RESEARCH ARTICLE OPEN ACCESS

Analysis and Detection of Botnets and Encrypted Tunnels

Dharna [1], Pooja Singhal [2]
Department Of Information Security and Management
Indira Gandhi Delhi Technical University for Women
Delhi - India

ABSTRACT

A botnet is a collection of compromised systems. A botnet has a bot-master which identifies the vulnerable systems and compromises them by injecting a malware code and remotely controls all these compromised systems using Command-and-Control Infrastructure. These compromised systems are bots. Thus, a botnet is a network of bots. These bots receive commands from bot-master to perform various malicious activities like Distributed Denial of Service (DDoS) attack, phishing, sending spam emails etc. Nowadays, Botnets have become a major threat to online ecosystems. Thus, Analysis and detection of the botnets has become a major research topic in recent years. Nowadays, botnets are relying on anonymous networks to hide their existence. Anonymous communication implies that no one will be able to retrieve the identity of the users in the network. The Tor browser is the most widely used anonymous network among botnets. The Tor aims to eliminate the mapping between user and services or servers by hiding the user's IP address and thereby blocks the user identification and communication tracking. The Tor browser provides anonymity to all of its users. Botnets are now using tor anonymity due to which they hide themselves and it becomes difficult to detect them. The proxy servers are also used to hide the identity of the users. Browsing using Proxy server changes the IP-address of its user due to which backtracking is difficult and it becomes extremely difficult to detect if a chain of proxies is used. This paper aims to identify several ways to reveal the identities of the bots and their bot-master that are using tor or any other anonymous network.

***Keywords* :—** Bot-master, Botnet, Bots, Command-and-Control Infrastructure, Distributed Denial of Service (DDoS) attack, Tor Network.

I. INTRODUCTION

Botnets are becoming one of the major threats to Internet security. Internet users have been attacked by widespread viruses earlier, but now scenario has changed. Now attackers are no more interested in just infecting large number of computers on the network, in-fact their interest has been shifted to compromising and controlling the infected computers for their personal profits. This new attack brings the concept of Botnets over the global network of computers. The term "Botnet" comprises of two terms "Bot + Net". Bot can also be called as a zombie. A Botnet is a network of computers in which a Bot-master (attacker) compromises vulnerable systems by injecting a malware code and after compromising, these infected systems can be controlled by the Bot-master remotely via commands. Bots can receive commands from Bot-master and perform many cyber-crimes like phishing, DDoS attack etc. The main difference between Botnet and other kind of malwares is the existence of Command-and-Control (C&C) infrastructure. The C&C allows Bots to receive commands and malicious capabilities from Bot-Master. Thus, a network of Bots is formed which is called "Botnet". The Bot-master is a person who controls and manages the whole network of Bots. Types of Botnet based on **Architecture-**

1. Centralized Botnets: The old approach used by Botnet for their Command and control (C&C) architecture was the centralized mechanism (hierarchical). In this approach, the Bot-master (attacker) distributes the command over the Botnet via various Bot-Controllers in order to hide attacker's real identity. The bot-controller receives the commands from the bot-master and then these bot-controllers distribute the commands to all bots in Botnet.
2. P2P Structured Botnets: This new approach has no C&C server (bot controller) in P2P botnet architecture. The bot-master directly communicates with a peer bot and then the peer bot sends the commands to other bots in the network.

With many countries limiting both freedom of speech and the press, and with privacy concerns being paramount, assuring anonymous Internet communication to provide anonymity and privacy, has become important. But at the same time, anonymity has been leveraged for shadowy activities- such as drug trafficking through Silk Road, publicizing classified information, and planning and coordinating terrorist activities like the November 2015 Paris attack. All this has increased the interest in breaking the anonymized network communication. Surveillance organizations are trying to identify the strength of popular anonymous communication services such as Tor. Anonymous communication implies not being able to identify the originator's IP address and thus his or her location.

Section II covers the literature survey. Section III covers anonymous network- tor, how tor works and OnionBot. Section IV covers how tor provides anonymity to its bot users. Section V covers Results and Section VI covers Conclusion.

