Botnets: An Unsolvable Problem?

R. F. McDowell and A. M. Colarik
Centre for Defence and Security Studies, Massey University, Albany, New Zealand

Abstract - *Botnets, large numbers of infected machines used to conduct malicious activity, are posing a great threat to the Internet. As the numbers of infected machines increases, the risk for both financial loss and the reliability of data transmission grows in tandem. In this paper we contend that sustaining this problem has been possible due to both weaknesses in traditional anti-malware software and bot exploitation of communication protocols. We maintain that as the botnet capability has expanded, independent research has suffered and furthered the difficulty with developing a solution. This paper explores the botnet problem, the potential reasons why the issue prolongs, and proposes a possible way to develop a greater research capability in order to eventually solve the problem*.

Keywords: Botnet, DDoS, Experimentation, Collaboration, Education.

1 Introduction

The popularity of the Internet and subsequent interconnectedness has given a greater ability for cyber-criminals to conduct malicious activity [1]. Larger numbers of connected and more easily reachable individuals has increased the capacity for malicious entities to exploit both security weaknesses and general technological naivety, equally for financial gain and cyber-vandalism. In-line with this interconnectedness, malicious activity is now regularly performed in a more networked and distributed manner [1]. To achieve this, malicious users exploit a network of infected ‘zombie’ machines known as a botnet which can target large numbers of individuals quickly and proficiently [2]. One ‘bot-master’ could potentially control thousands of machines all conducting individual malicious tasks thus increasing not only the potential victim pool, but attack capabilities. As examples, botnets have the capability to perform attacks such as distributed denial of service (DDoS), click fraud, phishing, malware distribution, spam emails, and illegitimate exchange of information [3]. The problem is exacerbated by bot-masters’ aptitude at concealing their activities which poses a risk not only for financial loss but for the integrity of data transmission. As bots are concealed, often through the purview of legitimate means, it becomes more difficult to differentiate between normal and abnormal.

Bot-masters achieve concealment through two functions: the use of polymorphic malware code at the machine level and the exploitation of data protocols at the communication level [1][4]. Firstly, polymorphic code changes with every instantiation so traditional signature based anti-malware will not detect it [5]. When a ‘new’ malware is analyzed by researchers its signature is placed within a database and anti-malware software will use these signatures as a means of identifying whether a code is malicious or not [6]. Polymorphic malware code naturally restricts this ability. Mainstream reliance upon signature based anti-malware has the potential to further prolong the issue, particularly if the victim pool is rising [7]. Inexperienced users are effectively lulled into a false sense of security if no threats are found, and thus ensuring that bot malware is not removed and making it more difficult to counter the botnet threat.

Secondly, botnet command and control (C&C) is often transmitted through standard communication protocols notably through internet relay chat (IRC), hypertext transfer protocol (HTTP), and peer to peer (P2P) in order to remain virtually anonymous [4]. A C&C server is created using one of these protocols and bots will connect, waiting for commands to perform malicious activity [8]. With protocols such as HTTP, bots will often hijack legitimate communication in order to bypass traditional firewall based security [3]. The issue therefore lies within the difficulty of distinguishing between malicious botnet traffic and normal traffic, primarily because behavior differs only in intent and not in content [9]. Whereas IRC and HTTP have a primarily centralized structure and therefore have a single point of failure if the C&C is removed, P2P is decentralized giving greater difficulty in removing the entire botnet. Likewise to the use of polymorphic malware code, a P2P structure is an additional way that bot-masters adapt their techniques to maintain their botnets and frustrate the abilities of researchers to deduce solutions. This is exacerbated additionally as attack techniques such as DDoS further exploit inherent flaws within communication protocols. For example, bot-masters may direct individual bots to surf through a website and access multiple web-pages in what is known as an HTTP application-layer attack [10]. Whilst the website gets overwhelmed, bots can remain anonymous as it often appears as though the website is going through a period of heightened traffic. This protocol exploitation makes it very difficult to detect a botnet attack, and thus difficult to analyze and develop an adequate solution.

