

Botnet Detection and Perimeter Defence

Niall A - 40417661

Coursework Submission for CSN08703

March 2019

# Section 1 - Research

## Botnets

It is known within the cyber security industry that in recent years, botnets have emerged as one, if not the, major threat to information security within businesses. This is primarily due to the fact that botnets, and subsequently the coders behind them, have been evolving in both size, sophistication, and knowledge of evasive techniques. *(Aussems, Noë, and Rivera, 2014, pp. 1-8)*

The purpose of botnets can be seen below:

*“Special Trojan viruses to breach the security of several users’ computers, take control of each computer and organise all of the infected machines into a network of ‘bots’ that the criminal can remotely manage.” (Kaspersky, 2019)*

A botnet works by having multiple bots (zombies) listening to a C&C channel, and upon receiving an instruction, will carry out a specific task. These tasks can vary from positive impacts to negative impacts on end uses and businesses. (The Honeynet Project, 2008)

The strengths of botnets consist of, but are not limited to, their wide range of capabilities; they can be used for phishing via spam email, as well as theft of end user confidential information, enabled by keystroke logging, as well as DDoS capabilities, provided that the botnet is large scale. In addition to this strength, botnets can adapt; one stage in the bot lifecycle (can be referred to as *egg downloading (Vuong and Alam, 2011, p.56)*)allows the bot to download new executables, for example a payload to deactivate the antivirus capabilities of a system. Thus, making botnets an extremely robust and sophisticated malware.

The weaknesses of a botnet consist of, but again are not limited to, how well it is implemented, and the topology used to create connections to the bots. The implementation, while in some cases being successful, can be a weakness; IDS’ are only as good as the botnets are bad, meaning that if a botnet is put together without planning or carefulness, it can become easy to spot and block, allowing it to be taken down. In addition to this, the topology used while setting up and, in turn, running the botnet can also be a weakness; more centralised structures, having a central C&C device connected to the array of bots, creates a single point of failure for the botnet – if the C&C device goes down, or is detected, it becomes easy to disrupt, and as a result destroy the botnet. *(Aussems, Noë, and Rivera, 2014, pp. 1-8)*

Botnets as a whole could become more sophisticated as time goes on; the introduction of more IoT devices offers a wider range of potential bot devices, as more and more brands are becoming commonplace, and it can be assumed that security is not as big of an issue as other features, such as usability and consumer cost in order to gain more market share; more security focus during development is likely to increase the cost, and subsequently drives down sales as the market for IoT is flexible. Due to this, it could be suggested that botnets can become more sophisticated to evade IDS’s through hiding activity via IoT protocols and listening to multiple C&C channels, such as RSS feeds, IRC Servers and HTTP pages whilst incorporating a distributed topology to avoid being destroyed from a single point of failure.

## IDS – Intrusion Detection Systems

An intrusion can be described as a potential possibility of a deliberate unauthorised attempt made in order to access/manipulate information, as well as render a user system unusable or unreliable. Subsequently, an IDS (intrusion detection system) can be described as a piece of hardware / software in place to detect abnormal activity on a device / network (HIDS / NIDS) which is created by an attempted intrusion. The purpose of an IDS is to point out when an intrusion occurs, and can categorise into 6 different types of intrusion:

|  |  |
| --- | --- |
| Attempted break ins | Denial of service |
| Leakage | Malicious use |
| Masquerade attacks | Penetration of the security control system |

An advantage of IDS systems is that through anomaly-based detection, the IDS can detect unseen attacks as its aim is to detect unusual (compared to normal) activity. This has a disadvantage however in that it will subsequently result in a higher volume of false positives as not all traffic will be malicious but it can still be classed as unusual. However, by using both signature detection and anomaly-based detection, the IDS can become very powerful. *(Chawla, Lee, Fallon and Jacob, 2018)*

# Section 2 – Botnet Analysis

Configure a working perimeter network topology with a firewall, DMZ, and host systems

as a testbed for the coursework, based on the addressing specified (see Moodle). For

example an annotated network diagram, and some basic configuration/connectivity testing

shown and discussed very briefly  
Use this to then analyse the operation of the running Bot agent and Botnet controller,

including any network scanning by the bot, connections created, and any communications

between the bot and controller. For example screen shots and brief discussion for: botnet

components running, analysis tools, outputs and interesting data, tools and outputs of

cracking codes

## Static Analysis

Through static analysis, it is possible to gain an overview of all the bots' capabilities. In this case, using tools such as dnSpy, and de4dot allowed the source code of the bot and controller to be deobfuscated and reversed so the contents of the source code for each can be seen. An issue that occurred however was the fact that some string tokens were still obfuscated, however using de4dot’s command line argument “-–strtyp delegate” allowed further deobfuscation so that all commands and methods were made clear.

