Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.036587
Article

Monitoring Peer-to-Peer Botnets: Requirements, Challenges, and Future Works

Arkan Hammoodi Hasan Kabla, Mohammed Anbar, Selvakumar Manickam, Alwan Ahmed Abdulrahman Alwan and Shankar Karuppayah*

National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Pulau Pinang, 11800, Malaysia
*Corresponding Author: Shankar Karuppayah. Email: kshankar@usm.my
Received: 05 October 2022; Accepted: 06 January 2023

Abstract: The cyber-criminal compromises end-hosts (bots) to configure a network of bots (botnet). The cyber-criminals are also looking for an evolved architecture that makes their techniques more resilient and stealthier such as Peer-to-Peer (P2P) networks. The P2P botnets leverage the privileges of the decentralized nature of P2P networks. Consequently, the P2P botnets exploit the resilience of this architecture to be arduous against take-down procedures. Some P2P botnets are smarter to be stealthy in their Command-and-Control mechanisms (C2) and elude the standard discovery mechanisms. Therefore, the other side of this cyberwar is the monitor. The P2P botnet monitoring is an exacting mission because the monitoring must care about many aspects simultaneously. Some aspects pertain to the existing monitoring approaches, some pertain to the nature of P2P networks, and some to counter the botnets, i.e., the anti-monitoring mechanisms. All these challenges should be considered in P2P botnet monitoring. To begin with, this paper provides an anatomy of P2P botnets. Thereafter, this paper exhaustively reviews the existing monitoring approaches of P2P botnets and thoroughly discusses each to reveal its advantages and disadvantages. In addition, this paper groups the monitoring approaches into three groups: passive, active, and hybrid monitoring approaches. Furthermore, this paper also discusses the functional and non-functional requirements of advanced monitoring. In conclusion, this paper ends by epitomizing the challenges of various aspects and gives future avenues for better monitoring of P2P botnets.

Keywords: P2P networks; botnet; P2P botnet; botnet monitoring; honeypot; crawlers

1 Introduction

If we look backwards at how network topologies have evolved, we can notice that each topology comes to solve an issue in the existing topologies. In other words, the contemporary network avoided issues of previous network topologies and completed a lack or need in the present technologies. Peer-to-Peer (P2P) networks have been applied and developed in state-of-the-art technologies such

as cryptocurrencies [1]. This kind of network operates distributed, which takes it to a higher level of resilience [2]. The high resilience and openness make the P2P networks favorable for contemporary file-sharing applications where everyone can freely join and leave without permission [3]. In addition, the P2P networks enable fast and efficient lookups of key-value pairs [4]. However, these interesting properties attract cybercriminals when we think otherwise; they could also leverage the same privileges.

Cybercriminals target not only individuals but also governments, organizations, institutions, banks, companies, etc. Nowadays, all internet-connected devices are susceptible to being attacked. Many cybercrimes originated from botnets. A 'bot' is a compromised machine remotely controlled by a botmaster [5]. A network of such infected machines under the command of a botmaster is called a 'botnet' [6]. Botmasters have leveraged the resilience offered by P2P networks to construct botnets. Consequently, the risk is that the botmaster leverages the same privileges of the P2P network, such as scalability, efficiency, and resilience. Therefore, the compromised end-hosts are exploited to steal data or Distributed Denial of Service (DDoS) attacks [7,8]. Hence, there is a necessity to invest efforts in detecting and taking down P2P botnets.

Examples of P2P botnets are Nugache [9], Zeus GameOver [2], Mozi [10], Slapper [11], Storm [12], ZeroAccess [13,14], and the sophisticated FritzFrog [15], etc. These botnets have a terrible history with many victims, leading to big financial losses [16]. So far, there are two P2P botnets: P2P botnets that either uses specific P2P protocol for a special purpose or adopt the public P2P protocols [17]. When P2P botnets utilize specifically built private P2P protocol, these botnets are easier to detect or monitor. At the same time, the second type of P2P botnets adopt the existing protocols and become stealthy and harder to detect and monitor [17].

Although P2P botnets are resilient, countering this botnet is possible, starting with monitoring. Monitoring gives a vision, vision gives understanding, and understanding gives the ability to determine a vulnerability within the botnet's design or communication protocol. To reach that level of determining the vulnerability or taking procedures against botnets, monitoring must be efficient. Efficient monitoring leads to an accurate vision of the botnet structure. As a consequence, the right decisions are made.

Monitoring the P2P botnets requires enumeration information of all bots in the botnet. The most common P2P monitoring approaches are Honeypots, Sensors, and Crawlers [18]. This paper explains the P2P botnet monitoring approaches as passive, active, and hybrid monitoring approaches. In addition, this paper exhaustively covers the related works based on honeypots, sensors, crawlers, and hybrid approaches to monitor the P2P botnets. Finally, this paper evaluates the satisfaction level of each work and then gives the challenges and future works in P2P botnet monitoring.

The remainder of this paper is organized as follows: Section 2 presents background about the key concepts of this paper. Section 3 explains the functional and non-functional requirements of P2P botnet monitoring. Section 4 describes the P2P monitoring approaches in three categories: passive, active, and hybrid. Section 5 thoroughly discusses the related works. Section 6 presents the challenges in monitoring the P2P botnets. Finally, Section 7 concludes this article and provides future directions.

2 Background

This section provides background knowledge and terminology about the key concepts of this paper, which are the P2P botnets and the types of P2P botnets. Before going further with the P2P botnets, it is important to introduce the concept of the P2P network and how the botnets have evolved based on the essence of P2P networks to become one of the most challenging issues.

In general, there are three types of networks in terms of authorization, as shown in Fig. 1, and all are still utilized hitherto. First is a centralized network with a centralized server to handle the major processes of the network. Second, a decentralized network where there are many servers to avoid single-point-of-failure. Third, in P2P networks, there is no authority responsible for the major process, where all the nodes in this network play the same role, and this is the reason behind naming the nodes as peers. A P2P network is a set of distributed systems that have equal capabilities and roles. Each node or peer in this network can directly exchange information [19]. The peers pool together in P2P networks to take advantage of utilizing a large number of resources. In addition, the strength of the P2P network is that all the peers have the same characteristics, including fault tolerance, load-balancing, and self-organization [3]. More so, the scalability of this kind of network gives it priority over centralized and even decentralized networks in many applications of distributed systems such as web caches, multicast systems, and anonymous communications systems. [19,20].

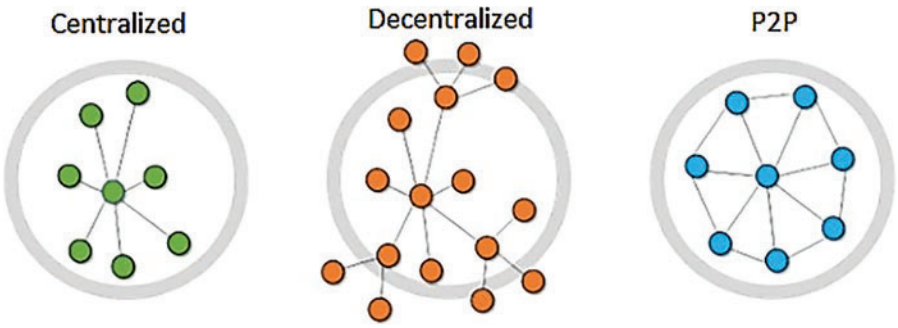

Figure 1: Types of networks

The P2P network is subdivided into two types based on node connection: structured and unstructured P2P networks. In structured P2P networks, there are restrictions on the network topologies and content placement. It is mostly used to implement algorithms to provide certain connectivity among nodes. Although structured P2P networks are more complex, these networks are more efficient [21]. Examples: Distributed Hash Table (DHT) and Hyper Cup [22]. In an unstructured P2P network, there are no restrictions on the network topologies. In addition, content placement is not related to the network topologies. This type of network performs better when it comes to dynamic environments.

2.1 Anatomy of P2P Botnet

Before going further into the reason behind developing P2P botnets, it is preferable to introduce the botnet itself through its architecture and C2 channels. Botnets substantially consist of three main components, namely: Bot (master/operator), Command-and-Control mechanism (C2), and Malware (malicious software). The botmaster remotely commands and controls the bots with the latest updates through the C2 server, where the bots are interconnected. This centralized architecture entirely relies on the C2 server to control the bots individually, which puts the botnet at risk of a single point of failure. Intuitively, illegitimates have worked hard to tackle this threat using the decentralized botnet, where bots can reach each other without C2 channels, i.e., when more than one bot represents a C2 channel. However, the C2 servers are still under threat of being crawled since each one indicates to others as a result of being effectively interconnected to each other. In other words, centralized and decentralized botnet architectures were susceptible to detection and monitoring.

As a consequence, P2P architecture in botnet comes to tackle these weaknesses in terms of enhancing the network traffic concealment and avoiding the single point of failure [14]. The P2P botnets use the P2P networks as a vector to recruit the peer nodes as C2 channels. In addition, P2P botnets utilize either standards or customized P2P protocols. Fig. 2 shows the P2P botnet (a) compared to the centralized botnet (b).