2 Existing Research

The ability for botnets to command a large and diverse range of attack techniques, to remain anonymous whilst doing so and to adapt to the changing cyber security environment has created significant difficulty for researchers in developing a solution. No current mitigation or detection technique has been able to offer anything fully adequate or permanent [1]. As a result, there are a number of passive solutions that do not sufficiently address the botnet threat environment [11]. Honeypots as an example allow researchers to study and analyze behavioral characteristics of malicious entities within a controlled environment. However, whilst successfully giving the ability to collect and analyze bot malware, they suffer due to a number of passive features [11][12]. Not only are they required to wait for an attacker, but they are only able to report information about the infected machines placed as traps so there is very little room for in-depth analysis [8]. This approach unfortunately offers little deterrent to malicious behavior.

As discovered largely in the botnet solution literature, this limitation with honeypots is also exacerbated by the nature of the research itself, much of which is conducted on a limited collaborative basis. The resourcing needed to fully research every propagation, attack, and communication method available to a botnet hinders the ability of singular or small-group research. As both the potential victim pool and attack capability pool rises, this type of research will suffer further. If the research centered approach of honeypots is adapted to a more encompassing and a more collaborative model however, a solution (or solutions) may be easier to develop. To widen the scope of research, testing should be conducted on not only the victim machine but also the attacking machine. This is why a framework to develop a more research focused and collaborative approach that could potentially be adopted on a wider-scale basis is necessary. This would seek to mitigate some of the issues that are currently faced, particularly with honeypot research, and would allow a much larger pool of educational knowledge to be collated in order to develop a solution. After all, thorough, in-depth research and education is realistically the only way a solution will be developed.

It is equally necessary to target the two fundamental and prolonging reasons for the botnet problem: vulnerabilities with anti-malware and exploitation of communication protocols. By directing research and education at these two functions, more should be learnt about how attackers successfully conceal their activities. If the passivity of honeypots is removed, perhaps by creating a type of honeypot that, rather than being on the receiving end would also be on the delivery end, deeper research could take place. Effectively, through a controlled environment, a researcher could become an attacker and behavioral characteristics of both the malicious entity and the victim machine could be analyzed more thoroughly.

Honeypots are essentially passive not only in their inability to ‘attack’, but their physical restriction limits the extent of research. In other words, because a honeypot must ‘wait’ for an attack, research results can be drawn out over longer periods of time. Given the variants and proliferations of current botnet technology, this becomes a less effective means of studying the problem. If an alternative, more robust honeypot was provided in an ‘open-source’ manner to cyber security researchers and educational facilities, a larger degree of testing and analysis could take place. Rather than the traditional notion of a honeypot, it would instead be an educational botnet. Researchers who have access to the educational botnet, along with the source code by which its created, would not only be able to test a range of attack capabilities on differing ‘victims’, but would be able to see a more ‘real-world’ view of behavioral characteristics in an accelerated period of time. Researchers would have the potential to think like an attacker, giving a greater ability to think ‘outside the box’, and thus provide more holistic data than what is currently available. This effectively alleviates the biggest drawbacks to botnet research: passivity and non-collaboration.

3 An Educational Botnet

In order for an educational botnet to be successfully implemented and to ensure there is the capability to gather relevant educational data, three provisions should be considered.

First, as the botnet is open-source, the source code by which the malware is initially created should be provided giving the possibility to extend it, manipulate it, and alter it. Due to the polymorphic nature of bot malware, this would give a better understanding of its real-world application [5]. Analysis of the code would give an idea of its behavior during its polymorphism, particularly if the end-user is able to manipulate it and extend it themselves. By concurrently using a behavior analysis tool, behavioral characteristics could further be analyzed within different machine processes. For example, the next generation of researchers need to understand how malware embeds itself into a system upon infection, through process monitoring that observes malicious processes, through network monitoring to observe network traffic, and through change detection that highlights key changes malware made to the file system and registry [13]. Through monitoring of key processes, a better ability to understand how bot-malware code obfuscates itself against anti-malware software could develop throughout both the machine and the network allowing deeper research. This in turn, would help to facilitate a more botnet specific behavioral-based anti-malware tool. As multiple users could extend the source code, wide-ranging comparative analysis and research could take place that focusses on the changes polymorphic bot-malware goes through. With multiple entities analyzing the behavior of bot-malware, the capability to determine if there are (or are not) certain characteristics will be enhanced.