Bot Controller

The bot controller when statically analysed is easy to follow. At the start of its execution, it creates a new instance of “class1” uaing port 5001 to act as a listener for the bots that will try to connect to it, this is done with the “IPAddress.Any” argument meaning it will listen to any IP trying to connect. From this, Method0 is called, which is responsible for printing to output the connection made, accepting the client and string Method1. Method1 is responsible for creating the strings that will be asent over the network (responses to commands) by using streamWriters and Readers bound to the tcp client that was created on bot connection (ASCII encoded). Arrays created here are array, array2, array3 and array4. (Appendix botnet.1) It then goes to interpret any communications from the bots by using a variable called text, which is assigned the value of the streamReader of the tcp client. Each response specified by the controller will result in a certain response from the controller, using method5 to transmit the signal back to the bot. (Appendix botnet.2 – botnet.3). Below can be found the 15 commands for the bot, proving the dynamic analysis by showing the responses, and subsequently the functions behind them.

1. “Hello”
   1. As found in the dynamic analysis, the hello command when sent from the bot, as well as custom (Appendix Botnet.4), will return in "Welcome to the gang...". The code snippets in Appendix Botnet.1 shows this, as well as the Wireshark capture screenshotted in Appendix Wireshark.1.
2. “Get”
   1. As found in the dynamic analysis, the Get command when sent from the bot, as well as custom (Appendix Botnet.4), will return in a base64 string that can be rendered into an image (Appendix Botnet.5). Dynamic analysis would suggest that the bot sends only one base64 string, however static analysis shows that two base64 strings are parsed together, using a space as a delimiter. This is shown in Appendix Botnet.6.
3. “Test”
   1. As found in the dynamic analysis, the Test command when sent from the bot, as well as custom (Appendix Botnet.4), will return either an MD5, SHA1 or SHA256 encrypted string of a random word from the array “the, owls, are, not, what, they, seem”. This random is decided by the result of a modulus calculation; if divisible by three completely it will use an SHA1 hash, if MOD 1 is the result then SHA256 will be used, otherwise MD5 is used. The proof of this can be seen in Appendix Botnet.7.
4. “Failover”
5. “Connect”
   1. As found in the dynamic analysis, the Connect command when sent from the bot, as well as custom (Appendix Botnet.4), will return one of 6 sites (apple.com, microsoft.com, ibm.com, twitter.com, hpe.com, and bbc.co.uk).
6. “Takedown”
   1. As found in the dynamic analysis, the Takedown command when sent from the bot, as well as custom (Appendix Botnet.4), will return one of three ciphers; mentioning Flashpoint in relation to the Mirai Malware, mentioning both types of malware (lockscreen and encryption), and mentioning Cryptolocker ransomware. The keys are Apple, Orange and Peach respectively. (Appendix Botnet.8)
7. “Capture”
8. “Keepalive”
9. “Look”
10. “Code”
11. “Generate”
12. “Snoop”
13. “Loop”
14. “Goodbye”
15. “Hint”

# Section 3 – Prototype Defences Implementation and Testing

# Section 4 - References

# Bibliography

Aussems, E., Noë, B. & Rivera, N., 2014. *Botnets - A tenacious Web Technology,* Leiden: Leiden University.

Chawla, A., Lee, B., Fallon, S. & Jacob, P., 2018. *Host based Intrusion Detection System with,* Ireland: Athlone Institute of Technology.

