Brief Report

A Review on Botnet Attacks

Mia Deeks *

Posted Date: 6 July 2023

doi: 10.20944/preprints202307.0366.v1

Keywords: honeypot; botnet; blockchain; security; IoT; auto-encoder

Brief Report

Botnet Attacks: Characteristics and Detection Techniques

Mia Deeks

Engineering and Informatics, University of Bradford, Bradford BD7 1DP, England; mdeeksho@bradford.ac.uk

Abstract: Botnet Attacks are one of the many types of ways to attack users in their day to day browsing, often of different trends and techniques to take your information without your control. Most firewalls can amply protect you against these trends and techniques, but scarily enough, recent articles and research have shown that they are starting to bypass said firewalls through layers of attacks, confusing your best line of defense. In this Report, we will be looking into exactly what are Botnets and the different subsets, what we can do to combat Botnet Attacks, where do they come from, their origin, the analysis of their kit and what we can put together from information collected. Through this we will be able to reach a conclusion of what we can use to possibly mitigate the effect they have on the internet and stop them from enacting their purpose.

Keywords: honeypot; botnet; blockchain; security; IoT; auto-encoder

1. Introduction

Botnet Attacks are as previously stated, one of the many subsections of technological attacks one can encounter on the internet, although they are preferable to company attacks, that doesn't mean they can't single out a singular person. Their process spans a simple yet wide view, with the tools given to them attackers and hackers can host these 'bots' into overpowering even the most reliable of services given to consumers on the internet.

Relating to a report named 'Bad Bot Report', it was detected that up to 25.6% of internet traffic was attributed to Bots that are sold to different attackers and hosts. [1] 25.6% of users are stated to be inhuman and made of code. We will look closer to who these bots are in section II.

2. What Are Botnet Attacks?

Botnets are a dangerous hivemind of fake accounts spun together to create a web, hoping to catch the unsuspecting victim within from a series of attacks which will be categorized within second III. For now, let's look at the ongoing botnet crisis, and understand what they are. Botnets are comparable to worms, travelling at hi speeds through different computers, the difference being that bots can cooperate and target effectively, keeping in mind they are there for a malicious purpose. [2] When Botnets spread, they render normal computers compromised through the 'bot' program, which then takes orders from the host behind the attack and can even communicate with other bots within the area/LAN/WAN. [3] They are directly controlled unlike most viruses, trojans and harmful applications, but because of this they don't have the tools needed to directly 'hack' into anything, only to steal information.

2.1. Toolkit of a Botnet Lifecycle

- "Bot – 'Zombified'/Infected machine that waits for commands from the Botmaster.
- Botnet – A network/web of bots under control of the Botmaster and used for individual or group purposes, they are essentially a 'horde' of zombie computers.
- C&C – Command and Control channel which the Botmaster uses to contact the bots under their control.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2023 doi:10.20944/preprints202307.0366.v1

- IRC – Internet Relay Chat, used commonly on the internet and provides one to one/one to many instant messaging, with different channels for each network over a varying set of subjects." [10,12] (1)

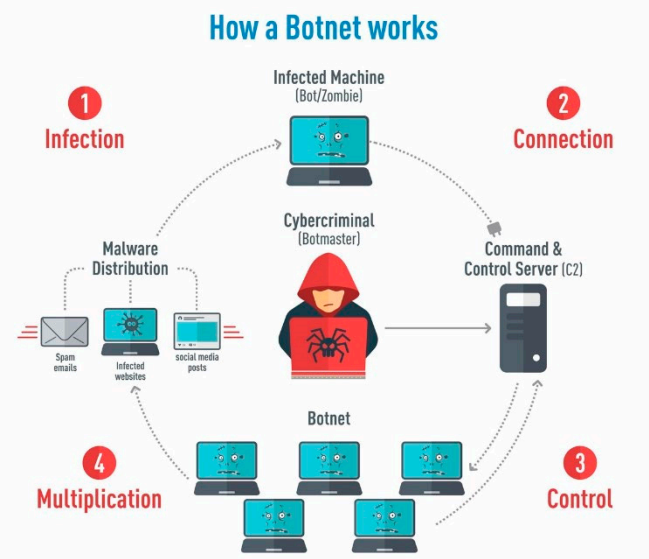

Figure 1. Steps taken to 'infect' other Computers and create new Botnets [13].

