Botnet: Evolution, Life Cycle, Architecture and Detection Techniques

Jagdish Yadav
Department of Computer Science, Himachal Pradesh University, Shimla, Himachal Pradesh, India
yadavjagdish65@gmail.com

Jawahar Thakur
Department of Computer Science, Himachal Pradesh University, Shimla, Himachal Pradesh, India
jawahar.hpu@gmail.com

ABSTRACT

Botnet has become thorn for the Internet and the cyber security. Botnets are network of zombies controlled by botherder for their malicious and nefarious activities. These activities include distributed denial of service (DDoS) attack, click fraud, phishing, spamming, malware dissemination, traffic sniffing etc. The botnets are capable of bringing down the whole network within seconds. A number of techniques have been developed to detect the botnet and dismantle them but attackers have shown that they are always ahead of these detection techniques. This paper discusses the evolution of botnet, life cycle of botnet, its architecture and detection techniques. There are numerous detection techniques proposed by researchers that are reviewed on the basis of various parameters in this work. In spite of having such vast amount of detection techniques botnet can't be tackled because of the dynamic nature of the Internet. The war against botnet can be seen as cat and mouse which is never-ending.

Keywords: Architecture, Botnet, Detection Techniques, Life Cycle.

1. INTRODUCTION

With the passage of time the technology has evolved to a large extent. But the drawback of the technology remains the same i.e., security. Many techniques have built to provide the security from unauthorized access, malicious and nefarious intent. But the hackers and nefarious intended people proved themselves always ahead of the security experts and give them a tough challenge.

One of the major security breach and nefarious activities are done by using Botnets. Botnet is formed by melding a large number of threats into one [1]. A typical botnet consists of a bot-master/botherder, bot server and bot clients or zombies. The concept of botnet came into existence with the concept of IRC that was mainly designed for group communication [1]. Bots were originally designed as a virtual individual that could perform actions on IRC in the absence of his owner. And the example of this kind of bot was GM created in 1989.

Botnet consists of large network of compromised computers that are remotely managed by the attacker (bot-master) used to attack other computers for malicious or nefarious intent. Some of these computers may be participating willingly or some are hijacked by Trojan or Malware [2]. The malicious activities performed by botnet includes sending spams, phishing, DDoS, click fraud, key logging, sniffing traffic and spreading new malware [3]. The taxonomy of the botnet is provided in Fig (1).

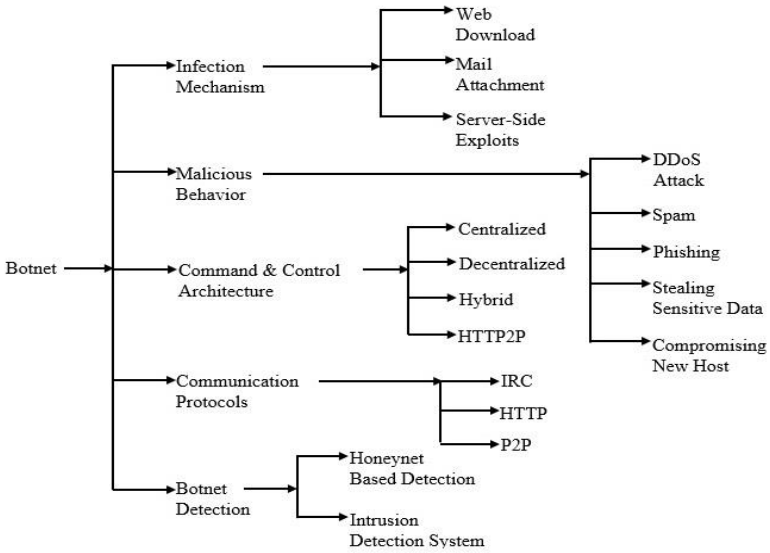

Figure 1. Taxonomy of botnet.

Botnet is different from the hacker's break-in in two ways. Firstly, the clients in botnet must be able to perform specified task. Secondly, many clients are required to act in a coordinated manner to accomplish a common task. In both the scenarios little or no intervention is expected from the attacker [1]. How the attacker launches an DDoS attack against an individual is shown in Fig (2).

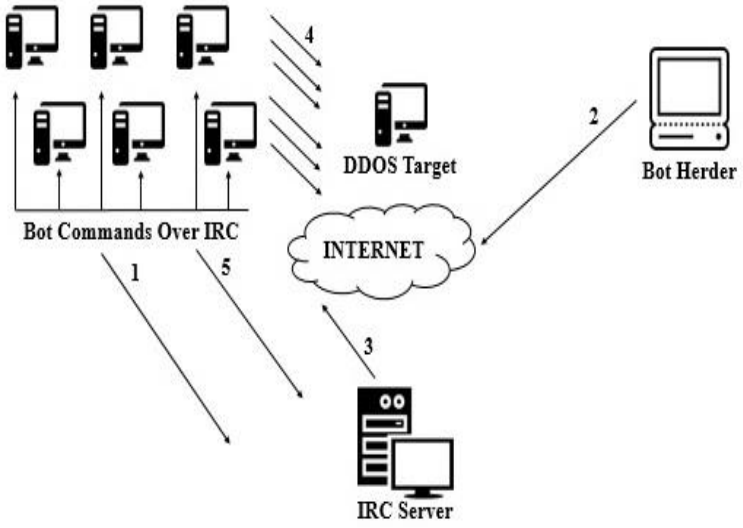

Figure 2. Botnet launching a DDoS attack.

In the first step, when a new bot-client is created it needs to rally back to C&C server and wait for the commands. Secondly, the botherder sends the command to the C&C server specifying the target, time of attack and type of attack and which bot-clients are allowed to participate in the attack.

The next step is to continuously monitor the C&C server to get the commands from the botherder. In the fourth step, the bot-clients perform the specified action at the given time interval and start sending network traffic to the target. The traffic sent is so enormous that the web site is unable to differentiate between the benign and attack traffic and soon starts to process the attack traffic. The last step is to report back the results to the C&C server.

