ISSN 2395-1621

BotTracer: Tracing Botnet based on DDoS activity analysis for big data environments

[#1]Sanket T., [#2]Kasturi H., [#3]Sanket S., [#4]Bhagyashree S., [#5]B. Padmavathi

[1]sanket95turkewadkar@gmail.com
[2]deepikahadap25@gmail.com
[3]sanketsawantj11@gmail.com
[4]bhagyashreeshinde2012@gmail.com
[5]b.padmavathi@raisoni.net

[#1234]BE Computer Department Student
[#5]Asst. Prof. Computer Department

GHRCEM, Wagholi, Pune.

ABSTRACT

In today's world data is increasing rapidly and relatively the account of attacks to access that data also increasing day by day. Botnet is one of power impacting attack of distributed denial of service (DDoS) its hard detect in network. In normal attacks we can identify the error but bots cannot create problems it's just overlook the activities in network. When some valuable information is transfer through packets it attacks over[8]. There is bot masters who control all those activities through C&C server (commands & control) which play important role in attack. This attack mostly targets the application layer for their control in network and this type of network is called as botnet. And it's complicated to resolve online when the target is the web server. The difficult part of this attack to identify bots in system so we need to emphasize on how to trace the bots. A efficient and innovative algorithm is needed to tracing bots in network and enable them to cause harm to network.. The DDoS attack traffic is sampled to locate suspicious hosts firstly, then the hosts' packets are collected and analyzed by DPI technology with some DDoS parameters, such as victim, start time of the attack etc. for finding C&C and Servers. This detection model has been implemented, named BTS (Botnet tracking system) at a POP of CERNET. The tests showed the practicability of this model.

ARTICLE INFO

Article History

Received: 20th May 2017

Received in revised form :

20th May 2017

Accepted: 22nd May 2017

Published online :

29th May 2017

Keywords: Bots, Botnet, C&C(Command and Control), BotMaster.

I. INTRODUCTION

In modern world of software development where computer is basic need of professional person. Communication is a first initial step and network is support. In cyber-crimes there many attacks distributed denial of service (DDoS) is one of the major attack[2]. Botmaster control all activities C&C channel is used for communication control several protocols are available. IRC (Internet relay chat) protocol which C&C used. Another protocol is HTTP on which few botnets are reliable[1]. But know it's too old and peer to peer (P2P) replace the place of both protocols[4]. (P2P) protocol is more powerful protocol which is used. Currently, Strom worm and Nugache are most popular P2P botnets. As bots are the performing malicious activities through C&C channel. For attack bots need to be connected to each other by which they form their team called as botnet. This botnet provides the flexible and efficient service during attack. For monitoring the event or process is going on in system which can inform us that something is wrong in system and for standard security the term intrusion detection is refer. Network layer becomes the strong point for botnet in (DDoS) attack[3]. Particularly UDP and ICMP flooding which affect the bandwidth thus facilitating the denial[3]. In Network service traditional botnets used centralized command and control (C&C) architecture for communication between bot master & bots. The current generation of bots started using the peer to peer (P2P) structure to run a more resilient C&C architecture, to disseminate commands through the botnet access[4]. In web server botnet take the advantage of

application level for flooding purpose through which attacker can easily attack on system. The remainder of the paper is organized. Botnet based DDoS attack tools are described. Classification of Botnet based DDoS attacks. Various Botnet based DDoS attack incidents. Our approach is based on the observation that, because of the pre- programmed activities related to C&C[2] , bots within the same botnet will likely demonstrate spatial-temporal correlation and similarity. For example, they engage in a coordinated communication, propagation, attack and fraudulent activities. A really efficient botnets would be structured as an old fashioned spy networks, with decentralized cell botnet machines would not know the C&C address[3], but merely how to cell, as well as one or two machines in the from other cells. Messages would be broadcast by hopping from cell to cell that kind of structure required a lot more thinking at the design phase, but would be lot more resilient to dismantlement by law enforcement forces.