II. LITERATURE SURVEY

Botnet is a collection of infected hosts (bots) and is controlled remotely by a bot-master through C&C channel.[1]The bots stay hidden until they are informed by their bot-master to perform an attack. Botnets can perform various malicious activities from DDoS, to spamming, phishing, identity theft. The main difference between Botnet and other kind of malware is the existence of Command-and-Control(C&C) infrastructure. The C&C allows bots to receive commands from bot-master.

According to Cooke et al. [2], the control mechanism of botnet can be classified into centralized and Peer-to-Peer. A centralized one has a central point for bot-master forwarding commands and messages to bots, while its weakness is the single point of failure. But P2P topology overcome this weakness. The centralized mechanism has made them vulnerable to being detected and disabled. The first generation of Botnets utilized the IRC channels as their C&C centres. Thus, new generation of Botnet emerged, Peer to Peer based botnet, which can hide their C&C communication.

[3] This paper has detailed the two architectures of botnet- Centralized and Peer-to-Peer. In centralized approach, the bot-master distributes the command over the botnet via various bot controllers(C&C server) in order to hide attacker's real identity. In Peer-to-Peer, there is no C&C server (or bot controllers). Bot-master directly communicates to a single Bot peer and then that bot spreads the command sent by bot-master to other bots. P2P botnet is not easily manageable, because transferring commands is slow.

[1], [3] Different approaches have been proposed for detection of botnets. These are: Honeypot-based, signature-based, anomaly-based, DNS-based, Mining-based and Network-based. Honeypot-based detection technique has been considered the most efficient technique among all.

[4] Their proposed framework for detection is based on monitoring network traffics. The architecture of the proposed botnet detection system consists of 4 main components: Filtering, Application Classifier, Traffic monitoring, malicious activity detector. Filtering is responsible to filter out irrelevant traffic flows. This stage reduces the traffic workload. Application classifier is responsible for separating IRC and HTTP traffics from rest of traffics. Malicious activity detector is responsible to analyse the traffics carefully and try to detect malicious activities. Traffic monitoring detect the group of hosts that have similar behaviour and communication pattern.

[5] The main contribution of this paper is that Peershark works on the detailed evaluation of conversation-based approach which is clearly advantageous over traditional flow-based approaches. PeerShark correctly categorize different types of P2P applications-whether malicious or benign-with good accuracy. But the accuracy obtained with classification of benign P2P applications is relatively lower as compared to accuracy of detection of P2P botnet. Being flow-oblivious (i.e. port and protocol oblivious), many lower-level details (transport layer protocol) are neglected.

[6] Their approach was to detect IRC-based botnet. This approach observes the IRC traffic within an organization network domain and identifies infected hosts and IRC server. From observing the real traffic, they observed that bot-master uses IRC channels to control his botnet and traffic is not encrypted. Their proposed IRC-based botnet detection system can detect not only infected hosts but also C&C servers.

[7] This paper focusses on how botnets are constantly searching for new ways to evade detection. To mitigate botnets, we have to detect the nodes (bots) themselves. Once found, we can hijack and shut down their command and control servers through a number of different methods. Tor is excellent at maintaining the anonymous identity of the sender. Each node only knows where it should send the data next, so it's impossible to track its chain to the original sender. There is a real chance that Tor-based botnet would be very tough to mitigate in the near future.

[8]This paper has investigated on how anonymous is the tor network. The Tor network reroute the traffic through several nodes: an entry node, which sends the traffic to the relay node, then relay node sends it to the exit node. Then from the exit node, it is transferred to the final destination. While sending data through Tor, the client encrypts it multiple times with the node's keys, including predecessor's and successor's IP-addresses. Each node has the key only for one layer, uses the key to remove that layer, and then forwards the data. In this way, it sees only the IP addresses of nodes from where the packet has come and where it has to go. The exit node sends the packet to its final destination, which only sees exit node's IP-address.