2.2. Infection of consumer computers and ongoing evolution

Within Figure 2 we can translate the general process with the definitions we now know from section II subject A. Within the left lifecycle: The Botnet Lifecycle; The Botmaster or in this case the Bot herder sets up the first bot within the network, adjusting their settings before sending them off with a newly registered DDNS connection. Through this they multiply, launching DoS attacks onto different victims [2], the botnets harvest information through a term labelled 'traffic sniffing'.

What is Traffic Sniffing? Most often its attributed to Bluetooth connection, impractical and practical uses. It listens in to conversations like a herd technique, observing whilst it corners a target into being 'captured' and analysed. [15] Sometimes the botnet is lost to another horde of botnets, as they are all not strictly on the same team, but owned by different individuals with again, malicious intentions [17]. Once the botnet has been routed, or become useless where it has been positioned, it abandons its post and unregisters from the DDNS. The Single Bot Lifestyle on the right from the left diagram, is a more exclusive look to the bots themselves and what they do when left in an idle state, after being used appropriately.

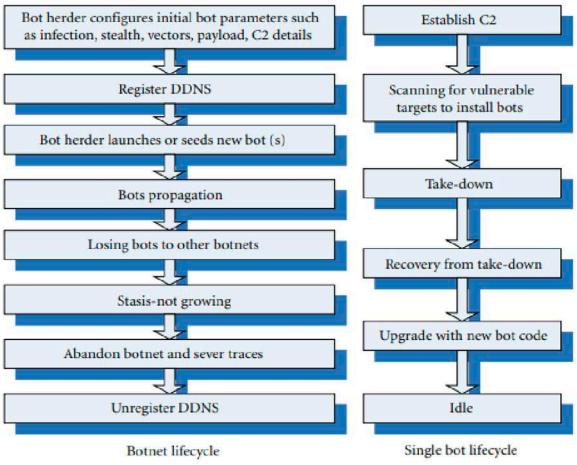

Figure 2: Cycle of a Botnet and single bot [2,14]

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2023 doi:10.20944/preprints202307.0366.v1

2.3. Evaluation of a Recent Botnet Attack

Now let's take a look at one of the best examples of botnet 'invasions' from recent years with the newly acquired understanding. One of the most well-known botnet attacks was the 'Mirai' botnet, which overwhelmed the internet from September 2016 and onwards. [6] The Mirai Botnet has been described as a 'outbreak', calling back to the multiple times botnet is described as a zombification of computers, when looking at the statistics of the time, it can be seen that the botnet infected approximately 65,000 IoT devices in the first 20 hours of its 'invasion'. Overwhelming the servers and creating irreparable damage. [6]

From what we've learned so far, how did they manage to do this? By attaching malware to their network of bots, the Mirai botnet was successful with DDoS attacks, they were able to target "DNS provider Dyn and Lonestar Cell, a Liberian telecom." [5] This was because they were being fed information through a unique set of code with a network telescope, giving them eyes to see hundreds of thousands of lines. Tracking their hosts from afar. "On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute during this period." [6,16].

Calling back again to a previous statement, we see now how they got their information, when they were collecting it from the 'network telescope', Mirai Botnets were sifting through a network known as Merit Network over a seven-month period. Lasting from July 18th, 2016, to February 28, 2017 [6] Knowing this, it is plain to see just how they found their hosts, the botnets lied in wait for the perfect moment to strike from months of preparation from the attacker's/hosts, before being sent out to properly commence orders by the bot herders/ bot masters [18]. Infiltrating an entire company from malware expansion, all high profile. They were able to control and a good chunk of information. From this, we can deduct that bots are harmful on their own, and even doubly so when given tools. I.E 'network telescope'.

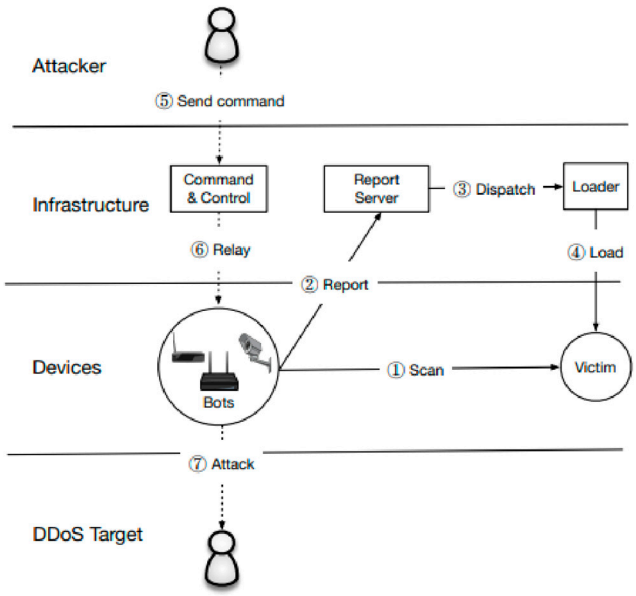

Figure 3: Mirai Operation visualised. [6]