Second, researchers should be able to access a C&C server in order to task bots as if one were a bot-master, and thus to give a better ability to monitor communication behavior. By providing a capability to analyze all traffic, both normal and abnormal, behavioral patterns could be detected, which would enhance the ability to create a mitigation tool. This, of course, would allow researchers to understand how bots exploit communication protocols to obfuscate their traffic as normal [9]. A further discussion will be had on the possible structure of this later in the paper.

Third, and possibly most important of all, all data and results generated through the educational botnet should be uploaded to a central database, accessible and usable by all researchers and education facilities to give a greater degree of analysis. This could potentially generate a huge amount of information dependent on the number of researchers taking part, and therefore give a much richer and robust activity data set for a prevention and detection tool to be developed. The generated data would be accessible by a wealth of researchers, all able to process and analyze it in differing ways, but all seeking the same end goal. The amount of effort and resourcing that could be provided to solve the botnet problem would be much higher than what is currently available.

3.1 Implementation Considerations

Whilst in theory an educational botnet sounds like a practical way to analyze botnets on a large-scale basis, there needs to be initial consideration for whether there would be enough interest in order for it to be worthwhile. A large computing capability would be necessary, along with active volunteers that are willing to conduct testing and research. If there is a lack of research machines available, virtual or not, the botnet capability would naturally be limited. The fewer machines available, the smaller the scope of research, which, in-turn, would limit the number of researchers that would likely be willing to participate.

Nevertheless, there are two potential options to mitigate this. Either the originating research center provides the machines or volunteers allow their own machine (or machines) to become segmented and ‘infected’. The first option, although potentially limited in scope, would negate some possibility that the botnet could be used for malicious rather than educational purposes. The second option on the other-hand would give a much larger capability for testing and as the machines wouldn’t be centered in one place, a more realistic depiction of a real-world botnet. Indeed, dependent on the amount of researchers, the botnet has the potential to be developed on a large-scale basis, thus giving a greater capability for analysis of varying means, inclusive of different infection and attack techniques, differing communication pathways, and differing ‘victims’.

The volunteer-based structure may also struggle to find willing participants who would allow their machine(s) to be infected with bot malware based on the risks involved. Malware of course causes systems issues and performance degradation, and removing it may require a complete system clean; a worrying prospect for a researcher. Nevertheless, studies on open-source software have shown that developers choose to participate in open-source research that is directly aligned with their motivations and attitudes [14][15][16][17]. IT researchers, educators, and professionals would be the primary target audience; those that would directly benefit from such a tool to aid them in their research. Bot-malware code is often intended to remain hidden anyway, and real-life examples provide that individuals are often unaware that their machine has become hijacked, so the likelihood of noticeable performance degradation is low [7]. As long as volunteers fully understand and fully agree to, effectively, infecting their machine with malware, the risk associated would be greatly reduced. To alleviate individual concerns, the software could intuitively ‘know’ where the malware is hiding itself and be able to remove itself if requested by the user through an uninstaller. Other tactics to alleviate the potential risk would be determined by the researcher; the use of virtualization software or use of ‘stand-alone’ machines as examples.

In line with the second requirement, botnet hierarchy and topology needs to be considered. Whilst a single sever topology (i.e. all ‘zombies’ receiving communication through one C&C server) would be adequate if the research was conducted on a smaller scale basis with limited users, consideration of multiple agents, all requiring access to a C&C for more thorough research is necessary [2]. A multi-server topology gives a more distributed and less centralized formation, but also gives the potential for the botnet to be hijacked, thus removing control, and giving a much higher possibility of malicious use. Conversely, a hierarchical topology has a central server which also controls a number of ‘proxy’ servers, giving a greater ability to provide small portions of the larger botnet to researchers [2]. As in Fig. 1, whilst the top ‘hub’ acts as the central server controlling the entirety of the botnet, the two hubs beneath could be provided to independent researchers, thus giving them control of their own botnet, yet still being very much connected to the larger network. This is important, not only because it allows centralized control and allows all research, data, and analysis to be fed up through the lower hubs into a database, but it will give greater flexibility for researchers to have access to a C&C server, and to better analyze communication patterns. Versions of this topology are often used by bot-masters to separate ‘chunks’ of their botnet for rent or resale [18].