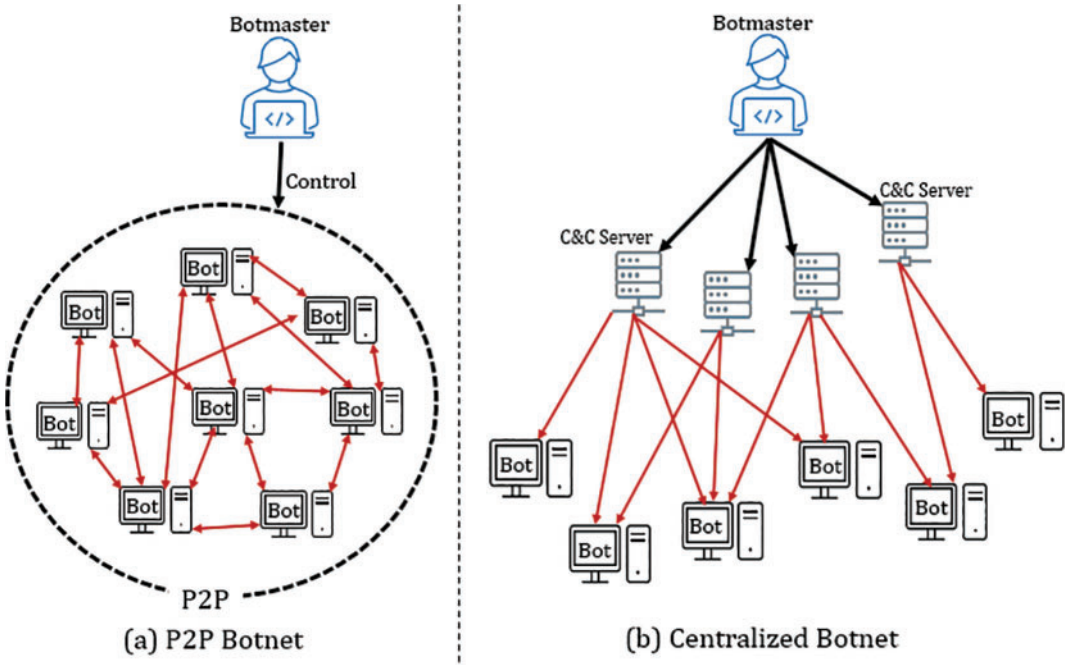

Figure 2: (a) P2P botnet, and (b) centralized botnet

The peers in P2P networks periodically announce themselves by lookup resources. In addition, each peer can know more about other peers once this peer sends a request to find a target identifier through the routing tables of other peers. Recursively, the peers in DHT protocols announce their addresses. Additionally, the peers also have automatic mechanisms to discover the addresses of others to insert them into their routing tables. Consequently, each peer knows about the active participants by its Neighbour List (NL) and can directly look for a specific peer (or target) in its NL.

2.2 Types of P2P Botnet

There are three types of P2P botnets: Structured, Unstructured, and Hybrid P2P botnets.

2.2.1 Structured P2P Botnet

The structured P2P botnets operate in a systematic P2P topology. The structured P2P botnets are susceptible to being tracked despite leveraging efficient lookups through reliable routing. Most structured P2P botnets are based on structured overlays such as Kademlia [23]. In addition, the botnets use public protocols and then mix their C2 messages with the P2P application traffic. Rotärmel classified the structured P2P botnets into two types: Pure structured P2P botnets and Parasitic P2P botnets, as explained in the following subsections [4].

a) *Pure Structured P2P Botnets*. In simple, each bot has an ID, which determines the bot's location in the network, and other bots are inserted using their IDs to route all other peer nodes. In this structured P2P botnet type, the botnets utilize a customized P2P protocol specifically designed for this purpose (for the botnet). Additionally, the overlay network is constructed with DHT, unlike the parasitic structured P2P botnets where there is no underlying network. An example of this P2P botnet is the Storm botnet [24].
b) *Parasitic Structured P2P Botnets*. This structured P2P botnet is considered one of the most dangerous P2P botnets because it can operate alongside benign participants on top of the existing DHT protocol. The scenario of this botnet is that the botmaster exploits the existing architecture to inject some superpeers (or C2 resources) with malware. Other bots coexist with this architecture to lookup the injected superpeers. The danger of these botnets is that the botnet messages are disguised as normal traffic of DHT, i.e., it is challenging to detect the parasitic structured P2P botnets [4]. Moreover, these botnets can freely join and leave the network without affecting the superpeers availability. Examples of these P2P botnets are the Hajime and IPStorm botnets [22,25].

2.2.2 Unstructured P2P Botnet

Unlike the structured P2P botnets where the ID is optional. This type of botnet is more flexible and does not require any predefined layout, i.e., it operates on any topology to select neighbouring peers and routing mechanisms. The scenario of this botnet is when a bot joins the network for the first time; a bot connects to a few bootstrap peers in order to know about the other peers via exchanging the addresses of neighbour lists. It is noteworthy that the publicly routable and stable bots represent the *superpeers* [18]. Karuppayah illustrated that bots use the Membership Maintenance mechanisms (MM) to maintain their neighbour lists and inform others that they are still connected to the network [18]. Furthermore, the unstructured P2P botnets are difficult to crawl or detect since they do not operate on specific structures that may be exploited. Some references mentioned *Superpeer Overlay Botnets*, but it is the same as the unstructured P2P botnets [23].

Nevertheless, to create the C&C architecture, select the top of the globally accessible compromised systems. Then, the compromised peers behind the Network Address Translation (NAT) present the normal peer bots, and then they connect to any *superpeers* to grab the published commands. However, these botnets are vulnerable to detection. The worst part is that the detection or removal of a *superpeer* does not significantly impact the botnet because the communication is redirectable to new *superpeers* [23].

2.2.3 Hybrid P2P Botnet

These botnets are to overcome the limitations of centralized and decentralized architectures. Recently, many discovered that P2P botnets had multi-layered hybrid P2P architecture. The nature of this structure appoints the top-layer bots as master C2 servers. Consequently, the P2P network acts as relay bots connecting top servers and bottom vassals as peer bots [23]. An example of this botnet is the Gameover Zeus botnet [2].

Fig. 3 summarizes the botnet's taxonomy.

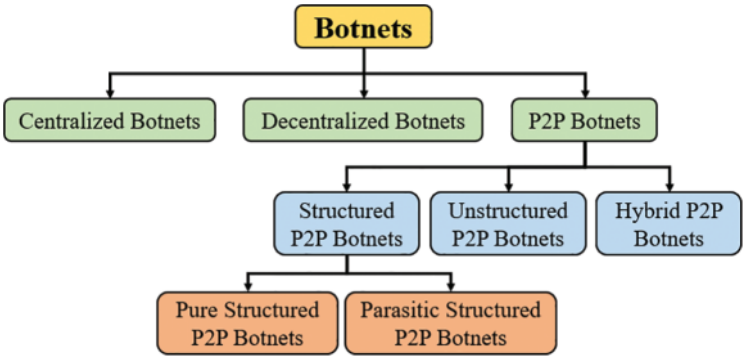

Figure 3: Botnets taxonomy

3 Requirements of P2P Botnet Monitoring

Any botnet monitoring mechanism should conform to certain functional requirements, regardless of some non-functional requirements that improve monitoring quality. This paper adopts the categorizing of Karuppayah [18] in terms of the functionality of P2P botnet monitoring requirements.

3.1 1 Functional Requirements

The functional requirements play a pivotal role in any monitoring mechanism or the main purpose of monitoring mechanisms. According to Karuppayah [18], the functional requirements are as follows.

a) *Logging*. Refers to the main purpose of monitoring which is gathering the information alongside the timestamps. The logging purpose is to provide additional botnet-specific metadata. In brief, the logged information is enough to report the infection to the network administrators or Internet Service Providers (ISPs).
b) *Protocol Compliance*. This refers to the necessity that the monitoring mechanism complies with botnet protocol since botnets respond only to the valid messages under scrutiny.
c) *Neutrality*. Refers to the fact that the monitoring mechanism should be straight to its purpose and does not contribute to the execution commands of the botmaster. If so, that contributes to avoiding artificial noise, which somehow leads to jumbling the observed behaviour of the botnet. Otherwise, the monitoring mechanism unobtrusively becomes a part of the botnet. After that, a bias increase causes an inaccurate conclusion of the botnet's nature.
d) *Genericity*. Refers to the applicability of any mechanism over different botnets, i.e., the proposed mechanism should be generally adaptable starting with the design and continuing its development.
e) *Bots Enumerating*. Enumeration capability represents the main aim of the monitoring mechanism. This capability gives an idea about the population of the botnet size.

