On the Internal Workings of Botnets: A Review

Emmanuel C. Ogu
Department of Computer Science, School of Computing and Engineering Sciences Babcock University, Nigeria

Nikos Vrakas
Department of Digital Systems University of Piraeus, Greece

Ogu Chiemela
Department of Computer Science, School of Computing and Engineering Sciences Babcock University, Nigeria

Ajose-Ismail B. M.
Department of Computer Science, School of Applied Science Federal Polytechnic, Ilaro, Ogun State, Nigeria

ABSTRACT
Denial of Service and Distributed Denial of Service Attacks have significantly shackled the development of computer networks and the internet, and masked their innumerable benefits behind many hours of service unavailability. These attacks are fostered, especially in their distributed variant, by networks of compromised machines (known as botnets, that is, a network of bots) that are taken over by a hacker / attacker, and coordinated in such a way as to channel overwhelming loads of malicious or useless traffic towards resource-providing / request-servicing servers. In the long run, a sufficient load of these traffic, overwhelm target servers and constitute them unable to service the requests of legitimate users that have subscribed legally to use these resources. This army of compromised systems have also been recently linked to various malicious and nefarious activities that have been taking place on computer networks and the internet in recent times; such activities relate to malware injection / infiltration, fraud, espionage, amongst others. This paper reviews the operations and coordination of botnets and the interactions that take place within the botnet during such malicious activities. New, valuable insights are provided towards the detection of such malicious networks through the introduction of the reverse life cycle of botnets.

General Terms
Information Security, Network Security, Network & Information Security, Botnets, Malware.

Keywords
Botnets, Cybercrimes, Information Security, Malware.

1. INTRODUCTION
Bots are malicious network entities that facilitate the workings of Denial of Service (DoS) attacks, especially in its distributed variant. A bot is a computer program which once installed gives an attacker ("master") remote control over a compromised machine (which becomes a "zombie" or "slave") via a secure channel. A network of zombies that are controlled by a single coordinating force (attacker) form a botnet (bot-network or network of bots). Botnets are ever-ready threats to any network infrastructure, and this is primarily because of two reasons: they greatly obfuscate the task of detection and simplify evasion such that firewalls and intrusion detection systems (IDS) are unable to handle; and also because a sufficient amount of bots in a botnet can generate traffic in overwhelming volumes enough to threaten even the best and most advanced servers [1] [2]. This is illustrated in Figure 1.

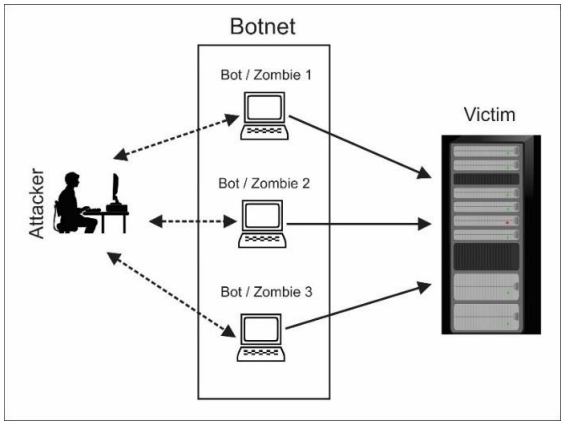

Figure 1: A Typical Botnet Attack Structure

Every botnet has the following generic participants or action points: the bot (the compromised machines – "zombies"), the bot controller (the malicious code that controls the bots in the network) and the bot master (the attacker who controls the botnet) [2].

Botnets are becoming a new generation of global threats to the internet and basically any other network, that is still yet to be properly understood. The philosophy behind botnets constitutes them flexible enough to be able to threaten any network topology, from a conventional infrastructure to Mobile Ad-hoc networks (MANET) [3], Voice over IP (VoIP) deployments [4], and Vehicular ad hoc networks (VANET) [5]. There has so far been investigated a three-step mitigation and control procedure for botnets. These include:

1. Prevent the bot from infecting other systems on the network;
2. Try to determine the command and control communication links among bot associates and between bots and controllers; and
3. Detect any other secondary features that the bot may be carrying, such as deadline propagations, target number of systems needed to further strengthen the botnet, etc. [2].