The infected system must be able to talk to the C&C. this entails getting some data out of the machine, through whatever network access it has. The easy case is when machine has full internet access (possibly through some NAT) however, there are some networks (especially business environments) where machines have only their local addresses, there is no NAT, and outbound communications must go through a HTTP proxy[3]. Possibly, the proxy will need some authentication, that may be automatically provided by the OS (windows) provided that OS libraries are used to do so. Thus, HTTP increases the probability of succeeding at connecting an external server, where as IRC can not really handle the "proxy only" case.

Plain HTTP requests may be inspected by nosy firewalls. HTTP may help you here; decrypting HTTPS requests requires installing and maintaining a custom certification authority that produces the fly fake certificates for contacted websites. There are appliances that offer that kind of services, but this is not yet fully wide spread. You want to hide the true C&C server where abouts[5]. If you have a large botnet, then it is a certainly that some sys admin, somewhere, will notice something fishy and try to guess where the C&C is. The HTTP protocol is stateless by main configuration, which means complex communication will be harder to the implement (you have to make some session for handling). On IRC, you can simply open a thread for your clients and then you will have a procedural and single flow of control. Although IRC can be extended by ssl as well, it is not so common as in the HTTP[3]. You will probably have to embed some IRC communication library for the task, it configuring for use ssl on a such way that it will be undetected will be harder. As I know, in the circles of the currently running botnets https is much more common, but most of them is a nightmare from a programmer point of view. For research I would be use unencrypted IRC (easier to program, no need for encryption)[6].

A botnet or robot network consists of thousands or sometimes millions of computers infected with the robot program. This piece of code enable attacker a.k.a bot header to remotely control the infected computer and make them perform tasks. Because the zombie computer continue to appear acting normal to their users.

II. RELATED WORK

As botnet continue to gain momentum in the form of spam, bots, click fraud, large-scale identity of thefts and proxy nets and a large-scale distributed denial of service (DDoS) attacks. Generally, a botnet may display following behaviour patterns likes signature-based, anomaly-based, DSN-based and mining-based. 1) Signature-based: Ntop is a very powerful network sniffing and statistics gathering tool. When used along with the darknet, we can analyze botnets and detect them. The idea of darknet evolved from honeynets, and is quite often underestimated[1]. Snort is used along with Ntop and darknet to a log alerts. Snort is used for detecting botnets based on their own signatures. If the traffic matches any of these botnet signature, then an alert is recorded for the log file. The particular packet can be further analysed in wireshark. 2) Anomaly-based botnet: Generally, antivirus find it very difficult to detect worms that use dynamic codes[1]. In this case, a DDoS attack can be detected using cisco NetFlow analyser. Although based botnet detection techniques is detect unknown botnets as well as unlike a signature based detection, sometimes an IRC network may be detected. 3) DNS-based : DNS-based botnet detection technique is based on domain name system information generated by a botnet.4) mining-based: A recent development in classification and clustering is models of botnet detection saw the rise of botminer.

BotTracer to detect these three phases with the assistance of a virtual machine techniques. To validate BotTracer we implement prototype of BotTracer based on VMware and window XP professional. The results show that BotTracer has successfully detected all the bots in experiments without any false negatives[4]. Detecting botnets not easy to trace. However, in existing system designing an effective P2P botnet detection system is face with several challenges. First the P2P file sharing and communication applications such as emulae and skype are very popular and hence C&C traffic of P2P botnets can easily blend into the background P2P traffic[4]. This challenge is further compounded by the fact that bot compromised host may exhibit mixed patterns of both the legitimate and botnet P2P traffic (eg. due to the coexistence of a file-sharing P2P application and a P2P bot on the same host)[3]. Second, modern botnets tend to use increasingly stealthy ways to perform the malicious activities that are very hard to be observed in the network traffic. For eg. some botnets send spam through the large popular webmail services such as hot mail which is likely transparent to the network detectors due to encryption and overlap with legitimate email use patterns. Third, as amount of the network traffic grows rapidly, the deployed detection system is required to process huge amount of information efficiently[8]. And need the improvement in techniques.