Section 2 describes the evolution of botnet, how they evolved from a gaming bot to such a nefarious thing on the Internet. Section 3 discusses the life cycle of the botnet and various steps involved in it. Section 4 presents various architectures used by the botnet for its propagation and communication. Section 5 deals with various botnet detection techniques. In last, section 6 concludes the whole paper along with the future scope.

2. BOTNET EVOLUTION

The term Botnet doesn't exit till 1988. The term came into existence after the development of first application layer protocol named Internet Relay Chat (IRC). The IRC provides communication in the form of text and works on a client/server model. They are intended for group communication on channels but also provide one-on-one communication and data transfer capability to share files [1].

The first IRC bot was GM Bot developed in 1989 [1]. The GM Bot play the game 'Hunt the Wumpus' with the IRC users. After that the bots begun to evolve from user friendly bots to malicious one. In the early stages of the bot development, their functionality was restricted to redirection, uploading, downloading and remote access of the victim [1]. Then they started launching DoS attack, port scanning, cloning and nowadays they gain the full control over the victim's system. Since that day after the occurrence of first bot many bots have emerged till date and are given in the Table (1).

Table (1): Botnet Evolution.

S. No.	Name	Type	Spam Capacity	Aliases	Year
1	GM Bot [1]	IRC	-	-	1989
2	Eggdrop [4]	IRC	-	-	1993
3	GTbot [5]	mIRC	-	-	1998
4	Pretty Park [8]	IRC	-	Trojan.PSW.-CHV, I-Worm.Pretty-Park	1999
5	SubSeven [8]	-	-	-	
6	Agobot [1]	IRC	-	Gaobot	2002
7	SDbot [1]	IRC	678,000	IRC-SD-bot	
8	Spybot [1]	IRC	-	Milkit	2003
9	Sinit [6]	P2P	-	Calypso	
10	Rbot [1]	IRC	1,900,000	-	
11	Polybot [1]	-	-	-	2004
12	Bagle [9]	-	230,000	Beagle, Mitglieder, Lodeight	
13	Mytob [9]	Hybrid	-	-	2005
14	Rustock [1]	IRC	150,000	RKRustok, Costrat	2006
15	Zeus [5]	-	3,600,000(US Only)	Zbot, PRG, Wsnpoem, Gorhax, Kneber	
16	Storm [5]	P2P	160,000	Nuwar, Peacomm, Zhelatin	2007
17	Srizbi [5]	IRC	450,000	Cbeplay, Exchanger	

S. No.	Name	Type	Spam Capacity	Aliases	Year
18	Akbot [5]	IRC	1,300,000	-	2007
19	Cutwail [5]	SMTP	1,500,000	Pushdo/ Pandex/Mutant	
20	Waledac [5]	P2P	80,000	Waled, Waledpak	2008
21	Kraken [5]	-	495,000	Kracken	
22	Conficker [5]	HTTP/P2P	10,500,000+	DownUp, DownAndUp, DownAdUp, Kido	
23	Mariposa [5]	P2P	12,000,000	-	
24	Festi [9]	-	250,000	Spamnost	2009
25	Mega-D [9]	-	509,000	Ozdok	
26	Grum [9]	SMTP	560,000	Tedroo	
27	BredoLab [9]	-	30,000,000	Oficla	
28	Keliho [9]	-	300,000+	Hlux	2010
29	TDL4 [9]	IRC	4,500,000	TDSS, Alureon	
30	Gameover Zeus [1]	P2P	-	-	2011
31	Ramnit [9]	-	3,000,000	-	
32	Chameleon [5]	HTTP	120,000	-	2012
33	Zer0n3t [9]	-	200+ server	Fib3rl0g1c, Zer0Log1x	2013
34	Necurs [9]	-	6,000,000	-	2014
35	Mirai [9]	-	380,000	-	2016
36	Smominru [7]	-	500,000(in 2018)	-	2017

The ‘year’ field shows the commencement year of each botnet, the ‘name’ field specifies the name of that botnet. Then comes ‘spam capacity’ that describes the number of anticipated bots in that botnet. The ‘type’ defines the protocol used by the botnet and the ‘aliases’ provides the alternative names by which they were known.

3. BOTNET LIFE CYCLE

The botnet follows a set of similar steps throughout their life cycle. The prospective bot-client (victim) can be exploited in many ways including backdoors by Trojans, malicious code, attacks against unpatched vulnerabilities, password guessing and brute force attacks [1]. The botnets can be created and

maintained in five phases during their life cycle: initial infection, secondary injection, connection, malicious command and control, and update and maintenance phase [10], as depicted in Fig (3).

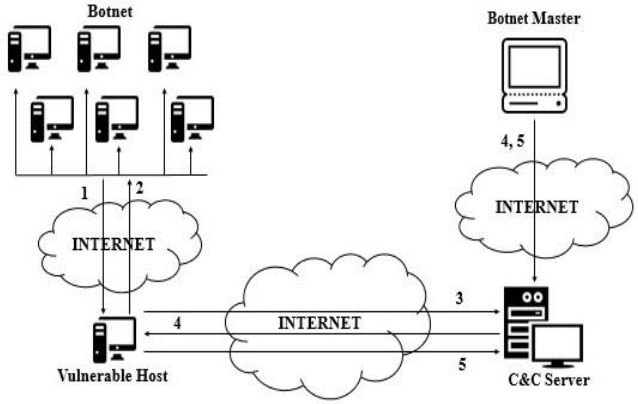

Figure 3. Phases in botnet life cycle.

In the first phase, the attacker scans the victim for known vulnerabilities and infects their machines using different exploitation mechanisms. After this initial infection, the secondary injection comes into play. Here the infected hosts execute a script known as shell-code that fetches the actual bot binary from the specified location via. FTP, HTTP or P2P. Once this bot binary is installed on the victim's computer it gets turned into 'zombie' and starts running the malicious code.

During the connection phase, the bot program establishes a communication channel and connects the zombie to the command and control (C&C) server. Once the channel is established the zombie becomes the part of the botnet army. Now the bot-master makes use of this C&C channel to disseminate commands to the bot-clients to perform the intended task. The C&C allows the bot-master to remotely control the bot-clients and perform various malicious and illicit tasks.

