International Journal of Latest Trends in Engineering and Technology
Vol.(10)Issue(2), pp.096-102
DOI: http://dx.doi.org/10.21172/1.102.15
e-ISSN:2278-621X

DBOT-CCC: DETECTION OF BOTNET COMMAND AND CONTROL IN CLOUD NETWORK

Parneet Kaur[1], Dr. Anuj Gupta[2]

Abstract- In recent time, the business companies are leveraging the cloud service that significantly improve their business oppotunities by enabling the usage of technological benefits of cloud network. Companies can host their business over the cloud network connected with internet that offer access to theem in a scalable, reliable computing resources. However instead of technological growth and benefits of the cloud resources, a less significant research efforts has been put in place to protect the cloud networks and resources hosted over it. Whenever a end user is connected with the internet try to access the cloud resources, the attacker may intercept into it and may perfome complete damage to the company resources.

In this research, the DBOT-CCC is presented that is a Detection of Botnet Command and Control in Cloud network by applying the novel approach of honeypot and DNS traffic analysis. Most of the botnet detection techniques are either Honeypot based or Rule based techniques which are not well suitable in cloud infrastructure scenario because of their resource intensive nature but by integrating the light-weight application of honeypot application for data logging and then by applying the monitoring of outbound DNS traffic inspection is lead to the detection of botnets in cloud network. The DBOT-CCC takes the advantages of Honeypots that is a proactive security approach for detection of attacks.

Keywords- Cloud Computing, Botnets, Honeypots, Cloud Security, IDS, Reputation Databases.

1. INTRODUCTION

The provisiong of providing the cloud's resources hosted somewhere on the internet [1]. When we talk about the Cloud Service Providers aka CSP which typically offers the services in both ways- softwares and hardwares, the refined software services such as databases and rough resources such as storage and computing processing power required for high end computing applications. The customers often access these resources based on a model called pay-as-you-go. The cloud service providers make charges on the provided resources and their usage by the end customer. Moreover in cloud services, the companies can even chose or in effect rent a computer resources rather than to buy or invest on them. For example, if a customer of cloud resource has came to know that the requirement is fulfilled and has thus scale down his demands of cloud services. The case in reverse scenario is also true, if a customer feels that the requirement is under-provisioned, he can scale up the demains of cloud usage [2].

In internet history, the attacks have grown exponentially that lead to the emegence of more advance defense techniques to protect the network such as firewalls, intrusion detection and prevention system, anti-viruse softwares, unified threat management appliances. However as observed in recent trends in atatcks such as WannaCry, SambaCry, Petya[3-7] depict the need of more powerfull security to prevent and mitigateagainst these kind of advance breace in the network. Moreover these attacks not only targeting the specific services, Operating system and applications , their target is more on the weakness running over the infrastructures since a long time. For example, there is a belief in internet user community that Linux OS is secured than other operating systems but in recent times, the attacker also breach the security of linux even. SambaCry malware hijacked the Linux servers. In more technical manner, The SambaCry malware exploited the linux samba server vulenrability that would allow the attacker to upload a piece of malious program on it and execute it to gain the full access of the server. Recent trends in cyber attacks depict new attacks are hard to detect [8], [9]. Thus, forensics is required for understanding these attacks and measuring their impacts. This is helpful to recover the system back to a safe state and counter them in the future. From a network point of view, attacks are more distributed. Botnets are one of the most major threat [10] and have evolved from a centralized model towards a decentralized, highly scalable architecture [11] based on peer-to-peer (P2P) networks [12].

For detection and collection of wide spreading attacks, the analysis has been shifted from the end user's PC to the large coverage of network such as Internet Service Provider's router, Gateway traffic inspection etc. However to monitor the large segment of network, there is need to handle the large volume of data, but many forensic tools and techniques are still manual process which build the need to new analytical tools to be applied to analyse large volume of data [13-16].

This paper discusses the detection of botnet attacks in cloud based infrastructure. The growing rate of cyber attacks is also applicabe in the segment of cloud networks. Section 2 gives the overview of botnets including attacks, structure and concepts. The detailed design of proposed system and developed modules are presented in section 3. The experimental results in the form of case studies are presented in section 4 and then the research is concluded in last section.