Despite these and many other approaches and techniques that have been proposed in literature, the challenge of botnets have remained a nightmare for many organizations and network infrastructure administrators. According to [6], the total number of bot infected systems on the internet was estimated to be between 800,000 and 900,000, with some botnets having more than 100,000 members. By 2004, the number of new bots discovered daily increased from below 2,000 to over

30,000 just within the first six months of 2004 [7]. Fast forwarding to 2011, a single botnet known as ZeroAccess had amassed a bot-army strength of between one and two million; generating millions of dollars of annual profit for their botmaster through click frauds and bitcoin mining [8].

Bots were, however, not always as dreadful as they now are. They were originally used in the management of Internet Relay Chat (IRC) channels. "IRC is a chat system that provides one-to-one and one-to-many instant messaging over the Internet. Users can join a named channel on an IRC network and communicate with groups of other users". The task of administering these busy chat channels soon became rather tasking and time consuming, channel operators therefore created bots to help with managing the operations of popular IRC channels. One of the first of such bots that was developed was Eggdrop which was written in 1993. Today, bots have evolved with very potent capabilities for disaster and damage to any network infrastructure [9].

2. BOTNETS LIFECYCLE ANATOMY

The fact that botnets were originally created to be used within legal jurisdictions, has now been put aside since botnets are now used to facilitate several cybercrimes and pose threats to cybersecurity infrastructure. Researchers have confirmed the involvement of botnets in cybercrimes such as DoS and DDoS attacks against critical infrastructure, the dissemination of various computer malware, phishing attacks, and various types of frauds ranging from financial frauds to Pay-per-click (PPC) frauds, Search Engine Optimization (SEO) poisoning, Corporate and Industrial Espionage, Bitcoin Mining, etc. [10], [11], [12], [8] [21].

A plethora of sources have attempted to propose various ways of detecting, isolating and classifying botnets within a network [13]. These proposals are focused on (a) observing network activities for familiar behavioural patterns that are associated with previously known botnets (Signature and DNS based), (b) checking for a deviation from the normal network operation, interactions and behaviour (Anomaly based), or (c) investigating their command and control (C&C) interactions and parameters (Mining and Machine Learning based) [13]. However, the rapid growth of botnets on the internet keeps increasing annually by very worrisome margins. Recent statistics from [8] and [14] insight that millions of botnets have infiltrated the internet and are being used to send millions of spam messages, malicious malware and ransomware payloads, amongst others. Hence, it would be no gainsaying that there may be, arguably yet provably, at least one compromised machine hibernates in every home and office all around the world, with snippets of codes (dead or alive), waiting in them to be awakened by their C&C botmaster.

Ideally, botnet's eradication may indeed lie in personal and individual security awareness; and for personal and individual security to be effective and efficient, there must be an understanding of the way botnets operate within themselves. Botnets have a very interesting lifecycle, and a lot of interesting, sometimes complex interactions take place within the botnet during its lifecycle. The generic lifecycle of interactions that take place within a botnet is illustrated in figure 2.

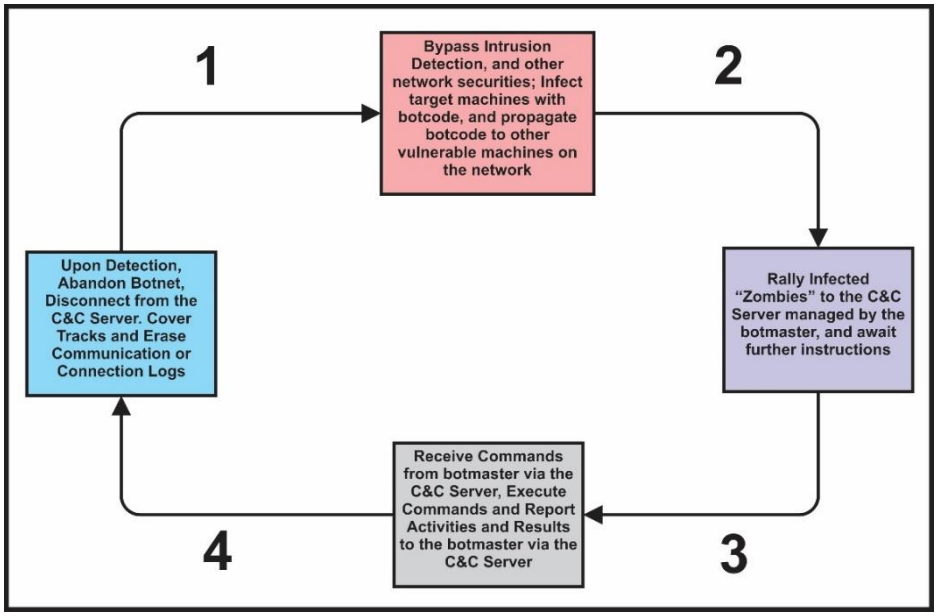

Figure 2: The Lifecycle Schema of a typical Botnet

Based on the botnet lifecycle illustrated in figure 2, this research would describe three generic phases that occur in this cycle. These phases are: Infection / Doping, Recruiting, and Synchronization / Rallying.

