMANCE: MONITORING AND DETECTION P2P BOTNET

Here we are designing an architecture using peer to peer architecture. The communication takes place through the peer list in the botnet architecture. Botmaster will starts communication with other bots using the infect command. Once the connection has been made the botmaster ask the peer list information using report command, this will be easy to monitor an entire botnet. The sensor is also a node in the botnet which is randomly elected.

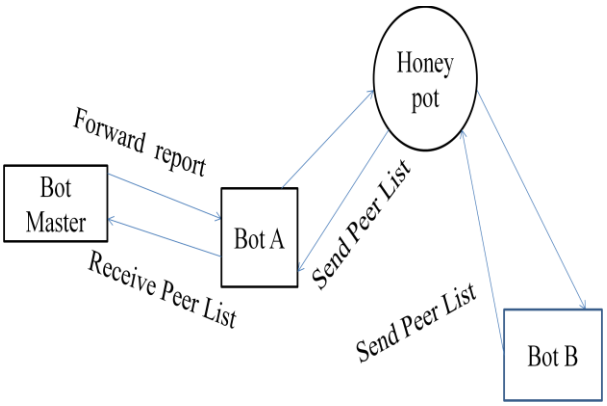

Fig 4: Data Flow Diagram for peer list

This will have the peer list information that is asked by the botmaster. The honeypot is a major advantage in this paper. Whenever communication is takes place between the two bots honeypot will monitor the communication as depicted in the following diagram as illustrated the flow of the communication.

Honeypot need not have any special algorithm to design. Whenever the honeypot is updated it will block the details that are going to send to the botmaster.

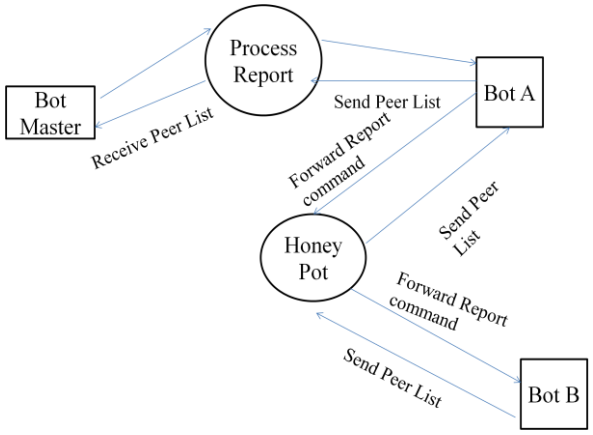

Fig 5: Data Flow Diagram for process report

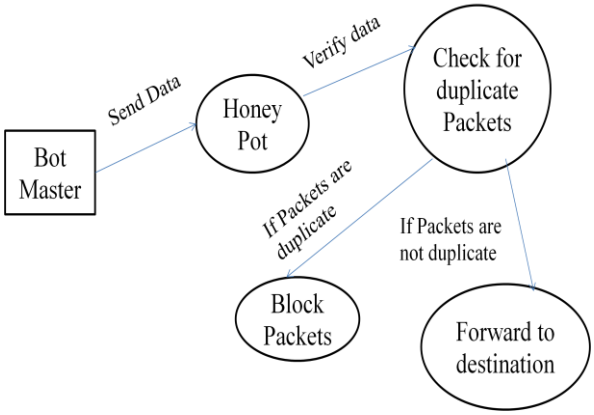

Fig 6: Data Flow Diagram for Honeypot

This will block the details and intimate us who are the attacker. Here the servent bots and the client bots. The servent bot is for looking at the self-determined service and also having the self-generated symmetric key encryption. The individual encryption and service port is the hard place to be detected by the botnet through the network.

Botmaster monitoring the botnet:

A system is intended to detect botnets that use propelled order and control system by associating auxiliary discovery information from numerous sources. This unnerving new class of assaults straightforwardly affects the everyday existences of a huge number of individuals and jeopardizes organizations around the globe. For instance, new attacks take individual data that can be utilized to harm reputation or lead to huge money related troubles. Present mitigation methods are emphasis on the indications of the issue, sifting the spam, solidifying web programs, or building applications that caution against phishing traps.