[1] Department of CSE, CGC College of Engineering, Mohali, Punjab, India
[2] Department of CSE, CGC College of Engineering, Mohali, Punjab, India

2. BOTNET PRELIMNIARIES

Figure 1 illustrates an example of attack scenario using botnet. First, an Internet-accessible computer is infected by a botnet malware while reading emails, downloading files and browsing web pages, etc, in the Internet. Then the bot-infected computer searches for the corresponding C&C server to obtain further instructions for continuous actions such as attack target and time.

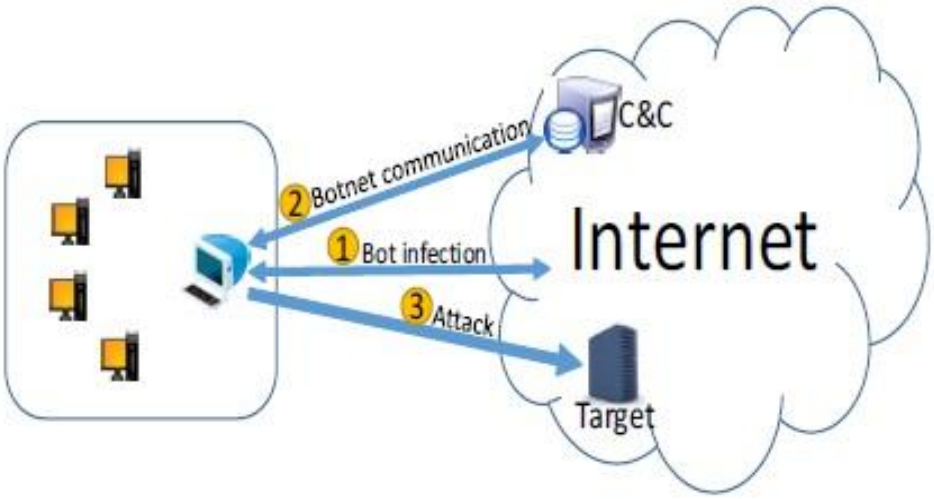

Figure 1. Typical steps in Botnets

In a typical scenario, when a end user access the cloud resources, he simply need a PC connected over the internet. When the user is trying to access the the cloud-enabled resources, an intruder can hijack the connection and can gain the control of the user's resources hosted over the internet. The figure 2 depicts the such attack case that can be quite possible in cloud network.

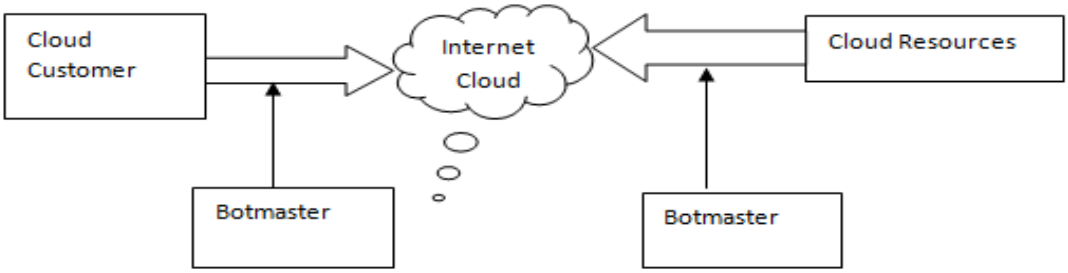

Figure 2. Attacker can intrude the cloud network

3. DBOT-CCC PROPOSED DESIGN

Here we discuss the proposed system that is implemented over the cloud network to detect the botnet command and control domains/Ips. Figure 3 depict the system design of developed framework that contain eight building blocks of the developed system:

3.1 M1- OwnCloud Infrastructure

In the module, the cloud infrastructure is created and establised using open source desktop based cloud- OwnCloud [17-18]. It is client-server architecture based software for hosting the file services over the internet. The OwnCloud is quite similar with the DropBox with the prime difference is that its open source and free to use. On this cloud setup, the lighet weight honeypot application are developed and build over it to log the attack data.