3.2 Non-Functional Requirements

The non-functional requirements directly contribute to improving the quality of monitoring. These requirements play a vital role in improving the quality of monitoring. According to Karuppayah [18], the non-functional requirements are as follows.

a) *Scalability*. As known, the botnet is a moving and changeable target. Thus, the monitoring mechanism must be scalable to contain the increased number of bots in the botnet. Otherwise,

any increase in bots causes a deterioration in the performance of the monitoring mechanism. In addition, scalability includes system resources such as memory, bandwidth, or computational resources. For that, more scalability gives better monitoring quality.

b) *Efficiency*. It differs from the scalability. This requirement refers to a two-fold standard about probing and resource efficiency. For probing, how can the monitoring mechanism minimize the noise from the resulting monitoring flow? e.g., sometimes crawlers cause a delay, leading to bias in the resulting data [18]. At the same time, the efficiency of the source refers to how able the mechanism is to perform efficient monitoring with minimal resources.

c) *Accuracy*. Refers to a two-fold standard, how accurate the enumeration and how accurate the (inter-) connectivity is. For enumeration, accuracy represents the ability to cover all bots, whether online or offline, at a given time. Whereas for (inter-) connectivity, accuracy refers to how exact the captured botnet topology is at a given time.

d) *Minimal Noise*. (Or minimal overhead) refers to a critical observation regarding how the monitoring mechanism does not negatively affect the botnet or the monitoring approaches by altering the nature of each behaviour. Regarding any monitoring mechanism bringing extra noise, anti-noise procedures should be taken. Otherwise, bias rises, and that causes an inaccurate understanding of that botnet.

e) *Stealthiness*. Refers to the confidentiality of the monitoring, i.e., the monitoring mechanism must not be identifiable by the botnet. Intuitively, botmasters might retaliate against the monitoring mechanism because it is a direct threat. The retaliation could be in different ways, such as DDoS attacks.

Karuppayah categorized the requirements into functional and non-functional to facilitate the monitoring evaluation for other researchers in an obvious manner. Ignoring the functional requirements leads to non-satisfying results or inaccurate presentation of the botnet, e.g., skipping the Neutrality can corrupt the whole monitoring because non-neutral approaches delude the monitors. In addition, what is the monitoring's purpose without Logging or Bots Enumeration? Because unless there is accurate information about the botnet topology or the interconnectivity, there will be a lack in the representation of the botnets, and then inaccurate procedures will be taken. Another necessary standard is Protocol Compliance; the monitoring mechanism looks isolated unless it complies with the botnet protocol [18]. As a consequence, we adopt this categorizing to evaluate the satisfying level of the related works (Section 5).

Moreover, non-functional requirements play another vital role in evaluating the quality of the monitoring mechanism. For that, they also are involved in evaluating the related works. For example, ignoring the Stealthiness or Minimal noise causes a quick reveal of the monitoring approach, then the botmaster might quickly retaliate with an attack like DDoS. Besides, the botmasters might react with a countermeasure against the monitoring approaches by avoiding being crawled or monitored. To this end, some requirements might seem direct-affect, and the rest might seem indirect-affect. However, each requirement plays an important role, and ignoring this requirement causes harm or accumulated harm.

4 P2P Botnet Monitoring

Before diving deeper into P2P monitoring, we would like to distinguish between monitoring and detecting approaches. The detection mechanisms are often highly customized, i.e., they are unable to detect all P2P botnets. Relatively, detection approaches provide limited information compared to monitoring approaches. At the same time, the monitoring approaches are more accurate in terms of

tracking the bots. An example of a common detection approach is the Intrusion Detection System (IDS) [26]. IDS is software that can identify attacks by distinguishing their intrusions as abnormal traffic [7,27]. Examples of common monitoring approaches are Sensors, Honeypots, or Crawlers, which will be discussed in the following subsections.

The main idea of monitoring is to monitor the botnet by approaches that disguise as a bot to not let the botnet recognize them as an intruder to the network of bots. Then, joining the botnets invisibly allows to identify and numerate the bots. Consequently, identifying and enumerating the bots provides more valuable information that gives further understanding of the botnet. More so, the open nature of the P2P botnets enables the monitors to know communication protocol.

Monitoring the P2P botnet is a challenging mission for two reasons. First, the existing monitoring approaches still have some limitations. For that, the most common and effective mechanisms are based on hybridization, i.e., to make each approach complete the lack of another approach. The second reason is that a set of functional and non-functional requirements must be considered before installing an efficient monitoring mechanism (See Section 3).

Monitoring significantly relies on the information collection approach. The information collection approach plays a vital role in discovering the P2P botnets in terms of how that approach provides a complete vision. As much as an approach gives more information about the botnets, more understanding is gained, and more accurate reaction procedures can be taken. In general, there are three categories of monitoring approaches: Passive, Active, and Hybrid approaches. The passive approaches include Honeypots and Sensors, the active approaches include Crawlers, and the hybrid approaches may include a combination of two approaches. This categorizing is based on the traceability of an approach to explore more about the P2P botnets. For example, some monitoring approaches cover a wide range of the systems' demographics but only gain shallow information. Still, some monitoring approaches cover limited capacities, but they can discover peers behind the NAT or firewalls. The following subsections show the most common approaches to monitoring P2P botnets.

4.1 Passive Approaches

Practically, the attribute of exchanging the NLs recursively among peers can be exploited to install a set of custom nodes that monitor the exchanged requests among other participants. To be specific, these custom nodes record the metadata of all messages of other peers. Holz et al. [24] were the first who utilize this method to collect information about other peers using custom nodes. Karuppayah refers to these nodes as sensors [18].

a) Sensors. Technically, each sensor announces itself to all other reachable peers and records the received messages. Moreover, after sensors track all the observed peer IDs and addresses, sensors present the DHT population. To conclude, this approach leverages the nature of the DHT protocol by injecting the network with custom nodes for a special purpose. Another advantage of sensors is that they are able to discover and track peers that operate behind a Network Address Translation (NAT) [18]. However, there are also some limitations to using sensors. As aforementioned, sensors use DHT messages to announce or update their routing tables. Therefore, sensors are passive approaches that can learn about the demographics of the network, but they are slow in the large and highly dynamic P2P network where all peers are allowed to join and leave at a very high frequency. For that, sensors are passive approaches since they passively wait for received messages and cannot completely represent the P2P network.

b) Honeypots. Honeypots and honeynets are also considered passive monitoring approaches. Such machines or network of machines aim to be infected to monitor and discover the

malicious activities of any attack. These honeypots passively monitor the network and the intrusions regardless of knowledge of the discovered malware or the communication protocol. Its limitation of conducting only monitoring makes it classified as a passive approach that still requires other monitoring and discovery approaches.

Stallings et al. [28] considered the honeypots a component of the Intrusion Detection System (IDS). A honeypot is a system that is used to lure potential attackers away. At the same time, the honeynet is a monitored network that contains many honeypot systems. These systems are designed to: collect information about the attackers, divert the attackers from reaching the critical systems, and give an administrator enough time to respond by encouraging the attackers to stay longer on the system. Indeed, the honeypot system is designed to appear to be valuable information, but legitimate users cannot reach that information [29]. In other words, honeypots are sources that include no production value. Intuitively, any attempt to communicate with the honeypot system is suspect. In addition, any attack on the honeypot seems successful. Then, an administrator has enough time to mobilize to react to the attack. In contrast, when the honeypots initiate outbound communication, the system is already compromised [29]. In contrast, the term Padded Cell Systems refers to a protected honeypot that cannot easily be compromised [29].

Regarding the interaction level with attackers, there are three categories of honeypots: Low-level interaction, Medium-level interaction, and High-level interaction honeypots. The low-level interaction honeypots can log a vast amount of botnet data. However, this honeypot category is time-consuming and has a higher risk probability than other categories [30]. Some examples are Honeyd, HoneyRJ [31], BotMiner [32], BotGrep, and BotTrack [33]. The medium-level interaction honeypot is more challenging for the botmaster to detect [34]. Meanwhile, it requires a longer time for implementation and expertise. In addition, the medium-level interaction increases the subject of a security vulnerability [35]. Some examples are Mwcollect, Honeytrap, Nepenthes [36], HoneyBOT [31], and Kippo Honeypot Distro [37]. At the same time, high-level interaction honeypots are easy to install and require little expertise [38]. However, this category of honeypots gives limited information about a specific botnet [28]. Moreover, the high-level interaction honeypots require a nearly-complete set of features [28,39]. Some examples are Specter [31,37], Minos, ManTrap, HoneyWall, and Argos [36]. Fig. 4 depicts the honeypots based on the interaction level.

