information

MDPI

Review

A Botnets Circumspection: The Current Threat Landscape, and What We Know So Far

Emmanuel C. Ogu [1,*], Olusegun A. Ojesanmi [2], Oludele Awodele [1] and 'Shade Kuyoro [1]

[1] Department of Computer Science, School of Computing and Engineering Sciences, Babcock University, Ilishan-Remo 121103, Ogun State, Nigeria; delealways@yahoo.com (O.A.); afolashadeng@gmail.com ('S.K.)
[2] Department of Computer Science, College of Sciences, Federal University of Agriculture, Abeokuta PMB. 2240, Ogun State, Nigeria; dejioje@yahoo.com
* Correspondence: oguemc@gmail.com

Received: 2 October 2019; Accepted: 19 October 2019; Published: 30 October 2019

Abstract: Botnets have carved a niche in contemporary networking and cybersecurity due to the impact of their operations. The botnet threat continues to evolve and adapt to countermeasures as the security landscape continues to shift. As research efforts attempt to seek a deeper and robust understanding of the nature of the threat for more effective solutions, it becomes necessary to again traverse the threat landscape, and consolidate what is known so far about botnets, that future research directions could be more easily visualised. This research uses the general exploratory approach of the qualitative methodology to survey the current botnet threat landscape: Covering the typology of botnets and their owners, the structure and lifecycle of botnets, botnet attack modes and control architectures, existing countermeasure solutions and limitations, as well as the prospects of a botnet threat. The product is a consolidation of knowledge pertaining the nature of the botnet threat; which also informs future research directions into aspects of the threat landscape where work still needs to be done.

Keywords: botnets; cyber security; network security; information security; threat landscape

1. Introduction

Botnets (or, a network of bots) are an army of compromised machines that are often under the control and coordination of a single source of (direct/indirect) influence via a remote secure channel. They are generally able to propagate themselves on a network and infect vulnerable machines. They typically rely either on maintaining contact with the bot master or owner of the botnet for command and control, or on certain modules within the bot code architecture that perform the same function. Over time, however, bot codes could now be engineered to be able to recruit other vulnerable systems as bots into the botnet, report the status of the operations of individual bots in the botnet, and protect the botnet and its member bots from infiltration [1]. The design and anatomy of botnets often makes them flexible and robust enough to be able to threaten and even successfully subvert any network topology, from conventional cyber infrastructure and devices to Mobile Ad-hoc networks (MANET) [2], Voice over IP (VoIP) infrastructures [3], as well as Autonomous Vehicular Networks [4]

Every botnet essentially has four general participants: These are *the bots or zombies* (the compromised machines) [5], *the bot controller or bot code* (the malicious code that infects the vulnerable hosts/targets in the network), *the bot master or bot herder* (the attacker/hacker/cybercriminal who engineers the bot code and controls the botnet) [6,7], and the *command and control (C&C) server* (the central rendezvous point for all the bots in the botnet [8].

As adapted from [9], and in the light of insights received from the existing literature, Figure 1 illustrates the generic structure of a botnet, showing the interactions between participants in a typical botnet setup.

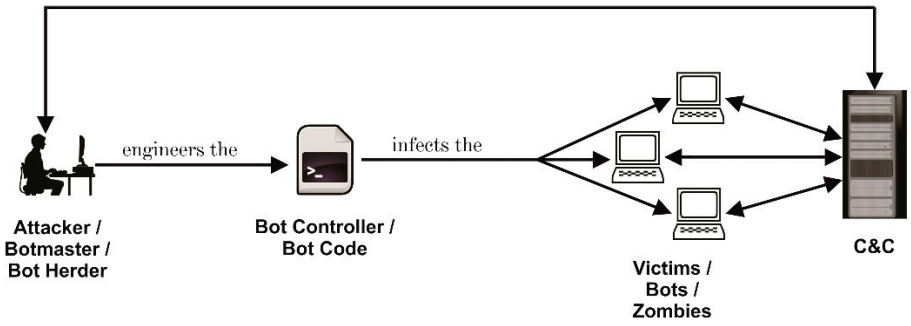

Figure 1. Generic Structure of a Botnet.

The power that drives botnets to achieve their various goals lies in the structure and resilience of its architecture. At the centre of this is the Command and Control (C&C) mechanism. Without this mechanism, a botnet is just a collection of infected machines that lack coherence and coordination [8]. This mechanism is put in place by the bot herder as a central source from which bots in the botnets receive instructions regarding their various operations, and to which they also report the results and status of their various activities.

Botnets have been repeatedly fingered as the major culprits in a type of subversive cyber-attack, known as Denial of Service (DoS) attacks. A DoS attack is a class of cybercrime attack that aims at maxing out the critical computing resources of servers (*CPU, Bandwidth and Memory resources*), thus making it impossible for clients who have legitimately subscribed to use these resources to have access to them. A Distributed Denial of Service (DDoS) attack involves the use of compromised machines (known as "*zombies*") to orchestrate a DoS attack. This class of attacks aims at incapacitating service-based infrastructures that deliver various services to subscribers over the Internet by overwhelming the servers with useless or malicious traffic, such that they are unable to service legitimate requests from authorised subscribers.

1.1. The Origin of Cybercrimes and the Rise of Modern Botnets

The Internet has continued to develop. Today, it is able to directly impact the existence and subsistence of mankind. However, it all began as a small-scale research project originally intended to engender research collaboration and the sharing of information between researchers in the United States. Security was not a concern for the nascent Internet, or any of its component networks in particular—as evidenced in these statements by the father of the Internet himself, Vinton G. Cerf (1943–): "The problem is that early on no one wanted to pay much attention to security ... " "We didn't focus on how you could wreck this system intentionally ... " [10,11]—until around the mid 1990s. As a matter of fact, the architecture of the early Internet was described as "fast, open and frictionless" by [11]; deductively having no form of inhibitions, or resistance to any form of attack or malicious activity.

This led to an Internet system that seemed vulnerable at the core; and because nothing had been formalised at the time, it was difficult to chart logical paths towards lasting solutions to the security challenges that were now plaguing this evidently very beneficial system. This was partly due to the fact that in a bid to mitigate some of these security challenges, a lot of similar ad-hoc solutions emerged. This was coupled with the fact that because component networks of the Internet were increasing, expanding and growing exponentially in their reach at the same time, it was difficult to implement solutions at the lower/operational levels of the network architecture that could actually circumvent and solve these emerging problems on the wider scale.

Because the early Internet was built for research, it found its first use in the hands of researchers and academicians who only used it for sparsely intermittent research activities [12]. With the advent of the Internet came the possibility of remote activities. It was now possible to interact, access, and participate actively without being physically present within the location or environment of influence. This possibility held great promises and prospects for many, amongst many reasons because files and information could now be accessed from miles away, conversations could also now be actively engaged in, regardless of geographical boundaries, and everything else had been brought within reach of a mouse-click or a keyboard command.

In the early years of the Internet, there soon became a massive interest and intrigue in trying to see whether systems and communication infrastructure could be brought down. For instance, an attacker might want to get control of an IRC channel by performing a Denial of Service (DoS) attack against the channel owner just out of "intrigue"; or try to get recognition in underground hacker communities for taking down popular web sites [13,14]. In those early years, there were not many penalties for such crimes, as standardisation was still ongoing regarding various aspects of the Internet; so these crimes were perpetrated at will, either for 'interest', 'intrigue' or 'petty revenge' [15]. All of these acts unarguably evolved to become the cybercrimes we know today.

Kshetri in [16] broadly defined cybercrime as a crime that employs a computer or a network at any phase, such as attacks to critical infrastructure, Internet fraud and money laundering, fraudulent interactions over the Internet, identity theft, to mention a few. In recent times, cybercrimes in the United States alone have been estimated to cost losses of up to $100 billion annually [17] (many of which are usually formally unreported). Symantec (2011) in [18], one of the world's largest cybercrime studies, revealed that the estimated annual global financial and time losses due to cybercrimes ($388 billion) now significantly exceeds the combined financial losses due to marijuana, cocaine and heroin ($288 billion) on the global black market.

Over the course of the past decade, the number of reported cybercrimes rose up to around the millionth mark for cybercrimes that are reported annually, resulting in economic and financial losses that far exceed the annual budget designation (in monetary values) of some entire countries. The United States' Internet Crime Complaint Centre (IC3) on May 10, 2014 received its three millionth complaint, since its establishment, regarding Internet crimes. This brought the total financial losses due to Internet crimes throughout the lifetime of the IC3 up till that point to an excess of $2 billion [19]. Even more recently, the Global Threat Report of the World Economic Forum (WEF) has put financial losses due to cybercrimes to date at over $1 trillion [20]. Perhaps, as Kshetri in [16] opined, the scale of these losses is associated with the fact that cybercrimes differ from other real-world crimes, in that they are savvy in their skillset requirements, globalised in their operations, and relatively uncharted in their landscape.

Figure 2 provides a visualisation of the statistics released by the IC3 from 2000–2018, relating to the reported financial losses due to cybercrimes, as well as the number of cybercrime complaints received for each of the years; herein, the almost exponential increase in these areas is made more evident.