1. Botnet life cycle:

Life or duration of a botnet depends on purpose of creation of the botnet. The lifecycle of a botnet can be defined in the following stages.

1) Attacker (bot-herder) decides on initial bot parameter like infection vector , payload, C&C detail.
2) Register a DDNS (Dynamic DNS).So that the botnet controller can change IP address and create new bot controller or malicious content host whenever required.
3) Register a static IP.
4) Bot-herder launches or seeds new bot(s).
5) Bots spread.
6) Attack a victim machine by DDoS or spam or phishing etc.
7) Losing bots to other botnets.

Botnets can change their command and control (C&C) content, protocol, and even C&C servers- ofter termed as fast flux (a techniques to hide the phishing and malware delivery behind the network of a compromised hosts, which keeps on getting modified due to new zombie machines.) botminer monitors two planes for botnet detection : namely C-place(C&C communication place) and A-plane (malicious activity plane)[5]. Clustering occurs in cplane by finding statistical distribution per unit time or per unit packets. Generally the communication between a local host and a remote server consist of protocol, source IP, destination IP[7].

2. Different types of bots:

During research it's found that there are many of bots in the wild. In this section we present some widespread and well-known bots. Here introduce new concepts of each piece of malware and further more describes some of features in more detail. In addition, show the examples of source code from bots and list parts of their command set.

1. Agobot/Phatbot/Forbot/Xtreambot
   This is probably known bot. Itself is written in C++.
2. SDbot/Rbot/UrXbot/….
   This family of malware is the moment the most active one : Sophos lists currently
   Written in very poor C.
3. MIRC-based bots GT Bots
   Since there are so many differents versions of them that it is hard to get an overview of all forks.

III.PROPOSED SYSTEM

A botnet tracking system(BTS) [1] is designed for traceing botnet this idea is proposed. The DDoS attack data is provided by NBOS which is a network performance and threat detection system that can detect DDoS attacks in real-time on the border of a network.

Here we show the structure and workflow of the system. BTS represent the three parts: Bot Locator, Packet Collector and Data Analyzer. The main function of Bot Locator is to analyze DDoS attack information in NBOS' DDoS database, filter the suspicious hosts according the procedure described then send the hosts' IP to Packet Collector. After that, related DDoS attack's parameters is send to Data Analyzer as triple(SIP, Target_IP, StartTime) by Bot Locator also. The triple is sent if and only if SIP equals to a suspicious host in a DDoS attack, that means the suspicious host participates another attack; Packet Collector collects non-attack packets of suspicious hosts; Data Analyzer analyzes the packets provided by Packet Collector and the triple DDoS parameters[1].

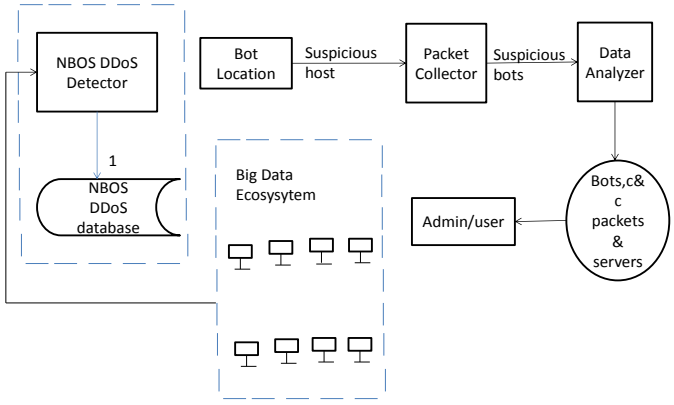
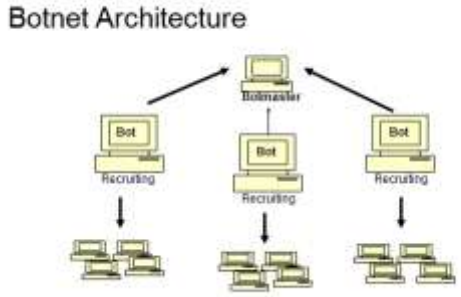


