THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Tracing Back The Botmaster

Sneha Leslie
Information Security and Cyber Forensics SRM University, India

Abstract:*Nowadays, cyber-attacks from botnets are increasing at a faster rate than any other malware spread. Detecting the botmaster who commands the tasks has become more difficult. Most of the detecting methods are based on the features of any communication protocol or the history of the network traffic. In this paper, a rational approach is brought for the live detection of the botmaster in the internal network. The victim machine monitors its packets and compromises the bots in the network and finds the traces to the botmaster. This approach works independent of the structure of the botnet, and will be a better option for online detection of the botmaster.*

Keywords: *Botnet, DDOS, detection methods, network security, metasploit framework.*

1. Introduction

Cyber-crimes are increasing day by day in various forms and with various effects. Thus Internet has become the most vulnerable area. Attacks done behind the legitimate machines have got a public name “Botnet”, where the botmaster who is the real attacker compromises set of host machines in the network called zombies and makes the zombies to perform the various malicious tasks against a victim. Thus botmaster hides his real identity. The term “bot” is derived from the word “ro-bot” which is said to be nay scripts or programs that would run repeatedly and automatically once triggered by an external or internal operation in the system[1].The first bot “Eggdrop” , was designed by Jeff Fisher in 1993,as a useful feature in IRC channels (*Green and et al – 2000*).

Mainly bots are of two types: benevolent bots and malicious bots. Bots that are used for legitimate and automated tasks are put in benevolent group. While the ones that carry out malicious codes and spread infection and cyber- crimes are put in malicious bot group. The internal working of botnet can be either as centralized one or as a peer-to-peer one. There are varieties of attack vectors performed by these botnets which include DDoS Distributed Denial of Service attack, sending spam emails, spreading malicious codes, phishing, and identity theft and click fraud and so on. There are different detection mechanisms proposed by different specialists. Most of them do work on the principle of network flow characteristics and protocol featuresand most of the approaches share the idea of detecting patterns based on communication flow [2].

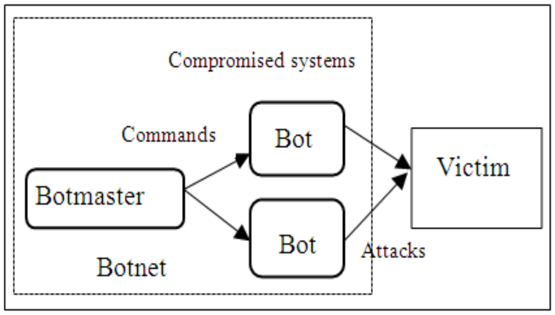

Fig1. Basic Botnet structure

Due to the dynamic nature of the botnets, ways already suggested are finding new approaches each and every day. The security of digital computing infrastructure has become a crucial and essential thing. The proposed work shows the live detection of the botmaster in the internal network by using concepts of compromising the zombies, packet capturing and packet analysing. A strong framework “Metasploit” is been used for the purpose of tracing the botmaster.

2. Related Work

Though there are many mechanisms all around, some of them have proved to be identifying the botnets well. Neural network applications are widely used in the design of botnet detection mechanism because of their high capability of finding abnormal patterns in traffic and they could handle non-linearity [3].Even websites has suggested various methods like response to

CAPTCHA as prevention against bots and malicious agents. Preventing a system from falling as a victim to botnet attack requires great awareness about the confidentiality, integrity and availability of the network infrastructure.

The malicious attackers had then changed their methods of communicating to the bots in various ways. This is when DNS links came into the scene. The technique of DNS tunnelling in which the attackers send the malicious code in the resource record fields of DNS message was carried

out [4]. For detecting suck abnormalities, Shannon entropy methods were proposed. Many applications have the option of grouping systems that show some behaviour of botnets though they are not, and isolating them for neutralizing the probability of any further unknown attacks [5].Where a pipelined approach is used, with incorporated filtering of white and black list. Basically chat-based botnet architecture is characterized in many areas which includes the role of IRC server, code server, and controller and so on.

An efficient method was brought in by using sniffing program. Sniffer program is designed to capture and analyse the communication between botnet master and its clients. It uses the following three parameters for the processing which are the file name, interface name and the count of packets to be captured [6,7]. Nepentheses is a special kind of honeypot used for automatic malware sample collection [8]. The next generation botnet detection systems must be completely independent on command and control mechanisms.