As the Internet continued to expand and garner controlling powers in day-to-day human activities, including but not limited to, areas of healthcare and medicine, and in the affairs and workings of various governments and national economies (ranging from national economic activities, to civic debates, business operations and transactions, urban and regional planning and management, transportation and navigation, pedagogy and education, judicial proceedings, healthcare administration, critical infrastructure administration and most other areas of human endeavour), the issue of cybercrimes moved beyond the domain of mere individual wrongdoings, intrigue and pettiness [15], to become a matter of national, as well as global, security. Better laws were needed to discourage criminal activities on the Internet.

In the wake of this reality, various countries began to institute and tighten cyber laws, beginning with several European countries, and then extending to such countries as the United States, Japan, and Australia, so as to be able to appropriately prosecute and convict criminals who were charged with cybercrimes. The United States Internet Crime Complaint Center (IC3), formerly the Internet Fraud

Complaint Center (IFCC) (until December 2003), was set up on 8 May 2000 as a collaborative centre with the National White Collar Crime Center (NW3C) / Bureau of Justice Assistance (BJA) and the Federal Bureau of Investigation (FBI), for receiving, developing and referring complaints relating to cybercriminal activities; creating an easy and convenient mechanism for reporting cybercrimes, so that the appropriate authorities can be alerted quickly where civil/criminal violations are suspected [21]. This has become one of the foremost platforms for cybercrime reporting, analytics and investigations.

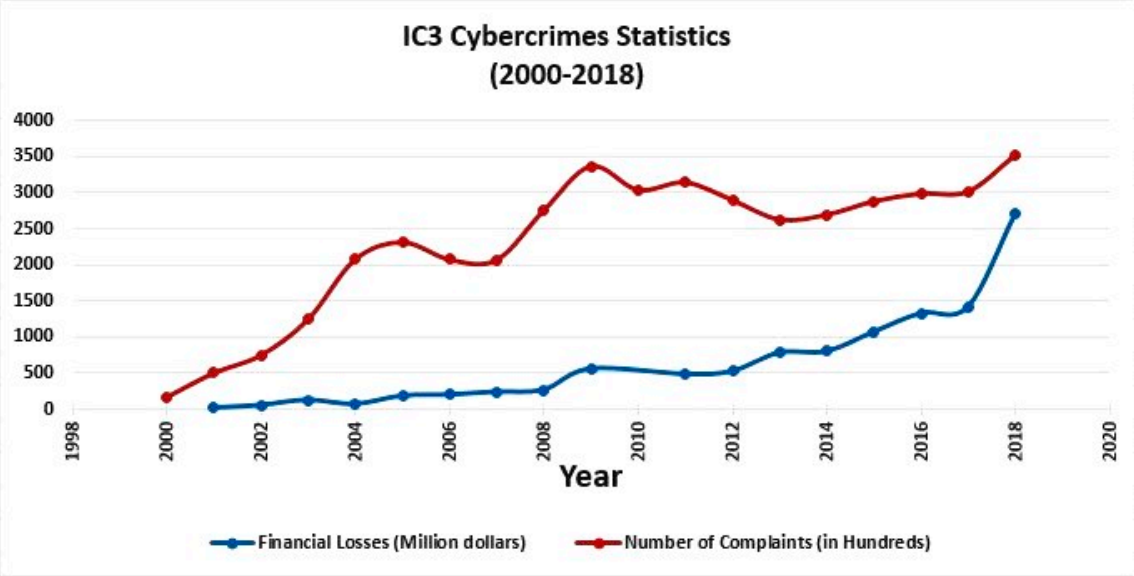

Figure 2. Internet Crime Complaint Centre (IC3) Cybercrime Statistics (2000–2018).

In 2001, the first international treaty on cybercrime was enacted. It was a sequel to the proceedings and recommendations of the Budapest Convention on Cybercrime. This treaty, amongst other provisions, sought to harmonise national laws (of the various member states) on cybercrimes, improve the investigative techniques that were being adopted for cybercriminal prosecution, and increase international cooperation amongst member nations in the Council of Europe [22]. The Committee of Ministers of the Council of Europe adopted the report of the Convention at its 109th Session, which was held on 8 November 2001. The treaty was opened for endorsement in Budapest, on 23 November 2001, and it became a law binding (signatory) member nations on the 1st of July 2004 [23]. This meant that some countries had officially become red zones for cybercriminal activities, even though most were still free zones.

However, as at 23 January 2018, of the 196 countries of the world, 50 (≈26%) nations (comprising the 46 member nations of the Council of Europe, and four non-member nations spanning America, Asia, and Africa) were signatories to the Budapest Convention on Cybercrime treaty; 56 (≈29%) nations (comprising 43 member nations of the Council of Europe, and 13 non-member nations spanning America and Asia) had ratified the treaty, and its provisions and specifications had entered into force as law within their territories [24]. The following are the realities posed by the implications of these statistics:

(a) Approximately 77% of the world is still a relatively safe haven for cybercriminals and their activities; therefore,
(b) The treaty does not yet bear the force of the law in at least four (8%) of the countries that are signatories to it;
(c) At least 10 (amounting to 5%) countries of the world that are not members of the Council of Europe have ratified the provisions and specifications of the treaty into law, even though they are not signatories to the treaty; therefore,

(d) There is still no globally acceptable (by democratic standards of at least a majority of the countries of the world) legal standard through which cybercriminals could be unilaterally prosecuted.

In the words of Sinrod and Reilly in [25], "These countries (that have failed to adopt a globally unified approach to dealing with cybercrimes), inadvertently or not, present the cyber-criminal with a safe haven to operate." Because cybercriminals can hijack machines from these nations and use them for cybercrimes in other nations across cyberspace, they are then able to escape proper prosecution in some cases, even when they are found out. In fact, as legislation and prosecution became more penal and stringent across various countries of the world, hackers and cybercriminals had to devise other means to carry out their malicious activities, preserve or increase the force of their attack impact and devastation, neatly cover up their tracks, complicate the task of detection, and then transfer culpability for their activities on the Internet. One of such means which they resorted to was the use of botnets (literally, hijacking and taking remote control of other peoples' computers, either to take them out, or to use them to orchestrate cybercrimes against other users and machines). "Never before has it been so easy to commit a crime in one jurisdiction while hiding behind the jurisdiction of another" [25]. This way, the rewards and benefits come to the hackers; while culpability rests with the victims.

Perhaps one of the earliest discovered examples of malware that featured take-out operations emerged on the Internet nearly three decades ago; even though some sources may rightly argue that the malware did not show characteristics back then that are conventional of known botnets today. On 3 November 1988, Robert Tappan Morris (1965–), a graduate student of Cornell University (Ithaca, NY, USA), was reported to have accidentally released one of the first worms on the Internet that showed some bot-like attributes—the Morris/Internet worm. This worm had the ability to propagate itself on a computer network and infect other systems without any external aid. About 6,200 computers in the United States alone were taken out by this worm. The Morris worm worked by exploiting known vulnerabilities in computer networks, then installing itself onto computers that were connected to the Advanced Research Projects Agency Network (ARPANET) at the time, crudely taking them out. Productivity losses valued at between $200 and $53,000 were incurred by the direct effect of the worm's activities. Morris was sentenced in December 1990 to three years of probation, 400 h of community service, and a fine of $10,050, including the costs of his supervision [26]. This seemingly innocent act of Morris arguably took away the innocence of the Internet from then hence, because the possibility of remote access to systems had now been made clear.

Following Morris' prosecution, the use of bots still remained unpopular until around the end of the 20th century when Eggdrop—perhaps the very first actual bot code—was written in 1993. Eggdrop was a legal bot code written to assist with the automated administration and management of busy Internet Relay Chat (IRC) channels, since the task of doing this manually had become tasking and time-consuming for IRC operators [27].

Other instances of the legitimate use of bots are incorporated into Internet search engines (Google, Yahoo, Ask, and others), which use these bots to pool together search results to user queries from across the web. So, bot codes were originally not designed for malicious purposes, and botnets were basically not associated with cybercriminal activities until around early 2004, when the likes of Bagle [28], Marina [29], and Torpig [30] emerged on the Internet. The Marina botnet was considered the most devastating of the three, with a bot-army capacity of 6,215,000 that was able to deliver approximately 92 billion spams daily; while the Torpig botnet with a capacity of 180,000 was able to gather over 70 gigabytes of sensitive financial information from the Internet. Today, botnets have become active tools in orchestrating cybercrimes around the world.

The exploitation of Internet users (individuals and corporations) and infrastructure for various (primarily, financial) gains by hackers has continued to gain rapid prominence. Unfortunately also, the level of expertise required to successfully orchestrate cybercrimes (especially of the intrusive and take-over type) in the present era is rapidly declining, partly due to the vast availability of easy-to-use take-over, hacking and cracking tools [31]. In essence, setting up and owning a botnet on the Internet is now something that even unskilled, naïve users can achieve, due to the free availability of tools

that could be used to remotely infiltrate machines on the Internet. Today, botnet activities have been directly/indirectly linked to cybercrimes such as: Denial of Service, Phishing Attacks, Spam, Ransomware, Click Frauds, Bitcoin mining, amongst others [1,32]; and attackers are finding botnets to be valuable tools for fulfilling their various nefarious purposes, which often include: Information Gathering, Distributed Attacks, (Non-Financial) Cyber-Frauds, Malware Distribution, Unsolicited Marketing and Advertising, Denial of Service (DoS) [8], Advanced Persistent Threats, Financial Frauds of various types, and more critically, Cyberterrorism and Cyberwarfare, amongst others.