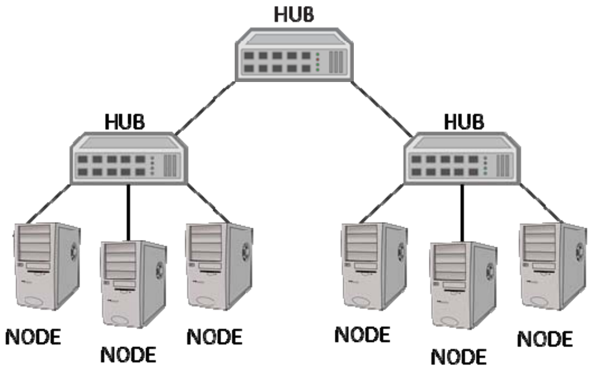
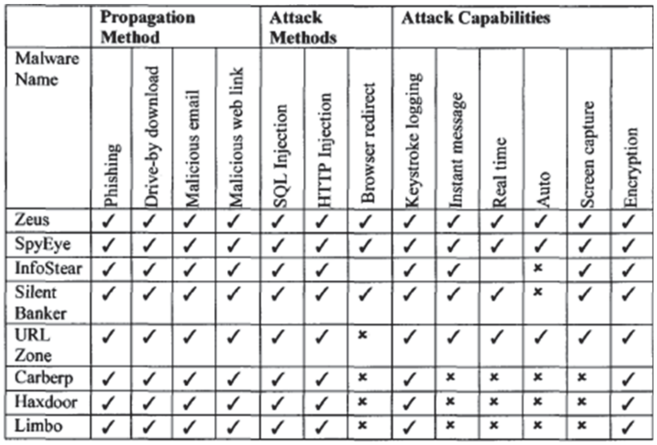


Figure 1. Hierarchical topology.

Despite the malicious motive of bot-masters who rent or resell segments of their botnet, the tools they provide could offer a model for what features are provided to researchers. Fig. 2 lists the most common toolkits widely available and details their component parts, namely propagation (infection) methods, attack methods, and attack capabilities. Providing a usable botnet matching the features of these toolkits, could give a better research capability toward studying typical malicious tools used within botnets. As can be seen, some features regularly appear such as phishing (i.e. attempting to obtain sensitive information such as usernames, passwords, and credit card details) and keystroke logging (i.e. the action of recording the keys struck on a keyboard) [19]. By providing the ability to conduct such attack and propagation methods within the educational botnet, more research and analysis could be generated toward targeting these prolific, malicious features, and a more robust solution could be implemented quicker. This would not only help to prevent singular instances of malicious intent, but also prevent the key abilities of botnets and botnet toolkits.



	Propagation Method				Attack Methods			Attack Capabilities					
Malware Name	Phishing	Drive-by download	Malicious email	Malicious web link	SQL Injection	HTTP Injection	Browser redirect	Keystroke logging	Instant message	Real time	Auto	Screen capture	Encryption
Zeus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SpyEye	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
InfoStear	✓	✓	✓	✓	✓	✓		✓	✓		×	✓	✓
Silent Banker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓
URL Zone	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
Carberp	✓	✓	✓	✓	✓	✓	×	✓	×	×	×	×	✓
Haxdoor	✓	✓	✓	✓	✓	✓	×	✓	×	×	×	×	✓
Limbo	✓	✓	✓	✓	✓	✓	×	✓	×	×	×	×	✓

Figure 2. Crime toolkits characteristics [19]

Fig. 2 also emphasizes the adaptability of these tools, further exemplifying the difficulty of developing adequate prevention software. The numerous propagation methods, attack methods, and attack capabilities requires constant research and analysis to understand changing and differing techniques. This is not always possible, particularly through small-group research. For example, whilst at one-point creation of illegitimate websites was a key propagation tool in a bot-master's arsenal, now legitimate websites are often hijacked as well, with malware covertly inserted into HTTP responses (often called Web Injects or HTTP Injection as in Fig. 2) [19][20]. This has reduced the trustworthiness of data transmission, and due to the vastness of the Internet almost impossible to fully prevent. Whilst the educational botnet may not be able to solve this problem overtly, it should be able to put more resourcing into understanding and recognizing the behaviors of propagation techniques.