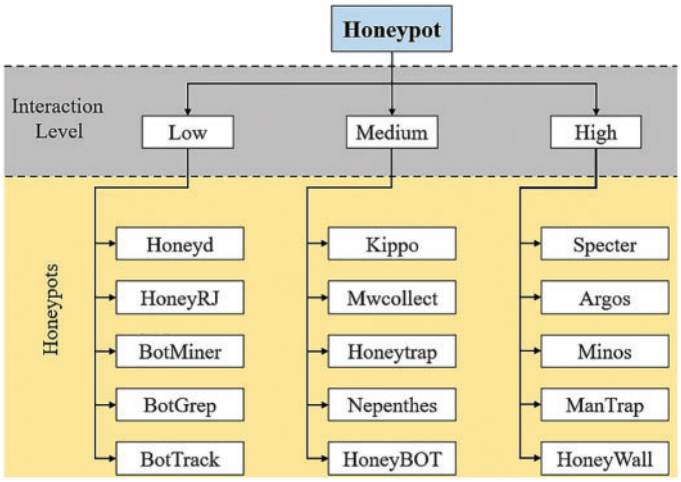

Figure 4: Honeypots are based on the interaction level

c) Software. The malware Sandbox is an environment to execute and inspect the malware in a dynamic execution using a virtualized environment such as Virtual Machine Ware (VMWare) or VirtualBox. An efficient example of malware Sandbox is Cuckoo [40]. Moreover, some applications play a passive role in monitoring the P2P botnets. For example, some applications only record information about outbound or inbound communications, namely Pen Register [29].

4.2 Active Approaches

Unlike the traditional strategies, the malware defence community started utilizing more active approaches against the P2P botnets because this botnet employs the P2P overlay network, making them more resilient and difficult to track.

a) Crawlers. Kang et al. [41] were the first who proposed utilizing crawlers to identify the bot-infected hosts. Crawlers are also used to enumerate the IP addresses of the bot-infected hosts in order to determine the local infection. Actively, crawlers contact reachable peers to query their routing tables by bootstrap peers. The bootstrap peers recursively repeat this process for each newly discovered peer [18]. Crawlers can obtain a more complete and accurate system view since they can crawl the whole network [4]. Now, there are two reasons to consider the crawlers as active approaches: i) crawlers improve visibility, unlike the honeypots where there is limited visibility, and ii) crawlers actively solve the problem of slow propagation of sensors. However, according to Karuppayah [18], crawlers also have a limitation: they cannot discover all the bots. Although the crawlers cover wider demography, they cannot contact or even discover the peers and the NAT or firewall [18] once they exist in public peers' routing tables.

4.3 Hybrid Approaches

We can observe each approach's main purpose, pros, and limitations. In the previous subsections (4.1/4.2), this paper generally categorized the most common existing approaches into Passive and Active, depending on their traceability. The main goal of hybrid approaches to monitor the P2P botnets is that each approach works to fill a gap in another. Consequently, the new hybrid approach leveraged the pros of both approaches.

A good example, Karuppayah utilized custom nodes to collect information about the botnet passively, and he refers to those nodes as sensors [18]. In addition, Karuppayah also used crawlers to explore the bot-infected hosts [17] actively. This combination was because each approach tackles one issue in another. For instance, crawlers cannot reach nodes behind NAT or firewalls, but sensors can. Meanwhile, sensors are slow and gather shallow information, whereas the crawlers are faster and gather more important information. For that, this couple works efficiently together. Otherwise, each approach is inefficient enough to monitor the P2P botnets effectively.

5 Related Works

This section groups the related works into three groups following the structure of the previous section (Section 4): Passive monitoring approaches-based related works, Active monitoring approaches-based related works, and hybridization-based related works.

5.1 Passive Monitoring Approaches

As previously categorized, Honeypots and Sensors are passive approaches to monitoring the P2P botnets; this section includes revising only the related ones based on leveraging the honeypots and sensors in their solutions.

> *a) Sensors-based Monitoring*. As aforementioned, sensors are nodes that passively participate in the botnets waiting for connection from other bots. The sensor can be upgraded to become a superpeer once it keeps responding to other bots. Thereafter, bots share information about this sensor node as a superpeer. As a consequence, it is possible to observe and analyze the botnet messages and then identify which nodes are involved in the botnets [42]. Taken together, the main purpose of sensors is to increase the visibility of the botnet as high as possible by being popular participants in that botnet [39].

Kang et al. [41] were the first to use sensors as a monitoring mechanism; the authors passively monitored the Storm P2P botnet by using nodes that seemed legitimate bots. The utilized node is a sensor that has been particularly designed to monitor the Storm P2P botnet. However, this monitoring tool can provide limited information about the whole network because of its passive nature, i.e., it does not contact other nodes but only listen [39]. In contrast, it can provide information about nodes that are even behind NAT or firewalls.

Kelihos botnet sensor is a node to track the growing population of hosting IPs. In addition, this sensor detects fast-flow domains hosted by the Kelihos botnet if those domains match with the monitored Domain Name System (DNS) authoritative traffic [43]. Moreover, this sensor also includes a filtration function to avoid false-positive results.

Rossow et al. [44] monitored the P2P botnets by peers that can contact and be contacted with other neighbouring peers during regular peer list verification cycles. Unlike the crawler, sensors can be contacted by non-routable peers, which means sensors can enumerate more peers than crawlers. However, sensors take a longer time to enumerate more bots.

Rodríguez-Gómez et al. [17] proposed a detection approach based on a number of peers that share resources in a P2P network. The proposed approach detects the parasite P2P botnets by identifying abnormal behaviours. Although promising results were obtained, the parasite P2P botnets were tentatively discovered [17].

Böck et al. [45] proposed a new autonomously detecting sensor, TrustBotMC. The authors studied and evaluated the proposed mechanism using different computational trust models in order to configure a local autonomous mechanism that avoids the P2P botnet tracking solutions.

> *b) Honeypots-based Monitoring*. The honeypots are lucrative targets that are designed to be compromised. The honeypots gather critical information about the botnets. The honeypots are the oldest monitoring approach compared to the other approaches. For that, the majority of related works are based on honeypots. In addition, honeypots are considered easier to apply since they do not require prior knowledge about the malware or the communication protocol, i.e., straightforward implementation. After the honeypot is infected with the malware, it contacts its C2.

There are many works to detect botnets using the honeypots, such as Kippo [46], Cowry [47], Dionaea [48], Telnet-Internet of Things (IoT)-Honipat [49], and IoTPOT [50]. However, this paper focuses more on utilizing honeypots to monitor the P2P botnets.

Gu et al. [32] implemented a new honeypot called BotMiner to detect the P2P botnets in real-time networks with a low false-positive rate. The proposed BotMiner clusters the communication traffic and similar malicious traffic and then applies the cross-cluster correlation to detect hosts with similar communication traffic and malicious patterns. Finally, the identified hosts represent the bots in the monitored network [32].

Nagaraja et al. [51] devised a honeypot to localize the bots using unique communication patterns arising from the overlay topologies for their C2. The devised technique is resilient to incomplete visibility to localize the bots with a low positive rate [51].

François et al. [52] proposed an approach to track the botnets using NetFlow and PageRank. The proposed approach adapted analysing of the behavioral communication patterns to infer potential activities [52].

Frederic Giroire et al. [53] proposed a method to detect the C2 of P2P botnets. The proposed method tracks the persistence of new destination atoms that are not whitelisted yet to identify the C2 destinations. This method does not require prior information about the protocol or destinations used by C2 communication. This method incorrectly identified the C2 traffic considering the stealthiness to achieve a low positive rate [53].

Rahbarinia et al. [54] proposed PeerRush, a novel system to identify unwanted P2P traffic. The proposed prototype goes beyond P2P traffic detection to accurately identify malicious applications such as P2P botnets. PeerRush showed promising results in detecting malicious P2P traffic with a misclassification rate of 0.68%.

The point is that the honeypots have less control over the infected machines. Then, this limitation led to the development of more advanced approaches such as Crawlers. These modern approaches can selectively refuse to respond or forward certain messages, giving these approaches more control over the monitoring. Moreover, these approaches also can communicate with the bots.

5.2 Active Monitoring Approaches

a) *Crawlers-based Monitoring*. As explained earlier, Crawlers are an active approach to monitoring P2P botnets. Recursively, crawlers keep requesting the active nodes for more nodes until all nodes are discovered, including bots, or the action is terminated. Crawling approaches are designed mainly by using either Breadth-First-Search (BFS) or Depth-First-Search (DFS) [39]. In addition, the information collected by crawlers assists in estimating and enumerating the botnet size, and this is because the crawlers utilize a graph traversal technique to request nodes for connectivity. Thus, this feedback allows the analyst to reconstruct the topology of the botnet and the connectivity graph.

Dittrich et al. [55] have designed a Nugache Crawler to enumerate and then estimate the Nugache botnet size without being noticed by the botmaster. This crawler utilizes the DFS method to establish the next level of nodes until there are no nodes. The authors adapted the Last-In-First-Out (LIFO) algorithm for crawling [55].

Holz et al. [24] designed the Storm Crawler to understand the botnets' connectivity further. The Storm Crawler has been designed based on the BFS method to locate the bots. This crawler also adopted the Last-In-First-Out (LIFO) algorithm [24].

KADemlia-like networks crawler (KAD) crawler has been designed by Salah et al. [56] to crawl the KAD network in a distributed manner by utilizing the design of the KAD network itself. This crawling method does not depend on online peers, unlike the Storm Crawler (previously explained).