Last phase in the botnet life cycle is the maintenance and updation phase. It is required to keep the botnet updated to evade from the latest detection techniques. Also, this updation is necessary when the server migration needs to be implemented to keep the bots alive and invisible. The steps involved in botnet lifecycle [1], are given in Fig (4).

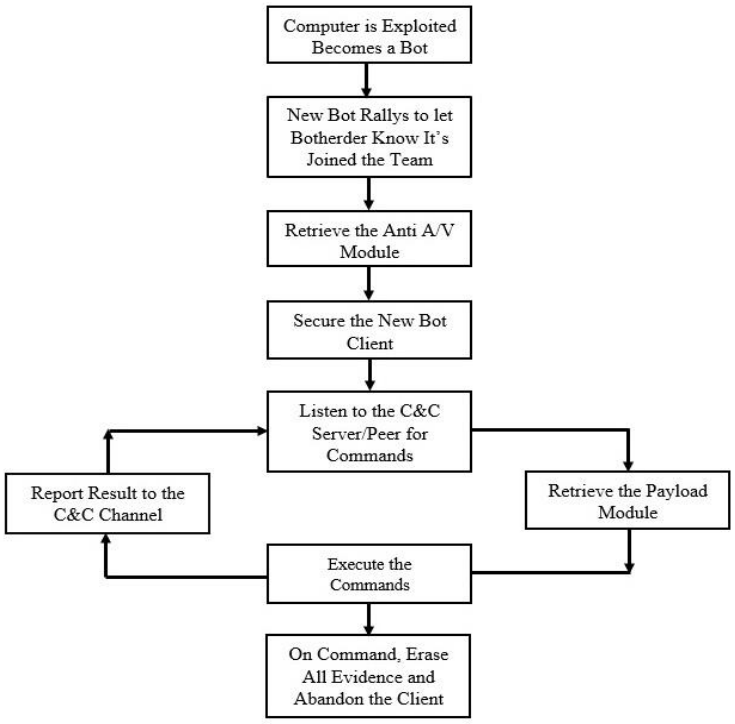

Figure 4. Flow chart for botnet life cycle.

When the computers are exploited, they become bots. After that the bot clients initiate contact with the botnet command and control server to let the botherder knew that they have joined the bot network. The above process is called rallying. Rallying is basically the term given when for the first time a botnet client login in to a C&C server.

When rallying is done the next step is to secure the new bot client from removal. This is done by using the anti-antivirus tool which helps them to hide from the antivirus. This helps in securing the new bot client. The next step is to listen the commands on the C&C and retrieve the commands for their execution. After execution the results needs to be sent back to C&C server. The last and the most important step is to erase all the evidences.

4. BOTNET ARCHITECTURE

The botnets are developed using four types of architectures: Centralized Architecture, Decentralized Architecture, Hybrid Architecture and Hypertext Transfer Protocol Peer to Peer (HTTP2P) Architecture [2]. The first one is not that much secure because of central controlling server. The second one is hard to detect but on the other hand it is hard to manage, while the third one is combination of the first two.

A. Centralized Architecture

It is the oldest and the easiest architecture to control and manage. But the major drawback of this architecture is that it consists of a central controlling unit. If this central unit gets detected than the whole network can be taken down within a blink of an eye. The centralized architecture is shown in Fig (5).

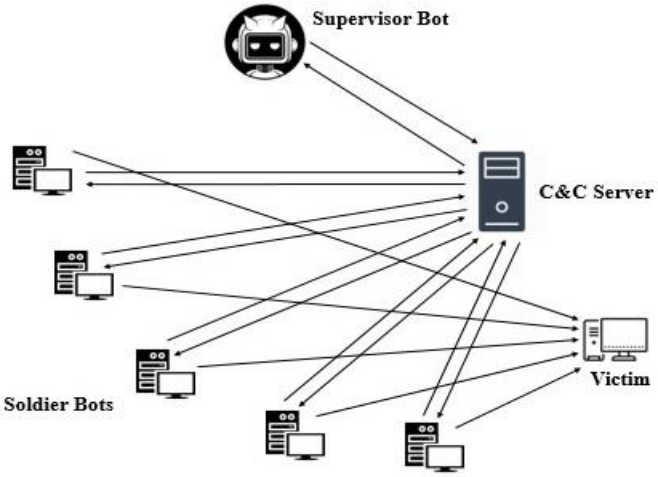

Figure 5. Centralized Architecture.

The centralized architecture can be categorized into two types: push and pull style as given in Fig (6), that determines how the bot-master's commands reach the clients. In push style C&C, the bots are connected to the IRC based C&C server and wait for the commands from the bot-master. Here the bot-master have a real-time control over the bots. Examples of IRC based botnets are Agobot, Spybot, Rbot, SDbot and GTbot [11].

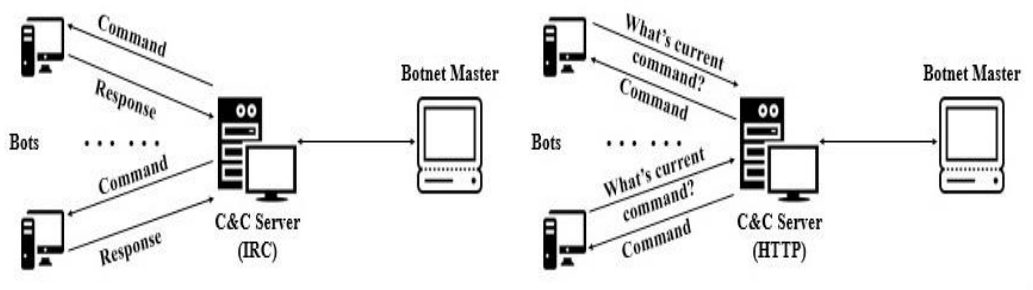

Figure 6. Push and pull style C&C servers.