Fig. 3 is an example of what an actual control panel of a botnet toolkit looks like, what is included, and what could be included in an educational botnet. Whilst it is anticipated that potential users of an educational botnet would have advanced computing knowledge, ease-of-usability should still be a key component in order for researchers to more efficiently understand its usage. 'SpyEye' which is able to conduct all of the features as listed in Fig. 2, has a number of intuitive applications which could be implemented into an educational botnet. As can be seen, it has a relatively user-friendly control panel that handles administrative tasks such as tasking infected bots, viewing statistical information such as displaying the amount of successfully infected machines within the botnet, and viewing stolen data from a database called FTP backconnect [20]. Fig. 4 depicts the search function of this database known as the FormGrabber Admin Panel (FAP), which is used to search for stolen credit card numbers, account numbers, and screenshots of hacked victim machines [20]. It also has a 'plugins' manager which effectively controls the varying different attack capabilities listed within Fig. 2, inclusive of DDoS. A similar type of control-panel with similar features for the educational botnet would allow ease of accessibility and a range of different statistical, monitoring, and application type tools for control of individual bots and the wider botnet. Researchers could direct their botnet to conduct a spam exercise test, or view how many machines have successfully been infected very easily. The database implementation could also be used as a way to upload and view statistical and behavioral data, also accessible by other researchers.

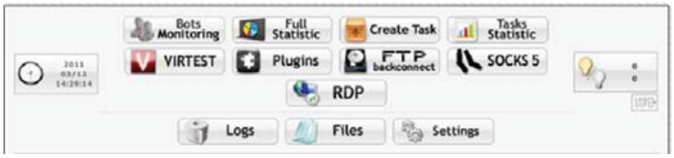


Figure 3. SpyEye toolkit characteristics [20]

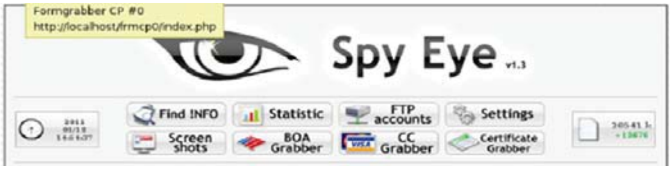


Figure 4. SpyEye's FormGrabber Admin Panel in action [20]

Notably, 'Virtest', an anti-virus testing module that is able to find the detection rate of SpyEye could also be implemented in an educational botnet [20]. Along with conducting simulated attacks, a testing mechanism for anti-botnet and anti-malware software would be a key component. It is certainly feasible that products like it will likely facilitate the creation of a potential solution and by providing a 'Virtest' mechanism, more thorough testing can take place. Multiple different propagation and attack techniques can be conducted, and testing will allow us to see how well it can withstand such attacks.

There is one major drawback to this type of software however. There needs to be robust consideration for how certain attacks would take place as it is not feasible to actually steal and store data. An educational botnet would not conduct malicious activity per se; it will create more of a simulated environment that tests the behavioral characteristics of malicious activity. If one is testing bot behavior when perpetrating a phishing exercise for instance, where does the malicious email go? It certainly cannot go to real-life individuals as this would cross an ethical line. Would researchers be required to bulk create a number of fake email addresses? Would there be a number of victim machines outside the botnet? Or would researchers run virtualization software? Lastly, how strong of an authentication process will be required to ensure only ethical researchers engage the educational botnet? These are a few ethical and practical questions that need to be considered in order to ensure such software does not come under intense scrutiny. Malicious testing is the backbone of the educational botnet and without it, there would clearly be a lack of adequate research, but it needs to be done through a highly controlled process.