The other challenge that takes place in designing the botnet is making sure that the defenders cannot easily monitor its botnet.

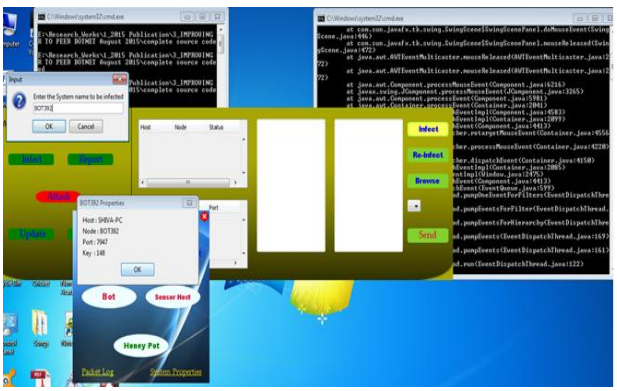

Fig 7: Connection with bot

But the botmaster can easily monitor, using the detailed information about the botnet, the botmaster can 1) effectively perform the attacks in accordance with the bot population, delivery, off/on position IP address. 2) While facing the counter attacks from defenders, we should have tightly control over the botnet; here in this section we

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

present a modest and effective technique to monitor the botnet using the botmasters.

Monitoring through a dynamically changeable sensor: The botmaster issued a special command known as report command to the botnet for monitoring the proposed P2P botnet. This command will instruct all the bots to forward its information to a specific machine which is compromised and controlled by the attacker. Here the sensor is the data collecting machine. The report command has the specifications about the IP address or the domain name of the integrated sensor host. Before knowing the actual report command this would prevent the defenders from knowing the actual identity.

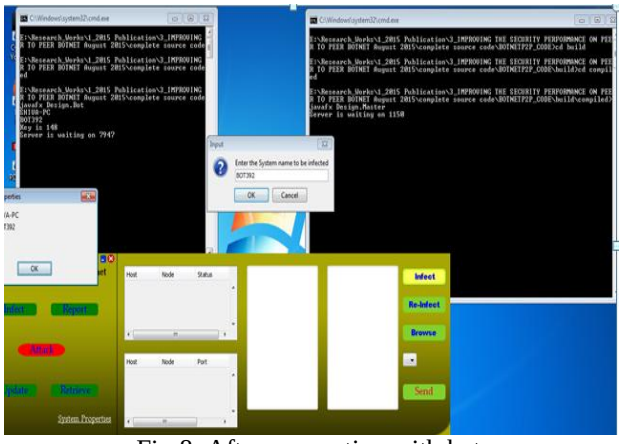

Fig 8: After connection with bot

There is also having possibility, that the defenders may quickly identify the identity of the sensor host. To deal with these threats the botmaster can implement any of these upcoming procedures.

1) Using the internet service like HTTP, E-mail for reporting to sensor. The sensor is chosen so that it will provide the service to avoid abnormal network traffic.
2) Instead of using single sensor it is better to use a several sensor machines.
3) Choosing the sensor host which is hard to shut down, monitor.
4) Verifying the sensor host manually is not the honey.
5) After retrieving the report data, wiping out the hard drive on a sensor host immediately is important
6) Specification of expiration time in the report command for preventing the bots exposing itself after that time.
7) Once the botmaster knows that the sensor host has captured by the defenders issue the other command to the bot net to cancel the previous report command.

Monitoring information:
The botnet size and the topology are only wants to know by the botmaster.

IP address type: IP address is used for the identification of the internet computers. But the identification of IP address is made difficult by DHCP and NAT. The botmaster may implement the ID based identification. It simplifies the dimension of the NAT.

Botnet construction
The construction procedure is as follows,

1) Newly infected. Bot A will pass the peerlist to the host B. If the servent bot is A, B will add A in its peer list.
2) If "A" is identified "B" then the servent bot A add B in its peer list in the similar way.
3) Reinfection - If there is a possibility of reinfection bot A infects bot B, so the bot B will be replaced.