Since the beginning of the last decade, botnets have grown in popularity, and have become the preferred media for carrying out the various aforementioned cybercrimes. Today, many sophisticated and high-profile botnets continue to exist in cyberspace; notably the likes of Zeus, Storm, Stuxnet and Conficker, among others. Security professionals have identified the biggest and most popular myth and risk preluding botnet and cybercrime victimisation as being "the illusion of 100% security" that is often the result of relatively high investments in security and security products, such as operating systems, anti-spyware and anti-malware applications, firewalls or intrusion detection systems, amongst others, with its associated complacency [33]. In reality, no system can be 100% secure, thereby making this a rather unrealistic goal even for cybersecurity. Rather, and because the global security landscape is constantly evolving and cybercrimes are beginning to look more organised, what works best is to have in place a system that provides the capability to deal with security threats and incidents in such a way as to mitigate losses and other impacts of the threat [34].

Morris' worm, which was arguably one of the first malware to remotely cause damage to computer systems, effectively took away the innocence of the nascent Internet at the time. Cyber laws became necessary, since malicious individuals no longer had to be present at their target locations to actually cause damage to the machines.

1.2. Typology of Attackers & Botnet Owners

An array of free and easy-to-use attack tools on the Internet implement methods and techniques, using which, computers and networks could be easily taken down, even by novices. However, this reality differs remarkably from the case of botnets. The types of attackers that control modern botnets as owners can be distinguished based on the strength of their resources (e.g., composition, coordination, finance, insider capabilities, power and influence, popularity, etc.). There are three broad types/classes of such attackers:

(a) Hackers/Skilled Individuals: This class of attackers are basically individuals that possess an extensive knowledge and skillset in the art of scripting and coding. They are often loners who understand the internal workings of information and communication technologies, systems, and networks well enough to be able to obfuscate and bypass routine processes and procedures, in order to gain access and privileges within secure environments where they are not authorised.

(b) This class of botnet owners tend to use their botnets primarily for financial crimes and identity thefts, and often wield a rather limited amount of resources. They usually cover their tracks very carefully for fear of being found out and prosecuted in line with their jurisdictional cyber laws.

(c) Hacker Groups: This class of attackers are organised groups of hackers that share a common vision, mission and/or ideology. They possess more resource strength when compared to skilled individual hackers, as they are able to pool skillset, influence and finances to orchestrate coordinated attacks with more organisation, precision and impact. In recent times, we have seen a rise in the numbers and popularity of such hacker groups, such as: Anonymous, Chaos Computer Club and Legion of Doom, amongst many others. Because the composition of these groups typically span across several nationalities and jurisdictions with variations in cybercrime legislation, the individual members often seem insulated from the consequences of the group's activities due to the technical complexities associated with the trans-jurisdictional policing of cybercrimes.

(d) Government/Nation-state Actors: Until recently, it was not commonplace to see government and nation-state actors as active players in the activities of cyber offence. Today we have seen several governments come up to claim responsibility for the actions of botnets and similar other malware that have featured in the annals of cyberspace offensives. One popular example was the 2014 Denial of Service attack launched against Sony Pictures by the North Korean Government [35]. Such offensives and attacks are usually of a high-impact nature, owing to the very vast and seemingly limitless resource base that governments and nation-states are usually able to access.

While skilled individual hackers would often own botnets primarily for financial reasons, hacker groups are more inclined to align the activities of their (botnet) operations with the ideologies and missions that inspire the existence of the group. Hacker groups could launch offensives against rival groups to gain popularity, or against government services and infrastructure to prove a point, or complicate the task of governance. Government/Nation-state actors, on the other hand, are more likely to deliver botnet-enabled attacks against entities that tend to threaten or undermine their sovereignty.

Even though financial benefits would sometimes follow as by-products of their operations, hacker groups are typically not financially-motivated as the primary goal of their operations. In recent times, however, the tracks and footprints left behind by cyber incidents tend to suggest that hacker groups can be hired by mafia organisations, government agencies, or wealthy individuals to launder money and use their botnets for various other cybercrimes that may or may not be financially-motivated. Also, there has been evidence suggesting that hackers groups sometimes voluntarily align their solidarity and operations in support of other propagandist, conspiracy and hacktivist ideologies.

2. Synchronous & Asynchronous Botnet Attacks

Botnets engage in two modes of attacks, which could either be synchronous or asynchronous: Synchronous attacks generally rely upon coordinated commands from C&C servers, from which attackers/botherders simultaneously issue commands to all bots in the network. These commands could be for immediate or post-dated action. In this mode of attacks, botnets have to connect to the C&C servers every time to receive the instructions that make synchronous attacks possible. The operations and activities of many common and popular botnets follow after this synchronous approach.

On the other hand, however, there exists certain botnets that are self-sufficient in their binaries and activities, and are able to deliver on operations without first rallying to the C&C servers every time. Asynchronous botnets are distributed in their operations and attacks, and the absence of a C&C server makes it difficult to track and monitor; reverse engineering now becomes the most feasible way to track the footprints and activities of such botnets [36]. Stuxnet was one example of such a botnet. Salamatian et al. in [37] demonstrate an example of how asynchronous attacks are no less potent that synchronous botnet attacks.

However, recent evidence in the footprints left behind by certain high-profile new-age botnets suggest that they incorporate, and are able to switch between both modes of attacks: They adopt the synchronous attack mode when they are online or have Internet access; then they switch to the asynchronous mode when they are offline and without Internet connectivity. The known botnet C&C architectures that feature in the synchronous modes of attack are discussed next.

Known Botnet Control Architectures

There are three main architectures by which existing botnets have been known to be controlled and coordinated. These are shown in Figure 3.

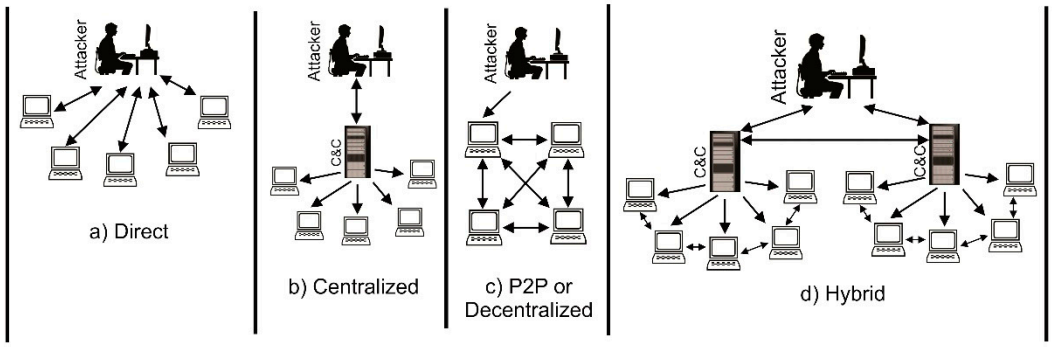

Figure 3. Known Botnet command and control (C&C) Architectures.

As adapted from Eslahi et al. [6] and Barnett [9], and in light of insights received from existing literature, Figure 3 shows the four basic methods by which attackers may choose to deploy the C&C mechanisms for their botnets, as discovered in the literature. These are (a) Direct, (b) Centralised, (c) P2P or Decentralised, and (d) Hybrid.

(a) Direct: This is shown in Figure 3a. In this architecture, botmasters exert direct control over their botnets and the individual bots that compose it. The botmasters are able to directly recruit and interact with the bots and disseminate commands directly, either towards achieving the same (coordinated) or different goals. In this method, only the botmasters are able to know all of the bots in the botnet, because the various bot machines in the botnet typically do not have any form of interaction with each other. Even though this architecture lost popularity as cybercriminal laws became more stringent and penal, because it was possible to trace bots and botnets directly to the botmaster, its popularity is now beginning to rise as more sophisticated machine identity obfuscation techniques are emerging.

(b) Centralised: In the centralised C&C architecture shown in Figure 3b, all the bots in the botnet rely on connection to a centralised C&C server in order to remain a part of the botnet, receive commands and updates, and also make status and operations reports. This method is gradually ceasing to be the preferred method by botmasters because of the ease of takeout. Once the centralised C&C server is located and taken out, the botnet is dislodged, and its operations can no longer cohere. However, botmasters have devised a way of making this method more sophisticated by distributing and deepening the hierarchy of the centralisation, primarily through layering. This sophistication is similar to what was discussed as the Hierarchical (variant) by Marupally and Paruchuri in [38].

(c) Peer-to-Peer (P2P) or Decentralised: In the P2P C&C architecture, each infected host possesses the capability to serve both as a bot and as a C&C server to at least one other computer connected to it. Using the P2P method, the botmaster has no need to maintain forward communication with all of the bots and the botnet, especially at the infiltration level, because the bot code is engineered to be a self-sufficient unit; once released onto the network, the bot code is both able to recruit new bots to join the botnet, and also infuse each bot with the C&C capability, such that only reverse communication (which may then be redirected through several external servers for added layers of anonymity) to the botmaster is necessary to make status and operations reports. It is the lack of a single point of the failure of the botnet modelled in this architecture that makes them more resilient to most modern take down measures [6,39,40]. This architecture is shown in Figure 3c.

