www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 7 Issue 5 May 2018, Page No. 23926-23930
Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i5.08

Bot Net of Things – A Survey

[1]M. Shanmughapriya [2]G. Sumathi [3]K.C.Aarthi
Assistant Professor Velammal Engineering College Chennai – 600066

Abstract:
Internet of things is talk of the town now a days but the potential threat IoT has over the cyber safety is less emphasized. The possibility for attackers with all systems interconnected with no or less security measures installed in them, makes them vulnerable to all kinds of security attacks. Botnet consists of collection of private computers interconnected together and affected by malicious software, which can be controlled as a group without the owner's knowledge. BotNet is roBot and Network combination, the bot here is the compromised device. Denial of service, spyware, email spam, click fraud, bit coin etc., and are some of the well-known attacks by botnet. Botnet control itself has become a community, which focuses on prevention, control and repair services. This paper focuses on detailed survey of botnet and it's regarding features

Keywords: BotNet, Cyber Security, IoT

Introduction

BotNet is a collection of devices interconnected logically. The devices include range of handheld, household and other smart devices that are connected via internet. One of these devices in the collection should be compromised by a malicious malware, which in turn acts as a bot and controls all other devices connected to it.

Terminology Used

The core components of botnet of things uses many technological jargons that needs to be understood for clarity in the field.

Terminology	Meaning
A botnet's originator	Known as a "bot herder" or "bot master" controls the botnet remotely.
Command-and-Control (C&C)	Controls the botnet remotely.
Covert channel	Type of computer security attack that creates a capability to transfer information objects between processes that are not supposed to be allowed to communicate by the computer security policy.
Internet Relay Chat (IRC)	It is an application layer protocol that facilitates communication in the form of text
Zombie computer	It is a computer connected to the Internet that has been compromised by a hacker, computer virus or trojan horse and can be used to perform malicious tasks.

DDoS	A Distributed Denial of Service (DDoS) attack is an attempt to make an online service unavailable by overwhelming it with traffic from multiple sources.
Scrumping	The process of stealing computing resources as a result of a system being joined to a "botnet" is sometimes referred to as "scrumping"
Bulletproof hosting	It is a service provided by some domain hosting or web hosting firms that allows their customer considerable leniency in the kinds of material they may upload and distribute.
Denial-of-service attack	It is a cyber-attack where the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet.
Fast flux DNS	Fast flux is a DNS technique used by botnets to hide phishing and malware delivery sites behind an ever-changing network of compromised hosts acting as proxies.

Botnet Topologies

The two things needed to set up a botnet are an addressing mechanism to identify and reach a command-and-control instance, and a communication protocol to distribute commands to the bots. The latter is often referred to as anoverlay network that forms the botnet's communication channel. Different botnets are using different strategies here which is reflected in the topology used: We differentiate between centralized, decentralized and locomotive botnets. The kind of topology is extremely important for the selection of containment strategies. Centralized topologies as depicted in figure 1 are the classical botnet structures.

The classical botnet structures.

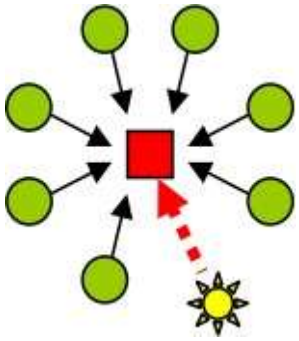

Figure 1: Centralized botnet with 7 bots and one commander

The box in the middle denotes the central C&C server with seven connected bots and a commander (the star symbol). Examples are the IRC-based Agobot, Rbot, and Sdbot families [1]. A static command-and control server is contacted by bots via its IP address (which generally requires resolving a DNS name first).Centralized botnet infrastructures often rely on existing network protocols on top of IP that implement standard client-server architectures, like IRC or HTTP. For this reason, they are obviously completely extinguishable by taking down their C&C server. Figure 1. A centralized botnet with seven bots and a commander The communication in a centralized botnet can either follow a push strategy (as in IRC- based

communication) where each bot stays connected to a server which then distributes commands simultaneously to all hosts in a broadcast-like manner. Or the server has to be polled by the clients on a regular basis (as in HTTP-based botnets). In the latter scenario, the general method is to set up and update a central resource like a web page which can be browsed by the bots. Both approaches have their advantages, e.g., IRC botnets can be built upon an existing IRC infrastructure with multiple self-synchronizing servers, providing load-balancing and reliability. HTTP, on the other hand, is more stealthy and better suited for bypassing security gateways and hiding amongst regular traffic patterns.