Stutzbach et al [57] used Cruiser crawler to crawl the Gnutella file-sharing systems. The authors captured a complete and accurate snapshot of the Gnutella network in a few minutes with more than one million peers, i.e., this is a fast-crawling method.

Another crawler, Less Invasive Crawling Algorithm (LICA), has been designed to adapt to different environments using parameter calculation. LICA crawls from a node in the bootstrap list and then limits the crawl number to a parameter. Intuitively, crawling ends once all the connected nodes are discovered or the limit set is reached. Otherwise, LICA recursively keeps repeating more iterations of crawling. In short, this crawler often crawls those popular nodes and ignores those with less connectivity, but not in detail [58].

P2P Graph Search Method is another crawler that aims to reconstruct the P2P botnet graph by discovering all peers and asking them for their peer lists [44]. This crawling method includes additional action taken against the P2P botnet. One good thing about this crawler is its capability to operate in real time.

Crawlers such as P2P Crawler, Storm Crawler, Nugache Crawler, and LICA Crawler cannot handle anti-monitoring mechanisms [39]. However, these crawlers are to design and fast crawl the botnets.

5.3 Hybridization-Based Monitoring

According to [39,41,44], more than 40% of bots contact the sensors behind the NAT or firewall. Thus, this is enough evidence for the new direction of using hybrid mechanisms in one monitoring as performed by [18]. For instance, crawlers and sensors complement each other for better monitoring results.

P2P Zeus Crawler is based on BFS to crawl the P2P Zeus botnet. This crawler starts crawling from the seed nodes and keeps repeating until all the nodes are crawled [44]. This crawler mainly focuses on capturing the intrinsic properties of P2P botnets. The experiment showed promising results in estimating the population size of the P2P botnet. In addition, the authors evaluated the disruption resilience of the P2P botnet.

Herwig et al. [22] used both passive and active measurements to analyze the operation of the Hajime P2P botnet. Active measurement of Hajime P2P botnet was target scanning the botnet infrastructure. At the same time, the passive measurement is to collect the root DNS backscatter traffic. Finally, the authors provided a representation of the Hajime botnet's behavior and what kind of devices they are more vulnerable to being compromised. In addition, they provided statistics on what countries are more or less susceptible to Hajime botnet [22]

Karuppayah [18] performed a monograph about P2P monitoring approaches besides highlighting the advantages and disadvantages of each. In addition, Karuppayah has formulated the functional and non-functional requirements for the P2P monitoring, which this paper utilizes as criteria to rate the satisfaction of each work under three levels: does not satisfy, partially satisfy, and satisfy. Therefore, Karuppayah designed a dual-perspective advanced monitoring system to monitor the P2P botnets. The proposed monitoring system exploited passive and active approaches to configure a robust and generic monitoring system. Sensors were passively utilized, and crawlers were actively used for monitoring purposes [18]. As a result, each approach completes the other one. Therefore, the performance showed

superiority over the existing P2P botnet monitoring mechanisms in considering a wide range of critical requirements for better monitoring.

Another hybrid approach is called *Trap-and-Trace* (Let us call it TAT); this approach combines the function of honeypots with the capability to track the botnet back through the network [59]. TAT is an attractant technology that is still in use. This approach combines techniques to detect intrusions and trace them back to their sources. The trap consists of a *Padded Cell System* or *Honeynets* to attract the intruders; once the intruders are trapped, the approach notifies the administrators of the intrusion presence [29]. One professional and popular example of TAT is Symantec ManHunt.

Tables 1–3 summarize the related works based on passive monitoring, active monitoring, and hybrid monitoring approaches, respectively. Besides, the findings and limitations of each work, and lastly, the satisfaction status of each work based on achieving the requirements of P2P monitoring that have earlier been discussed in Section 3.

Table 1: Summary of related works based on passive monitoring

Article	Approach	Findings	Limitations	Satisfaction
[41]	Sensors	-Monitoring the Strom P2P botnet. -Providing information about non-routable bots, i.e., behind NAT or firewall.	This approach could not be used to monitor other P2P botnets, i.e., it is about genericity.	Does not satisfy
[43]	Sensors	-Monitoring the Kelihos fast-flux P2P botnet in real time. -Tracking the growth of the infected bot's population using passive DNS.	This approach could not be used to monitor other profiles of fast-flux, nor even detect other malicious domains such as ransomware or trojan-dropping fields, i.e., it is about genericity. This approach also did not consider logging requirements.	Partially satisfy
[44]	Sensors	-Proposing a new graph-based model to capture the properties that refer to fundamental vulnerabilities of P2P botnets. -Genericity has been considered in this model since the proposed approach was applicable over 11 P2P botnets.	-Some functional requirements were not considered, such as the enumeration of bots or logging. -Some non-functional requirements, such as the accuracy of enumeration and connectivity, were not considered. In addition, resource efficiency also was not considered in this model. -The proposed model was evaluated on only four P2P botnets.	Partially satisfy

(Continued)

Table 1: Continued

Article	Approach	Findings	Limitations	Satisfaction
[17]	Sensors	-Determining the events of parasite botnets in the P2P networks. -Claiming the P2P resources that are occasionally shared differently than those corresponding to the botnet's behavior.	-In those used P2P resources, some functional requirements, such as protocol compliance, enumeration of bots, and genericity, were not considered. -In addition, some non-functional requirements were also not considered, such as scalability, reducing the overload, or even stealthiness, since it is known how difficult it is to monitor and detect the parasite botnets.	Partially satisfy
[45]	Sensors	-Presenting novel autonomous detecting sensors in P2P botnets that play as botmasters to anticipate their behavior. -Using TrustBotMC can reduce the benefits of sensor monitoring. -The experimental results showed satisfactory results in reducing the gathered intelligence by 53% compared to techniques that existed at that date.	-Such a mechanism is only applicable to some botnets, as the authors mentioned, i.e., it lacks genericity. -Adding more sensors to obtain more information causes extra overload. -Resource efficiency has yet to be considered in the proposed solution.	Partially satisfy
[32]	Honeypot	-Detecting P2P botnets with a low positive rate in real-time networks using an anomaly-based system.	-Evading the used C-plane monitoring and clustering. -Evading the used A-plane monitoring and clustering, i.e., the stealthiness requirement has not been considered. -Evading the used cross-plane analysis, i.e., it could be avoided once they use only one day, not several days. As a consequence, this might cause a longer time to deal with an immediate event once it is evaluated and applied on a real-time network.	Partially satisfy
[51]	Honeypot	-The authors advised BotGrep to Localize the botnet members.	-The achieved accuracy was not satisfactory, although there was a low false-positive rate. -BotGrep requires huge amounts of information till it can observe how parts of the communication graph change occasionally.	Does not satisfy

(Continued)

Table 1: Continued

Article	Approach	Findings	Limitations	Satisfaction
[52]	Honeypot	-Proposing a novel approach to track the P2P botnets using NetFlow and PageRank.	-Some functional requirements were not considered, such as the enumeration of bots or the logging. -Some non-functional requirements, such as accuracy or resource efficiency, were also not considered.	Does not satisfy
[53]	Honeypot	-Exploiting temporal persistence to detect the bots. -The proposed method incurs low overload. -The proposed method considers a very stealth botnet.	-The experience had limited scalability. -Some functional requirements were not considered, such as protocol compliance, enumeration of bots, or even logging.	Partially satisfy
[54]	Honeypot	-Proposing a novel system to identify unwanted P2P traffic, such as P2P botnets.	-Some functional requirements were not considered in this prototype, such as protocol compliance, logging, or even the enumeration of bots. -Some non-functional requirements, such as scalability, were also not considered since they were evaluated using existing P2P traffic datasets.	Does not satisfy

Table 2: Summary of related works based on active monitoring

Article	Approach	Findings	Limitations	Satisfaction
[55]	Crawler	-Designing a Nugache crawler to enumerate and estimate the size of the Nugache P2P botnet. -The authors considered functional requirements such as the enumeration of bots. In addition, stealthiness also was considered in this crawler.	-Genericity has not been considered since this crawler was evaluated to crawl the Nugache P2P botnet. -The outcomes gave an adequate estimate of the P2P botnet but needed an accurate count.	Partially satisfy
[24]	Crawler	-Designing a crawler to analyze and mitigate the P2P botnets. -Proposing two different ways to disrupt the communication between the botmaster and compromised machines to mitigate the botnet.	-The proposed methodology was evaluated only on Storm botnet, which does not reflect this solution's genericity. -Some functional requirements were not considered, such as enumeration of the bot and logging. -Some non-functional requirements such as accuracy, scalability, and stealthiness were also not considered in this crawling.	Partially satisfy

(Continued)