3.2 M2- Honeypot application

Honeypot is a resources which are to be attacked by the attackers. In this design, the low interaction honeypot- Dionaea is customied by removal of unnecessary application over it such as GNOME , X-server etc.

3.3 M3- Network Logs

The network communication are captured and stored in raw dump files in a automatic manner. The shell script is build to sniff the network packets using TCPDUMP utilitu and log them into raw dump files which are further taken as input yo M4 modules.

3.4. M4- PCAP parser

This module is completely self-developed that parse the captured network logs using python DPKT library. The main function of this module to replicate the network packets and stored them into a database including metadata of a packet such as source IP, destination IP , source port, destination port, number of bytes exchanged between source and destination.

3.5 M5- DNS Outbound Query/Response Parser

This module is also self-developed using python DPKT library. The outbound DNS queries and their corresponding resolved IP address is extracted and logged into a database. During the botnet communications, it is a possibility that multiple domains are queried but only few domains are resolved into IP address who are active at that particular time window. Those domains are interesting part for our our further research. The comulative sum (CUSUM) [19] algorithm is applied here that determine the number of domains queires and answered in pre and post infection of botnet communication. The main characterstic of the botnet is that when they try to communication with the botnet domains, there is slight increase of the traffic and when their taks is finised, there is decrease volume of traffic, this is so called the change point detection in botnet outbound network dialogs which lead us to detection of botnet domains.

$$S0 = 0$$

$$Sn + 1 = max\ (0, Sn + Xn - Wn)$$

Certain threshold value has been set, whenever the value of S exceed the defined threshold value that indicate the change point detection.

3.6. M6-VT Labelling

In this module, the domains names and response extracted from Moule M5 are checked with passive DNS records in websense database that flag the domain as blacklisted domain. These blacklisted flag is a indication of further need to apply analytical techniques on pre and post infection traffic corresponding that domain.

3.7. M7-Reputation Checker

In the desined and developed system, the botnet feeds from global feeds are integrated from intel cricial stack API. The feed form various repoutation databases are regularly fetched and updated into reputation databases. The cron tab entry has been made in linux to regularly pull the list of reputation feed into linux which are further parsed to make it machine digestible format and directly usable to match the outcome of domain to reputation database.

H. M8-Feed Parse- The python code is developed which parse the system files pulled from intel crticial stack API. The data pulled from this API is raw data which further need to be parsed to make it directly usable and machine digestible. The outcome in the form of outbound domain queried and resolved are matched with the global reputation databases to make it validated.