3. Detection Methodology

The proposed work is established for the internal network mainly in the case of any organization, institution etc. Here we depict the situation of DDoS attack by the botnet on to the targeted victim machine. The botmaster sends malicious codes and payloads to the legitimate host machines in the network. These malicious payloads once sent to the host machines are made to be downloaded or activated by the systems itself. Thus the host machines get affected by the bot malware and join the botnet as clients to the botmaster. Later, using these compromised systems known as zombies, botmaster would perform various attacks on other target systems. Here the botmaster is forcing the zombies to perform a distributed denial of service attack on the target system by flooding the victim's NIC using illegal ICMP packets as in Fig 1.

The proposed work is established for the internal network mainly in the case of any organization, institution etc. Here we depict the situation of DDoS attack by the botnet on to the targeted victim machine. The botmaster sends malicious codes and payloads to the legitimate host machines in the network. These malicious payloads once sent to the host machines are made to be downloaded or activated by the systems itself. Thus the host machines get affected by the bot malware and join the botnet as clients to the botmaster. Later, using these compromised systems known as zombies, botmaster would perform various attacks on other target systems. Here the botmaster is forcing the zombies to perform a distributed denial of service attack on the target system by flooding the victim's NIC using illegal ICMP packets as in Fig 1.

The details obtained at each steps are tabulated and stored for future reference. Then the victim machine would scan and find the OS versions, vulnerabilities, services, open ports details from the IPs that are been tabulated based on their count of packets sent. Steps are taken to compromise the systems found harmful from the table information. Here a strong and open source tool "Metasploit" is used for the whole process of tracing back the botmaster. The whole process can be defined using mainly three processes as in Fig 2.

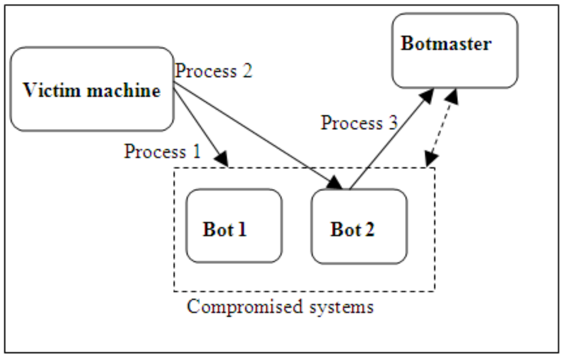

Fig 2.Structureof tracing the botmaster

The following processes can be detailed as:

i. Process1: victim machine willcontinuously
monitor its own interfaces for incoming packets. When it sees more number of ICMP packets coming from any IP(s), then those IPs will be noted. Scanning and other information gathering will be done on the zombies and their environment. Vulnerabilities will be listed based on their OS version, services etc.

ii. Process 2: Customized attack vectors and the
payloads are designed using metasploit framework by run control files. The attack is launched on the zombies and they are compromised by gaining their access. Privilege is escalated accordingly.

iii. Process 3: Sniffing tools are run remotely on
the compromised zombies, other applications can also be run their accordingly. Finally, the incoming packets of the compromised machines are taken and analysed to get the source address of the attacker. There is an easy chance of getting the same source address in the interfaces of different compromised machines, which makes it confirmed to be the botmaster's IP address.

In the process of the work, there are certain assumptions made and they are as follows: The proposed idea of detection is to be implemented in an internal network. When it comes to external network with public IP, it has the limitation of tracing back only in systems that uses Java enabled browsers. The approach then has to be in a different fashion.

The scope of the botnet problem is to find and quantify the highly covert nature of botnets which makes them harder to measure. The above mentioned processes or steps will be automated for the analysing of the incoming packets to the host system and for the detection of the botmaster from them.

4. System Architecture

The system architecture will be the overall

design of the proposed work with its modules shown in Fig 3. It will be tasks done by the victim machine for the whole scenario of botmaster detection. Certain tools and applications are needed for the same.