[9] This paper has presented different anonymity technologies that enable Internet users to maintain a level of privacy. They have covered anonymity technologies including proxy servers, remailers, JAP (Java Anon Proxy), I2P (Invisible Internet Project), and Tor with the geo-location of deployed servers. Among these systems, proxy servers, Tor and I2P are actively used, while remailers and JAP have minimal usage

[10] This paper has presented two contributions to break Tor anonymity, a Data Mining driven solution to recover the browsing history of Tor users and optimal configuration settings based on game theory for Tor users and operators, as well taking malicious nodes into account. They have used a malware "Torinj" that is targeted against Tor exit nodes as it is expected that exit nodes are large in number and as well as more vulnerable and less protected than entry nodes. The "Torinj" has the ability to recover a user browsing history even when a trusted entry node is used.

[11] This paper has introduced the illusion of privacy of botnets over Tor. This paper showed that P2P botnets using Tor are still vulnerable to the same kind of attacks such as crawling and centralized botnets are vulnerable to the vulnerability of tor itself. The bots using Tor network are detectable due to the network traffic characteristics and the ports used by them. Centralized C&C servers also attract a lot of communication from all their bots. This behaviour exposes the botnet and this anomaly is not difficult to identify in the network.

[12] This paper has introduced a novel system called 'TorWard for the study and the identification of malicious traffic over Tor. An Intrusion Detection System (IDS) was used to analyse the traffic flowing. Malicious traffic over Tor includes P2P traffic, malware traffic (like worms, viruses, bots), Denial of service attack traffic, spam traffic and many other. The paper showed around 10% of tor traffic triggered IDS alerts.

[13] In this paper, potential attacks on the anonymity networks that can compromise user identities and communication links have been discussed. They have also summarized protection mechanisms against such attacks. It states that while using Tor, a user can browse the web without leaving a trace of his/her IP address in the logs of any web servers. They have surveyed the de-anonymization approaches so that it is easy to understand vulnerabilities in the anonymity networks.

[14]This paper present the techniques that exploit the Tor exit policy to greatly simplify the traffic analysis. The fundamental vulnerability exposed by this paper is not specific to Tor browser but rather to the problem of anonymous web browsing itself. There are two security problems that this paper exploits: HTTP's vulnerability to man-in-the-middle attacks and web browser's code execution feature. Thus Tor may actually decrease the anonymity of users by making them vulnerable to man-in-the-middle attacks. The web browsers execute malicious code which allows for arbitrary communication back to HTTP server and this pattern can be can be detected by an external observer using traffic analysis.
Thus, the Tor creates a tunnel, and then anyone can access the restricted web content. This is the genuine problem and one can't mitigate this.

III. ANONYMOUS NETWORK

A. *Tor*

In the Tor network (Tor's original name, The Onion Router), the traffic has to be rerouted through several nodes: an entry node, relay node and an exit node. Entry node sends the traffic to relay node, which sends it to the exit node. The tor traffic is encrypted. The source's identity is anonymous because the destination can see only the exit node's IP address. Tor randomly selects exit nodes to prevent any traffic analysis attack. Tor has to minimize the communication latency to avoid any degradation in performance and selection of exit node is also not equally random and thus does not produce uniformity in distribution of exit nodes. Thus, the exit nodes actually used might be more heavily concentrated in a particular area or to a particular ISP. Various investigative results shows significant imbalance between number of available exit nodes and those actually used. Moreover, most of the exit nodes are concentrated in a particular ISP or a particular small area. Consequently, the effects of exit-node distribution and their selection could erode network security and anonymity.

B. *How Tor Works*

1. Tor aims to eliminate the mapping between user and servers by hiding the user's IP address and thus prevents user identification and communication tracking.
2. To accomplish this, Tor has to generate an overlay network in which each node (entry, relay, and exit) maintains Transport Layer Security (TLS) connection to every other node. Thus, Tor traffic is encrypted with TLS. Tor has to establish a "circuit"- a random path through the network by selecting entry, relay and exit nodes.
3. Tor can extend this "circuit" by adding more relay nodes to it, but generally a circuit has only one relay node so that the communication latency is at acceptable level.
4. To select an exit node, Tor uses weighted random selection: It traverses the entire connection from source to the destination and in order to maximize the number of pending exit streams and considering the exit node's capacity and uptime as selection parameters, it selects the exit node.
5. To avoid delays, Tor builds "circuits" pre-emptively and regularly within every 30 seconds.
6. When data is sent through Tor, it is encrypted multiple times with node's keys, including the predecessor's and successor's addresses for each node.
7. Each node has the key only to decrypt the data for one layer. Each node uses that key to remove that layer and then forwards the data to the next node in path. In this way, one node can see only the IP address of where the packet came from and where it has to go.
8. The exit node sends the packet to the final destination, thus the destination only sees the IP address of the exit node. When reply returns from the destination, each node add its encryption layer and then only the sender can finally remove them all and thus read the reply that came from the destination.
9. Each circuit formed is used for 10 minutes and is not rotated after each access, an exit node's IP address could

be recorded multiple times and the same exit node might get selected repeatedly.