Taking some time to look at the diagram (3) we can see it follows a similar hierarchy to the ones previously spoken of when we compare back. However, there is small changes that show their bots worked during the initial incursions.

Mirai bots scan for the IPv4 server, and use a hardcoded dictionary of IoT credentials stolen from other compromised computers, when successfully logged in, it 'reports' back and sends for a loader(malware) to infect the device, creating another bot. [6] The only way it lacks difference is when calling back to what we previously established, there was no prime directive, but here within the Mirai Botnets we can see there is a motive. They had pre-planned and adjusted the bots for invasion.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2023 doi:10.20944/preprints202307.0366.v1

3. Botnet Detection Avoidance

Within this section, we will be looking at what Botnets and Bots can use to hide their malicious activities and intentions from the eyes of security Automated Security or people assigned to finding the botnets themselves, and what these techniques do/can do to have an effect on the internet and its surroundings [19]. Most of these techniques are to do with evading detection, as that is the most important and grave weakness to the bots, once one is discovered, it is easy to link them to the rest of the Botnet, and eventually, the Bot Herder.

3.1. Fast Flux

FF or Fast Flux is a technique used by bots to evade botnet detections and revealing the rest of the horde they are either with or without, by using IP fast fluxing, they change their IP between multiple addresses, alongside their domain names. [7,9] Fast flux is a common and easy technique for bots to enact, all it takes is some coding and soon the bot will be able to mask its true placement along the network to security systems, causing confusion between security AI, and needing human interaction at times.

3.2. Double Flux

Double-Flux is a more consistent and put together way of counter-detection against being routed as a bot within a botnet. Similarly, it works like Fast Flux, only the difference is that it changes the of flux agents and of the DNS servers, leaving it entirely unknown even to those hosting the bots with this technique. [9] AI would be able to find the bot within the flux of interchanging domains and Ips, if given enough time, but it would also be easier to get human minds into working through and finding the bots specifically [20]. As the bots go through more security to encrypt their location whilst they corrupt other computers, it becomes increasingly difficult to find them. That is until you see the 'zombified' computers left in the wake of its code running.

3.3. Domain Flux

Domain Flux or DF is the best bots can get from being protected against detection algorithms, essentially the idea is that the algorithm must make sure all bots can generate domains via the same seed, using the auto-generated names to contact the C&C server. If the client doesn't go through, the bots in question will eventually try to hard-code their way through with configuration files. [9] Usage of Domain Flux is the again one of the best types of counter-detection a botnet could use, the dedicated servers often fail against this technique through its widespread range.

4. How Can We Defend Against Botnets

4.1. Defending against the Mirai Botnet

There are many different ways to defend against Botnets, and different uses of 3rd party approved software guaranteed to protect your computer from being compromised by the 'horde' of bots. First, let's look back at the set of problems we first faced and analyzed in section II subject C.

Mirai exploited open ports with hard-coded credentials stolen via other bots within their botnet network, the bots probing unsuspecting computers and files for information it can get for finding ways into previously secured companies. [5] This caused 65k IoT devices to become rampant and useless, another zombified computer/machine. [5] How do we combat this system of unique scanning and fastworking ai? Block-chains.

Within [5,10] there is a proposal to use block-chains to retaliate against Mirai based botnet code, after it became open source and was available to all attackers and hosts willing to use it for their own measures. The Block-chain is used to continuously exchange information once banks are full, swapping them back and forth to create a fluidity within the autonomous system, refreshing consistently so that Bots do not have the chance to take the information they want, and only that

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2023 doi:10.20944/preprints202307.0366.v1

which is swapped. Who is to say, however, that the bots would get to that stage? There are two other processes used in this system to ensure that bots would not have the chance to access the block-chain and its consecutively changing information [21].