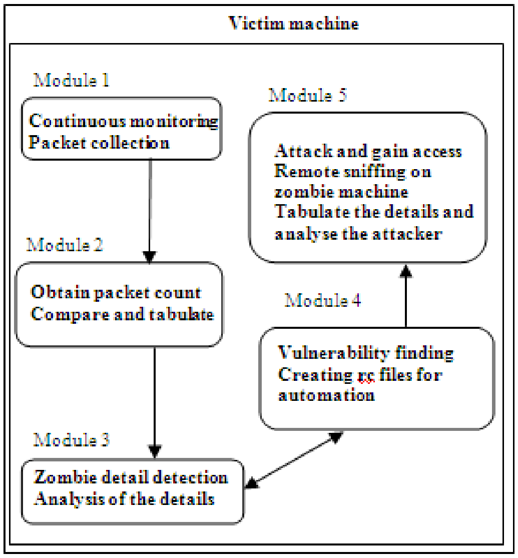

Fig3. Architecture diagram for Botmaster detection

5. Module Description

Module 1 describes the first and the initial phase of the work, where the target machine monitors its incoming packets using packet capturing and sniffing tools like Scapy or Winpcap module, even Wireshark can be used. Thus the output pcap file will be obtained with the IPs of different hosts and their send packets, with the type of protocols. The content of the pcap file is then translated to another output file for convenient parsing and finding the details. Usually the sources sending more ICMP packets are noted for the reason that they may probe pinging of large sized packets which may lead to flooding. The flow of the module 1 can be depicted through the Fig 4 below.

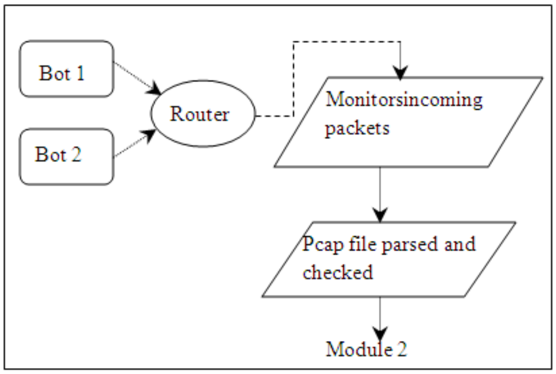

Fig 4.Packet capturing and monitoring

Module 2 describes the process after packet capturing. A threshold value will be set, for the packet capturing in the predefined program which will be linked to the metasploit framework. The output of the pcap file which is parsed is analysed for the count of packet from each IP. Accordingly, the count of incoming packet is compared with the threshold value. The case where the count either exceeds the already set value or can be approximated to that value is noted. And IPs of such hosts machines are noted with their count of packets send.

The details that are obtained are tabulated and stored in the database of the framework. Common vulnerabilities and attack vectors are listed in the tabular form for future reference that has been occurred.

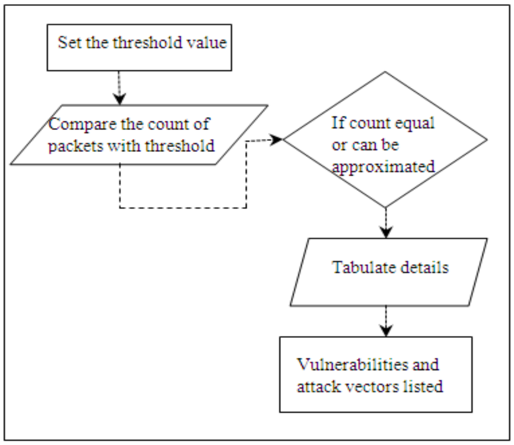
Set the threshold value
Compare the count of packets with threshold
If count equal or can be approximated
Tabulate details
Vulnerabilities and attack vectors listed

Fig5. Setting threshold value and comparing result

Module 3 reveals the information gathering phase done by the target machine on the zombies based on the IPs obtained. Nmap scan is done to know whether the host machines are alive or not, as well as to gather information on their OS versions, services, open ports etc. Thus the environments of the zombie systems are studied. Vulnerability testing tools like Vulcan can be used for finding out the common weaknesses and all the details so far obtained are tabulated.

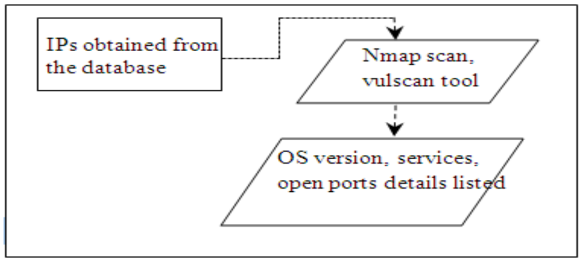
IPs obtained from the database
Nmap scan, vulscan tool
OS version, services, open ports details listed

Fig6. Scanning and vulnerabilities testing

The main activities of Module 4 include adding the existing payloads, exploits to the table in the database. Customized attack vectors are designed by the help of resource (.rc) files. These files are run to automate the attacks. The whole process can be automated according to the OS versions of the different zombie machines. The above mentioned files can be designed in a way where exploit, payload, target parameters, everything can be customized. They are been called from the metasploit module.