10. Exit node selection is weighted to favour the nodes with higher data rate and capacity, thus, it is natural to assume that a country's Internet data rate would heavily influence its exit node's use.

C. OnionBot, a Botnet utilizing Tor

OnionBot is a peer-to-peer botnet that relies on Tor network for the communication among nodes. No bot knows the IP address of any other bot. To communicate with each other, they only know the onion address of the bot to which the message has to be sent. Therefore, tracking the bot chain is actually impossible.
Onion Bot operates in 4 stages:

1) Infection: It is the phase where vulnerable users are infected through phishing spams, drive-by-download, zero day vulnerability, remote exploitation etc.

2) Rally: Once a computer is infected, it enters into the rally stage in which the infected computer, which is now a bot, will look for other bots in the network. To do this, this bot bootstraps into the network with the help of a hardcoded peer list of onion addresses, which are periodically updated.

3) Waiting: After connecting to the OnionBot network, this bot enters into the waiting stage where it is ready to receive commands from the bot-master.

4) Execution: After receiving commands from bot-master and identifying the target, it enters the execution phase, where it sends out spams or perform DDoS (Distributed Denial of Service) attacks etc.

5) Especially with users already using Tor to enhance their privacy, OnionBot, which operates within Tor, can have the potential to easily infect the other connected Tor users.

IV. HOW TOR PROVIDES ANONYMITY TO ITS BOT USERS

A. Using Sniffers

Sniffers are used to capture the packets over the network. The idea here is to browse any link (say, "whatis myipaddress.com") and analyze the destination IP-address. It should match with IP-address of "whatismyipaddress.com".

1. Browsing with any normal browser: In destination address, it shows the IP-address of "whatismyipaddress.com".

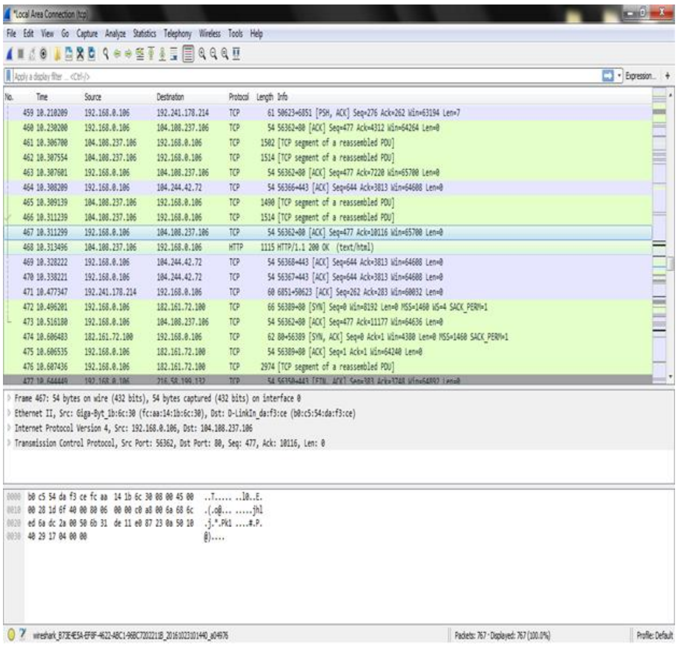

Fig. 1 Sniffer captures the packet while normal browsing.

2. Browsing with Tor browser: For the same request, it shows some other IP-address (other than that of "whatismyipaddress.com"). It can be the IP-address of tor-browser.