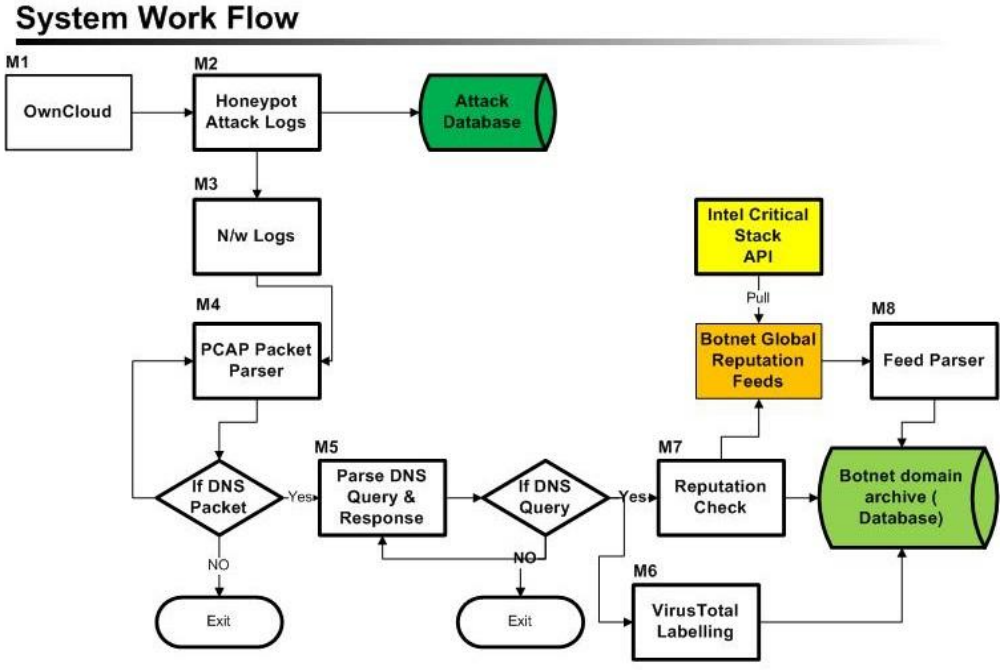

Figure 3. System Design Flow

4. WORKFLOW OF THE IMPLEMENTED SYSTEM:

1) The honeypot application is implemented over the OwnCloud resources for attack data capturing and logging into a database.
2) The network traffic is logged in the form of PCAP logs that are further processed through packet decoder module using DPKT library for parsing DNS packet query and responese
3) The network logs in the form of pcap dump files (raw files) are feed as input to file crawler module that would search for pcap dumps in a directory
4) The pcap dumps are parsed through the DNS parser module which will check for DNS communication by checking port 53 communications in packets and making the flag as standard query and response
5) If it is a standard query, the DNS query and responses are extracted using dpkt library.
6) The DNS rcode (response code) are checked and feed to next module for domain check.
7) The Feed engine will extract the global reputation feeds from Intel critical stack databases, parse them and log into database as archive.
8) The extracted domain query and responses are matched with reputation databases and flag as botnet domains.
9) The database archive is list of botnet domains extracted from global feeds and our network communications.

5. DATA CAPTURING- NETWORK DIAGRAM

Figure 4 depict the network diagram of data capturing and logging the network traffic as PCAP dump files. The OwnCloud machine is placed inside the network of the organization and the network tarffic orginated from the OwnCloud is captured using the tcpdump utility. The attack events are logged using light weight honeypot application running on the same machine.

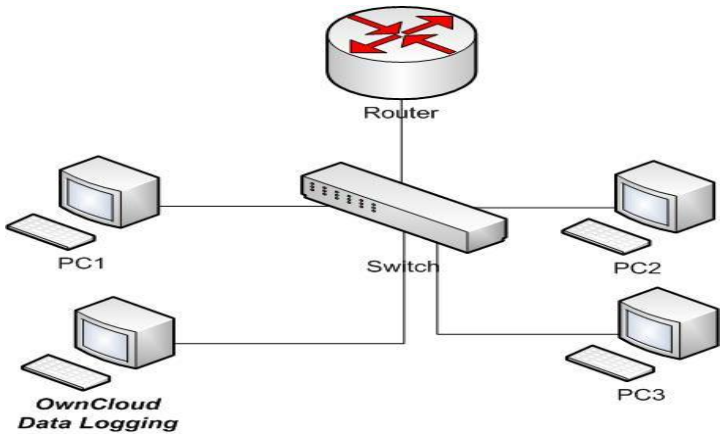

Figure 4. Data Capturing Network Diagram

6. EXPERIMENTAL RESULTS

Here the experimental results in the form of botnet domain queries responsed are highlighted that are detected by the developed system. The developed system is tested on the ubuntu 16.04 linux distribution, 32-bit architecture with 2GB of memory. The idea behind using the linux ubuntu OS is that software used in the development are open source and free to use.

6.1. Running OwnCloud

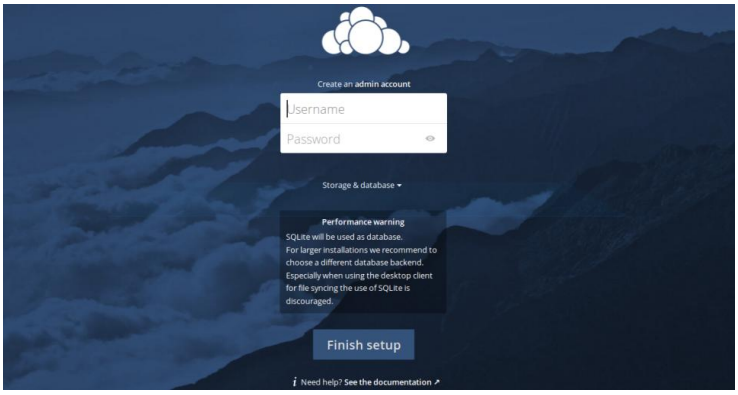

Figure 5. OwnCloud Interface

During the deployment for duration of two months, among all the traffic logged and captured, a sample pcap of size 97.3MB were analysed as case study of bot infection analysis. Figure 6 depict the port wise distribution of network usage during that

time window. It is clearly visible the change points in network communications. Somewhere the network connections were becoming very high and somewhere these are normal running as smooth network communications.