While in the pull style C&C, the bots are connected to the HTTP based C&C server. The bots read the commands from the file stored at C&C server. They do not provide real-time control [11]. Examples of HTTP based botnets are Rustock, Clickbot and Zeus [12].

B. Decentralized Architecture

In order to overcome the drawbacks of the centralized architecture the attackers designed the decentralized architecture. In this architecture there are more than one C&C server. Here the bot-master sends the commands to an infected machine (zombie) which then transfers it to other zombies, acting as command and control server as well as client at the same time. That's why it is complex to detect as compared to centralized architecture. The decentralized architecture is depicted in Fig (7).

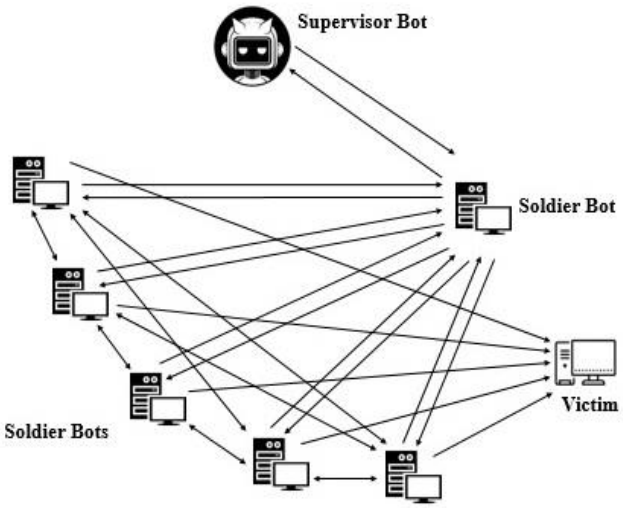

Figure 7. Decentralized Architecture.

This architecture is based on the peer to peer (P2P) protocol and also known as peer to peer architecture. The message latency and survivability of this architecture is higher as compared to that of centralized architecture [13]. Data mining techniques provides promising results when it comes to detect P2P attacks [2].

C. Hybrid Architecture

It is formed by the combination of centralized and decentralized architecture. Each bot-master in this architecture has its own list of peers and does not share it with others due to the security reasons [2]. The hybrid architecture is depicted in Fig (8).

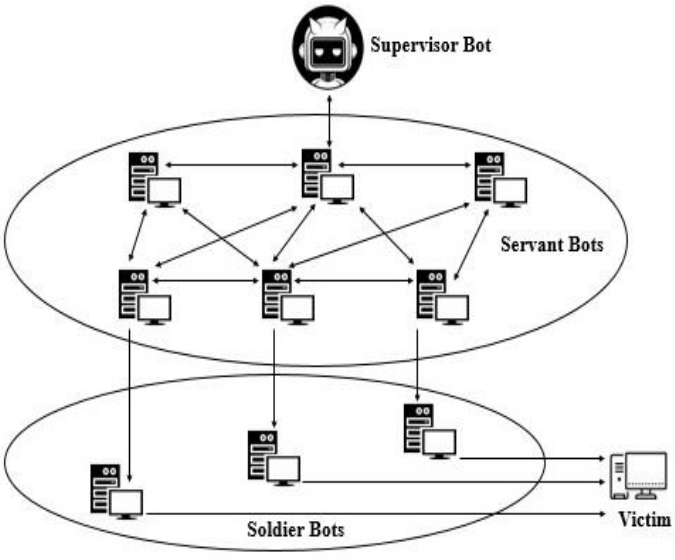

Figure 8. Hybrid Architecture.

This hybrid architecture has the advantages of both centralized and decentralized architectures. In this the C&C servers have a decentralized networks and bots connect to the server in typical client-server manner.

D. Hypertext Transfer Protocol Peer to Peer (HTTP2P) Architecture

To overcome the drawback of P2P architecture i.e. threat of Sybil attack [2], and make more complex and harder to detect architecture the attackers combine the HTTP and P2P architecture and formed the HTTP2P architecture.

In this architecture the bot-master cipher the message and continuously searches for the bot-client. When the client is found delivers the message to it. While the bot-clients are not allowed to contact the bot-master or other bot-clients until there's a call from the bot-master.

5. BOTNET DETECTION

The botnet spreads like an epidemic and this needs to be dealt with. The botnet detection plays a crucial role here as they act as an initial step towards the botnet remedy. Different techniques have been proposed by the IT community and cyber security to prevent or escape from the notorious activities of the botnets. But still these techniques are not so good and needs a lot of work to be done in the botnet prevention, detection and mitigation. A lot of work needs to be done to stop the botnet in its early stages. The researchers are continuously working on this area and trying to give a new and improved technique from the previous one. The classification of botnet detection techniques [5, 10, 14, 15], are given in Fig (9).

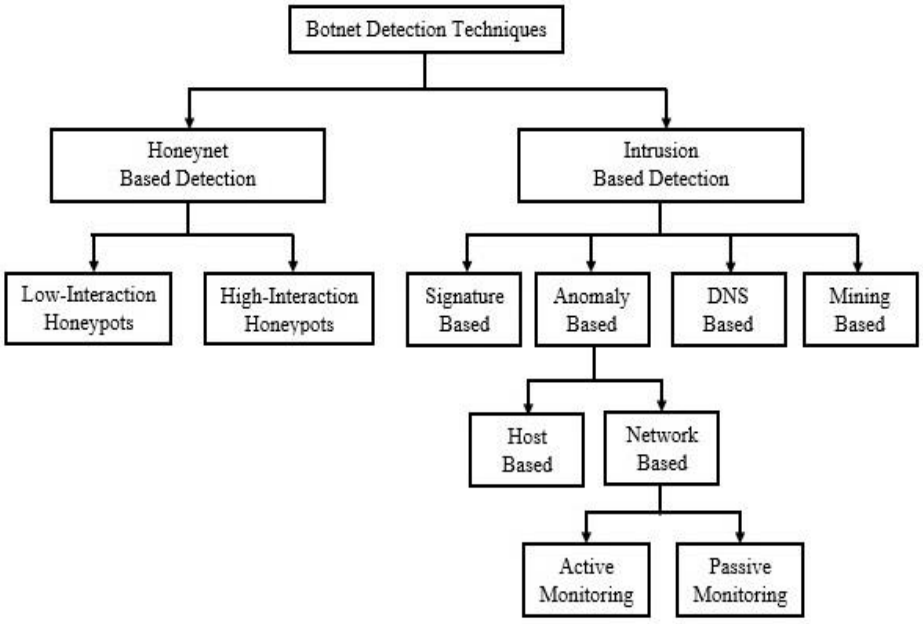

Figure 9. Botnet detection techniques.