(d) Hybrid: As shown in Figure 3d, in the hybrid approach, the strengths of the Direct, P2P and Centralised methods may be combined to create the most resilient deployment of a botnet C&C server. Most modern botnets that have threatened the Internet in recent years (such as Conficker) have been discovered to feature a C&C mechanism [41] that exhibits some form of hybridisation, which has made them quite difficult to exterminate. Marupally and Paruchuri

discuss the Multi-Server P2P Model [38], which illustrates the working principles of the hybrid botnet C&C architecture.

These are more general, over-the-top classes/categories of the communications structures featured by botnets; Vormayr, Zseby and Fabini [42], present a more detailed survey of the communication patterns featured by botnets. However, Table 1 comparatively highlights the differences between the C&C architectures discussed above, based on some characteristics.

Table 1. Comparison of Botnets C&C Architectures.

S/N	Characteristics	C&C Architectures			
		Direct	**Centralised**	**P2P/Decentralised**	**Hybrid**
1	Setup	Easiest	Easy	Fairly Difficult	Difficult (difficulty increases with hybridisation)
2	Administration	Difficult	Less Difficult	Easy	Easier
3	Resilience	Least	Fair	Moderate	High
4	Ease of takeout	Easiest	Easy	Moderately Difficult	Difficult
5	Ease and Accuracy of Traceback	Easiest	Easy	Difficult	Very Difficult
6	Command Dissemination Latency (the time it would take a command issued by the botmaster to travel through to the very last bot in the botnet)	Instant	Fast	Moderately Slow	Slow (speed of dissemination decreases further with depth and level of hybridisation)
7	Possibility of Botnet Failure	Instant	Easier	Easy	Difficult
8	Botnet Enumeration	Near Impossible	Easier	Easy	Difficult
9	Botnet Franchisement	Difficult	Easiest (and more structured)	Easier	Easy

As summarised in Table 1, the direct botnet C&C architecture is the easiest to setup, using some dozens to few hundred lines of bot codes, because no sophistication is involved, and the botmaster directly controls all the bots in the botnet, which makes it more difficult to administer and manage; while the hybrid C&C architecture is the most difficult to setup, due to the required bot code sophistication, but it is also the easiest to administer, because commands just have to be released to any one bot in the botnet, and it can be circulated round the entire botnet machines.

This makes the direct botnet C&C architecture the least resilient and easiest to take out, because it is easy to traceback commands to the point of origin, which is often the machine under the direct control of the botmaster; however, the hybrid architecture presents better resilience and greater difficulty to take out, because there is very low likelihood that the machine which the commands are traced back to is actually the originating (attackers') machine.

Commands are disseminated the slowest in the hybrid C&C architecture compared to other architectures, with the direct architecture being the fastest, because all bots in the botnet receive commands directly from the botmaster; but then, the possibility of the hybrid architecture failing permanently is near impossible compared to other architectures, and with the direct architecture having a likelihood of instant failure as soon as the botmaster's machine is traced and taken out.

In addition, it is more difficult for security professionals to be able to enumerate (know the actual strength of numbers) of a direct C&C botnet structure, as only the botmaster is likely to have the true picture of such; whereas, it is most difficult for enumeration to be done correctly in a hybrid C&C architecture, and relatively easiest in a centralised architecture. But then again, it is very difficult for a botmaster to franchise the whole or parts of the botnet in a direct C&C architecture, because there is no coordinated structure; however, it is easiest for such to happen in the centralised architecture.

3. Lifecycle of Botnets

The generic stages involved in the lifecycle of a typical botnet are shown in Figure 4.

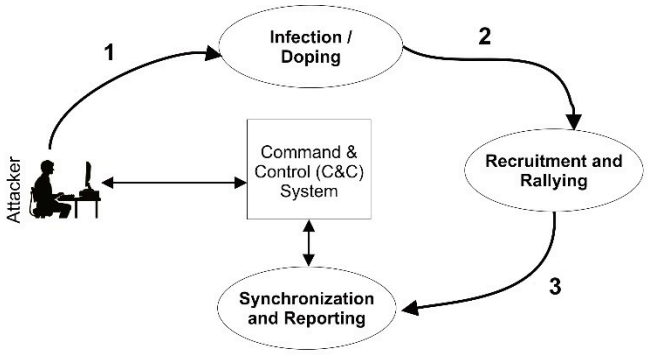

Figure 4. Generic Botnet Lifecycle.

Botnets generally undergo the stages of Infection/Doping, Recruitment & Rallying, and Synchronisation & Reporting during its lifetime, as shown in Figure 4, which is adapted from insights provided by Ogu, et al. [43].

(a) Infection/Doping: This is the first stage of the botnet lifecycle. The botmaster releases a carefully engineered and structured bot code into the network. This code then seeks to exploit certain vulnerabilities in software or network configurations that may already be known to the botmaster (following proper reconnaissance). Once machines are located on the network that feature (these) vulnerabilities, they are infused with the bot code; turning them into zombies, whereby control of these machines are remotely ceded to the botmaster. These machines have now been doped.
Infection/Doping of vulnerable machines could employ either active procedures (such as scanning, flooding, war driving and injection, or physical trans-loading/infusion, amongst others) or passive procedures (such as drive-by downloads, trans-loading from various removable media, or social engineering, emails, ads, cloned URLs, games, bugged/pirated Software, amongst others) [1,8,32].

(b) Recruitment & Rallying: This is the second stage of the botnet lifecycle, and one which the botmaster arguably may consider as the most important stage. This is because the strength of a botnet has been discovered to be directly proportional to its bot-army strength [44]. At this stage of the botnet lifecycle, newer targets with similar vulnerabilities are acquired and enumerated as members of the botnet [6,8].
Rallying mechanisms used in recent botnets include: Hard coded or generated Domain Name Services (DNS) commands; or hardcoded IP Addresses [8].

(c) Synchronisation & Reporting: This is the third stage of the botnet lifecycle. At this stage, the enumerated members of the botnet would be synchronised with the C&C centre, from whence they would henceforth receive commands and directives for action, and also report their status and results of their operations [6,8]. Attackers could decide to either use existing protocols or neoteric protocols for C&C [8]. Following this stage, the bots need to maintain synchronisation with the C&C system at all times in order to receive new commands, infiltration parameters and takeover specifications, which they readily execute. Next, backdoors are installed on the zombies,

unused ports are opened up and/or hijacked, such that even after firewalls upgrades and security patch updates, these would still remain difficult to shut off [1]. These guarantee future access to the bot by the C&C server and the botmaster when the need arises.

After these three stages have been completed, the botnet is said to have reached full maturity. At this point, it becomes difficult to dislodge such a botnet; because dislodgement at this point would mean that either all the bots have been detected and disinfected, the botnet has been abandoned by the botmaster, or the C&C system has been traced and taken out/blocked [6,8].

4. Typology of Existing Botnets

Different types of botnets have been discovered in existence within cyberspace today. Amongst these include:

(a) Spam Botnets: This class of botnets are involved in sending and disseminating large amounts of spamware daily, and seeking to exploit naïve users typically by emails. Popular examples of botnets that belong to the class are the Necrus and Gamut botnets of June 2016 and around early 2013 respectively, which were reported by the McAfee Labs March 2018 Threat Report to comprise a combined 97% of the global spam botnet traffic [45]. Others include Bagle of early 2004, the Storm botnet of early 2007, and the Marina botnet, amongst preponderant others. Xie, et al. [46] discovered that this class of botnets feature a lot of similarities in bot IP address distribution, email sending patterns and behaviours, email properties and sending time.
Information Gathering/Reconnaissance Botnets: This class of botnets are used to mine information over the Internet in large quantities on a daily basis. They also feature in the espionage operations of coordinated cybercrime syndicates. A popular example of a botnet that belongs in this class is the Mirai botnet that was discovered in August 2016 to have been scanning the Internet for the IP addresses of vulnerable devices that are part of the Internet of Things (IoT) [47], and then goes on to infect them to be enlisted as part of the botnet; which was later discovered to have been behind the October 2016 Dyn cyber-attack [48]. Kolias, et al. [49] and Kambourakis, et al. [50] present a detailed analysis of the Mirai botnet, covering its internal structure, system of operations, variants of the Mirai botnet and the realities that Mirai and related botnets portend for the future of the IoT.

(b) The Satori botnet is a more dreaded variant of the Mirai botnet that was discovered in May 2018 to feature operations similar to its parent form, but was instead focused on mining information pertaining to vulnerable cryptocurrency remote management infrastructures for the purpose of later infiltrating user wallets to steal cryptocurrencies [51]. Another example was the Asprox botnet that hit the Internet around 2008.