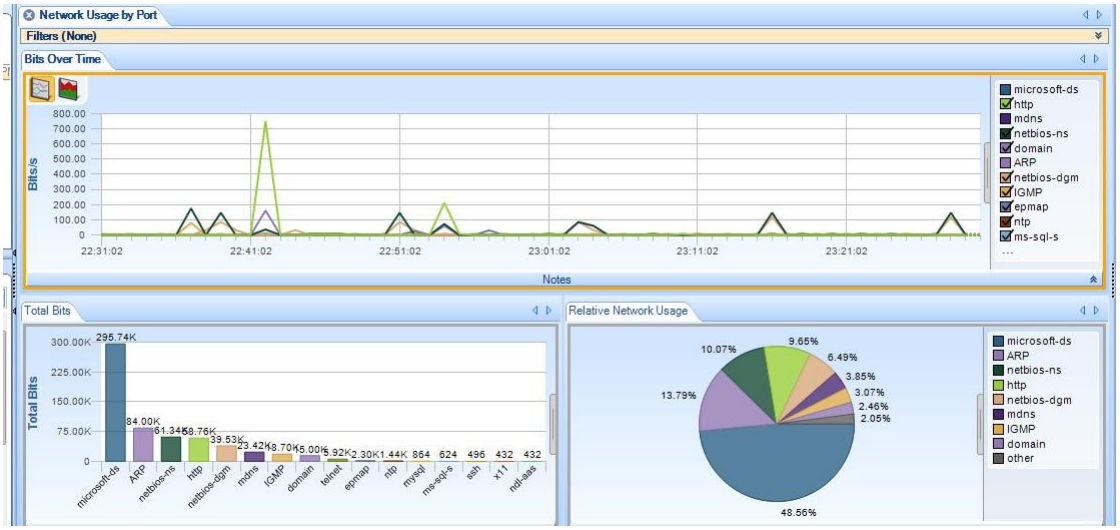

Figure 6. Network usage distribution

6.2. Domain Queried and Responses

Figure 7 depict the DNS communication recored in the sample pcap file. It is clearly observed the peaks of DNS packets in certain time intervals. The packets related to these peaks are very much intereted for further analysis. The pre and post traffic during the observed peak window are analysed to extract the possible botnet communication.