Fig:Architecture Diagram for Botnet

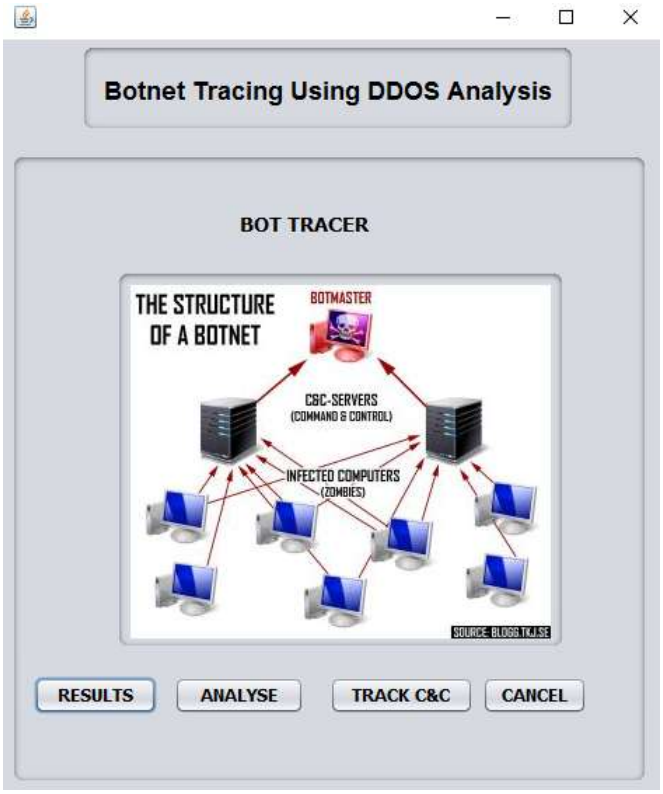
The significant contributions of the proposed system approach are listed as follows:

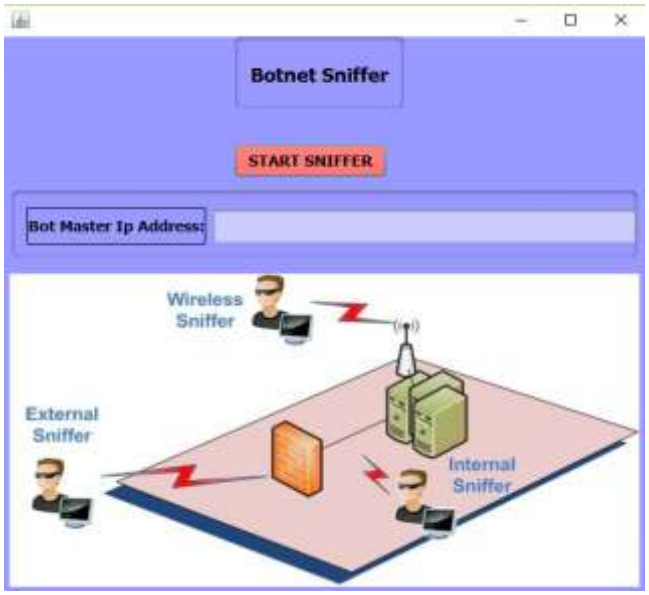
1. A unique bot detection system is built on the essential characteristics of the bots, namely, failure traffic and communication traffic. Each phase detects the presence of bots separately in parallel, hence reducing the overall detection time with high accuracy.
2. An online bot detection system is developed that processes the network data in the form of the data stream.
3. An effective and efficient algorithm for the detection of bots using the network failure traffic only is proposed that be capable of detecting newly infected hosts before they start communicating with the botnet.
4. Distributed Hadoop Map Reduce paradigm is adapted to process big data to achieve the scalability[7].
5. A model framework is implemented and evaluated based to realworld network traffic, which has demonstrated high accuracy.

Architecture:



IMPLEMENTATION WORK:





NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.

NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X , Linux , Solaris and other platforms supporting a compatible JVM.

1. Programming Language:

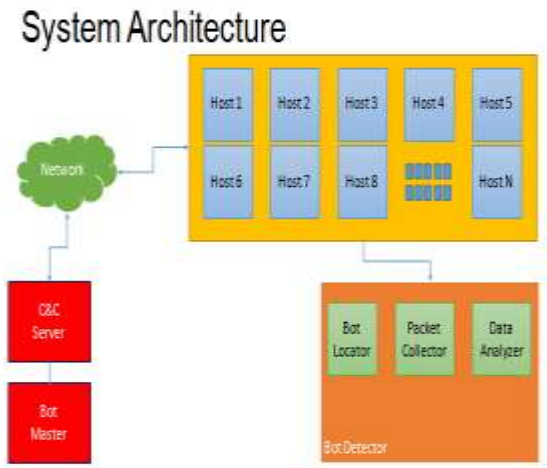
The Java programming language used and latest version of jdk8 is used for development.
It is developed in Eclipse IDE platform.
Windows is the operating system chosen for the module.
The database used in the project is MySQL and Hadoop

2. MySQL :

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.
The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software.

Component Design

A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.



Introduction:

In the project implementation phase of the document, we discussed here about database required for the project. Important module design for the implementation of the project. The algorithm used in the project. Software required for implemen-tation of the application. All we have discussed in the Software implementation phase of the document.

Tools And Technologies Used:

- Eclipse provides tools for developing system.
- My SQL for storing databases.
- Apache Hadoop Server for server side services.

1. JDBC:
Java Database Connectivity, commonly referred to as JDBC, is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the JVM host environment.

2.Jdk 8:
Oracle released a new version of Java as Java 8 in March 18, 2014. It was a revolutionary release of the Java for software development platform. It includes various upgrades to the Java programming for improving JVM, Tools and libraries

BOTNET TRACING WITH DDOS ATTACK DATA:

Detection Procedure:

According to the analysis above, we proposed a method for locating C&C traffic by DDoS attack parameters and DPI. The procedure consists of two phases mainly:
Phase 1: Locate the suspicious hosts with DDoS attack data provided by IDS.
Phase 2: Analyze the suspicious hosts non-attack traffic and DDoS data to find C&C.

Bot Filtering:
Firstly we assume a bot uses its genuine IP to participate in attacks. We focus on this kind of DDoS activities as the start point of research work. It is not very difficult to filter bots address if DDoS parameters are available on this occasion. As bots are recycled, and the bots number of a botnet in a concerned network (A certain IP address, such campus network, notes as X_Space) is limited, a host within the network could be times (more than M1 supposed) and each of the attck has only a few SIP (less than P1 supposed). The definitions and procedure of the filtering are as following:

Definition:
DDoS_size: the number of SIP involved in a DDoS attack within X_Space
DDoS_num[IP]: the accumulated count of DDoS attacks participated by an IP address

Procedure:
obtain a DDoS data from IDS
if DDoS_size< P1 then
for each SIP in DDoS do DDoS_num[SIP] =
DDoS_num[SIP] + 1
for each DDoS_num do
if DDoS_num[IP] > M1 then yield IP as a suspicious host

Detection Rule:
After a suspicious host is located, its non-attack traffic can be collected and the C&C packets should be in the collected traffic if the host is a bot. If the maximum interval between about receives C&C and starts the attack is T, only the packets within the interval T are needed to be inspected. When a traffic-collected suspicious host A is reported being involved in a DDoS attack as SIP, rules are needed to detect C&C packets from the traffic in the interval. To construct the rules, another DDoS parameter Target_IP is needed, the victim of a DDoS (section 3). After that, we give
out two rules to find C&C packets follows:
Rule #1
Perform DPI in all the non-attack traffic in the interval[Start Time-T, Start Time] with the characteristic Target_IP in packet content. If a match is found, the packet probably contains a C&C command. Further manual check could verify the conclusion. If the C&C packet uses TCP protocol, its SIP must be the C&C server. Rule #1 may be ineffective if the C&C is encrypted. Generally, the behavior of hosts communicating each other is random, while there must be communications between botnetcontrollers and bots before the start of

DDoS attack, so thedetection rule #2 is constructed with this deduction.