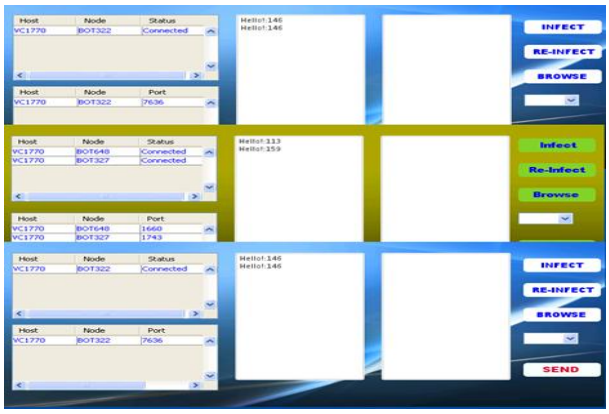

Fig 9: Report

Whenever reinfection takes place frequently the reinfection can inter connect different infection path together, to make all the bots to connect evenly. The important among these are, a bot does not provide the peer list information to the one who reinfect it. This is important because if the defenders are not recursively infect all servent bots based on the seized bot in the honeypot. Firewall is used by the defenders for redirecting the leaving infection from taken bot A to reinfect the servent bots in the A peer list. At that point it will routinely get the peer list from the servent bots.
In order to analysis the construction of botnet topology net via simulation, we need to determine its settings. First, bhagwan[14] studied P2P files sharing systems. He witnessed that about 50% of computers will change the IP address within 4-5 days [14]. Second as pointed out and the botnets have fallen their sizes to an approximate of 20,000 yet the possible population is much higher [15,16]. In addition to that we thought that the peer list has a size of N = 20 and there are 21 servent host to spread the botnet.

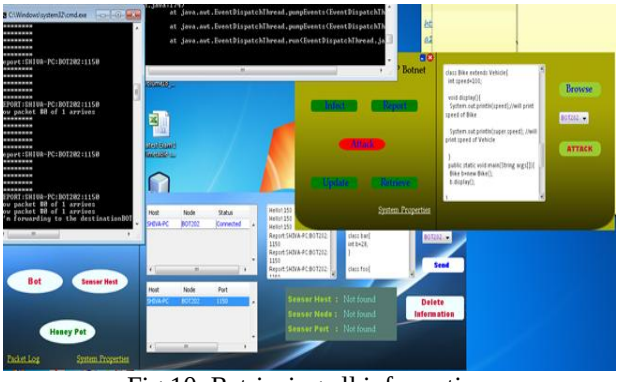

Fig 10: Retrieving all information

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

Advanced construction procedure:
Allowing the bots to exchanging and updating the peer list frequently is the best method to develop the network connectivity. But this type of design makes an easy way for protectors to acquire the characters of each servent bots, in the event that anybody is caught by defenders. Utilizing the botnet data, the botmaster can without much of a stretch overhaul its peer list for having solid and adjusted availability. Once the botnet spreads out, the botmaster will issue the report command for the purpose of obtaining the information about all presently accessible servent bots. This type of servent bots is called as the peer list updating servent bots. Then the command is issued by the botmaster called update command for allowing all bots to get the updated peerlist from the sensor host. This sensor host will choose N servent bots randomly, to encomposes the updated peer list and it will send back to the requested bot. This procedure is could run by the botmaster either a single time or few times through or adjust botnet transmission after each running process each present bots will have identical and well-poised connection for peer list updating.


Fig 11: Botmaster sending code to attack bot

[
According to the botmaster, when and how the peer list updating procedure can run? First after the release of the botnet the procedure should be executed once for preventing defenders from eliminating all primary bots. Second each phase of the updating process creates the botnet strong and balanced. In addition to that the botmaster could run the process to update the topology of botnet.