Ruby language is used for framing the metasploit framework, Python is also supported. Input parameters necessary are added to the rc files for the execution on the target. These inputs can be inferred from the tables that were listed in the beginning. Thus this phase is all about keeping everything ready for attack or compromising the zombie machines. Fig 7 depicts the flow of this module.

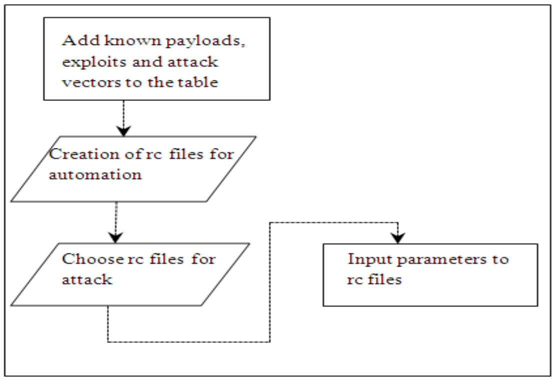
Add known payloads, exploits and attack vectors to the table
Creation of rc files for automation
Choose rc files for attack
Input parameters to rc files

Fig7. Customizing attack vectors

At last the time for compromising the zombies and gaining their access has. As the zombies are already compromised machines by the botmaster due to some vulnerability in them, it is sure that those loopholes do still exist in them. These ways do reveal a path for the target system to exploit and gain access. In module 5, the attack is launched against the zombies by running the rc files. On successful execution of the contents in the rc file, access to the admin privilege or the shell of the compromised system is gained. Techniques to escalate privileges can be handled in many ways.
Backdoor creation will be performed. Metasploit framework is used efficiently for this module. Most of the payloads and exploits are predefined, which has been tested and given results. Sessions are maintained by forcing them to run background in metasploit. Similarly, different rc files are made to run on different zombie machines and they are compromised. The next step is to run the packet capturing and sniffing tool like Wireshark or Winpcap or Scapy module remotely in-order to find the incoming packets on the zombie machines. Thus the clear idea of host machines that are sending packets to these zombies can be drawn. Finally, the source address of the attacker can be traced by the output of this sniffing tool. Mostly, in different zombies when the packets are captured from their NIC, there are possibilities for the same source address appearing repeatedly which gives the clear guess of the botmaster's address.

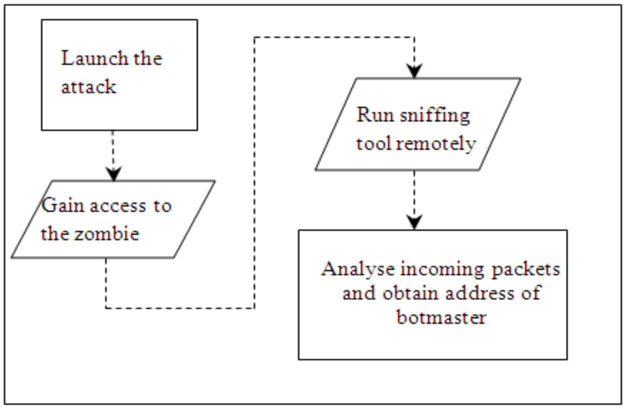
Launch the attack
Gain access to the zombie
Run sniffing tool remotely
Analyse incoming packets and obtain address of botmaster

Fig8. Compromising and tracing the botmaster

It is always to be noted that the counting of zombies from the C&C commands always will lead to vague idea if the botnet size. This is because nowadays, zombies are instructed from different servers and C&C channels [9]. The rise of their presence are also increasing as in new detection technologies are discovered. Yet there are more techniques both known pattern based as well as anomaly based detection schemes [10]. Fig 9 depicts the graphical representation of the quantifying aspect of botnets and their effects.