Rule #2
Create set B which contains hosts communicated withsuspicious host A from StartTime-T to StartTime. K attackscorrespond K sets, i.e. B1,...,Bk. If the intersection of Bi(i=1,…,K) contains one IP only, then the IP is very likely abotnet C&C server. If the intersection of Bi (i=1,…,K) containsseveral IP, then increase K gradually to reduce the intersectionsize. Discard the current search when K exceeds thresholdMaxcount .
According section 3, if K<N-M, C&C packets should befound theoretically.
Apparently, achieving rule #1 needs only one attack'sinformation, while rule #2 needs K, so a certain sequence existsbetween rule #1 and rule #2. Rule #2 is enabled when rule #1 isineffective. Below are the detailed definitions and procedure.

Definition:
A: suspicious host being filtered;
Target_IP: victim's IP;
Payload(IP): packet payloads to IP;
Map(IP, Text): search IP in Text;
Gather(IP, T): the set of each address which has communication with IP in time T;
Intersection(Gather(IP, T), K): the intersection of K address
Sets

Procedure:
If Map(Target_IP , Payload(A)) is True then
bot , C&C Server = A , opposite IP of A
else while K <= Maxcount
If detecting position is at theborder of X_Space, the scenarios that both DDoS and C&Ctraffic could be captured are (c) and (f), that means our botnet
if Intersection(Gather(A,T),K).size equals 1 then
bot, C&C Server = A,
Intersection(Gather(A,T),K)
else K = K + 1

IV. CONCLUSION

Here we find the solution on problem of botnet how to trace the bots in working system. And there are several method to tracer based on honeynet. In which we detect the bots at the border and locate the C&C channel. Here we also prove the reverse tracking of botnet by finding the bots through C&C servers is possible. . A system named BTS were implemented and installed at a POP of CERNET to test the idea. Two algorithm are used here such as failure traffic based bot detection. And bot detection using communication traffic. In future, we will focus on the online processing of network data on Hadoop itself .

REFERENCES

1. A Research paper on BotTracer: Tracing Botnet based on DDoS activity analysis for big data environments.

2. Wei Ding, Wentao Ren Zhen Xia, Li Wang "Botnet Tracing Based on Distributed Denial of Service Activity Analysis", IEEE 8th International Conference on BioMedical Engineering and Informatics (BMEI 2015).

3. Wu Xianghua, Cao Lijun "Analysis and Design of Botnet Detection System", 2012 International Conference on Computer Science and Service System.

4. Esraa Alomari, B. B. Gupta, Shankar Karuppayah, "Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art", International Journal of Computer Applications (0975 – 8887) Volume 49– No.7, July 2012.

5. Perdisci, R. Wenke Lee." Building a Scalable System for Stealthy P2P-Botnet Detection". In proceedings of Information Forensics and Security, IEEE Transactions on Volume:9 , pages27 - 38,Nov 13. [9] W.T. Strayer.

6. Sajjad Arshad, Maghsoud Abbaspour, Mehdi Kharrazi, Hooman Sanatkar," An Anomaly-based Botnet Detection Approach for Identifying Stealthy Botnets", iccaie2011.

7. Dileep Kumar G , Dr CV Guru Rao, Dr Manoj Kumar Singh, Mohammed Kemal," Network-based IDS for Distributed Denial of Service Attacks", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) , Volume 3, Issue 1, January – February 2014 .

8. Thejiya V, N Radhika, B Thanudhas, "J-Botnet Detector: A Java Based Tool for HTTP Botnet Detection", International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391 Volume.

9. Muhammad Mazhar Ullah Rathore, Anand Paul, "Real-Time Big Data Analytical Architecture for Remote Sensing Application", IEEE journal of selected topics in applied.