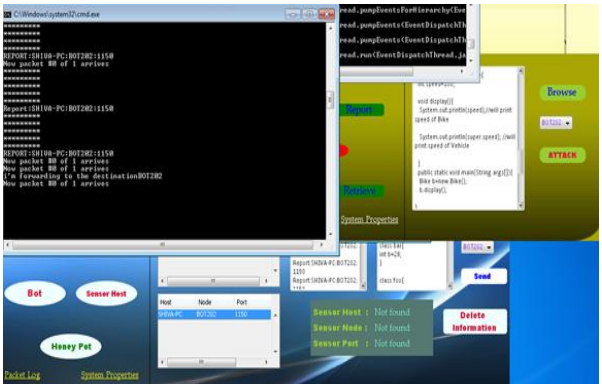
Fig 12: Finding attacker

There are two reasons that affect the connectivity of botnet. 1) Few of the bots are detached by the defenders. 2) Few are offline status. Despite the fact that these two elements are totally distinctive, they have same effect on botnet network at the time the botnet utilized by the botmaster.

Robustness based on two metric functions:
For the botnet to be connected together, the servent bots used in the peer list updating, procedure is the back bone of the two metric functions is represented to measure robustness. Here let C (p) is the connected ratio and D (p) is the degree ratio after removing top (p) fraction of almost connected bots among the peer list updating [2].

C(p) = Number of bots in largest connected graph / Number of remaining bots [2]

Defence against the p2p botnet:
As, the peer-to-peer botnet relies on servent bots. On the off chance that the botnet is not ready to get an extensive number of servent bots, this botnet will debase to conventional C&C servers which are less demanding to close down.

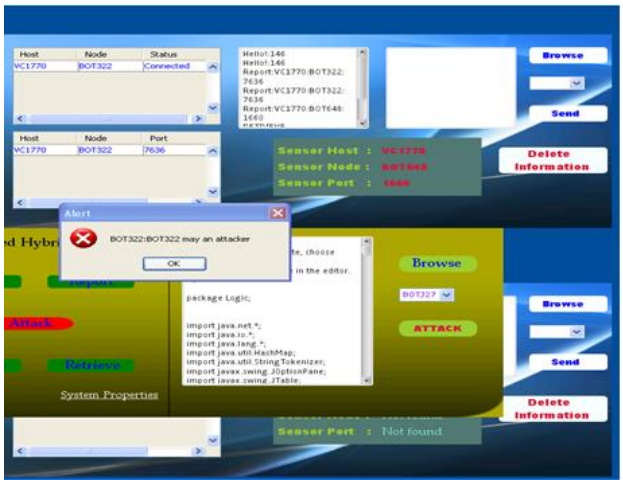
Fig 12.1: Finding attacker

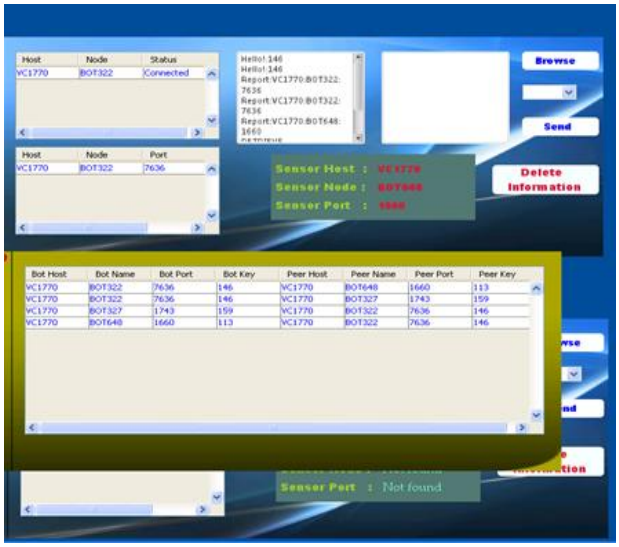
Fig 12.2: Finding attacker

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 5, May 2016

This is the motivation behind why the protectors concentrated on PCs with static worldwide IP address. Second before having an overhaul charge for the first from the botmaster, the bot expert is defenceless static since it is associated through the little arrangement of starting servent bots.

The defenders ought to grow fast detection systems empowering to close down rapidly. The third defence relies on honey bot technique. If the botnet cannot sense mechanisms of honeypots, the defenders try to poison its communication channel. It will let their infected honeypot link the botnet to have a static global IP address. They will be treated as a servent bots.