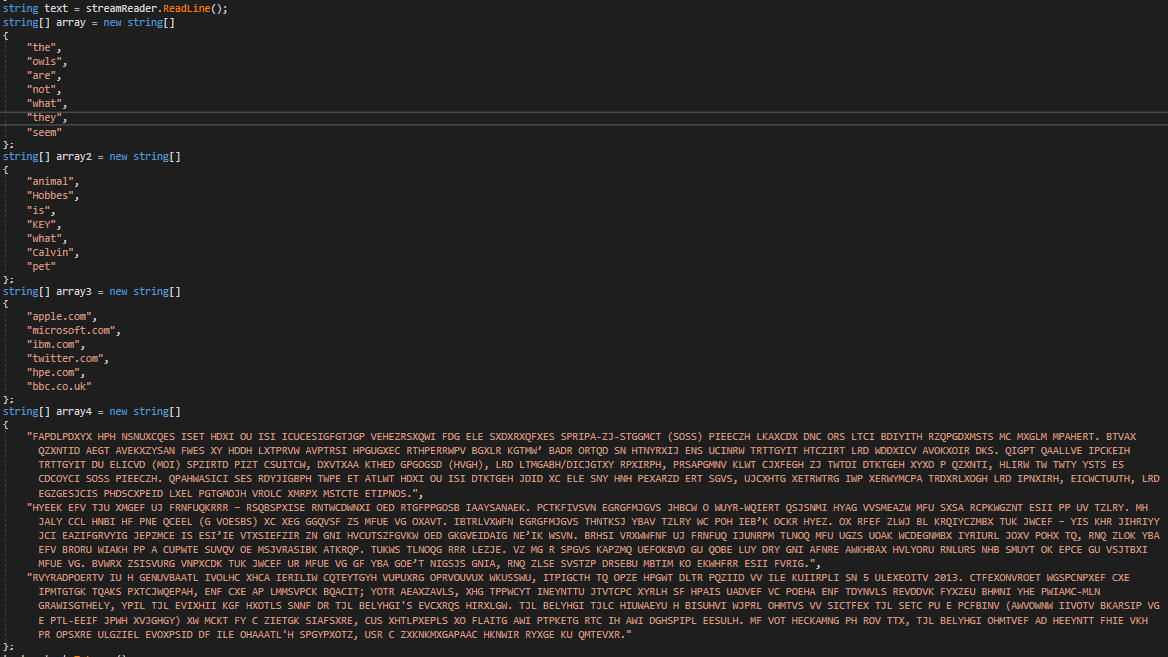
Kaspersky, 2019. *What is Botnet? | Preventing Botnet Attacks | Kaspersky Lab UK.* [Online]   
Available at: https://www.kaspersky.co.uk/resource-center/threats/botnet-attacks

The Honeynet Project, 2008. *The Honeynet Project, 2008. Uses of Botnets. [Online].* [Online]   
Available at: Available at: https://www.honeynet.org/node/52  
[Accessed 14 March 2019].

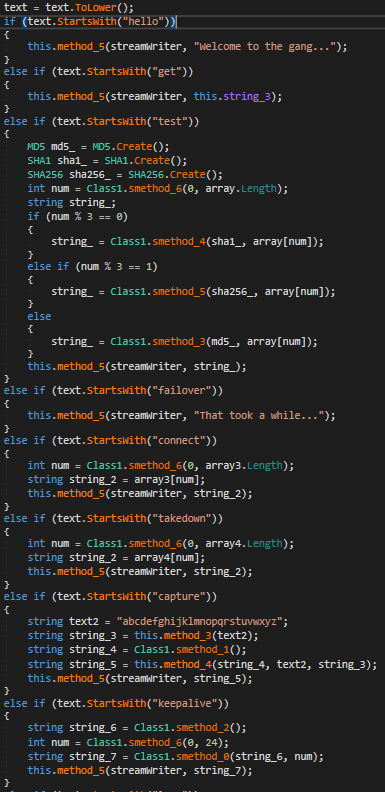
Vuong , S. & Alam , M., 2011. Advanced Methods for Botnet Intrusion Detection Systems. *Intrusion Detection Systems,* pp. 55-80.

### Appendix

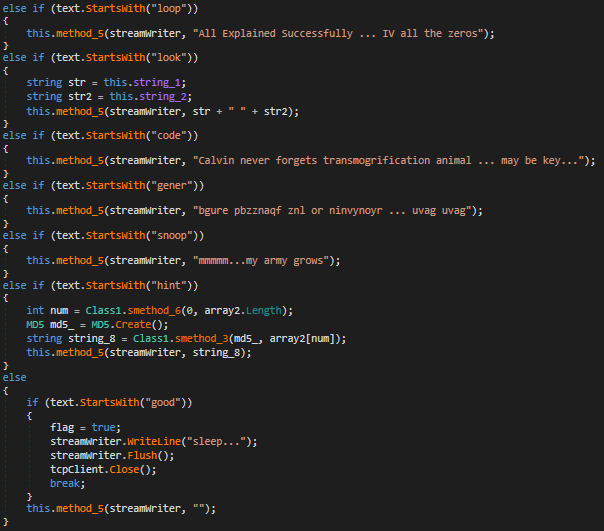
Botnet.1



Botnet.2



Botnet.3



Wireshark.1