1. Infection / Doping: This phase occurs when the botmaster releases the bot code / bot controller into the network or the internet, either as a (sometimes obviously) malicious whole, or as part of (*a dope*) of a seemingly harmless piece – emails, ads, URLs, games, etc. Other popular means that have been confirmed to be used in infecting and doping vulnerable machines include: Drive-by downloads, Pirated Software, etc. [8].
2. Recruiting: In this phase, the bot code / bot controller that has infiltrated the network infrastructure or internet is responsible for executing the recruitment procedure. Recruitment may initiated directly by the botmaster who serves the bot code to specific target hosts of interest, or the bot code could be a self-recruiting one – which roams the network, looking for vulnerable hosts to infect.

3. Synchronization / Rallying: This phase occurs after bots have been successfully recruited into the bot army. They are rallied back to a central C&C unit which could either be administered centrally (by the botmaster) or in a peer-to-peer manner (by other bots in the botnet) [15], but usually remotely via the internet. The bots maintain synchronization with the C&C unit at all times in order to receive new commands, infiltration parameters and takeover specifications, which they readily execute. Synchronization and Rallying are possible because during the process of the bot code installation, backdoors are installed on the zombies, unused ports are opened and/or hijacked such that even after firewalls upgrades and security patch updates, these would still remain difficult to shut off [8].

These phases illustrate what would be referred to in this research as the forward botnet lifecycle (See Figures 1 & 2).

Evidences from literature [12], [15], however, suggest that there exists a reverse botnet lifecycle which may be the reason why such threats lingered on the internet and remain a subject of critical discuss in various network security domains.

In essence, botnets never really die. Bots may, however, be temporarily dislodged and scattered apart from their botnets and C&C through the utilization of various security mechanisms, but they still lie hibernated within the network infrastructures, carrying within them bot codes/controllers and waiting for the next botmaster to awake them so the bot army can be re-assembled.

Command and Control mechanisms can easily be handed down generations of botmasters (or hijacked by other individuals) who can easily awaken whatever hibernated bots existed on a previous botnet. New sources [8] have also revealed a fierce botnet competition taking place in cyberspace in which botmasters seeks to takeover bots that have already been recruited as members of others (sometimes rival botnets). They achieve this simply by scanning the network to confirm that they have already been recruited as part of an existing botnet, then through the same backdoors and hijacked ports, they uninstall the existing bot codes on the victim and replace it with theirs, thereby taking over ownership of the bot and rallying back to the C&C server for further instructions [8].

Essentially, the reasons why botnet still linger and lurk around network infrastructures, and the reason why their effective mitigation remains a complex task based on four different facts: (a) inadequate information about their origins, (b) what motives them to drive their activities, (c) how they are created and deployed and (d) luck of effective screening and filtering of already compromised machines that are part of a botnet [8].

On top of that, botmasters have devised diverse means of coordinating their botnets in order to avoid detection or blocking even by state of the art security techniques. Evidences from literature [8] have also proven that the ultimate goal of an attacker in coordinating botnet activities is related to securing the C&C server, hiding it from the prying activities of firewalls and IDSs and masking it from being traceable by security professionals and other hackers too. This is important because whoever is in control of the C&C infrastructure, controls the botnet (in essence, owns the botnet).

Amongst the methods the attackers could employ to achieve the goal of retaining possession and control of their botnet C&C servers include: migrating between random C&C server addresses that are generated using a malware that incorporates Domain Generation Algorithms (DGAs), and using the Fast Flux [16] method to point several IP addresses to the domain names that the bot attempts to contact, thereby reducing the possibility of the actual C&C server being detected and taken down [8].