This paper mainly reviewed the work done by the researchers in the area of the botnet detection techniques. These detection techniques have been broadly classified into two categories: honeynet based detection and intrusion detection system. Further these two techniques were sub-categorized for the better understanding.

A. Review on Honeynet-Based Detection Techniques

Honeynet is a collection of virtual servers on one physical server. The servers in these honeynet are termed as honeypot. They are used as a trap where vulnerabilities are left intentionally. The main purpose is to invite the attackers to attack that system so that to gather: bot signature for content-based detection, techniques and tools that were used by attackers, know the security holes through which bots penetrate in the network and also the mechanism of the C&C server [16].

Honeypots are categorized as: high-interaction and low-interaction based on their emulation capacity. The high-interaction honeypots can simulate all aspects of the real system and allow perpetrators to gain full access, while the low-interaction honeypots provide simulation of some features and allow partial access to the perpetrators [17].

Table (2): Tabulated review on honeynet-based detection techniques.

S. No.	Proposed Model/Method	Authors (Year)	Detection Type	Pros	Cons
1	Honeyd [18]	N. Provos (2004)	Low-Interaction Virtual Honeypot	• Can deceive the fingerprinting tools like Nmap & Xprobe. • Provides arbitrary routing topologies.	• Not possible for nemesis to gain full access of a system even if it is compromised.
2	Potemkin [19]	M. Vrable et al. (2005)	High-Fidelity Honeypot	• Honeyfarm, that offers high-fidelity host emulation for a large number of IP addresses with few physical servers can be implemented.	• Based on assumption that the honeypot will attract all traffic which is far away from reality.
3	Shadow Honeypots [20]	K. G. Anagnostakis et al. (2005)	Hybrid Approach i.e. combination of honeypots and anomaly detection.	• Pertinent for protection against client-side attacks. • Easy integration with additional detection modules.	• The fruitfulness of the rollback mechanism depends upon the calls and the detector's latency.
4	Double Honeypot System [21]	Y. Tang & S. Chen (2005)	-	• Can detect polymorphic worms without any human expert intervention.	• The system needs to be tested in a live environment. • Iterative algorithms need to be enhanced.
5	Hardware & Software Independent Honeypot Detection Methodology [22]	C. C. Zou & R. Cunningham (2006)	-	• Automatically detect and remove compromised honeypots.	• The technique was not peculiar for botnet detection but intended for honeypot detection.
6	HoneyBow [23]	J. Zhuge et al. (2007)	Combination of Low & High-Interaction Honeypots.	• Comprises of three malware collection tools: MwWatcher, MwFetcher & MwHunter. • All of them allows different mechanism.	• Cannot differentiate between the versions of polymorphic worm.

S. No.	Proposed Model/Method	Authors (Year)	Detection Type	Pros	Cons
7	Network of High Interaction Honeypots [24]	T. Bajtoš et al. (2018)	High-Interaction Honeypots	• Botnet can be analyzed in the infection phase.	• Detection based entirely on signatures.

The Table (2) lists various honeynet based detection techniques. These techniques are reviewed on the basis of their types i.e. whether they are low-interaction based, high-interaction based or hybrid, their pros and cons. These techniques are not so useful as they are not intended towards detection but are used for simulation purposes. Also, botherders have mechanisms to detect these honeypots.

B. Review on Intrusion Based Detection Techniques

These systems monitor the network traffic and checks for any deviation in the network. They are categorized into four categories as: signature-based detection, anomaly-based detection, DNS-based detection and mining-based detection.

- **Review on Signature-Based Detection Techniques:** These detection techniques make use of predefined behavior and knowledge of existing bot signatures in order to detect botnet [10]. Snort which is an open source Intrusion Detection/Prevention System comprises of set of rules that monitors the network traffic in order to detect intrusion. It also has the proficiency to perform real-time traffic analysis and packet logging on Internet Protocol networks along with protocol analysis [25, 26].

Table (3): Tabulated review on signature-based techniques.

S. No.	Proposed Model/Method	Authors (Year)	Dataset	Pros	Cons
1	BotHunter [27]	G. Gu et al. (2007)	Georgia Tech Campus Network & SRI Laboratory Network.	• Method can be extended with anomaly based "Entropy Detectors" for the identification of encrypted channels.	• Encrypted bots cannot be detected.
2	AutoRE [28]	Y. Xie et al. (2008)	Email dataset from MSN Hotmail.	• Tracks the sending behavior and the associated email content pattern.	• Success rate is totally dependent on the generation of signatures. • The model is not implemented on real-time data.
3	N-EDPS [29]	S. Behal et al. (2010)	SBSCET Network	Provides prevention against botnet attack along with detection.	• Unable to detect encrypted C&C channels.

The Table (3) describes some of the techniques based on signature. These techniques are reviewed on the basis of dataset used and their pros and cons. Apart from these, signature-based detection techniques are only useful in detection of known botnets. They can't detect botnets if the predefined signatures are missing, which means they cannot detect unknown botnets.

- **Review on Anomaly-Based Detection Techniques:** Anomaly based detection techniques detect botnet by analyzing network traffic for anomalies such as traffic on unusual ports, high network latency, large amount of traffic and abnormal system behavior [30]. These techniques are expensive because of their higher computation but are capable of detecting unknown botnets. They are further categorized into host-based and network-based detection. The Table (4) gives some of the techniques used for anomaly-based detection along with their accuracy rate, false positive rate (FPR), dataset used and their drawbacks.

Table (4): Tabulated review on anomaly-based detection techniques.