Secondly, Hosts are integral to the newly proposed system. We have two types of hosts: Normal Hosts and IoT Hosts, within the system, they will use exclusively IoT Hosts, due to the fact that they can be remotely monitored and controlled [5] and are less liable to compromise unlike your normal standard computer when put near a botnet, IoT Hosts are better by one advantage: You can monitor them during a botnet attack. Finally, we have the Autonomous System (AS), used in both Block-Chain and IoT technologically, however it will be responsible for storing a list of IP addresses, communicating who has been compromised, who will be compromised, who hasn't, and who could be. [5] The best detection against Botnets is to know who is dangerous, and who isn't, which is exactly what the Autonomous Systems do naturally.

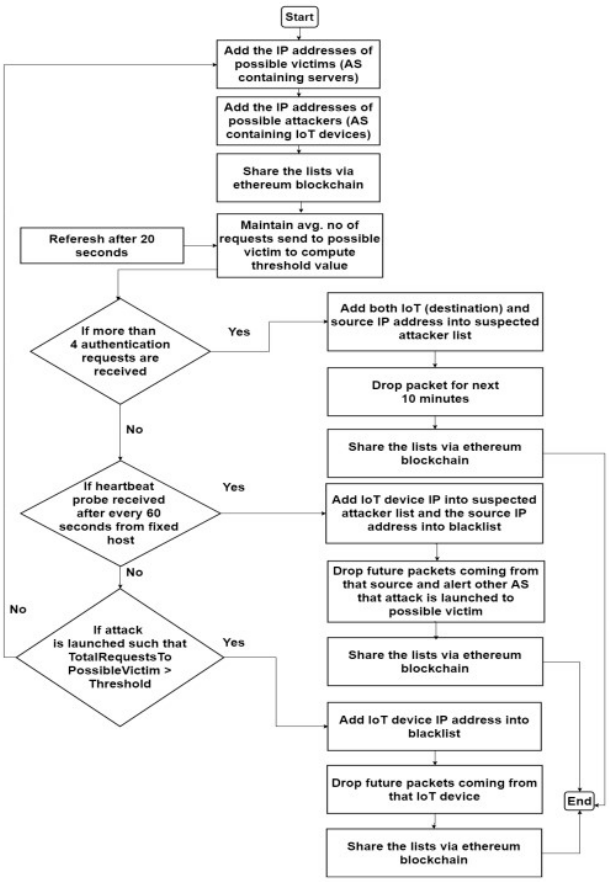

Figure 4: Flow Chart of the proposed approach [5]

4.2. Using Honeypots to detect Botnet Attacks

Honeypot is one of the few most important ways to counteract bot-netting and bots as a whole from their horde-lifecycles and mindset.

The basic principle if we follow (5) is that honeypots act as bots that a bot controller would send forward to bring malicious applications to unsuspecting users and computers, essentially granting us an 'inside man' to the botnet as a whole, using fire with fire to take the information of the bot herder, and shut down their operations. Within the Figure 5 [3], we see that the bot controller needs to verify the fake bot before he lets them into the 'horde', after seeing his sensor light up and send a message to him, letting him know there is a bot heading their direction. Most often the botmaster has set up multiple sensors to prevent failure and malicious traffic heading his way, and he cannot predict if the bots are again, his or another's.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2023 doi:10.20944/preprints202307.0366.v1

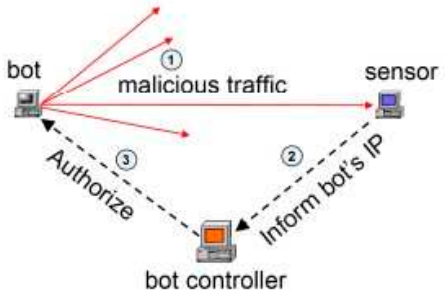

Figure 5: Detecting honeypot bots through a botnet [3]

4.3. Using DNS server Detection Techniques

DNS detection is used based on the DNS information outputted by a bot within a botnet, "bots typically initiate connection with C&C server to get commands" [11] From this we can gather that it would be best to monitor the C&C server from thereon out to make sure there is no malignant data trafficking through, alongside the fact that the DNS origins of each computer that passes through be immediately checked [22]. The best way to find out who is a bot and who isn't is to check how many times a query has been pinged for DDNS rates, this goes back to how bots often try to hide their data and where they are on the network through fluxing [11].