(c) Identity Theft Botnets: This class of botnets are involved in stealing large amounts of private user identity information, such as social security and credit card details, health record information, login usernames and passwords, among other forms of sensitive information, typically for fraudulent purposes. Popular examples of botnets that belong in this class include the Zeus botnet that "compromised over 74,000 FTP accounts on websites of such companies as the Bank of America, NASA, Monster.com, ABC, Oracle, Play.com, Cisco, Amazon, and BusinessWeek", stealing sensitive banking information through web browser keystroke logging and form grabbing [52], and also the Bredolab botnet, which was developed in 2009 by 27-year-old Russian Hacker Georgy Avanesov to siphon bank account passwords and other confidential information from infected computers [53]. Others include the Torpig botnet of since 2005, the Alureon botnet of around 2010, and the Mariposa botnet of December 2008, amongst others.

(d) Click-Fraud Botnets: This class of botnets attempt to mimic legitimate human click-ad behaviour in a bid to con Internet advertisers into believing that their online adverts have been engaged with by a legitimate, actual human audience, thereby accumulating financial revenue for the botherders as the botnet click-advertising web traffic continues to mount. One popular example of a botnet belonging to this class is the Chameleon botnet of February 2013, which was reported

to have amassed a monthly revenue of over $6 million USD for the botnet owners following an infection of over 120,000 Windows® machines [54].

(e) Crypto Botnets: This class of botnets are used by criminals in mining crypto currencies and resources for financial gain. Examples include the high-profile Smominru and ADB.Miner cryptomining botnets.

The typology of most known botnets can be broadly aligned with one or more of these type groups, based on the features and characteristics exhibited by the botnet.

5. Botnet Countermeasures

Botmasters have kept implementing newer methods of evading detection, both for the botmasters themselves (using methods like stepping stones, surrogacy, etc.), their bots (using methods like binary obfuscation, anti-analysis, security suppression, rootkit technology, amongst others), their botnet C&C servers (using such methods as IP and domain flux, rogue DNS servers, anonymisation, amongst others), and their botnet C&C traffic (using such methods as encryption, protocol and traffic tweaking and novel communication techniques, etc.) [8].

Towards the emergence and advancement of effective and efficient countermeasures to botnets, some sort of structured knowledge about the behaviours that characterise botnets is needed. Khattak et al. [8] and Plohmann, et al. [55] present two recent, very structured taxonomies and reviews in this regard—covering botnet behaviours, techniques for evading detection and hiding information, approaches to detecting botnets, as well as procedures for disinfection and measuring impact. In addition, Kambourakis et al. [56] further present an up-to-date research on the botnet threat and techniques that have been featured in the Internet of Things (IoT), as well as the economics of botnets, and blockchain capabilities for enhancing reliability, efficiency and the resilience of botnets and their operations.

The knowledge contained in these studies are valuably complementary and indeed supplementary to this research, because they are fundamental to understanding the principles of botnet operations in general, as well as the strategies that have informed the evolution of countermeasures to botnet activities. Nonetheless, these studies do not cover the systematics goals that various botnet countermeasures have sought to achieve on the rapidly evolving threat landscape. We consider this knowledge to be essential for understanding how research has evolved to solve various aspects of the botnets problem, and how the threat and threat landscape has progressively evolved (expectedly so) to respond to advancements in research. This knowledge is covered in this section of the research.

Sources such as FORTINET (2012) [1], SOPHOS (2014) [32], Stone-Gross et al. [57], Ianelli and Hackworth [58] and the Honeynet Project and Research Alliance (2005) [59] from up to a decade ago have directly linked botnet activities to cybercrimes like Denial of Service, Phishing Attacks, Spam, Ransomware, Click Frauds, Bitcoin mining, amongst others. Some sources have also proposed countermeasures to botnet activities on computer networks. These countermeasures are either deployed in the network or on the host in order for them to work effectively [8]. The procedures or countermeasures proposed in literature for combating botnets are usually aimed at one or more of the following broad goals: (1) Prevention [P], (2) Detection [D], (3) Offensive [O], (4) Reconnaissance (or, Recon.) [R], or (5) Mitigation [M].

(a) Prevention [P]: This goal is aimed at increasing the chances/possibility of averting the occurrence of a botnet attack in a network. Techniques that seek to achieve this goal are most often implemented around the edge (entry and exit) points of the network environment/infrastructure, so as to make sure that traffic and data coming into the network are legitimate and non-malicious. Examples of such proposed techniques include that proposed by Luo et al. [60], amongst others.

(b) Detection [D]: The goal here is to spot the presence of a botnet within a production network. Techniques that are tailored towards this goal typically focus on analysing packets, traffic and communications that take place within the network, so as to identify those of malicious

compositions, operations and intent, and to alert the administrators accordingly. These techniques may sometimes do little to prevent the botnets from attacking the network, but they basically aim to identify the presence of botnets in the network. However, in some deployment cases, these techniques are used alongside another technique(s) that focuses on one of the other goals, so as to make it more potent and sophisticated. For example, Böck et al. [36] propose a novel approach to detecting botnets that are fully-distributed and asynchronous in their operations, using a novel mechanism known as Trust Based Botnet Monitoring Countermeasure (TrustBotMC), and then proposing a follow-up mitigation strategy. Khattak et al. [8] offer further insight into the dimensions for botnet detection that have been proposed in literature.

The task of detecting botnets in actual implementations is often made further difficult by a phenomenon known as flash crowds. Flash Crowds occur when a large crowd of legitimate users repeatedly try to gain access to a server resource or service at the same time, often around a time that can be considered as the peak period(s) of such service(s) (known as a flash event), and may wrongly be flagged off as a persistent threat/attack situation. Indeed, Flash Crowds can also cause DoS to occur, and in fact go a long way to further complicate the task of detecting and controlling DoS attacks. This is so because flash crowd traffic and DoS attack traffic have certain characteristics in common, and distinguishing them under the rush and load of DoS traffic can be a really difficult task. Peng et al. [44] and Alsaleem et al. [61] proposed a rule-based mechanism by which HTTP denial of service (DoS) attacks could be detected and isolated during flash events, while in the same vein, Saad et al. [62] proposed a rule-based technique for the detection of anomalous ICMPv6 behaviours; all for the purpose of reducing the rates of false positives and negatives in threat situations. Also, Jazi et al. [63] proposed a technique for detecting HTTP-based DoS attacks at the application layers of web servers using sampling techniques, while Behal et al. [64] reviews existing strategies and methods for characterising and isolating Distributed Denial of Service (DDoS) attacks, even in the midst of flash events. Lonea [65] proposed a quantitative method for detecting DDoS attacks in cloud environments by analysing intrusion detection system alerts, while D'Cruze [66] proposed an efficient and flexible Software-Defined Networking (SDN) solution to mitigate DDoS attacks on Internet Service Provider (ISP) networks.

Furthermore, machine learning-based detection techniques have also been proposed for the detection of DDoS attacks that are widely known to feature botnets as the primary threat agent. Examples of such techniques are put forward by He et al. [67] and Nidhi et al. [68], amongst others.

(c) Offensive [O]: Here, the goal is to launch a form of counter-attack against the botnet/intrusion element, with the ultimate motive of taking it down (where possible), or forcing the individual bots to go against the commands they are receiving from the C&C server; effectively obfuscating and dislodging the botnet. One way by which this is often achieved is through sinkholing. Usually, countermeasures that are built towards this goal are designed to take advantage of freshly-discovered/already-known vulnerabilities in the botnet design architecture, by engaging with active research findings and discoveries that relate to the botnet under investigation; hence, offensive countermeasures are often not generic, but specific to certain botnet types/examples. Offensives could be direct (when they engage with the botnet directly, and are targeted towards dislodging/incapacitating specific botnet components or the botnet itself) or indirect (when they are they just targeted towards obscuring or redirecting particular botnet components, often through surrogate points in the network) [8].

(d) Reconnaissance [R]: Though considered to be one of the most passive of countermeasure goals, this goal is actually what should be the foundation of any countermeasure that seeks to effectively take down any modern engineered botnet. The goal here is to passively monitor a known botnet that has been detected on a network, and gather as much information as possible relating to its mode of operations, bot members/strength, C&C architecture, malicious capabilities and obfuscation techniques, amongst others. Most offensive countermeasures that

actually produce any result in real botnet scenarios rely largely on detailed and extensive ab initio/pre-engagement reconnaissance.

(e) Mitigation [M]: This goal is aimed at controlling and curtailing the extent of the damage to the network and hosts, whose environment has already been breached by a rampaging botnet; it is concerned with damage control. Countermeasures focused towards this goal typically involve disinfecting bots in real-time, stopping compromised services, reinforcing firewall defences, closing up unused ports on hosts, amongst others.

The more effective countermeasures discovered in literature often combine at best two or three of the above goals in a single countermeasure solution. Most classic and modern countermeasures did not emerge from a perfect synthesis of the above goals. This presents an insight into the fact that, perhaps the ultimate countermeasure(s) to botnets may be that (those) which is/are able to incorporate the most or all of these goals into one single countermeasure solution that is able to present a complete, robust and formidable front against botnet attacks; being able to switch between and synergise these goals effectively in the face of real attacks.

5.1. Categories & Limitations of Existing Botnet Countermeasures

Countermeasures developed for botnets can be grouped into the following categories:

5.1.1. Spoofing

Spoofing countermeasures are beginning to emerge as one of the most promising countermeasures to botnets. These techniques work by seeking to compromise certain characteristics of the bot code, such that the bot codes are unable to return the results pre-configured by the botmaster. In the work by Xiang et al. [41], a technique is proposed for spoofing the executable file path for persistent botnets, such that each time they run, they are forced to fire up a countermeasure solution against them. Also, in Liu et al. (2014) [69], an anti-spoofing technique is proposed against botnets that carry out flooding/DoS attacks against networks by spoofing the IP addresses of legitimate hosts within the network.