S. No.	Proposed Model/Method	Authors (Year)	Detection Accuracy (%)	False Positive Rate (%)	Dataset	Pros	Cons
1	Detection Based on Network Traffic Flow Characteristics [32]	G. Kirubavathi & R. Anitha (2016)	99	0.02	University of Victoria, CAIDA, CVUT University, University of Georgia, Dalhousie University & Centro University.	• Detect botnets regardless of their structures. • Detect unknown botnets.	• Inefficient to detect botnet in IOT.
2	Service-Based Profiling System [33]	W. Chen et al. (2017)	91	5	CTU-13	• Greater detection accuracy even without any prior knowledge.	• Results vary on different port numbers.
3	Machine Learning Using DNS Query Data [34]	X. D. Hoang & Q. C. Nguyen (2018)	90.80	9.30	Domain names by Alexa, Conficker botnet & DGA botnet.	• Detect DGA and FF botnets.	• High false positive rate.
4	Unsupervised Learning Model [35]	S. Nomm & H. Bahsi (2018)	90% & Above	-	Set of 115 numeric features from 9 IoT devices.	• One model can be used for training for all devices.	• Detection rates fluctuates with different IoT devices.
5	Hybrid Analysis [36]	Y. Shang et al. (2018)	96.62% (F-Score)*	-	Traffic from telecom company.	• Better than individual analysis.	• Detection can be evaded by utilizing a legitimate website as C&C communication. • Randomizing the pattern of communication also helps in evasion.

S. No.	Proposed Model/Method	Authors (Year)	Detection Accuracy (%)	False Positive Rate (%)	Dataset	Pros	Cons
6	LSTM Based Neural Network [37]	K. Sinha et al. (2019)	96.2	0.037	CTU-13	• Can detect encrypted botnets.	• Large amount of time required for feature extraction. • Detects particular type of botnet.
7	SDN Using Deep Learning [38]	S. Maeda et al. (2019)	99.2	-	CTU-13, ISOT & their own laboratory data.	• Network isolation is performed to prevent from internal infection.	• Experiment was not performed on machines that were infected by bots.

Even though anomaly-based detection techniques solved the problem of detecting unknown botnets, but they suffer the problem of detecting an IRC network that may be a botnet but has not performed any malicious activity, hence there are no anomalies. J. Binkley and S. Singh have combined TCP-based anomaly detection with IRC tokenization and IRC message statistics to create a system that can detect botnet clients [31]. Also, the technique can divulge bot servers.

- **Review on DNS-Based Detection Techniques:** These are based on DNS information generated by the botnet. While accessing the C&C server bots carry out certain DNS queries to pinpoint the particular C&C server typically hosted by a Dynamic DNS (DDNS) provider. So, by the means of DNS monitoring it is possible to detect botnet DNS traffic and botnet DNS traffic anomalies [33].

Table (5): Tabulated review on DNS-based detection techniques.

S. No.	Proposed Model/Method	Authors (Year)	Detection Accuracy (%)	False Positive Rate (%)	Dataset	Pros	Cons
1	PsyBoG [39]	J. Kwon et al. (2016)	95	0.19	DNS traces from real malware and recursive DNS server.	• Power spectral density analysis is used to detect botnet. • No prior knowledge is needed for detection.	• Modification in query patterns lead to evasion.
2	BotGRABBER [40]	O. Pomorova et al. (2016)	96	4	Campus network of Khmelnitsky National University.	• Bypass evasion techniques such as domain flux, cycling of IP mapping, fast flux & DNS-tunneling.	• High false positive rates: 4%-9%.

S. No.	Proposed Model/Method	Authors (Year)	Detection Accuracy (%)	False Positive Rate (%)	Dataset	Pros	Cons
3	DBod [41]	TS. Wang et al. (2017)	-	-	Education network of Tainan City, Taiwan.	• Detects new DGA-based botnet without any prior knowledge.	• Unable to detect bot-compromised host, if it has not been attacked earlier or never tried to connect to C&C server.
4	DNS Rule Based Approach [42]	K. Alieyan et al. (2019)	99.35	0.25	ISOT	• Botnet can be detected just by checking the abnormalities in DNS query and response behavior.	• Cannot detect P2P botnets.
5	DNS Based Profiling [43]	O. P. Dwyer et al. (2019)	99	-	Data collected from globally distributed honeypots.	• Low computational cost is required.	• Detect only Mirai-alike botnets.

DNS based detection techniques are given in Table (5). The tabular comparison consists of their accuracy, false positive rate, used dataset and their demerits. These techniques provide results with higher accuracy at the same time uses low computation cost as only DNS related information is to be processed for detection.

- **Review on Mining-Based Detection Techniques:** The aim of data mining techniques is to recognize various useful patterns to discover regularities and irregularities in large data sets [33]. An efficacious technique for botnet detection is to identify the botnet C&C traffic. But the identification of such traffic is arduous. Since the botnets deploy normal protocols for C&C communication, the traffic is similar to the benign traffic [10]. Neither the C&C traffic is enormous nor it causes high network latency.

Table (6): Tabulated review on mining-based detection techniques.

S. No.	Proposed Model/Method	Authors (Year)	Detection Accuracy (%)	False Positive Rate (%)	Dataset	Pros	Cons
1	Multiple Log-File Based Temporal Correlation technique [44]	M. M. Masud et al. (2008)	99.9	0.0	Data from SDBot & RBot infected machines.	• Not restricted to detection of IRC botnets. • Method works even in the absence of C&C payload.	• Detection was not performed on real-time traffic.

S. No.	Proposed Model/Method	Authors (Year)	Detection Accuracy (%)	False Positive Rate (%)	Dataset	Pros	Cons
2	Visual Threat Monitor [45]	A. Shahrestani et al. (2009)	-	-	Network flow between source and destination application.	• System provides greater flexibility and scalability as layers in architecture are independent to each other.	• More visualization techniques need to be implemented because a single misinterpretation may lead to false results.
3	P2P Botnet Detection using Data Mining [46]	WH. Liao & CC. Chang (2010)	98	-	PC's infected by bots.	• Detected P2P botnet just by examining the network traffic.	• Research was only conducted for LAN environments.