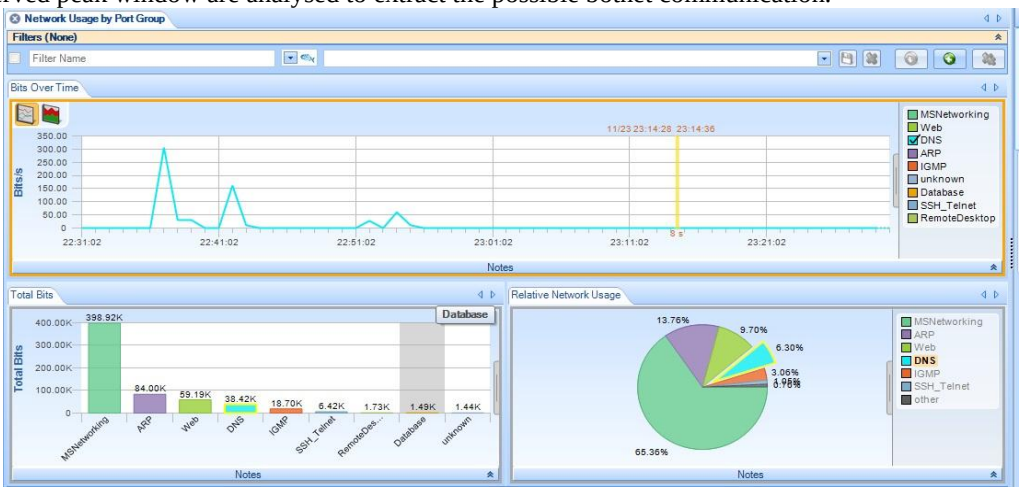

Figure 7. DNS Change Point Detection Visulization

6.3. Blacklisted Domains:

6.3.1. Feed from Intel Critical Stack:

Below snapshot indicate the feed of intel cricial stack API integrated with the system. As depicted these filed is not machine digestible format and can not be directlty used to check the reputation status
of botnet domains. In a similar way, 37 such global feeds are currently integrated and working along with the main system.

#fields	indicator	indicator_type	meta.source
158.69.x.x		Intel::ADDR	https://feodotracker.abuse.ch/feodotracker.rss
178.62.x.x		Intel::ADDR	https://feodotracker.abuse.ch/feodotracker.rss
8.8.x.x	Intel::ADDR	https://feodotracker.abuse.ch/feodotracker.rss	
107.170.x.x		Intel::ADDR	https://feodotracker.abuse.ch/feodotracker.rss

6.3.2. Processed Global Feed – Database

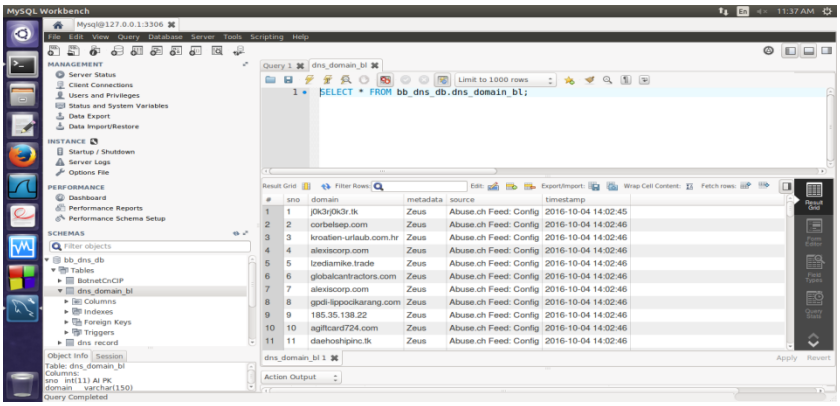

Figure 8. Blacklisted domain database

6.4. Botnet Command and Control domains:

Figure 9 clearly indicate the bot communication patterns exchanged between the botnet command and control domain/IP 60.10.x.x. The domain extracted are also macthed with blacklisted databases such as Zeus Tracker, Fedeo Tracker etc.