Table 2: Continued

Article	Approach	Findings	Limitations	Satisfaction
[56]	Crawler	-Designing a KAD crawler to crawl the KAD network in a distributed manner in order to capture a snapshot of the interconnectivity graph. -The proposed crawler is accurate, fast, and generic for Kademlia-like networks.	-Some functional requirements were not considered, such as Neutrality and Logging. -Some non-functional requirements, such as resource efficiency, stealthiness, and scalability, were also not considered.	Partially satisfy
[57]	Crawler	-Proposing a P2P crawler to configure a complete snapshot of the Gnutella file-sharing system.	-This crawler improved its efficiency in only Gnutella networks, i.e., genericity was not considered. -Functional requirements such as enumeration of bots or logging needed to be clearly discussed or considered in this crawler. -Some non-functional requirements, such as stealthiness or scalability, were also not considered.	Does not satisfy
[58]	Crawler	-Using the Less Invasive Crawling Algorithm (LICA) to crawl the unstructured P2P botnets through only the local information. -The results were better compared to Depth-first and Breadth-first search.	-This crawler still needs to include some features to handle the anti-monitoring mechanisms. -This crawler ignored some features that consider the churn and diurnal patterns. -This crawler does not provide a complete botnet enumeration. -This crawler is effective with the assumption that all the bots are online.	Partially satisfy

Table 3: Summary of related works based on hybrid monitoring

Article	Approach	Findings	Limitations	Satisfaction
[44]	Hybrid	-Using P2P Graph Search Method to construct the P2P botnet. -This crawler includes some additional procedures taken against the P2P botnets. -Genericity was considered in this crawling method.	-This crawler cannot give an accurate enumeration of bots. -Some critical requirements, such as logging, stealthiness, and resource efficiency, were not considered.	Partially satisfy
[22]	Hybrid	-Measurement and analysis of Hajime P2P botnet. -Disambiguate IP addresses from bots.	-A functional requirement, Genericity, was not considered. This method is applied only on the Hajime P2P botnet. -Enumeration of bots was not considered as well.	Partially satisfied

(Continued)

Table 3: Continued

Article	Approach	Findings	Limitations	Satisfaction
[18]	Hybrid	-Using sensors and crawlers simultaneously to monitor the P2P botnets. -Formulating the functional and non-functional to advanced monitoring of P2P botnets. -The performance is better than other related works considering critical functional requirements such as genericity, protocol compliance, enumeration of bots, neutrality, and logging.	-More features are needed to handle the noise from an unknown third party. -The long presence of sensors may skew the churn measurements, whereas most bots usually have shorter sessions [18].	Satisfy
[59] Chapter 7-[29]	Hybrid	-Trap-and Trace combines techniques to detect intrusions and trace them back to their sources	-As aforementioned, crawlers cannot track bots behind the NAT or firewalls. -This crawler ignored some features that consider the churn and diurnal patterns.	Partially satisfied

6 Challenges in P2P Botnet Monitoring

As defenders keep improving the countermeasures against security penetrations, the invaders, on the other hand, are also working on improving their penetrating mechanisms. Therefore, it is continuous strife. However, the competition is harder on the defenders than the invaders because the invaders often lead the competition direction, i.e., defenders are required to counter against what invaders attack. In addition, defenders are also required to develop security systems. To conclude, defenders have a much bigger responsibility that cannot be delayed. Monitoring is the best way to combine both directions for defenders, whether countering against invaders or attackers and improving the general security against P2P botnets.

There are still many challenges in monitoring P2P botnets, although some interesting and developable related works exist. This paper classifies the challenges into three classes: challenges about monitoring approaches, challenges about the nature of P2P botnets, and challenges of directly countering the botnets.

For the first class, these challenges are about considering and improving the functional and non-functional requirements of monitoring the P2P botnets. As aforementioned, the functional requirements are genericity, protocol compliance, enumeration of bots, neutrality, and logging. More lacks appear once we ignore more of these requirements. And for advanced monitoring, non-functional requirements should strictly be considered, including scalability, stealthiness, efficiency (probing, resource efficiency), accuracy (enumeration of bots, interconnectivity), and minimal noise. More clearly, monitoring approaches should comply with the existing protocol and keep neutral. Furthermore, logging and managing the bots' information is critical since it represents the main purpose of monitoring. Another challenge is the genericity of the monitoring approach, i.e., more generic effective monitoring approaches are still needed, not only adequate for one kind of network or one kind of botnet. Nevertheless, because the P2P bots can freely join and leave, the monitoring mission is more difficult, especially since we noticed that most existing monitoring approaches are experimentally practical but for a certain period. Accuracy also is a significant player in this game

because what is the purpose of inaccurate monitoring? And it is very challenging to enumerate the bots accurately and reveal the bot's interconnectivity because, as aforementioned that bots can freely join and leave the botnet.

For the second class, these challenges pertain to how botnets exploit the architecture of the P2P networks. In other words, we should consider and work on vulnerabilities that already exist in P2P architecture, such as churn. In the P2P botnets, the bots interconnect via an overlay that includes neighbourhood relationships between each bot with another subset of bots. This overlay is maintained using a Membership Maintenance mechanism. Besides, the P2P botnets also experience node churn like the traditional P2P networks. Churn term refers to the ability of peers to freely and frequently join and leave the P2P networks. Thus, Membership Maintenance works on withstanding churn by ensuring that the participating bots remain connected to the overlay and removing the offline ones from the NL of the bot to replace them with responsive ones. In monitoring, this is challenging when bots frequently join and leave the botnet. Consequently, monitoring should be done for a long time to have a complete insight into the botnet. Thereafter, long-time monitoring costs resources and causes noise. Thus, churn must be carefully considered in each monitoring approach.

For the third class, these challenges pertain to the anti-monitoring mechanisms and how to handle the anti-monitoring mechanisms. For example, RatBot [60] proposed a theoretical anti-enumeration P2P botnet. The proposed technique makes the complex crawling process. P2P Zeus implemented another anti-crawling technique that blacklists any node that frequently requests NLs [61,62]. There are more works related to anti-monitoring, such as [42]. For that, future monitoring needs to include features that can tolerate anti-monitoring mechanisms, whether anti-crawling, anti-enumeration of bots, etc.

Table 4: The advantages, disadvantages, and research gaps of P2P botnet monitoring approaches

Approach	Advantages	Disadvantages	Research gap
Honeypot-Honeynet	-High efficiency of collective data [63] -High flexibility -Applicable to various applications and systems [64]	-Difficult to set up and build [65] -Less control over the infected machines [18] -Slow information analysis [65]	-Botnet is highly efficient in collecting data and easy to build and manage detection. However, information analysis is slow. -Data collection and alerting against botnet attacks are quick. However, building a system for the first time takes much time.
Crawlers	-Identifying all the infected bots besides their interconnectivity.	-Crawlers still suffer from the anti-monitoring mechanisms. -Crawlers often need to enumerate all bots in the botnets [18]. -Crawlers cannot track peers that are behind the NAT or firewalls.	-Crawlers still need to include features that tolerate the anti-monitoring mechanisms. -Crawlers might also need churn consideration features, like the crawler mentioned in [44].

(Continued)

Table 4: Continued

Approach	Advantages	Disadvantages	Research gap
Sensors	-Non-routable peers can contact sensors. -In contrast to crawlers, sensors enumerate both *superpeers* and non-*superpeers*.	-Slow in gathering information [18]. -Sensors cannot gather fine-grained data that are enough to represent the interconnectivity of the botnets [39].	-Sensors still need some improvement against anti-monitoring mechanisms.

7 Conclusion and Future Works

7.1 Conclusion

The majority of effective detection systems against P2P botnets are based on monitoring the network traffic to identify the possibility of C2 existence between the bots and botmasters [17]. This paper covered the monitoring mechanisms from earlier than a decade. In addition, this paper categorized the monitoring mechanisms into three categories: Passive monitoring, Active monitoring, and Hybridization-based monitoring approaches. Each approach has its advantages and disadvantages. For that, Hybridization came out to leverage two approaches simultaneously. Furthermore, this paper also summarized the findings and limitations of each related work. Therefore, this paper evaluated each work in terms of satisfaction status by using functional and non-functional requirements, as explained in Section 3. To conclude this with a take-home message, Table 4 summarizes the advantages, disadvantages, and research gaps for all monitoring approaches of P2P botnets.

7.2 Recommendations and Future Works

This paper exhaustively covers the P2P botnet monitoring techniques, besides the advantages and limitations of each technique. Therefore, this paper proposes future avenues that fill the gaps of the existing challenges and unsolved issues. In general, the upcoming solutions must consider the functional and non-functional requirements (Section 3), which facilitates managing the new proposed solutions and their functionalities. Intuitively, achieving the functional and non-functional requirements assists in avoiding the limitations of the related works, as discussed in Table 1. For example, most related works proposed solutions on a specific botnet type regardless of the genericity, i.e., even efficient solutions have been created to monitor and detect only one type. For that reason, we encourage the upcoming researchers and workers to experimentally test and evaluate their solutions on many types of botnets in order to achieve genericity in the next proposed solutions. Furthermore, the upcoming researchers and workers must remember that adding more resources (e.g., sensors) to guarantee solving the problem, but indeed it increases the overload. Therefore, a hybridizing approach utilizing limited resources can achieve better results than repeatedly using the same approach.