3. NEW PERSPECTIVES ON BOTNETS

The problem of botnets have lingered far more than could initially have been foreseen when they emerged as a challenges on the scene of network security, several years ago. This challenge has defied even some of the most sophisticated and advanced solutions that have been proposed to try and mitigate them; they keep re-emerging time and again, and usually with a more sophisticated and advanced techniques.

Further, botnets can now be hired on the internet by individuals and (even government and political) organizations who have enough finances to motivate a hacker to deploy botnets in order to carry out various malicious and nefarious activities against their opponents, enemies, rivals and business competitors; ranging from DoS attacks to malware infiltration, espionage, amongst others [8].

While botnet's life cycle has been covered and detailed comprehensively in most modern literatures and reviewed in previous sections, figure 3 describes and illustrates the botnet's reverse life-cycle.

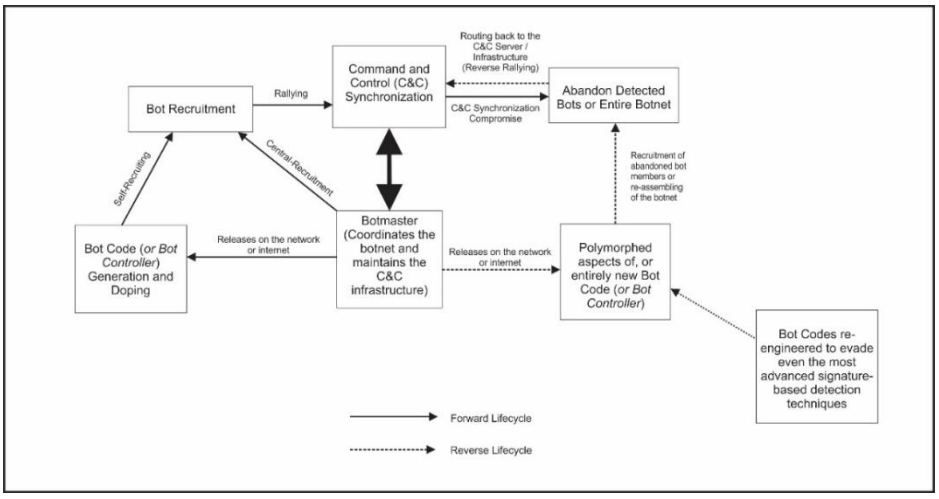

Figure 3: The forward and reverse life cycles of botnets

Similarly to the forward lifecycle, in the reverse lifecycle, a botmaster (who may not actually be the original owner of the botnet) releases a bot controller code on the internet and proselytize previously existing botnet members or previously compromised machines that may have possibly been dislodged from other botnets, abandoned by their botmasters or cut-off from a command and control source due to risk of possible detection. All these bot fragments are gathered and reverse-rallied back to the command and control source and a new botnet is emerged (see the broken arrows in figure 4).

This accounts for why even after botnets have been hopefully dislodged and mitigated by various security mechanisms, they still find a way of re-emerging with a new more complex or slightly different structure that evades from detection form signature based mechanisms.

4. PROPOSALS IN LITERATURE AND RELATED WORK

[17], proposed a new mobile botnet that is resilient to detection by conventional anomaly and mining-based detection methods, which exploits the push notification service of Google's Android mobile platform for disseminating commands using Google's cloud-based C2DM (Cloud to Device Messaging) service. Through evaluation, strategies are proposed to enhance the scalability, resilience, stealth, resource efficiency and controllability of the botnet. The authors go further by presenting methods of deploying a C2DM botnet for orchestrating SMS-Spam-and-Click attacks in such a generalized form that covers also the iOS and Windows mobile operating systems. Possible defence methods against the proposed mobile botnet are also discussed. The baseline architecture for the design of a C2DM botnet was also described with a prototype implementation of the architecture. In the specific implementation, a regular, trusted Android application is injected with the malicious botnet code and installed along with the regular .apk Android package, but with extended malicious capabilities and permissions. The botnet could be mitigated by the sandboxing application prior to installation, in order to observe the communication patterns of the application for seeming malicious activity. It could also be mitigate by disabling unnecessary push notification from third party applications on the end-users' mobile device. Furthermore, only permissions that are necessary and required for specified function(s) of third party applications should be granted; and the AndroidManifest.xml file should be checked regularly to ensure that there is only one authorised C2DM receiver.