In a decentralized topology, no single command-and control component exists. Instead, each bot seeks for a commander using some upstream query mechanism. A schematic structure is depicted in figure 2:

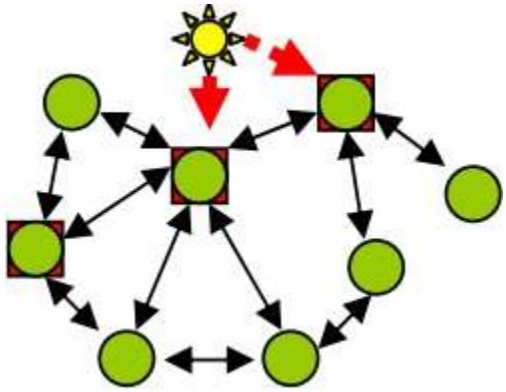

Figure 2. A decentralized botnet with three bots acting as C&C servers

Each bot knows some neighbors and receives and forwards commands. Three bots act as C&C servers and are advised to distribute commands in the network. Well-known representatives are the Storm Worm [3], or Conficker [5]. The two-tiered approach allows the botnet owner to easily change the C&C backbone, making it much harder to take it down. As in centralized botnets, commands can be pushed to bots, which requires that they can be reached instantly, or infected machines pull commands from their individual C&C server (the latter being the most common case). Bots can be implemented to automatically re-establish a C&C session on disconnects. Most decentralized botnets seen so far were based on peer-to peer (P2P) technology that allows for both information queries as well as host addressing, the two features needed for the communication between a bot and a command server. In a common P2P botnet some peers are controlled by the botnet owner and used to issue and propagate information (i.e. commands) to other peers. Taking advantage of the flexible self-organizing network infrastructure, these nodes are easily replaceable with other hosts. The decentralization can be taken even further by designing fluxy registration of C&C servers at the query layer (i.e., a pool of command servers returned to queries which is kept highly dynamic through automated subscriptions). This situation is visualized in figure 3 on the next page: The shaded structures are past C&C servers that have been replaced by other ones automatically. Bots recognize the change and contact the new server instead. In most cases these C&C servers are also infected hosts, temporarily playing the role of a commander. Another way would be to change the query interface, e.g., by choosing timedependent domain names. We call such botnets locomotive because of their constantly moving structure. One example is the HTTP-driven Torpig botnet [4]. Conficker, in addition to its P2P structure, also makes use of constantly changing DNS names [5-7]. There is no standard implementation of such botnets. In fact, the overall structure is often even more complex than outlined here. Figure 3. A locomotive botnet with C&C servers that move over time In reality the boundaries between centralized, decentralized, and locomotive botnets are blurred: A similar strategy was already commonly implemented in classical botnet infrastructures where a DNS entry was used to transparently switch between servers. However, this does not really provide more security as it only displaces the single point against which takeover attempts could be mounted.

Components of Botnet

a. **Command and Control Server**—Often abbreviated as C&C, a command and control server is the centralized computer that issues commands to and receives information back from the bots. Command

and control infrastructure frequently consists of several servers and other technical components. Most botnets use a client-server architecture, but some botnets are peer-to- peer (P2P), with the command-and-control functionality embedded in the botnet.

b. Peer-to-Peer Botnet—Peer-to-peer (P2P) botnets use a decentralized network of bots for added protection against takedowns. While P2P botnets can include a C&C server, they may also operate without one and be structured randomly to further obfuscate the botnet and its purpose. While P2P botnets are less likely to be identified, the botmaster cannot easily monitor command delivery and the implementation can be complex.

c. Botmaster— Alternatively called a botnet controller or bot herder, the botmaster is the botnet's operator. This individual remotely controls the botnet, issuing commands to the C&C server, or to individual bots within the network. A botmaster's name and location are heavily obfuscated to prevent identification and prosecution by law enforcement.

d. Bot—An Internet-connected individual device within the botnet is called a bot. A bot is most often a computer, but a smart phone, tablet, or Internet of Things device can also be part of a botnet. A bot receives operational instructions from a command and control server, directly from the botmaster, or sometimes from other bots within the network.