4 Distributed Denial of Service and Preventing Malicious Activity

Whilst the previous considerations cannot be answered for every attack and propagation technique, we can attempt to discuss potential mitigation tactics through the example of DDoS. DDoS is chosen because it exhibits the same characteristic protocol exploitation techniques as used by bot-masters, and is a prolific botnet attack tool both for financial gain and cyber vandalism. By providing a simple plug-in, similar to what is found within SpyEye, keying of an Internet Protocol (IP) address or Domain-Name Server (DNS) along with allocating a specified number of bots for traffic could be all that is required to conduct an attack. To ensure the plugin is even more beneficial for researchers, the type of DDoS attack (i.e. Transmission Control Protocol Flood, HTTP Flood, DNS Reflection etc.) could also be selected. This would help to analyze traffic patterns, noting the differences between each type of attack.

Of course, this function still has the potential to be used maliciously if restraints aren't placed on how researchers conduct DDoS attacks. Initially terms and conditions would need to be agreed to between individual researchers and the central research center, of which it needs to be clear that the educational botnet is not to be used for malicious purposes. Whilst it is conceded that this would not necessarily prevent malicious activity, it is deemed as unlikely that researchers would openly breach these terms and conditions. When conducting an attack, the researcher would either have to create their own dummy website or enter into an agreement with an organization that allows DDoS to be conducted on theirs. Similarly to the creation of the initial botnet, the second option would be better suited for testing purposes because it gives a better 'real-world' depiction and thus allowing better traffic analysis of normal versus abnormal to take place. It is anticipated that volunteers would have their own reasons for allowing DDoS attacks on their personal websites; they may have been a victim on previous occurrences and seek to prevent attacks in the future or they may have existing DDoS 'prevention' techniques they are seeking to test as examples. A further agreement between the researcher and the organization would also specify exactly what time the DDoS attack would take place; it could not for instance take place at peak business hours due to the potential for customer loss.

Regular auditing should take place to further ensure that malicious activity is not being conducted. In the case of DDoS, individual researchers should collate a list of the websites they intend to attack, and either provide proof that particular organizations have agreed for it to take place or prove that a targeted website is owned by them. As data will be fed back into a central database, scanning against the collated list, which could potentially become an automated process, should take place to ensure no malicious activity is being conducted. If an organization is found to be using the software for malicious activity, they should be automatically banned and access removed. Upon entering the DDoS plugin, or any other attack vector plugin for that matter, there could also be a pop-up message that re-iterates that malicious use will not be tolerated, auditing will take place, and anyone found to be using the software for malicious purposes will be banned. Again, this won't necessarily stop individuals conducting malicious activity, but would act as a deterrent to the vast majority of participants. Indeed, even if the originating research center due diligently checks every application from each researcher, there is always the potential for manipulation, particularly as there would be little control over organizations' employees that may be using the software. It is not really feasible to check each and every individual who may use the botnet at one time or another, but we need to do our best to ensure the integrity and reliability of such a system.

5 Conclusions

An educational botnet would be but a start to solving a larger, more encompassing problem. Research is the only way forward, and by providing an accessible tool for academics, a solution will be better facilitated. The framework for the educational botnet discussed within this paper is one design, but one that mimics existing botnets in such a way as to allow researchers to understand it within a 'real-world' picture. It also allows specific targeting to what we have argued are the two fundamental reasons for the prolonging nature of the botnet problem: vulnerabilities with anti-malware and exploitation of communication protocols. To further extend this idea, multiple entities controlling and 'attacking' and therefore collaboratively collating data on these issues will give a much more streamlined and dedicated researching capability than what has currently been available.

It is of course paramount that further research is conducted before implementation, particularly with the number of potentially adverse variables that need to be considered. Whilst certainly there could be a number of possible benefits enabled through the educational botnet, there are also a number of possible pitfalls, particularly surrounding the ethical nature of the botnet's attack capabilities. What needs to be understood however is that despite a wealth of independent research, very little has been accomplished thus far. This idea, whilst having the potential to be used unethically if checks and balances are not developed, does give a greater capability to perform actual attacks; and thereby eliciting a more robust data set for analysis. This in turn gives a capability for further, more in-depth research to take place, which may not only solve the botnet problem but also the wider malware issue.

6 References

[1] Karim, A., Salleh, R. B., Shiraz, M., Shah, S. A., Awan, I., & Anuar, N. B. (2014). Botnet detection techniques: review, future trends, and issues. *Journal of Zhejiang University - Science C (Computers, & Electronics), 15*(11), 943-983.