This mutual egress filtering (MEF) technique, implemented within the access control lists of the autonomous border routing systems of networks, works by fishing and filtering outbound packets, these having source addresses that are not members of the local autonomous system, and especially when the packet is set towards a spoofing attack against other MEF-enabled autonomous systems. Liu and Bi (2015) [70] present a succinct overview of spoofing techniques and their deployability, with the advantages and added security it could offer to networks.

Most spoofing techniques are built with detection capabilities, oftentimes only for the botnets that they are engineered to spoof. In some cases, they may also generate results that could help to prevent or immunise machines against certain botnets, such as that proposed by Xiang et al. [41]. However, this may not always be the case. Hence, they may prove impotent in the face of dynamic or polymorphic botnets.

5.1.2. Analysis-Based

A number of recent botnet detection countermeasures are based on one form of analysis of either botnet traffic, network traffic, log files, or the other; even when they are encrypted, as proposed in Ying, (2014) [71] and Zhang et al. [72]. The primary goal of most Analysis-Based techniques is to detect, (and in some cases, prevent) the presence of botnets or bot activity in the network. One way by which this is achieved is by comparing certain characteristics of network traffic with a database of attributes (signatures) that have already been associated with specific botnet behaviours, such as those proposed by Zand et al. [73], Bilge et al. [74] and Bhatia et al. [75]. Furthermore, as seen in the literature, another technique seeks to analyse network traffic for traffic/packets that exhibit behaviours that deviate from the normal, observable network behaviour (an anomaly), such as those proposed by Wang et al. [76], Boukhtouta et al. [77], Zhao et al. [78] and Caglayan et al. [79]. Anomalous behaviours could also

be analysed in relation to hypertext transfer protocol (HTTP) traffic, such as noted by Jia et al. [80], Mathew et al. [81], Choi et al. [82] and Eslahi et al. [83]; domain name service (DNS) query patterns and properties, such as by Seo et al. [84] and Futai et al. [85]; and transmission control protocol (TCP) requests, such as that looked into by Abdullah et al. [86]. Some other techniques discovered in the literature involve the graphical analysis and interpretation of the associations and interactions of bots, in a bid to forecast their potentials, understand the structure of their C&C, amongst others, as proposed by Wang and Paschalidis (2014) [87]; or the analysis of search engine log files, as proposed by Zhang et al. [88].

Honeypotting: Another strategy that is frequently featured in and alongside most detection techniques, is to gather relevant/needful information about botnets (reconnaissance). This involves the use of systems that are known as Honeypots. Honeypots are vulnerably configured machines that are set up as easy targets around the edges of, or within a network. They are used to pose as easy targets for attackers, such that when they are easily compromised and enumerated as members of a botnet, they can then be accessed and used to gather relevant information about a botnet that could be useful in defending against it, such as: Its C&C mechanism, signature characteristics and behaviours that could be implemented for content-based detection, attacker motivation, bot source codes, etc. [89], or the number of nodes present in the botnet [39]. In other deployments, honeypots are seen as tools that help to gather information about users and activities that take place within a monitored area. You may see them as surveillance cameras that observe and log the activities of entities that may have malicious intent without their notice. Honeypots are able to fool attackers into thinking they have succeeded with their attack intents by presenting them with effects and responses that suggest attack success, when in actual sense they are being observed, tracked and monitored (see Spitzner (2003). [90]).

Examples of implementations of the honeypot technique towards countering botnets can be found in sources such as Al-Hakbani and Dahshan (2015) [91], Daniel and Hongmei (2013) [92] and Moon et al. [93]. As adapted based on insights received from Barfar et al. [94], Figure 5 illustrates a basic (naïve) deployment of honeypots.

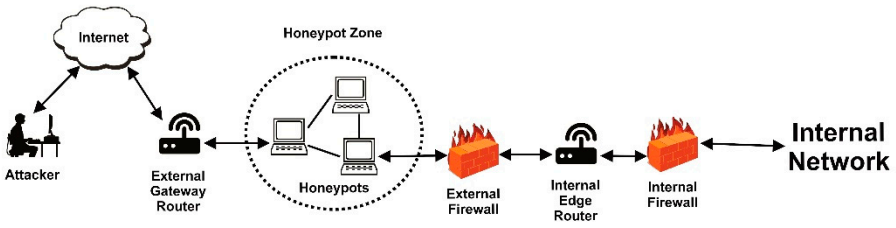

Figure 5. A basic deployment of Honeypots.

Darknets are fast becoming a popular alternative to the use of honeypots. A Darknet, which is essentially a derivative of a honeynet, refers "to a private or public chunk of a network that is empty of any servers/services." It can be seen as "a silent honeypot." Actually, a darknet has at least one silent host on its network, which is catching and inspecting every packet. This is based on the idea that because packets and traffic are not expected to be seen on that side of the network, therefore, any packet found there is suspected to be malicious and should be analysed [95]. Figure 6 illustrates a basic deployment of a darknet as adapted based on insights from Landecki et al. [95].

The various analysis-based techniques proposed in literature have pitfalls that are unique to each of these techniques. The advantages and pitfalls of each of these techniques have been reviewed and succinctly pointed out in Eslahi et al. [6], Raghava et al. [89], Panimalar et al. [96] and Seewald et al. [97]. However, it is also important to add to these that the extent of the analysis procedures used by most of these techniques could, if not controlled, add an extra overhead to the cost of network operation and management, and potentially result in a compromise of the quality of service (QoS) of the network.

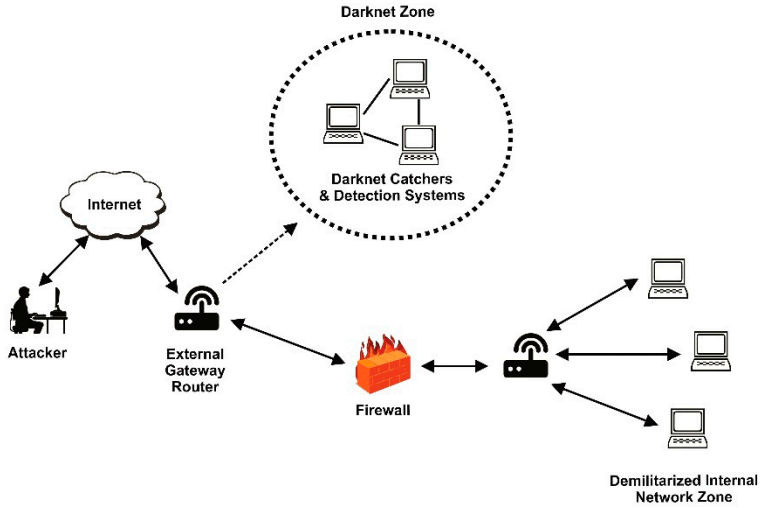

Figure 6. A basic deployment of a Darknet.

Some modern techniques have also proposed improvements to some of the pitfalls posed by some analysis-based techniques, in order to enhance their potency and reduce the number of false positives generated, such as Huang (2013) [98], Narang et al. [99] and in Narang et al. [100].

5.1.3. Exploit/Take Down

Some countermeasures are also proposed in literature that could attempt offensives at botnets. Usually, these techniques are built with capabilities that help them to be able to spot the presence of particular botnets on a network; and they are able to then mitigate the impact of these botnets by launching offensives/counters against them. Sometimes in a bid to overrun them, these botnets are kept busy trying to ward off these attacks, thereby reducing their spread and impact. Watkins et al. [101] proposed a technique for exploiting C&C vulnerabilities in botnets. This technique was experimented with the Zeus botnet, and involved the use of fuzzy logic to discover vulnerabilities in its C&C mechanism. The discovered vulnerabilities were now used to engineer a sinkholing countermeasure aimed at redirecting botnet traffic heading to the C&C server to another location known as a sinkhole, so as to cripple the C&C operations of the botnet, thereby dislodging it, and limiting the impact of simulated DoS attacks.

Furthermore, some of these techniques could also incorporate one form of analysis-based capabilities or the other, as part of their botnet detection procedures, while also providing offensive and defensive capabilities. A recent example of such was proposed by Yan et al. [102].

One other technique proposed in the work by Hangxia et al. [103] takes advantage of the vulnerability posed by the fact that nodes can join a P2P botnet without any way of confirming whether they were honest nodes or not. In this technique, the routing table of botnets are poisoned (compromised by infiltration) with nodes information; thereby causing it to re-route the command information from the attacker to possibly non-existent nodes, and disrupting C&C communication, thereby mitigating the impact of the botnet.

Major limitations in exploiting/taking down countermeasures are the ethical and legal concerns regarding the extent of involvement, meddling and access which these countermeasures often require in order to perform effectively.

5.1.4. Mining

Mining-based countermeasures use data mining/machine learning techniques to acquire relevant information about botnets from already acquired botnet data. These techniques are often used in collaboration with other methods that achieve other countermeasure goals. Recent examples of such

techniques include: Lin et al. [104], Eskandari and Raesi (2014) [105], Tsuruta and Shoudai (2013) [106], Garant and Lu (2013) [107] and Ohrui et al. [108]. Also, Monshizadeh et al. [109] and So-In et al. [110] present a recent in-depth evaluation and discussion of such mining techniques.