Therefore, anomaly-based detection techniques that relies on high network latency and network behavior anomalies, cannot be used to identify botnet C&C traffic. In such cases data mining techniques are used that extract sufficient data from network log files for analysis. Several data mining techniques that can efficiently detect botnet C&C traffic are classification, clustering, correlation, statistical analysis and aggregation [10, 33]. Mining based detection techniques are given in Table (6) with a comparison based on their accuracy, false positive rate, datasets and their drawbacks.

6. CONCLUSION

Botnets are a crucial security threat to the Internet as well as its users. They can mow down an entire network within a blink of an eye. The ever-changing nature of the Internet leverages the botherders to succeed in their ferocious intents.

The paper discusses the history and evolution of botnet over time. The life cycle of the botnet is explained along with various protocols used by botnet for its communication with the C&C servers. The architectures used by the botherders for the propagation of botnets are also explained. Then various botnet detection techniques are given. The researches have shown that DNS and mining based detection techniques along with machine learning algorithms have given promising results in detecting botnet as compared to other detection techniques.

But as the botnets keep on changing their propagation mechanism new detection techniques need to be developed and techniques that can stop botnet in the initial stages need to be determined. Also, techniques should be implemented in such a way that it becomes enigmatic for attackers and arduous to evade. In modern world scenario, IoT has led to massive growth and as a result it needs to be safeguard from such despicable attackers. The pivotal point is that the techniques should be developed at low computational cost.

REFERENCES

[1] C. A. Schiller, and J. Binkley, in *Botnets The Killer Web App*, 1st ed. Massachusetts, US: Syngress Publishing, 2007.

[2] I. Ullah, N. Khan, and H. A. Aboalsamh, "SURVEY ON BOTNET: ITS ARCHITECTURE, DETECTION, PREVENTION AND MITIGATION," in *2013 10th IEEE ICNSC, Evry, France*, 2013, doi: 10.1109/ICNSC.2013.6548817.

[3] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks", *Neural Computing and Applications*, vol. 29, issue 11, Jun. 2018, pp. 991-1004, doi: 10.1007/s00521-016-2564-5.

[4] S. S.C. Silva, R. M.P. Silva, R. C.G. Pinto, and R. M. Salles, "Botnets: A survey," in *Computer Networks*, vol. 57, issue 2, Feb. 2013, pp. 378-403, doi: 10.1016/j.comnet.2012.07.021.

[5] A. Karim, R. B. Salleh, M. Shiraz, S. A. A. Shah, I. Awan, and N. B. Anuar, "Botnet detection techniques: review, future trends, and issues," *J. Zhejiang Univ. – Sci. C*, vol. 15, issue 11, Nov. 2014, pp. 943-983, doi: 10.1631/jzus.C1300242.

[6] R. Ferguson, *The Botnet Chronicles A Journey to Infamy*, Trend Micro, Nov. 2010. Accessed on: Oct. 16, 2019. [Online]. Available: https://www.trendmicro.co.uk/media/wp/botnet-chronicles-whitepaper-en.pdf

[7] L. Abrams, *Smominru Mining Botnet In Cyber Turf War With Rival Malware*, Bleeping Computer, Sep. 18, 2019. Accessed on: Oct. 16, 2019. [Online]. Available: https://www.bleepingcomputer.com/news/security/smominru-mining-botnet-in-cyber-turf-war-with-rival-malware/

[8] *PrettyPark*, F-Secure, Accessed on: Oct. 16, 2019. [Online]. Available: https://www.f-secure.com/v-descs/prettyp.shtml#summary

[9] *Botnet*, Wikipedia The Free Encyclopedia, Accessed on: Oct. 16, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Botnet

[10] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of Botnet and Botnet Detection," in *2009 Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Glyfada, Greece*, 2009, doi: 10.1109/SECURWARE.2009.48.

[11] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," in *Proceedings of the 15th Annual Network and Distributed System Security Symposium, San Diego, California, USA*, Jan. 2008.

[12] Shodhganga, Accessed on: Oct. 16, 2019. [Online]. Available: https://shodhganga.inflibnet.ac.in/bitstream/10603/204333/11/11_chapter%203.pdf

[13] S. Anwar, J. B. M. Zain, M. F. B. Zulkipli, and Z. Inayat, "A Review Paper on Botnet and Botnet Detection Techniques in Cloud Computing," *ISCI 2014 – IEEE Symposium on Computers & Informatics, Kota Kinabalu, Sabah, Malaysia*, Sep. 2014.

[14] P. Kaur, and A. Gupta, "A Study on Botnet Detection in Cloud Network," *IJCSE*, vol. 6, no. 11, Nov.-Dec. 2017.

[15] K. Alieyan, A. ALmomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," *Neural Comput & Applic*, vol. 28, issue 7,2015, pp. 1541-1558, doi: 10.1007/s00521-015-2128-0.

[16] H. R. Zeidanloo, M. J. Z. Shooshtari, P. V. Amoli, M. Safari, and M. Zamani, "A Taxonomy of Botnet Detection Techniques," in *2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China*, July 2010, doi: 10.1109/ICCSIT.2010.5563555.

[17] Erdem Alparslan, Adem Karahoca, and Dilek Karahoca, *BotNet Detection: Enhancing Analysis by Using Data Mining Techniques*, Advances in Data Mining Knowledge Discovery and Applications, Adem Karahoca, IntechOpen, DOI: 10.5772/48804. Sep. 12 2012. Accessed on: Oct. 23, 2019. [Online]. Available: https://www.intechopen.com/books/advances-in-data-mining-knowledge-discovery-and-applications/botnet-detection-enhancing-analysis-by-using-data-mining-techniques

[18] N. Provos, "A virtual honeypot framework," *IEEE Secur. Priv. Mag.*, vol. 8, no. 3, pp. c4–c4, 2010.