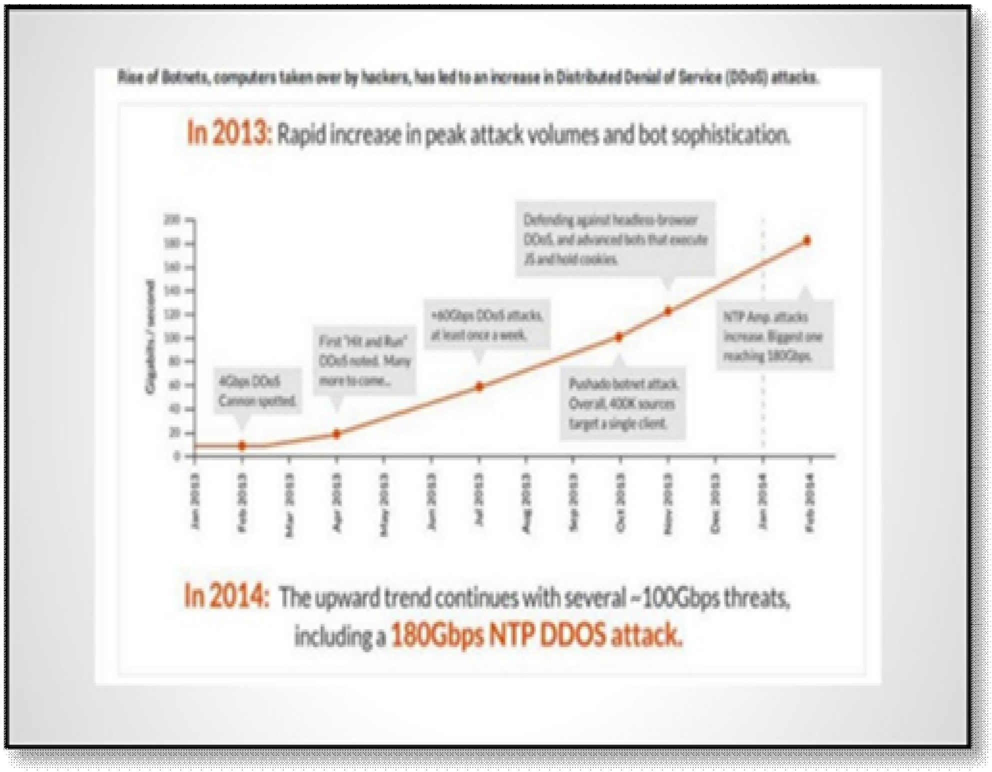

Fig9. Rise of botnets and their effects

6. Conclusion

The botnet era has changed the network and its users into fear of any attack at any time. The live detection of the botmaster using metasploit framework is an efficient method for finding the address of the source. It needs less knowledge on the working principle of the communication flows. By this method, it will be assured that no systems in the process will be put to shut-down or hanged state. Along with this proposed work, the idea of bringing ingress filtering on routers and other network devices is been looked upon for anti-spoofing mechanisms. As well as, access control list will have to be incorporated together in the design for the prevention of host systems getting hanged or crashed from the excessive traffic that arises at the network interface card. The above mentioned both processes will be considered as the future work of this project, to make it work in external networks. This brings in effective and high detection rate with low computational overhead.

7. References

1. Banday, M. Tariq, Jameel A. Qadri, and Nisar A. Shah. "Study of Botnets and their threats to Internet Security." (2009).
2. Feily, Maryam, AlirezaShahrestani, and SureswaranRamadass. "A survey of botnet and botnet detection." Emerging Security Information, Systems and Technologies, 2009.SECURWARE'09.Third International Conference on.IEEE, 2009.
3. Nogueira, António, Paulo Salvador, and FábioBlessa. "A botnet detection system based on neural networks." Digital Telecommunications (ICDT), 2010 Fifth International Conference on.IEEE, 2010.
4. Bos, Herbert, Maarten van Steen, and Norbert Pohlmann. "On Botnets that use DNS for Command and Control." (2011).
5. Strayer, W. Timothy, et al. "Detecting botnets with tight command and control."Local Computer Networks, Proceedings 2006 31st IEEE Conference on.IEEE, 2006.
6. Al-Ahmad, Walid, and Ayat Al-Ahmad. "Botnets Detection Using Message Sniffing." The International Conference on Digital Security and Forensics (DigitalSec2014).The Society of Digital Information and Wireless Communication, 2014.
7. Gu, Guofei, Junjie Zhang, and Wenke Lee. "BotSniffer: Detecting botnet command and control channels in network traffic." (2008).
8. Nazario, Jose. "Botnet tracking: Tools, techniques, and lessons learned." Black Hat (2007).
9. Cremonini, Marco, and Marco Riccardi. "The dorothy project: An open botnet analysis framework for automatic tracking and activity visualization." Computer Network Defense (EC2ND), 2009 European Conference on. IEEE, 2009.
10. Zeidanloo, HosseinRouhani, et al. "A taxonomy of botnet detection techniques." Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on.Vol. 2.IEEE, 2010.