Google's revolutionary Android Operating System (OS) is unarguably one of the smartphone Operating Systems that have helped to spring forward the evolution of mobile technology and capabilities in the 21st century. The Android OS brought with it a new level of openness and customizability that users had never experienced before. The growing popularity of the Android has also brought along an increase in the amount of mobile malwares and botnets targeted at the mobile operating system. [18], investigates the trends and behaviours that have characterized the evolution of Android botnets and malwares generally. An in-depth study of literature, relating to known malware applications discovered on the Android, was used to deduce generic behaviours and characteristics of Android botnets in terms of the Android Botnet Development Model and the Android Botnet Discovery Process, so as to aid a proper understanding of the activities of Android botnets and how they can be discovered. Common characteristics of Android malware discovered in this research relate to: bugged repackaged applications, receiving C&C commands, stealth messaging, stealing user information, applications obtained from third-party application stores and markets, downloading of additional content and manipulation of the AndroidManifest.xml File in order to escalate features and permissions.

As the "botnets" phenomenon continues to advance and evolve and gradually invading mobile infrastructures and networks, and as botmasters continue to implement newer methods for evading detection by even the most advanced heuristics and intrusion detection systems, the researches by [15] and [20] have proven to be of great importance. The paper presents botnets that have recently been discovered on mobile networks and infrastructures, emphasising on the new command and control mechanisms employed by these botnets in carrying out their malicious activities. The paper also reviews the challenges as well as the limitations that have trailed botnets detections methods and techniques within mobile environments, while also consider the solutions that already exist for combating and preventing mobile botnets. SMS, Bluetooth connections, HTTPS, and a hybrid of these have been identified as some of the most preferred methods by which botmasters, of mobile botnets, send C&C instructions to mobile "slaves" for the execution of malicious activities. Known challenges posed by these mobile botnets to detection schemes include:

1. Low computational capabilities of the mobile devices
2. Proprietary/specific security schemes on most mobile devices and platforms
3. Variations in the modes of infection and propagation of mobile botnets
4. Advanced evasion and fool-proof techniques incorporated into the botnets by the botmasters andAbsence of a central security management technique or system for mobile networks and devices.

Lately, centralized C&C botnet structures have proven to be an easy target for takedown by network and cyber security mechanisms. Consequently, botnet operators have reorganized their botnet C&C structures to become Peer-to-Peer (P2P) based. P2P botnets (responsible for node enumeration and poisoning attacks) have proven to be more resilient and difficult targets due to the absence of a single point of failure within the botnet structure. [19], proposed a formal graph model for capturing the very unique properties and intrinsic vulnerabilities of P2P botnets. Two aspects of resilience are highlighted in this model: (a) the intelligence gathering resilience, which tests how much malwares can deter analysts from fishing out bots on a network, and (b) the disruption resilience aimed at disrupting P2P botnets by sinkholing them (re-directing all of them towards one of the attacker-controlled machines) and partitioning them into smaller, sub-networks that are unusable and weaker in strength. The graph model is applied towards accessing the resilience of all active P2P botnets. Several strategies are further proposed towards evaluating strategies for mitigating and testing the resilience of P2P botnets. Upon testing and evaluation, results demonstrated that some P2P botnets became susceptible to disruption by the graph model, while others proved to be more robust due to their complex design.

The command and control protocols used in most modern botnet and malware families are beginning to show a sharp deviation from the traditional HTTP and IRC protocols. As botmasters have begun to evade most payload analysis IDS mechanisms by encrypting C&C traffic, [19] presented a

method for detecting botnets which use encrypted channels for command and control. They proposed PROVEX, a payload-based network intrusion detection system (NIDS) that automatically develops / derives probabilistic vectorised signatures. PROVEX is trained to learn values that characterize various fields (by incorporating a knowledge of known command and control encryption algorithms) within encrypted C&C protocols, by evaluating the probability of certain byte occurrences within traces of C&C traffic. Authors claim that this mechanism was able to identify C&C message syntaxes, for the families of malware that were studied, by decrypting all packets that were intercepted on their test-bed environment. However, even though PROVEX shows a relatively high detection accuracy and scalability indices in detecting encrypted malware command and control channels, it would perform poorly if it is made the target of a massive scaled DDoS attack, because it would result in a lot of resource utilization and wastage while legitimate client requests would be stalled.

5. CONCLUSIONS AND FUTURE WORK

This research has been focused on the internal workings of botnets and provided new perspectives concerning their reverse lifecycle that have not been previously discovered in literature. Also provided is a thorough analysis of how botnets operate, as well as a state-of-the-art review of the most significant scientific works in literature.