Moreover, we also encourage to utilize the hybridization to earn the advantages of two or more different approaches, which can cunningly fill the gaps of each approach when used separately. For example, crawlers cannot track bots behind the NAT, but sensors can. Meanwhile, sensors cannot cover wider demography, but crawlers can. Therefore, hybridizing these approaches can get more advantages simultaneously and fill the gaps of each one.

In the future, we recommend that researchers/workers consider the anti-monitoring issues where most existing works ignore the anti-monitoring mechanisms. Last but not least, researchers/workers who monitor and detect the P2P botnets need to spend more effort identifying the offline bots, not only

the online ones, and tracking the botnet growth. Finally, this topic will always bring new challenges when the dark side of this cyberwar is also working on improving its abilities. Still, the point is that the researchers should take care of the existing shortcomings and work on filling the gaps.

Funding Statement: This work was supported by the Ministry of Higher Education Malaysia's Fundamental Research Grant Scheme under Grant FRGS/1/2021/ICT07/USM/03/1.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, 2008. https://doi.org/10.1162/ARTL_a_00247

[2] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of gameover zeus," in *Proc. 2013 8th Int. Conf. Malicious Unwanted Softw. "the Am. MALWARE 2013*, Fajardo, PR, USA, pp. 116–123, 2013. https://doi.org/10.1109/MALWARE.2013.6703693

[3] S. Masood, M. Alyas Shahid, M. Sharif and M. Yasmin, "Comparative analysis of peer to peer networks," *Int. J. Advanced Networking and Applications*, vol. 9, no. 4, pp. 3477–3491, 2018.

[4] D. Rotärmel, "Detecting parasitic botnets in DHT-based peer-to-peer networks," B.Sc. Thesis, Technische Universitat Darmstadt, Germany, 2021.

[5] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz *et al.,* "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Applied Sciences*, vol. 9, no. 11, pp. 2375, 2019.

[6] P. Narang, S. Ray, C. Hota and V. Venkatakrishnan, "PeerShark: Detecting peer-to-peer botnets by tracking conversations," *IEEE Security and Privacy Workshops*, vol. 2014-January, pp. 108–115, 2014.

[7] A. H. Hasan, M. Anbar and T. A. Alamiedy, "Deep learning approach for detecting router advertisement flooding-based DDoS attacks," *Journal of Ambient Intelligence and Humanized Computing*, 2022. https://doi.org/10.1007/s12652-022-04437-0

[8] G. Sagirlar, B. Carminati and E. Ferrari, "Autobotcatcher: Blockchain-based P2P botnet detection for the internet of things," in *Proc.-4th IEEE Int. Conf. on Collaboration and Internet Computing, CIC 2018*, Philadelphia, PA, USA, pp. 1–8, 2018. https://doi.org/10.1109/CIC.2018.00-46

[9] S. Stover, D. Dittrich, J. Hernandez and S. Dietrich, "Analysis of the storm and nugache trojans: P2P is here," *Login Usenix Mag.*, vol. 32, no. 6, pp. 18–27, 2007.

[10] T. Tu, J. Qin, H. Zhang, M. Chen, T. Xu *et al.,* "A comprehensive study of mozi botnet," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 6877–6908, 2022.

[11] I. Arce and E. Levy, "An analysis of the slapper worm," *IEEE Security and Privacy*, vol. 1, no. 1, pp. 82–87, 2003.

[12] T. F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart," in *IEEE 30th Int. Conf. on Distributed Computing Systems*, Genoa, Italy, pp. 241–252, 2010. https://doi.org/10.1109/ICDCS.2010.76

[13] T. Seals, "Unique P2P architecture gives DDG botnet 'unstoppable' status | threatpost," *Threat Post*, 2020. https://threatpost.com/p2p-ddg-botnet-unstoppable/154650/ (accessed Nov. 08, 2022).

[14] A. H. H. Kabla, A. H. Thamrin, M. Anbar, S. Manickam and S. Karuppayah, "PeerAmbush: Multi-layer perceptron to detect peer-to-peer botnet," *Symmetry*, vol. 14, no. 12, pp. 2483, 2022.

[15] J. Aurand, "FritzFrog p2p botnet attacking healthcare, education and government sectors^{OBJ}-binary defense," *Binary Defense*, 2022. https://www.binarydefense.com/threat_watch/fritzfrog-p2p-botnet-attacking-healthcare-education-and-government-sectors/ (accessed Nov. 08, 2022).

[16] Priyanka and M. Dave, "PeerFox: Detecting parasite P2P botnets in their waiting stage," in *Int. Conf. on Signal Processing, Computing and Control (ISPCC2015)*, Waknaghat, India, pp. 350–355, 2016. https://doi.org/10.1109/ISPCC.2015.7375054
[17] R. A. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro, M. Steiner and D. Balzarotti, "Resource monitoring for the detection of parasite P2P botnets," *Computer Networks*, vol. 70, pp. 302–311, 2014.
[18] S. Karuppayah, "Springer briefs on advanced monitoring in P2P botnets a dual perspective," in *Springer Briefs on Cyber Security Systems and Networks*, Springer Nature, Singapore, 2018. [Online]. Available: https://link.springer.com/book/10.1007/978-981-10-9050-9
[19] B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer networks," in *Proc. 22nd Int. Conf. on Distributed Computing Systems*, Vienna, Austria, pp. 10, 2002.
[20] G. Fox, "Peer-to-peer networks," *Computing in Science & Engineering*, vol. 3, no. 3, pp. 75–77, 2001.
[21] S. Haridi and S. El-Ansary, "An overview of structured P2P overlay networks," in *Handbook on Theoretical Algorithmic Aspects of Sensor, Ad Hoc Wireless, Peer-to-Peer Networks*, Swedish Institute of Computer Science (SICS), Sweden, no. January 2014, 2005. https://doi.org/10.1201/9780203323687.ch39
[22] S. Herwig, K. Harvey, G. Hughey, R. Roberts and D. Levin, "Measurement and analysis of hajime, a peer-to-peer IoT botnet," in *Proc. 2019 Network and Distributed System Security Symp.*, San Diego, CA, USA, 2019. https://doi.org/10.14722/ndss.2019.23488
[23] R. S. Rawat, E. S. Pilli and R. C. Joshi, "Survey of peer-to-peer botnets and detection frameworks," *International Journal Network Security*, vol. 20, no. 3, pp. 547–557, 2018.
[24] T. Holz, M. Steiner, F. Dahl, E. Biersack and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm," in *LEET 2008-1st USENIX Work. Large-Scale Exploit. Emergent Threat*, Botnets, Spyware, Worms, More, San Francisco California, pp. 1–28, 2008.
[25] A. T. Research, "The InterPlanetary storm: New malware in wild using InterPlanetary file system's (IPFS) p2p network," 2019. https://www.anomali.com/blog/the-interplanetary-storm-new-malware-in-wild-using-interplanetary-file-systems-ipfs-p2p-network (accessed May 11, 2022).
[26] A. H. H. Kabla, M. Anbar, S. Manickam and S. Karupayah, "Eth-PSD: A machine learning-based phishing scam detection approach in ethereum," *IEEE Access*, vol. 10, pp. 118043–118057, 2022.
[27] A. H. H. Kabla, M. Anbar, S. Manickam, T. A. Alamiedy, pp. B. Cruspe *et al.,* "Applicability of intrusion detection system on ethereum attacks: A comprehensive review," *IEEE Access*, vol. 10, no. June, pp. 71632–71655, 2022.
[28] W. Stallings, M. Bauer and E. M. Hirsch, *Computer Security: Principles and Practive*, 2nd edition, New Jersey, USA: Pearson Education, 2015.
[29] M. E. Whitman and H. J. Mattord, *Principles of Information Security*, 6th edition, Boston, USA*:* Cengage Learning, 2018.
[30] Z. Wang, G. Li, Y. Chi, J. Zhang, Q. Liu *et al.,* "Honeynet construction based on intrusion detection," in *Proc. of the 3rd Int. Conf. on Computer Science and Application Engineering*, Sanya, China, Article 80, pp. 1–5, 2019. https://doi.org/10.1145/3331453.3360983
[31] E. Peter and T. Schiller, "A practical guide to honeypots," Washingt. Univerity, pp. 1–19, 2011. [Online]. Available: http://www.cs.wustl.edu/~jain/cse571-09/ftp/honey/
[32] G. Gu, R. Perdisci, J. Zhang and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. of the 17th USENIX Security Symp.*, San Jose, CA, USA, pp. 139–154, 2008.
[33] F. Haltas, E. Uzun, N. Siseci, A. Posul and B. Emre, "An automated bot detection system through honeypots for large-scale," in *Int. Conf. on Cyber Conflict, CYCON*, vol. 2014, no. July, Tallinn, Estonia, pp. 255–270, 2014.
[34] A. Belqruch and A. Maach, "SCADA security using SSH honeypot," in *Proc. of the 2nd Int. Conf. on Networking, Information Systems & Security*, vol. Part F1481, Article 2, Rabat, Morocco, pp. 1–5, 2019. https://doi.org/10.1145/3320326.3320328