[19] M. Vrable, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. C. Snoeren, G. M. Voelker, and S. Savage, "Scalability, Fidelity, and Containment in the Potemkin Virtual Honeyfarm" in *Proceedings of the Twentieth ACM Symposium on Operating Systems Priniciples-SOSP'05*, pp. 148–162, 2005, doi: 10.1145/1095810.1095825.

[20] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis, "Detecting Targeted Attacks Using Shadow Honeypots," *14th USENIX Security Symposium*, 2005.

[21] Y. Tang, and S. Chen, "Defending against Internet worms: A signature-based approach," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA*, 2005, doi: 10.1109/INFCOM.2005.1498363.

[22] C. C. Zou, and R. Cunningham, "Honeypot-aware Advanced Botnet Construction and Maintenance," *International Conference on Dependable Systems and Networks*, 2006, doi: 10.1109/DSN.2006.38.

[23] J. Zhuge, T. Holz, X. Han, C. Song, and W. Zou, "Collecting Autonomous Spreading Malware Using High-Interaction Honeypots," *International Conference on Information and Communication Security*, vol. 4861 LNCS, pp. 438–451, 2007.

[24] T. Bajtoš, P. Sokol, and T. Mézešová, "Virtual honeypots and detection of telnet botnets," in *Proceedings of the Central European Cybersecurity Conference*, 2018, doi: 10.1145/3277570.3277572.

[25] "Snort", Accessed on: Oct. 30, 2019. [Online]. Available: https://www.snort.org/#get-started

[26] "Snort (Software)", Accessed on: Oct. 30, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Snort_(software)

[27] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation," *in Proceedings 16th USENIX Security Symposium*, pp. 167-182, 2007.

[28] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: Signatures and characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 171–182, 2008, doi: 10.1145/1402946.1402979.

[29] S. Behal, A. S. Brar, and K. Kumar, "Signature-based Botnet Detection and Prevention," 2010. Accessed on: Oct. 10, 2019. [Online]. Available: https://www.researchgate.net/profile/Sunny_Behal/publication/267846973_Signature-based_Botnet_Detection_and_Prevention/links/565be66d08aefe619b2488f4.pdf

[30] J. Vania, A. Meniya, and H. B. Jethva, "A Review on Botnet and Detection Technique," in *International Journal of Computer Trends and Technology*, vol. 4, issue 1, 2013.

[31] J. R. Binkley, and S. Singh, "An Algorithm for Anomaly-based Botnet Detection," in *Proceedings USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI'06)*, pp. 43-48, 2006.

[32] G. Kirubavathi, and R. Anitha, "Botnet detection via mining of traffic flow characteristics," *Computers & Electrical Engineering*, vol. 50, pp. 91–101, 2016, doi: 10.1016/j.compeleceng.2016.01.012.

[33] W. R. Chen, X. Luo, and A. N. Zincir-Heywood, "Exploring a service-based normal behaviour profiling system for botnet detection," *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal*, 2017, doi: 10.23919/INM.2017.7987417.

[34] X. Hoang, and Q. Nguyen, "Botnet Detection Based On Machine Learning Techniques Using DNS Query Data," *Future Internet*, vol. 10, no. 5, p. 43, 2018, doi: 10.3390/fi10050043.

[35] S. Nomm, and H. Bahsi, "Unsupervised Anomaly Based Botnet Detection in IoT Networks," *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA*, 2019, doi: 10.1109/ICMLA.2018.00171.

[36] Y. Shang, S. Yang, and W. Wang, "Botnet Detection with Hybrid Analysis on Flow Based and Graph Based Features of Network Traffic," *International Conference on Cloud Computing and Security*, vol. 11064 LNCS, pp. 612-621, 2018, doi: 10.1007/978-3-030-00009-7_55.

[37] K. Sinha, A. Viswanathan, and J. Bunn, "Tracking Temporal Evolution of Network Activity for Botnet Detection," 2019. Accessed on: Oct. 10, 2019. [Online]. Available: https://arxiv.org/abs/1908.03443

[38] S. Maeda, A. Kanai, S. Tanimoto, T. Hatashima, and K. Ohkubo, "A Botnet Detection Method on SDN using Deep Learning," *2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA*, 2019, doi: 10.1109/ICCE.2019.8662080.

[39] J. Kwon, J. Lee, H. Lee, and A. Perrig, "PsyBoG: A scalable botnet detection method for large-scale DNS traffic," *Computer Networks*, vol. 97, pp. 48–73, 2016, doi: 10.1016/j.comnet.2015.12.008.

[40] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, and K. Bobrovnikova, "Anti-evasion technique for the botnets detection based on the passive DNS monitoring and active DNS probing," *International Conference on Computer Netwoks*, vol. 608 CCIS, pp. 83–95, 2016.

[41] T. S. Wang, H. T. Lin, W. T. Cheng, and C. Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis," *Computers & Security*, vol. 64, pp. 1–15, 2017, doi: 10.1016/j.cose.2016.10.001.

[42] K. Alieyan, A. Almomani, M. Anbar, M. Alauthman, R. Abdullah, and B. B. Gupta, "DNS rule-based schema to botnet detection," *Enterprise Information System*, 2019, doi: 10.1080/17517575.2019.1644673.

[43] O. P. Dwyer, A. K. Marnerides, V. Giotsas, and T. Mursch, "Profiling IoT-based Botnet Traffic using DNS," *IEEE GLOBECOM*, 2019.

[44] M. M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. W. Hamlen, "Flow-based identification of botnet traffic by mining multiple log files," *2008 First International Conference on Distributed Framework and Applications, Penang, Malaysia*, 2008, doi: 10.1109/ICDFMA.2008.4784437.

[45] A. Shahrestani, M. Feily, R. Ahmad, and S. Ramadass, "Architecture for applying data mining and visualization on network flow for botnet traffic detection," *2009 International Conference on Computer Technology and Development*, 2009, doi: 10.1109/ICCTD.2009.82.

[46] W. H. Liao, and C. C. Chang, "Peer to peer botnet detection using data mining scheme," *2010 International Conference on Internet Technology and Applications, Wuhan, China*, 2010, doi: 10.1109/ITAPP.2010.5566407.