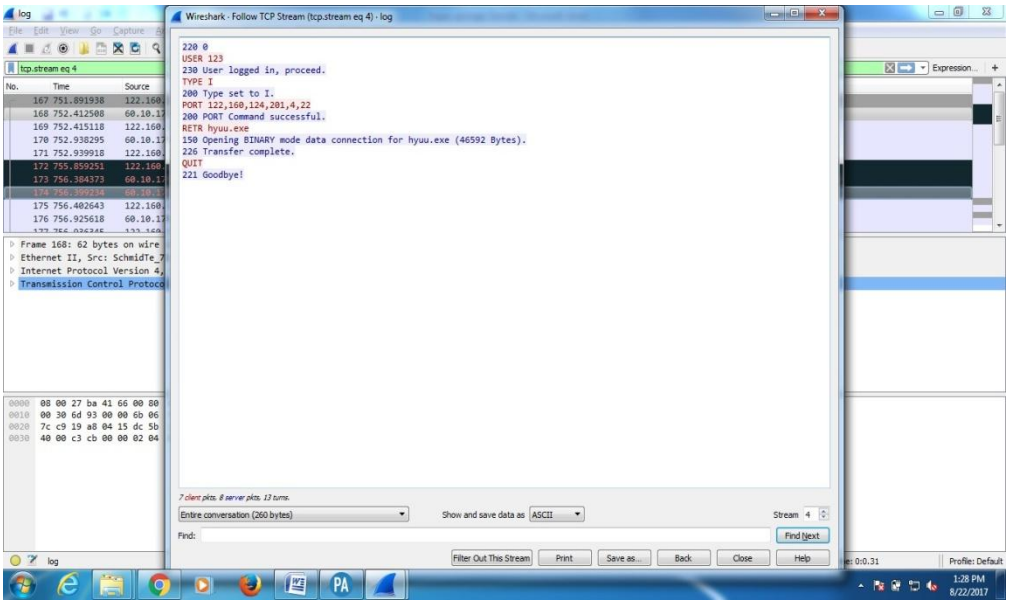

Figure 9. Botnet communication

7. CONCLUSİON AND FUTURE WORK:

The cloud computing technology is growing area that make easy life for the customers and the companies who want to run their business at low cost as the cloud computing enable the pay-as-per usage flexibility to the users. The user community start to shift their data on the cloud resources which is somewhere on the internet. Despite of its easy usage of technology, it is vulnerable to the attacker community as internet-connected resources can be hijacked by the attacker. One of these attacks is botnet which are remotely controlled by the botmaster. In today's internet enabled infrastructures, number of cyber crimes are increasing as noticed in recent attacks of WannaCry, SambaCry, Petya kind of attacks. To address these challenges in the field of cyber security, botnet detection system is developed for cloud network known as DBOT-CCC: Detection of Botnet command and control in Cloud network. This paper discusses the developed system and its capability of botnet detection and its validation through blacklisted databases. It is clearly indicated that the developed system is capable to detect the botnet command and control domains. The addition of more number of botnet features to detect advance botnets is remained as future work. Also clustering of similar botnet domains by applying the machine learning model is our future research into it.

8. REFERENCES:

[1]. Armbrust et al., 2010, A View of Cloud Computing
[2]. BOT-CLOUDS, The Future of Cloud-based Botnets
[3]. https://www.symantec.com/connect/blogs/what-you-need-know-about-wannacry-ransomware
[4]. https://en.wikipedia.org/wiki/WannaCry_ransomware_attack
[5]. https://en.wikipedia.org/wiki/Petya_(malware)
[6]. https://www.malwaretech.com/2017/06/petya-ransomware-attack-whats-known.html

[7]. https://www.bleepingcomputer.com/news/security/Linux-servers-hijacked-to-mine-cryptocurrency-via-sambacry-vulnerability/
[8]. W. Wang and T. E. Daniels, "A graph based approach toward network forensics analysis," ACM Trans. Inf. Syst. Secur., vol. 12, no. 1, pp. 1–33, 2008.
[9]. N. Liao, S. Tian, and T. Wang, "Network forensics based on fuzzy logic and expert system," Computer Communications, vol. 32, no. 17, pp. 1881–1892, 2009.
[10]. A. networks, "Worldwide infrastructure security report (2009 report),"Tech. Rep., 2010.
[11]. G. Masters, "Mariposa Botnet Mastermind Nabbed," July 2010. [Online]. Available: http://www.scmagazineus.com/mariposa-botnet-mastermind-nabbed/article/175721/
[12]. P. Porras, H. Sadi, and V. Yegneswaran, "A Multi-perspective Analysis of the Storm (Peacomm) Worm." [Online]. Available:http://www.cyber-ta.org/pubs/StormWorm/SRITechnical-Report-10-01-Storm-Analysis.pdf
[13]. "Safeback," http://www.forensics-intl.com/safeback.html.
[14]. J. McHugh, R. McLeod, and V. Nagaonkar, "Passive network forensics: behavioural classification of network hosts based on connection patterns," SIGOPS, vol. 42, no. 3, pp. 99–111, 2008.
[15]. V. Corey, C. Peterman, S. Shearin, M. S. Greenberg, and J. V. Bokkelen, "Network forensics analysis," IEEE Internet Computing, vol. 6, pp. 60–66, 2002.
[16]. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Oct. 2004.
[17]. https://owncloud.org/
[18]. https://en.wikipedia.org/wiki/OwnCloud
[19]. The CUSUM algorithm a small review, Pierre Granjon June 22, 2012