6. Botnets in Mobile and Cloud Environments

The portability and growing popularity of mobile devices also has effects on the threat landscape of botnets [111], and this has created inroads to a new area of research. Ref. [112] and Ref. [113] give a comprehensive review of mobile botnets; covering the state-of-the-art C&C architectures that are featured by contemporary mobile botnets. Anagnostopoulos et al. [113] further elucidates the advanced capabilities that Tor's hidden services and DNS protocols avails to hackers for masking their identities and footprints, while also upscaling the resilience of their bot army through optimisations to a TXT-based Tor fluxing scheme for DNS operations. In addition, sources such as Conti et al. [114], Kadir et al. [115], Farina et al. [116], Alzahrani et al. [117], Natarajan et al. [118], Liao and Li (2014) [119], Eslahi et al. [120], Mtibaa et al. [121], Abdullah et al. [122], Hamon (2014) [123], Mtibaa (2013) [124], Choi et al. [125] and Apvrille et al. [126] highlight some of the recent techniques and procedures used by attackers in setting up mobile botnets, as well as some countermeasures that have been recently developed for combating the challenge of mobile botnets. Today, malware featuring the likeness and actual characteristics of botnets have been found to compromise known operating systems that power mobile devices; including the popular Symbian Java (Symbian/Yxes worm of 2009), IoS (iKee.B bot client of 2009), and Android (Geinimi malware of 2010) mobile operating systems [112].

As mobile botnets continue to exhibit advanced capabilities for coordination and stealth communication, the have become a crucial concern for contemporary cybersecurity.

Botnets have also been discovered in traditional as well as mobile clouds, as a means of enhancing their strength, ubiquity and coverage. This has been discussed by recent studies such as Mtibaa et al. [127], Li et al. [128] and Zhaoet et al. [129]; and techniques for the detection of botnets in cloud environments have been proposed by Badiset al. [130] and Kebande and Venter (2014) [131], while countermeasures have also been reviewed and proposed to mitigate the impact of botnet-propelled attacks in cloud-based environments, as can be found in Alosaimi et al. [132] and Wahab et al. [133], who proposed an adaptive solution for detecting insider attacks in cloud environments, and also by Daffu and Kaur (2016) [134], respectively, amongst others.

7. The Botnets of the Future

Newer botnets are expected to keep emerging, as old ones keep metamorphosing in order to evade the current solutions landscape. This is because attackers see a new criminal empire building on the Internet, where botnets can now be sold/leased to individuals or corporations for various malicious and nefarious uses, in exchange for some financial benefits. Chang et al. [135], along with Bottazzi and Me (2014) [136] give insights into botnets-as-a-service, a new paradigm that describes this trend. Furthermore, the presence of botnets in mobile and cloud environments provides insights to the possibility that botnets of the future may soon on their own be able to learn and exploit vulnerabilities in the patterns of user interactions and operations by modifying/re-configuring themselves, through the capabilities of machine learning, to orchestrate malicious activities against such a user/machine, as well as other users and/or machines. Vasilomanolakis et al. [137] present such a futuristic botnet that implements computational trust models and is smart, cautious and able to learn from their past experiences.

One other reason why botnets may yet linger on the Internet is due to the ethical and legal implications and reservations that now stall the implementation of some already developed countermeasures. As has been earlier highlighted in this research, even blue chip malware defence and anti-virus manufacturers face trust issues and concerns in trying to implement malware solutions. The extent of access to a user's machine and information, as well as meddling, which these solutions require to produce desired results, are beginning to generate some ethical and legal debates. These are

expected to intensify in the future years, even as these solutions continue to get more sophisticated and interfering.

Botnets are also expected to more decisively hit mobile devices and cloud infrastructures in the future. The ubiquity provided by mobile devices, coupled with advancements in mobile communication technologies (LTE, 3G and WiFi), propose a new landscape with great potentials for botherders [8]. Furthermore, the Internet of Things presents another viable frontier for the proliferation of botnets and the perpetuation of their activities in cyberspace. In the aftermath of the Mirai botnet of August 2016, recent researches such as that by Kolias et al. [49] have emphasised the need for more pragmatic and holistic approaches to IoT security, in order to safeguard the future of the IoT.

As the global landscape is beginning to more tightly embrace the coming revolution of the Internet of things, it is important to point out that all of the devices that are expected to be interconnected by the Internet of Things could also serve to provide a wider pool of takeover victims to be enumerated as part of botnet activities. Kambourakis et al. [50] present a comprehensive state-of-the-art review of the IoT botnet threat landscape that is contemporaneous in relevance, and this emphasises that in a poorly-secured IoT landscape, every connected device is a potential hub/rallying point for cybercrimes and various nefarious cyber operations.

8. Summary

Based upon the progress of the literature survey so far, Figure 7 summarises the current state of the threat landscape of botnets as discovered by the foregoing research survey.

In summary, bot codes are propagated either by active (scanning, war-driving & injection, flooding, physical trans-loading, infusion, etc.) means, or passive (drive-by downloads, trans-loading, social engineering, bugged/pirated software, emails, games, ads, cloned URLs, etc.) means. Botnet participants include the attacker/botmaster/botherder/hacker, the bot code/controller, the zombies/bots/victims and the C&C server and communication traffic. Attackers evade detection using such methods as stepping-stones and surrogacy; bot controllers typically evade detection by encryption methods; the bots adopt the method of binary obfuscation, anti-analysis, security suppression or rootkit technology to hide their network presence; the C&C servers & communication traffic evade detection by using IP & domain flux, rogue servers, anonymisation, encryption, protocol and traffic tweaking, etc.

Botnet operations and attacks could be either synchronous or asynchronous. Synchronous attacks rely on a C&C interaction, while asynchronous attacks do not. Command & control architectures featured by modern botnets are classified as direct, centralised, P2P/decentralised, or hybrid. The typical stages in the lifecycle of a botnet include: Infection/doping, recruitment & rallying, synchronisation & reporting. Existing botnets could be broadly typified as spam botnets, information gathering/reconnaissance botnets, identity theft botnets, or click-fraud botnets; while the attackers/owners of these botnets could be skilled hackers, hacker groups, or government/nation-state actors. Existing botnets countermeasures either focus on prevention, detection, offence, reconnaissance and/or mitigation. These findings characterise the current threat landscape of botnets, generally.

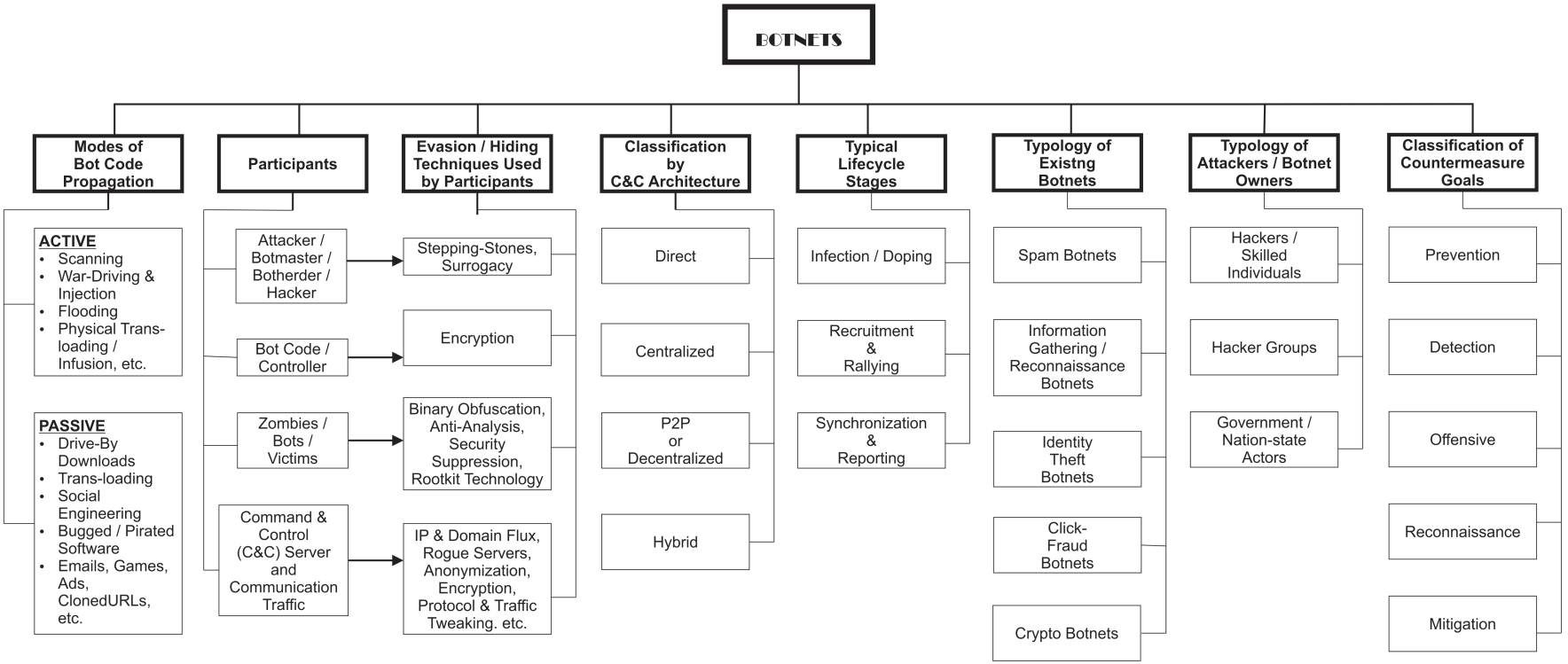

Figure 7. The Current Threat Landscape—What We Know So Far about Botnets.