5. Are There Flaws to This Defense

5.1. The Honeypot's general weaknesses

One of the biggest flaws to Honeypot is the fact that fake/compromised bots on 'our' side might be sent into the general public, in the completely opposite direction of the botmaster and his sensors. Causing an innocent person and their computer to have false malware spread to them and collecting information on the wrong person.[3]

Another flaw is that the botmaster could be aware that not all bots are his, and thus has encrypted a special message on his bots or tagged them in a specific way to ensure that they are his own and not anyone else's. This makes honeypots become blatantly obvious, and he can redirect them in the other direction again. Honeypot has also become in general outdated except for specific versions that are only continuously able to update [9].

5.2. Honeypot vs Reconaissance Worm

The dubbed 'two-stage reconnaissance worm' [3] has two parts to play, the first being to decide once a computer is compromised, whether it is a honeypot or not. The second part is "allowing the infected host to join the P2P botnet" [3]. On paper this sounds intriguing, great even when you connotate it with the fact that the spearhead forces the host to decide, however without the stages and the sudden 'spearhead' movement, the botmaster could predict that it is not a true 'other' bot, and not authorize it mingling with the rest of the bots. This poses a problem for the net since the hosts A would thereon be inaccessible with the other hosts.

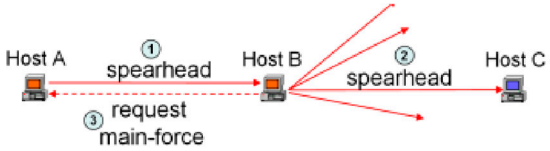

Figure 6: [3] Reconnaissance Worm Illustrated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2023 doi:10.20944/preprints202307.0366.v1

5.3. Honeypot vs Advanced Reconaissance Worm

Presenting the dual-honeypot worm, or two-stage honeypot worm, it is a direct upgrade from the previous one we covered and is immensely useful. However, it again has drawbacks that can't be relied upon concurrently and must be considered that it is a risk and may alert the botmaster that there are fake bots trying to access their host domain [23]. The spearhead must decide on its own whether the host is remote or not. Its complexity does not make it unique, as botmasters have been renowned for making anti-detection against this specific technique, being aware of its effectiveness has made it only used once [3,9].

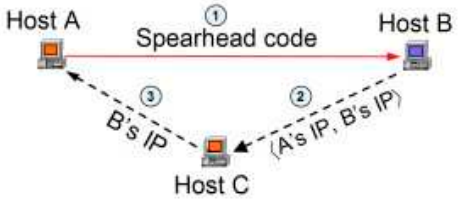

Figure 7: [3] Reconnaissance Worm Illustrated attacking a dual honeypot bot

5.4. Mirai Open Source Code

Mirai's Open-Source Code could be a massive downfall for the people trying to mitigate botnet attacks, as although Mirai's impact on the internet and IoT technology was immense and well-educating, it also became free for everyone to see and learn from, meaning those who had been needing an outlet for malicious intent. Giving them the keys to the source-code means they can perfect the already scary bots and improve on them into becoming worse than before when experienced by clients and compromised computers.

6. Conclusion

From evidence gathered and research taken, it is safe to assume that Botnets are a challenge we still currently face in the IoT and WWW (world-wide web) even to this day in 2022, the previous problems have all amounted to different solutions and new pathways. Both bad and good. Whenever something is learned about the inner workings of bots distributed online, there are countermeasures taken but often to our detriment with botmasters improving on previously effective code. However, this does not mean all hope is lost for the detection of botnets, new algorithms as demonstrated in the subjects talked about have proposed new and exciting ways of preventing mass amounts of 'zombified' computers, possibly nullifying the production and propagation. To do this it means we must build layers upon foundations of encryption and security algorithms, that bot AI simply cannot penetrate with hard-coding solutions anymore, information should be kept hidden and away from the secret observing eyes of bots. Again, this is no easy task, but it does not mean it is impossible to stop the ongoing hordes of fake profiles and crashing multiple sites with DNS overloads. The best course of action is to continue to mitigate attacks by finding out who will be attacked first, who is next, who may be at risk, who are the attackers and where are they sourced from, alongside the proposition of a new Block-Chain algorithm. The block-chain stands the best chance under layers of security to seal the bots from overloading servers and cornering computers into being compromised.