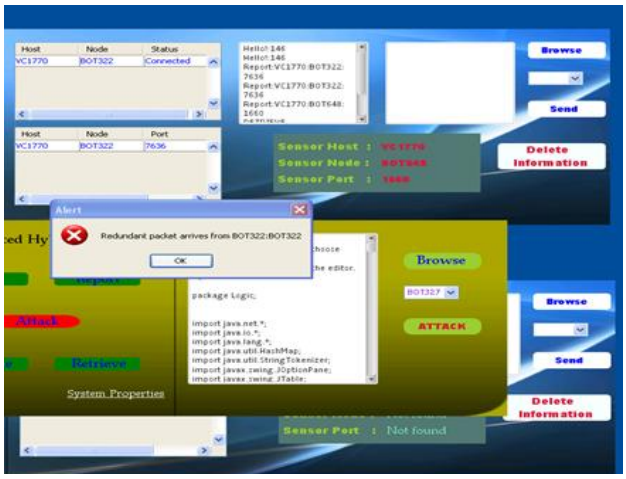

Fig 13: Honeypot prevention (a)

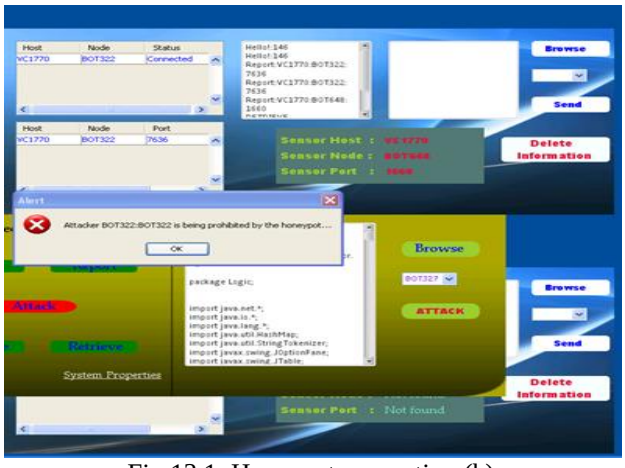

Fig 13.1: Honeypot prevention (b)

As a consequence of this they will involve part of positions in peer list, diminishing the quantity of substantial correspondence directs in botnet. Notwithstanding that the defenders will know their complete botnet correspondence and its individuals through spying honeypots.

Honeypot strategies:
Honeypot is a successful approach to trap and keep an eye on malware and vindictive exercises. It is a viable approach to utilize honeypot on botnet spying. Botnet observing on spying honeypots:

In the event that botnet cannot distinguish honeypots, protectors will let their honeypots to join botnets and screen its exercises. In light of the honeypot bots protectors can get the plain content of charges by botmaster. Once the command is comprehended, defenders can have the capacity to do the following;

1) Swiftly finding the sensor host used by botmaster in report command
2) Knowing the target in attack command, it will be easy to implement countermeasures quickly right before the attack

Alternative honeypot based observing happens amid peer list method. A honeypot can be designed to course its active movement to another honeypot; in the meantime the malignant code still trusts that it has some genuine machines.

Botnet detection and monitoring without honeypots:
Monitoring traffic to the botnet sensor:
Centralized monitoring sensor is the week point of proposed botnet. If the defenders are having a setup of decent traffic cataloguing systems it is potential to detention the traffic to a botnet sensor. This is called as botnet sensor monitor.

Detecting and monitoring the servant bots:
In the proposed peer-to-peer botnet, servent bots are especially used in peer list updating procedure. If the non-server host is affected and helps as servent bot, the host can easily be tracked by the defenders because of increase in the traffic in and out of the host.

VI. CONCLUSION

This paper has laid out the inceptions and structure of bots and botnets, and indicated how they have developed to end up strong weapons. We concentrated on strategies for recognizing P2P-based bots and showed three general command and control topologies to represent the trouble of centering identification endeavours on order and control activity. In view of this understanding, we portrayed a way to deal with recognize botnets by connecting auxiliary location data to pinpoint bots and botnet correspondence.