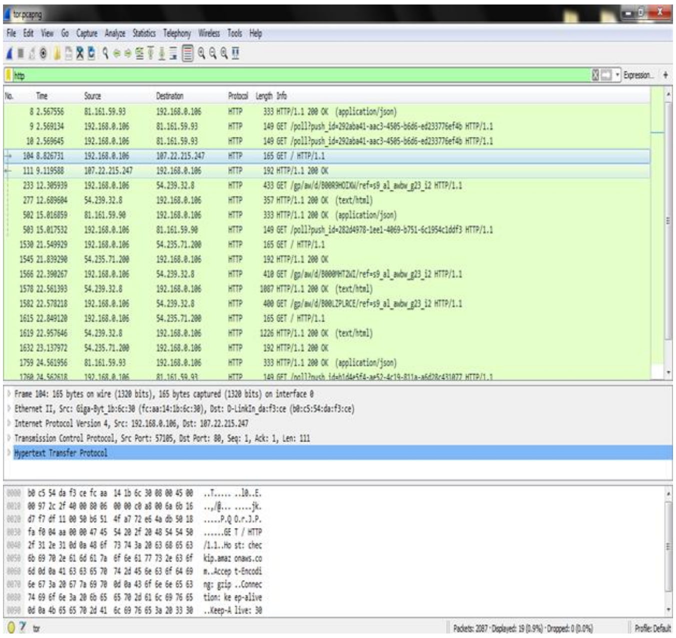

Fig. 2 Sniffer captures the packet while tor browsing.

B. Using Operating System Utilities

"Netstat" utility can also capture the destination IP-addresses while browsing, it also captures the packet details.

1. Result of netstat while normal browsing: It includes the IP-address of “whatismyipaddress.com” in the destination.

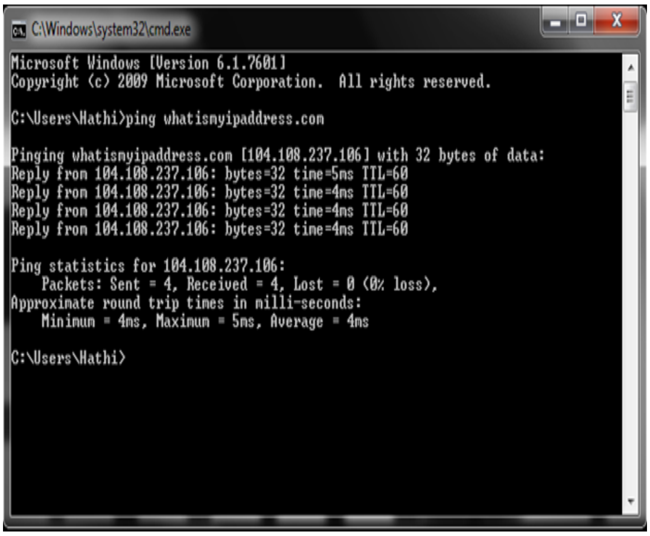

Fig.3 IP-address of “whatismyipaddress.com”.

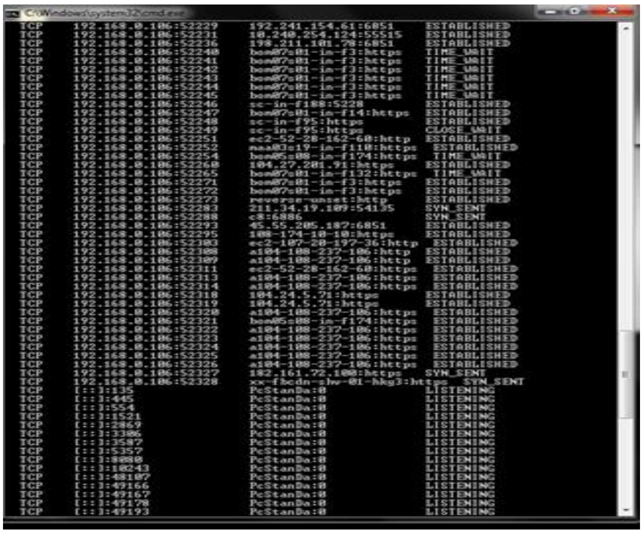

Fig. 4 Netstat captures addresses while normal browsing.