e. Zombie—Another name for a bot. Because the bot is controlled by an outside computing device or person, it is likened to a fictional 'zombie'. A botnet is also known as a "zombie army."

f. Botnet Attack

a. How C and C Distribute Malware

i. A botmaster develops a botnet by distributing bot malware to infect PCs or other devices. He may also rent an existing botnet from another criminal.

ii. The newly harvested bots or "zombies" report in to the botnet's command and control (C&C).

iii. The C&C now controls these bots and issues instructions for the bot to distribute executable malware files, as well as the email templates and potential victim address lists.

iv. The infected zombie bots receive the orders, each sending email messages carrying the malware payload to thousands of potential victims.

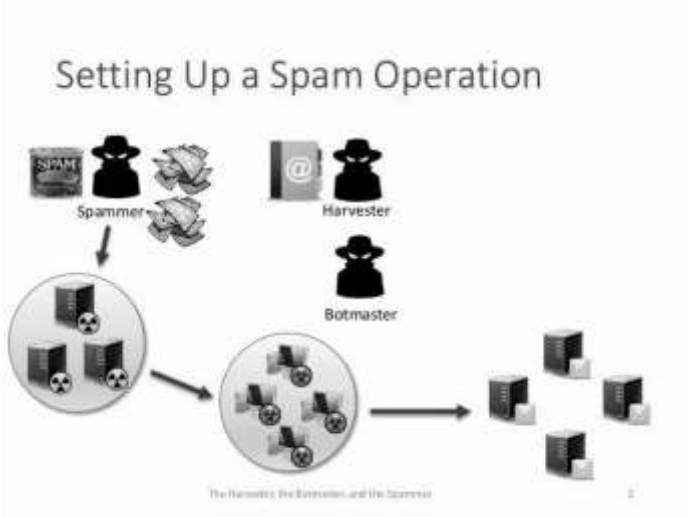

Figure 3 shows how the flow of attack works

Counter Measures for BotNet Attack

How to Identify the System influenced by Bot Net of things. If answer to the following questions is yes, then it is possible that system may be under an influence of a botnet.

1. Is your computer or internet connection running slower than normal?
2. Did your computer start behaving erratically? Does it crash frequently? Do you receive unexplained error messages?
3. Did the fan kick into overdrive when your computer is idle?
4. Did you notice unusual internet activity (like high network usage)?
5. Does your browser close frequently and unexpectedly?
6. Did your computer take a long time to start or shut down or didn't shut down properly?

These can indicate that a program is running without your knowledge and using a fair amount of

resources. The next step would be to check the Task Manager – see what's going on in there. You can also disconnect from the Internet and see if there are any differences. Of course, all these could also indicate that your fan is full of dust and it just needs to be cleaned. Or that your computer is obsolete and needs an upgrade. However, if this is not the case and you discover that your computer is part of a botnet, the standard advice would be to wipe it all out. Format it and reinstall the operating system. In order to minimize any potential damage, make sure that you always backup all your important files and folders. This is a piece of advice most people ignore, but I know you know better than that.

Conclusion

Thus to avoid botnet of things spread, measures to be taken in developing IoT applications with secure gateways. Protocols for message transfer and information sharing must be made rigid in all perspectives. This paper focuses on basic analogy to understand the newly evolving threat.

References

[1] N. Ianelli, A. Hackworth, "Botnets as a vehicle for online crime," CERT,

[2] Request for Comments (RFC) 1700, December 2005.

[3] The Honeynet Project & Research Alliance, "Know your enemy: Tracking botnets," http://www.honeynet.org, March 2005.

[4] K.J. Houle, G.M. Weaver, "Trends in denial of service attack technology,"CERT, October 2001.

[5] J. Kristoff, "Botnets," NANOG32, October 2004.

[6] S. Racine, "Analysis of internet relay chat usage of ddos zombies," Master's thesis, ETH Zurich, April 2004.

[7] E. Cooke, F. Jahanian, D. McPherson, "The zombie roundup: Understanding, detecting and disrupting botnets," in 1st Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), 2005.

[8] Ramachandran, M. Feamster, D. Dagon, "Revealing botnet membership using dnsbl counter-intelligence," in 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), 2006.

[9] M.A. Rajab, J. Zarfoss, F. Monrose, A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in Internet Measurements Conf. (IMC), 2006.