Further research in this area would focus on breaking the reverse life cycle of botnets. Right from the point of initial identification of bot culprits and initial dislodgement of the botnet, research efforts would be focused on discovering how bots can be completely isolated from all residual bot codes/controllers that could trigger a reverse life cycle for the bot, update its structure for dodging from new behavioural signatures and possibly regenerating the entire botnet.

6. REFERENCES

[1] Banks, S., & Martin, S. (2007). Bot Armies: An Introduction.

[2] Cooke, E., Farnam, J., & Danny, M. (2005). The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. Proceedings of the USENIX SRUTI Workshop, 39, p. 44.

[3] Hanafy, I. M., Salama, A. A., Abdelfattah, M., & Wazery, Y. M. (2013). AIS Model for Botnet Detection in MANET using Fuzzy Function. International Journal for Computer Networking, Wireless and Mobile Communications (IJCNWMC), 3(1).

[4] Geneiatakis, D., Vrakas, N., & Lambrinoudakis, C. (2009). Utilizing bloom filters for detecting flooding attacks against SIP based services. Computers and Security, 28(7).

[5] Garip, T. M., Gursoy, E. M., Reiher, P., & Gerla, M. (2015). Congestion Attacks to Autonomous Cars Using Vehicular Botnets.

[6] Allen, H., & Roman, D. (2003). Increased Activity Targeting Windows Shares. CERT Advisory CA-2003-08.

[7] Laurianne, M. (2004). Bot Software Spreads, Causes New Worries. IEEE Distributed Systems Online, 5(6).

[8] FORTINET. (2012). Anatomy of a Botnet. Carlifornia: Fortinet®.

[9] Egg Development Team. (1993). Eggdrop: Open source IRC bot. Retrieved from http://www.eggheads.org/

[10] Ianelli, N., & Hackworth, A. (2005). Botnets as a vehicle for online crime. FORENSIC COMPUTER SCIENCE IJoFCS, 19.

[11] Honeynet Project and Research Alliance. (2005). Know your enemy: Tracking Botnets. Honeynet Project and Research Alliance. Retrieved from http://www. honeynet.org/papers/bots/

[12] Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., . . . Vigna, G. (2009). Your botnet is my botnet: analysis of a botnet takeover. Proceedings of the 16th ACM conference on Computer and communications security (pp. 635-647). ACM.

[13] Maryam, F., Alireza, S., & Sureswaran, R. (2009). A Survey of Botnet and Botnet Detection. Proceedings of the Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE'09 (pp. 268-273). IEEE.

[14] SOPHOS. (2014). Security Threat Report 2014. Oxford, UK: SOPHOS.

[15] Eslahi, M., Salleh, R., & Anuar, N. (2012). Bots and botnets: An overview of characteristics, detection and challenges. Proceedings of the International Conference on Control System, Computing and Engineering (ICCSCE), 2012 (pp. 349-354). IEEE Press.

[16] The Honeynet Project. (2007). Know Your Enemy: Fast-Flux Service Networks. Retrieved from http://www.honeynet.org/papers/ff

[17] Zhao, S., Lee, P. P., Lui, J., Guan, X., Ma, X., & & Tao, J. (2012). Cloud-based push-styled mobile botnets: a case study of exploiting the cloud to device messaging service. Proceedings of the 28th Annual Computer Security Applications Conference (pp. 119-128). Association for Computing Machinery (ACM).

[18] Pieterse, H., & Olivier, M. S. (2012, August). Android botnets on the rise: Trends and characteristics. Information Security for South Africa (ISSA), 2012, 1-5.

[19] Rossow, C., Andriesse, D., Werner, T., Stone-Gross, B., Plohmann, D., Dietrich, C. J., & Bos, H. (2013). Sok: P2pwned-modeling and evaluating the resilience of peer-to-peer botnets. IEEE Symposium on Security and Privacy (SP), 2013 (pp. 97-111). IEEE.

[20] Banks, S. B., & Stytz, M. R. (2008). Challenges of modeling botnets for military and security simulations. Proceeding of SimTecT (Vol. 2008).

[21] Paxson, V. (2001, July). An analysis of using reflectors for distributed denial-of-service attacks. ACM SIGCOMM Computer Communication Review, 31(3), 38-47. doi:10.1145/505659.505664