2. Result of netstat while tor browsing: In the destination there is no IP-address of whatismyipaddress.com.

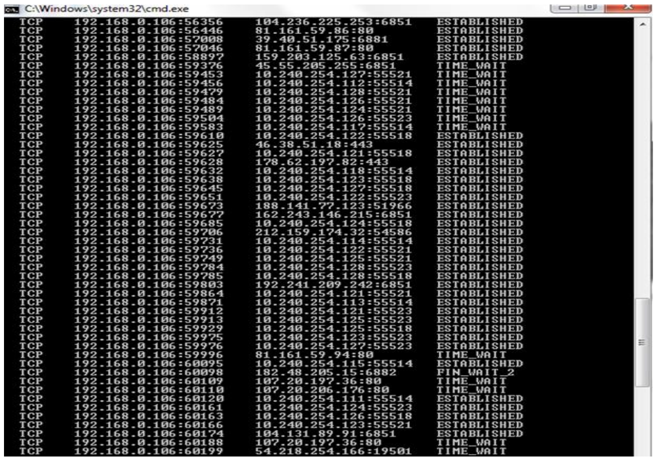

Fig. 5 Netstat captures addresses while tor browsing.

The system is communicating to tor and then tor is communicating to whatismyipaddress.com, thus, whatsimyipaddress.com shows IP-address of tor instead of the system. The destination part might be the IP-address of tor but that can never be the IP-address of “whatismyipaddress.com”.

C. Using Man-in-the-Middle Utility

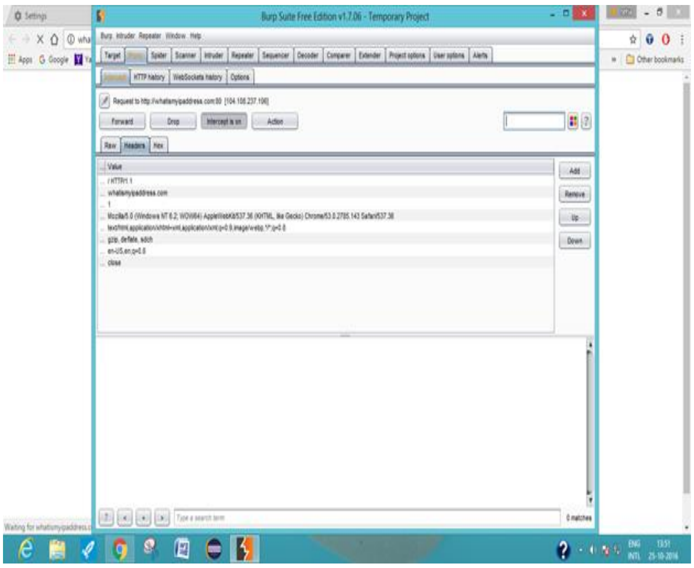

Fig. 6 It captures IP-address.

V. RESULTS

Thus, the survey involving sniffers, operating system utilities and man-in-the-middle utility proves that tor network actually hides the IP address of the source node. It eliminates the mapping between source and destination. The Tor network reroute the traffic through several nodes: an entry node, which sends the traffic to the relay node, then relay node sends it to the exit node. Then from the exit node, it is transferred to the final destination. While sending data through Tor, the client encrypts it multiple times with the node’s keys, including predecessor's and successor's IP-addresses. Each node has the key only for one layer, uses the key to remove that layer, and then forwards the data. In this way, it sees only the IP-addresses of nodes from where the packet has come and where it has to go. The exit node sends the packet to its final destination, which only sees exit node's IP-address. Backtracking to the source node is a little difficult. But, at the same time the tor network is still vulnerable. Various efforts have already been made to break its anonymity.

Type	Identify Anonymity?
Sniffers	Yes
OS Utility	Yes
Man-in-the-middle Utility	Yes

Table1: All types used to identify anonymity.

Type	Expected Destination IP-Address	Actual Destination IP-Address
Sniffers	104.108.237.106	104.108.237.106
OS Utility	104.108.237.106	104.108.237.106
Man-in-the-middle Utility	104.108.237.106	104.108.237.106

Table2: It shows that while normal browsing expected and actual destination IP-address values match.