[2] Hachem, N., Mustapha, Y. B., Granadillo, G. G., & Debar, H. (2011). Botnets: Lifecycle and Taxonomy. *Conference on Network and Information Systems Security.* Paris: Institut Telecom.

[3] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., & Garant, D. (2013). Botnet detection based on traffic behaviour analysis and flow intervals. *Computers & Security*(39), 2-16.

[4] Leder, F., Werner, T., & Martini, P. (2009). Proactive Botnet Countermeasures: An Offensive Approach. *Proceedings of the 1st Conference on Cyber Warfare.* Bonn, Germany: Nato Cooperative Cyber Defence Centre of Excellence.

[5] Chandrashekar, J., Orrin, S., Livadas, C., & Schooler, E. M. (2009). The Dark Cloud: Understanding and Defending Against Botnets and Stealthy Malware. *Intel Technology Journal,* 13(2), 130-146.

[6] Ask, K. (2006). *Automatic Malware Signature Generation.* Gecode. Retrieved from http://www.gecode.org/~schulte/teaching/theses/ICT-ECS-2006-122.pdf

[7] Greengard, S. (2012). The War Against Botnets. *Communications of the ACM, 55*(2), 16-18.

[8] Silva, S. S., Silva, R. M., Pinto, R. C., & Salles, R. M. (2013). Botnets: A Survey. *Computer Networks*(53), 378-403.

[9] Saravanan, R., Shanmuganathan, S., & Palanichamy, Y. (2016). Behaviour-based detection of application layer distributed denial of service attacks during flash events. *Turkish Journal of Electrical Engineering & Computer Sciences*(24), 510-523.

[10] Matta, V., Di Mauro, M., & Longo, M. (2016). *DDoS Attacks with Randomized Traffic Innovation: Botnet Identification Challenges and Strategies.* Fisciano, Italy: DIEM, University of Salerno.

[11] Hyslip, T. S., & Pittman, J. M. (2015). A Survey of Botnet Detection Techniques by Command and Control Infrastructure. *Journal of Digital Forensics, 10*(1), 7-26.

[12] Moon, Y. H., Kim, E., Hur, S. M., & Kim, H. K. (2012). Detection of botnets before activation: an enhanced honeypot system for intentional infection and behavioural observation of malware. *Security and Communication Networks*(5), 1094-1101.

[13] Zelster, L. (2015). *5 Steps to Building a Malware Analysis Toolkit Using Free Tools.* Retrieved October 9, 2016, from Zelster: https://zeltser.com/build-malware-analysis-toolkit/#install-behavioral-analysis-tools

[14] Bagozzi, R. P., & Dholakia, U. M. (2006). Open source software user communities: A study of participation in Linux user groups. *Management Science, 52*(7), 1099-1115.

[15] Roberts, J. A., Hann, I., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A

longitudinal study of the Apache Projects. *Management Science, 52*(7), 984-999.

[16] Sen, R., Subramaniam, C., & Nelson, M. L. (2009). Determinants of the Choice of Open Source Software License. *Journal of Management Information Systems, 25*(3), 207-239.

[17] Choi, N., Chengalur-Smith, I., & Nevo, S. (2014). Loyalty, Ideology, and Identification: An Empirical Study of the Attitudes and Behaviors of Passive Users of Open Source Software. *Journal of the Association for Information Systems, 16*(8), 674-706.

[18] Ollman, Gunter. (2009). *Botnet Communication Topologies Understanding the intricacies of botnet command-and-control.* Damballa. Retrieved from https://www.damballa.com/downloads/r_pubs/WP_Botnet_Communications_Primer.pdf.

[19] Alazab, A., Abawajy, J., Hobbs, M., & Khraisat, A. (2014). Crime Toolkits: The Curent Threats to Web Applications. *Journal of Information Privacy and Security, 9*(2), 21-39.

[20] Sood, A. K., Enbody, R. J., & Bansal, R. (2013). Dissecting SpyEye – Understanding the design of third generation botnets. *Computer Networks*(57), 436-450.