REFERENCES

[1] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis (2006) "A Multifaceted Approach to Understanding the Botnet Phenomenon" IMC'06, October 25–27, 2006, Rio de Janeiro, Brazil. ACM 1595935614/06/0010.
[2] Ping Wang, Sherri Sparks, and Cliff C. Zou (2010), "An Advanced Hybrid Peer-to-Peer Botnet" IEEE Transactions on dependable and secure computing, Digital Object Identifier no. 10.1109/TDSC.2008.35. Vol. 7, No. 2, April-June 2010.
[3] Felix C. Freiling, Thorsten Holz_, and Georg Wicherski (2005) "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks" S. De Capitani di Vimercati et al. (Eds.): ESORICS 2005, LNCS 3679, pp. 319–335, 2005. Springer-Verlag Berlin Heidelberg 2005.
[4] Zheng Bu, Pedro Bueno, Rahul Kashyap, and Adam Wosotowsky (2010) "The New Era of Botnets" McAfee Labs, Mission College Boulevard Santa Clara, CA 95054

[5] Ullah, I., N. Khan, and H. A. Aboalsamh. "Survey on botnet: Its architecture, detection,prevention and mitigation", 2013 10th IEEEINTERNATIONAL CONFERENCE ON NETWORKING SENSING AND CONTROL (ICNSC), 2013.

[6] Pandiaraja, P., and J. Manikandan. "Web proxy based detection and protection mechanisms against client based HTTP attacks", 2015 International Conference on Circuits Power and Computing Technologies [ICCPCT-2015], 2015.

[7] N. Provos, "A Virtual Honeypot Framework," Proc. 13th Conf. USENIX Security Symp. (SSYM '04), Aug. 2004

[8] Sivakumar Kalimuthu, Premylla J, Kannan P, Anatte Schaffer J (2015) "Susceptibilities Detection Approach in Network Connected Servers using the Composite Fault Prototypical", International Journal of Advanced Research in Computer and Communication Engineering, DOI 10.17148/IJARCCE.2015.4615, ISSN: 2278-1021, Vol. 4, Issue 6, pp.64-70, May-June 2015.

[9] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. The Internet Motion Sensor: A distributed blackhole monitoring system. In Proceedings of Network and Distributed System Security Symposium (NDSS '05), San Diego, CA, February 2005.

[10] S.Kandula, D.Katabi, M.Jacob and A.Berger (2005),"Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds " Proc.second symp.networked systems design and implementation (NSDI 05), May 2005.

[11] D.DOgon, C. Zou and W. Lee (2006), "Modeling botnet propagation using time zones" proc.13th Network and distributed system.

[12] I. Arce & E. Levy (2003), "An Analysis of the Slapper Worm," IEEE Security and Privacy Magazine, vol. 1, no. 1, pp. 82-87, Jan.-Feb. '03.

[13] C. Zou and R. Cunningham, "Honeypot-Aware Advanced Botnet Construction and Maintenance," Proc. Int'l Conf. Dependable Systems and Networks (DSN '06), June '06.

[14] B. McCarty, "Botnets: Big & Bigger (2003)," IEEE Security & Privacy Magazine, vol. 1, no. 4, pp. 87-90, July-Aug. '03.

[15] P. Barford and V. Yegneswaran, An Inside Look at Botnets, to appear in Series: Advances in Information Security. Springer, 2006.

[16] Dagon, C. Zou, and W. Lee, "Modeling Botnet Propagation Using Time Zones," Proc. 13th Ann. Network and Distributed System Security Symp. (NDSS '06), pp. 235-249

[17] R. Bhagwan, S. Savage, and G.M. Voelker, "Understanding Availability," Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS '03), Feb. 2003.

[18] CNET News: Bots Slim Down to Get Tough, http://news.com.com/2104-7355_3-5956143.html, Nov. 2005.

[19] Washington Post: The Botnet Trackers, http://www.washingtonpost.com/wp-dyn/content/article/2006/02/16

[20] R. A. Rodr´ıguez-G´omez, G. Maci´a-Fern´andez and P. Garc´ıa-Teodoro (2011) "ANALYSIS OF BOTNETS THROUGH LIFE-CYCLE", SECRYPT 2011 - International Conference on Security and Cryptography