References

1. The most recent Botnet Attacks: The 2022 Edition The Most Recent Botnet Attacks: 2022 Edition (clickguard.com)
2. Liu, J., Xiao, Y., Ghaboosi, K., Deng, H. and Zhang, J., 2009. Botnet: classification, attacks, detection, tracing, and preventive measures. EURASIP journal on wireless communications and networking, 2009, pp.1-11. [Figure 2]
3. I. Ghafir and V. Prenosil, "Malicious File Hash Detection and Driveby Download Attacks," International Conference on Computer and Communication Technologies, series Advances in Intelligent Systems and Computing. Hyderabad: Springer, vol. 379, pp. 661-669, 2016.
4. Wang, P., Wu, L., Cunningham, R. and Zou, C.C., 2010. Honeypot detection in advanced botnet attacks. International Journal of Information and Computer Security, 4(1), pp.30-51. [Figures 5,6,7]
5. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D. and Elovici, Y., 2018. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), pp.12-22.
6. M. Hammoudeh, I. Ghafir, A.Bounceur and T. Rawlinson, "Continuous Monitoring in Mission-Critical Applications Using the Internet of Things and Blockchain," International Conference on Future Networks and Distributed Systems. Paris, France, 2019.
7. Ahmed, Z., Danish, S.M., Qureshi, H.K. and Lestas, M., 2019, September. Protecting iots from mirai botnet attacks using blockchains. In 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD) (pp. 1-6). IEEE. [Figure 4]
8. I. Ghafir and V. Prenosil. "Proposed Approach for Targeted Attacks Detection," Advanced Computer and Communication Engineering Technology, Lecture Notes in Electrical Engineering. Phuket: Springer International Publishing, vol. 362, pp. 73-80, 9, 2016.
9. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M. and Kumar, D., 2017. Understanding the mirai botnet. In 26th USENIX security symposium (USENIX Security 17) (pp. 1093-1110).
10. Borgaonkar, R., 2010, July. An analysis of the asprox botnet. In 2010 Fourth International Conference on Emerging Security Information, Systems and Technologies (pp. 148-153). IEEE.
11. I. Ghafir and V. Prenosil, "Advanced Persistent Threat and Spear Phishing Emails." International Conference Distance Learning, Simulation and Communication. Brno, Czech Republic, pp. 34-41, 2015.
12. Zhang, L., Yu, S., Wu, D. and Watters, P., 2011, November. A survey on latest botnet attack and defense. In 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 53-60). IEEE.
13. Lee, S., Abdullah, A. and Jhanjhi, N.Z., 2020. A review on honeypotbased botnet detection models for smart factory. International Journal of Advanced Computer Science and Applications, 11(6).
14. I. Ghafir, J. Svoboda, V. Prenosil, "A Survey on Botnet Command and Control Traffic Detection," International Journal of Advances in Computer Networks and Its Security (IJCNS), vol. 5(2), pp. 75-80, 2015.
15. Gallopeni, G., Rodrigues, B., Franco, M. and Stiller, B., 2020, June. A practical analysis on mirai botnet traffic. In 2020 IFIP Networking Conference (Networking) (pp. 667-668). IEEE.
16. Feily, M., Shahrestani, A. and Ramadass, S., 2009, June. A survey of botnet and botnet detection. In 2009 Third International Conference on Emerging Security Information, Systems and Technologies (pp. 268273). IEEE.
17. I. Ghafir and V. Prenosil, "Blacklist-based Malicious IP Traffic Detection," Global Conference on Communication Technologies (GCCT). Thuckalay, India: pp. 229-233, 2015.
18. Li, C., Jiang, W. and Zou, X., 2009, December. Botnet: Survey and case study. In 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC) (pp. 1184-1187). IEEE.
19. Botnets: Dawn of the Connected Dead Botnets: Dawn of the connected dead (emsisoft.com) [Figure 1]
20. J. Govil, "Examining the criminology of bot zoo," in Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07), pp. 1–6, Singapore, December 2007.
21. S. Eltanani and I. Ghafir, "Aerial Wireless Networks: Proposed Solution for Coverage Optimisation," IEEE Conference on Computer Communications Workshops", IEEE, 2021.

 doi:10.20944/preprints202307.0366.v1

22. Albazrqaoe, W., Huang, J. and Xing, G., 2016, June. Practical bluetooth traffic sniffing: Systems and privacy implications. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (pp. 333-345).
23. M. Bailey, E. Cooke, F. Jahanian, and J. Nazario. The Internet Motion Sensor - A Distributed Blackhole Monitoring System. In 12th Network and Distributed Systems Security Symposium, 2005.