Type	Expected Destination IP-Address	Actual Destination IP-Address
Sniffers	104.108.237.106	107.22.215.247
OS Utility	104.108.237.106	162.243.146.215
Man-in-the-middle Utility	104.108.237.106	212.159.174.32

Table3: It shows that while tor browsing expected and actual destination IP-address values do not match.

VI. CONCLUSION

Thus, the Tor creates a tunnel, and then anyone can access the restricted web content. This is the genuine problem and one can't mitigate this. Even various network security appliances provider like "CyberRoam", Cisco and Juniper are also threatened with the power of Tor browsing and till now have not been able to break the Tor power. But at the same time, P2P botnets using Tor are still vulnerable to the same kind of attacks such as crawling and centralised botnets are vulnerable to the vulnerability of tor itself. The bots using Tor network are detectable due to the network traffic characteristics and the ports used by them. Centralized C&C servers also attract a lot of communication from all their bots. This behaviour exposes the botnet and this anomaly is not difficult to identify in the network. The "Torinj" has the ability to recover a user browsing history even when a trusted entry node is used. Similarly with this, it has been found that there are many other parameters like IP address, geographical location, number of hops in the path to identify the communication path and the source node.

REFERENCES

[1] Hossein Rouhani Zeidanloo, Mohammad Jorjor Zadeh shooshtari, Payam Vahdani Amoli,M. Safari, Mazdak Zamani, "*A Taxonomy of Botnet Detection Techniques*",2010 IEEE.

[2] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, Manish Karir, "*A survey of Botnet Technology and Defenses*", Conference For Homeland Security, 2009, Cybersecurity Applications & Technology,2009,IEEE.

[3] N.S.Raghava, Divya Sahgal,Seema Chandna, "*Classification of Botnet Detection Based on Botnet Architecture*", 2012 International Conference on Communication Systems and Network Technologies,IEEE,2012.

[4] Hossein Rouhani Zeidanloo, Azizah Bt Manaf,Payam Vahdani, Farzaneh Tabatabaei, Mazdak Zamani ,"*Botnet Detection Based on Traffic Monitoring*", International Conference on Networking and Information Technology,2010,IEEE.

[5] Pratik Narang, Subhajit Ray, Chittranjan Hota, Venkat Venkatakrishnan, "*PeerShark: Detecting Peer-to-Peer Botnets by tracking Conversations*", 2014 IEEE Security and Privacy Workshops, 2014, IEEE.

[6] Chia-Mei Chen, Hsiao-Chung Lin, "*Detecting botnet by anomalous traffic*", Journal of Information Security and Applications 2I (2015) 42-51, Elsevier, 2015.

[7] Yang Yang, Christophe Leung, "*Botnets Drilling away privacy infrastructure*", December 2015.

[8] Robert Koch, Mario Golling and Gabi Dreo Rodosek "*How anonymous is the Tor network? A long-term black-box investigation*", IEEE Computer society, 2016.

[9] Bingdong Li, Esra Erdin , Mehmet Hadi Gunes , George Bebis , Todd Shipley, "*An overview of anonymity technology usage*", Computer Communications 36(2013) 1269-1283, Elsevier,2013.

[10] Cynthia Wagner, Gerard Wagener, Radu State, Thomas Engel, Alexandre Dulaunoy, "*Breaking Tor Anonymity with Game Theory and Data Mining*", 2010 Fourth International Conference on Network and System Security, IEEE,2010.

[11] Matteo Casenove, Armando Miraglia, "*Botnet over Tor: The Illusion of Hiding*" , 2014 6th International Conference on Cyber Conflict, IEEE, 2014.

[12] Zhen Ling, Junzhou Luo, Kui Wu, Wei Yu and Xinwen Fu, "*TorWard: Discovery of Malicious Traffic over Tor*", IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014.

[13] Esra Erdin, Chris Zachor, and Mehmet Hadi Gunes, "*How to Find Hidden Users: A Survey of Attacks on Anonymity Networks",* IEEE Communication Surveys & Tutorials, VOL. 17, NO. 4, Fourth Quarter 2015.

[14] Timothy G. Abbott, Katherine J.Lai, Michael R. Lieberman, Eric C. Price, "*Browser-based Attacks on Tor*", 2016.