[35] A. Vetterl, R. Clayton and I. Walden, "Counting outdated honeypots: Legal and useful," in *IEEE Symp. on Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, pp. 224–229, 2019. https://doi.org/10.1109/SPW.2019.00049
[36] D. Akkaya and F. Thalgott, *Honeypots in Network Security How to Monitor and Keep Track of the Newest Cyber Attacks by Trapping Hackers*, 1st edition, CA, USA: LAP LAMBERT Academic Publishing, 2012.
[37] S. Dowling, M. Schukat and H. Melvin, "A ZigBee honeypot to assess IoT cyberattack behaviour," in *28th Irish Signals Systems Conf. ISSC*, Killarney, Ireland, pp. 1–6, 2017. https://doi.org/10.1109/ISSC.2017.7983603
[38] P. Sokol, J. Míšek and M. Husák, "Honeypots and honeynets: Issues of privacy," *Eurasip Journal on Information Security*, vol. 2017, no. 1, pp. 1–9, 2017.
[39] S. Manickam, "Botnet monitoring mechanisms on peer-to-peer (P2P) botnet," *SSRN Electronic Journal*, 2020. https://doi.org/10.2139/ssrn.3713662
[40] Cuckoo, "Cuckoo sandbox-automated malware analysis," 2022. https://cuckoosandbox.org/ (accessed May 12, 2022).
[41] B. B. Kang, E. Chan-Tin, C. P. Lee, J. Tyra, H. J. Kang *et al.,* "Towards complete node enumeration in a peer-to-peer botnet," in *Proc. of the 2009 ACM Symp. on Information, Computer and Communications Security, ASIACCS 2009*, Sydney, Australia, pp. 23–34, 2009. https://doi.org/10.1145/1533057.1533064
[42] S. Karuppayah, L. Böck, T. Grube, S. Manickam, M. Mühlhäuser *et al.,* "SensorBuster: On identifying sensor nodes in P2P botnets," in *Proc. of the 12th Int. Conf. on Availability, Reliability and Security*, New York, USA, Article 34, pp. 1–6, 2017. https://doi.org/10.1145/3098954.3098991
[43] D. Mahjoub, "Monitoring a fast flux botnet using recursive and passive DNS: A case study," in *eCrime Researchers Summit*, San Francisco, CA, USA, eCrime, pp. 1–9, 2013. https://doi.org/10.1109/eCRS.2013.6805783
[44] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann *et al.,* "SoK: P2PWNED-modeling and evaluating the resilience of peer-to-peer botnets," in *IEEE Symp. on Security and Privacy*, Berkeley, CA, USA, pp. 97–111, 2013. https://doi.org/10.1109/SP.2013.17
[45] L. Böck, E. Vasilomanolakis, J. H. Wolf and M. Mühlhäuser, "Autonomously detecting sensors in fully distributed botnets," *Computers and Security*, vol. 83, pp. 1–13, 2019.
[46] A. Pauna, I. Bica, F. Pop and A. Castiglione, "On the rewards of self-adaptive IoT honeypots," *Annals of Telecommunications*, vol. 74, no. 7–8, pp. 501–515, 2019.
[47] R. K. Shrivastava, B. Bashir and C. Hota, "Attack detection and forensics using honeypot in IoT environment," In: G. Fahrnberger, S. Gopinathan and L. Parida (Eds.), *Distributed Computing and Internet Technology. ICDCIT 2019. Lecture Notes in Computer Science*, vol. 11319, pp. 402–409, Cham: Springer. 2019. https://doi.org/10.1007/978-3-030-05366-6_33
[48] A. Amjad, A. Griffiths and M. Patwary, "QoI-aware unified framework for node classification and self-reconfiguration within heterogeneous visual sensor networks," *IEEE Access*, vol. 4, pp. 9027–9042, 2016.
[49] M. Wang, "Understanding security flaws of IoT protocols through honeypot technologies," M.Sc. Thesis, Delft University ofTechnology, Netherlands, 2017.
[50] T. Luo, Z. Xu, X. Jin, Y. Jia and X. Ouyang, "IoTCandyJar: Towards an intelligent-interaction honeypot for IoT devices," Black Hat 2017, pp. 1–11, 2017. [Online]. Available: https://paper.seebug.org/papers/SecurityConf/Blackhat/2017_us/us-17-Luo-Iotcandyjar-Towards-An-Intelligent-Interaction-Honeypot-For-IoT-Devices-wp.pdf
[51] S. Nagaraja, P. Mittal, C. Y. Hong, M. Caesar and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in *19th USENIX Security Symp. (USENIX Security 10)*, Washington, DC, USA, pp. 95–110, 2010.
[52] J. François, S. Wang, R. State and T. Engel, "Bottrack: Tracking botnets using netflow and pageRank," in *Lecture Notes in Computer Science, (Book Series LNCCN)*, Springer Berlin Heidelberg, Germany, vol. 6640, no. PART 1, pp. 1–14, 2011. https://doi.org/10.1007/978-3-642-20757-0_1

[53] F. Giroire, J. Chandrashekar, N. Taft, E. Schooler and D. Papagiannaki, "Exploiting Temporal Persistence to Detect Covert Botnet Channels," in *Channels*, Springer, Berlin, Heidelberg, pp. 326–345, 2009. [Online]. Available: http://www.springerlink.com/index/D537672Q64130684.pdf

[54] B. Rahbarinia, R. Perdisci, A. Lanzi and K. Li, "PeerRush: Mining for unwanted P2P traffic," in *Lecture Notes in Computer Science (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, ELSEVIER, England, vol. 7967 LNCS, pp. 62–82, 2013. https://doi.org/10.1007/978-3-642-39235-1_4

[55] D. Dittrich and S. Dietrich, "Discovery techniques for P2P botnets," *Stevens CS Technical Report 2008-4*, vol. 4, no. April, pp. 1–16, 2009. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.163.4163&rep=rep1&type=pdf

[56] H. Salah and T. Strufe, "Capturing connectivity graphs of a large-scale P2P overlay network," in *Proc. Int. Conf. on Distributed Computing Systems Workshop*, Philadelphia, PA, USA, pp. 172–177, 2013. https://doi.org/10.1109/ICDCSW.2013.35

[57] D. Stutzbach, R. Rejaie and S. Sen, "Characterizing unstructured overlay topologies in modern P2P file-sharing systems," in *Proc. of the ACM SIGCOMM Internet Measurement Conf., IMC*, New York, USA, pp. 49–62, 2005. https://doi.org/10.1145/1330107.1330114

[58] S. Karuppayah, M. Fischer, C. Rossow and M. Muhlhauser, "On advanced monitoring in resilient and unstructured P2P botnets," in *Proc. of the 2014 IEEE Int. Conf. on Communications ICC 2014*, Sydney, Australia, pp. 871–877, 2014. https://doi.org/10.1109/ICC.2014.6883429

[59] 18 USC, *Chapter 206: Pen Registers and Trap and Devices*, Legal Information Institute LII, U.S., 2016. https://uscode.house.gov/view.xhtml?path=/prelim@title18/part2/chapter206&edition=prelim (accessed Jun. 03, 2022).

[60] G. Yan, S. Chen and S. Eidenbenz, "RatBot : Anti-enume rat ion peer-to-peer botnets," *Lecture Notes in Computer Science*, vol. 7001, pp. 135–151, 2011.

[61] C. Polska, "ZeuS-P2p monitoring and analysis," *Technical Report*, pp. 42, 2013. [Online]. Available: www.cert.pl/PDF/2013-06-p2p-rap_en.pdf

[62] D. Andriesse and H. Bos, "An analysis of the zeus peer-to-peer protocol," *University Amsterdam, Technical Report IR-CS-74*, no. 1, pp. 10, 2014. [Online]. Available: http://www.few.vu.nl/~da.andriesse/papers/zeus-tech-report-2013.pdf

[63] A. Noaman, A. Abdel-Hamid and K. Eskaf, "A novel honeynet architecture using software agents," in *Int. Conf. on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakhier, Bahrain, pp. 1–6, 2019. https://doi.org/10.1109/3ICT.2019.8910299

[64] C. Dalamagkas, P. Sarigiannidis, D. Ioannidis, E. Itrube, O. Nikolis *et al.,* "A survey on honeypots, honeynets and their applications on smart grid," in *Proc. IEEE Conf. on Network Softwarization, NetSoft 2019*, Paris, France, pp. 93–100, 2019. https://doi.org/10.1109/NETSOFT.2019.8806693

[65] S. Morishita, T. Hoizumi, W. Ueno, R. Tanabe, C. Gañán *et al.,* "Detect me if you . . . Oh wait. an internet-wide view of self-revealing honeypots," in *IFIP/IEEE Int. Symp. on Integrated Network Management*, Arlington, VA, USA, pp. 134–143, 2019.