9. Future Research Directions

As we expect to see botnets become even more sophisticated in the near future, further research is needed to unpack the internal workings of botnets that especially feature the decentralised/P2P control architecture. This remains one of the known formidable features of modern botnets, and botnet owners and attackers are expected to begin exploring technologies such as blockchain and advanced crypography towards hardening these nefarious control structures and making them more impervious.

In addition, the exposition of the reverse lifecycle of botnets necessitates future research to explore the covert backdoor channels that help preserve the subsistence of botnets by acting as "open doors" for their reawakening. As modern approaches to sinkholing and darknet implementations continue to reveal valuable information regarding botnets and their key survival features, future research directions should hone these findings and provide robust insights that would help terminate the reverse lifecycle of botnets.

Furthermore, in the wake of brewing political tensions on the global scene, and fears of politically-motivated cyber-attacks, it is also crucial for future research to explore the power and operational dynamics that characterise the different types of attackers who own and control botnets. This is likely to lead to the discovery of new threat and attack categories/classes that would help the security community to better understand the distinguishable characteristics of cyber threats and attacks on a broader scale.

10. Conclusions

In conclusion, this research has been able to circumspect in a relatively extensive manner the current threat landscape of the botnet threat, presenting what is known so far about this threat. This research comes on the heels of exigency of research efforts, especially in light of the growing political tensions sweeping across various parts of the world, particularly those tensions that have accompanied the fear of politically-motivated cyber-attacks. It therefore has become important for knowledge pertaining the botnet threat to be put together in a way that helps to easily visualise how the threat has evolved over the years, as well as the research efforts that have been put forward towards combatting the threat. It is herein that this research makes its contribution. Future researches can now be able to see at a glance what areas of the threat landscape still requires concerted research efforts, and what areas still remain relatively uncharted.

Author Contributions: Conceptualisation, E.C.O.; investigation, E.C.O.; resources, All Authors; writing—original draft preparation, E.C.O.; writing—review and editing, E.C.O.; visualisation, E.C.O.; supervision, O.A.O., O.A. and 'S.K.; project administration, E.C.O.

Funding: This research received no external funding.

Acknowledgments: The inputs of the three expert peer reviewers, as well as journal editors, and colleagues, which helped to enhance the quality of this research, are acknowledged and highly appreciated.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. FORTINET. *Anatomy of a Botnet*; Fortinet®: Sunnyvale, CA, USA, 2012.
2. Hanafy, I.M.; Salama, A.A.; Abdelfattah, M.; Wazery, Y.M. AIS Model for botnet detection in MANET using fuzzy function. *Int. J. Comput. Netw. Wirel. Mob. Commun.* **2013**, *3*, 95–102.
3. Geneiatakis, D.; Vrakas, N.; Lambrinoudakis, C. Utilizing bloom filters for detecting flooding attacks against SIP based services. *Comput. Secur.* **2009**, *28*, 578–591. [CrossRef]
4. Garip, T.M.; Gursoy, E.M.; Reiher, P.; Gerla, M. Congestion Attacks to Autonomous Cars Using Vehicular Botnets. In Proceedings of the 2015 Network and Distributed System Security (NDSS) Workshop on Security of Emerging Networking Technologies, San Diego, CA, USA, 8 February 2015.

5. Tanwar, G.S.; Goar, V. Tools, Techniques & Analysis of Botnet. In Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies, Udaipur, India, 14–16 November 2014; ACM: New York, NY, USA, 2014; pp. 1–5.
6. Eslahi, M.; Salleh, R.; Anuar, N. Bots and botnets: An overview of characteristics, detection and challenges. In Proceedings of the International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 23–25 November 2012; IEEE Press: Piscataway, NJ, USA, 2012; pp. 349–354.
7. Bijalwan, A.; Pilli, E.S. Understanding botnet onInternet. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 5 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–5.
8. Khattak, S.; Ramay, N.R.; Khan, K.R.; Syed, A.A.; Khayam, S.A. A taxonomy of botnet behavior, detection, and defense. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 898–924. [CrossRef]
9. Barnett, R. Botnet Herders Targeting Web Servers. Tactical Web Application Security Blog. 14 May 2010. Available online: http://tacticalwebappsec.blogspot.com.ng/2010/05/botnet-herders-targeting-web-servers.html (accessed on 6 May 2018).
10. Greenemeier, L. Connecting with anInternet Pioneer, 40 Years Later. *Scientific American*. 4 December 2009. Available online: https://www.scientificamerican.com/article/internet-pioneer-cerf/ (accessed on 8 October 2017).
11. Timberg, C. Net of Insecurity: A Flaw in the Design. *Washington Post*. 30 May 2015. Available online: http://www.washingtonpost.com/sf/business/2015/05/30/net-of-insecurity-part-1/?utm_term=.798dc8fff3c9 (accessed on 8 October 2017).
12. Davies, A. Computational intermediation and the evolution of computation as a commodity. *Appl. Econ.* **2004**, *36*, 1131–1142. [CrossRef]
13. Dittrich, D. The DoS Project's "trinoo" Distributed Denial of Service Attack Tool. 2014. Available online: http://staff.washington.edu/dittrich/misc/trinoo.analysis (accessed on 1 February 1999).
14. Qijun, G.; Liu, P. Denial of Service Attacks. San Marcos. 2007. Available online: http://s2.ist.psu.edu/paper/DDoS-Chap-Gu-June-07.pdf (accessed on 21 Ocotober 2019).
15. Network Box UK Ltd. *Denial of Service Attacks (DoS)*; Managed Security Services; Network Box: London, UK, 2010; Available online: http://www.network-box.co.uk/sites/default/files/Denial%20of%20Service.pdf (accessed on 21 Ocotober 2019).
16. Kshetri, N. The simple economics of cybercrimes. *IEEE Secur. Priv.* **2006**, *4*, 33–39. [CrossRef]
17. Gorman, S. Annual U.S. Cybercrime Costs Estimated at $100 Billion; Study Casts Doubt on Previous, Higher Figures. *Wall Street Journal Publications*. 22 July 2013. Available online: https://www.wsj.com/articles/SB10001424127887324328904578621880966242990 (accessed on 3 March 2018).
18. Symantec. *Norton Study Calculates Cost of Global Cybercrime: $114 Billion Annually*; Symantec: Mountain View, CA, USA, 2011; Available online: http://www.symantec.com/about/news/release/article.jsp?prid=20110907_02 (accessed on 26 April 2015).
19. Internet Crime Complaint Centre (IC3). *TheInternet Crime Complaint Center Receives 3 Millionth Complaint*. Available online: http://www.ic3.gov/media/2014/140519.aspx (accessed on 24 April 2015).
20. World Economic Forum (WEF). *The Global Risks Report 2018*, 13th ed.; World Economic Forum: Geneva, Switzerland, 2018; Available online: http://www3.weforum.org/docs/WEF_GRR18_Report.pdf (accessed on 19 January 2018).
21. Internet Crime Complaint Centre (IC3). *2010 Internet Crime Report*. Available online: http://www.ic3.gov/media/annualreport/2010_IC3Report.pdf (accessed on 24 April 2015).
22. Council of Europe. *Convention on Cybercrime*; The Council of Europe's Official Treaty Office: Strasbourg, France, 2001; Available online: http://conventions.coe.int/Treaty/EN/Treaties/Html/185.htm (accessed on 24 April 2015).
23. United Nations Educational, Scientific and Cultural Organisation (UNESCO). *The COE International Convention on Cybercrime before Its Entry Into Force*; United Nations Educational, Scientific and Cultural Organisation: Paris, France, 2014; Available online: http://portal.unesco.org/culture/en/files/19556/11515912361coe_e.pdf/coe_e.pdf (accessed on 24 April 2015).
24. Council of Europe. *Convention on Cybercrime-CETS No.: 185*; The Council of Europe's Official Treaty Office: Strasbourg, France, 2001; Available online: https://www.coe.int/en/web/conventions/full-list/-/conventions/treaty/185/signatures (accessed on 23 January 2018).

25. Sinrod, E.J.; Reilly, W.P. Cyber-crimes: A practical approach to the application of federal computer crime laws. *St. Clara Comput. High Tech. LJ* **2000**, *16*, 177.
26. Lee, T.B. *How a Grad Student Trying to Build the First Botnet Brought the Internet to Its Knees*; The Washington Post: Washington, DC, USA, 2013.
27. Egg Development Team. *Eggdrop: Open Source IRC Bot*. Available online: http://www.eggheads.org/ (accessed on 21 Ocotober 2019).
28. Mashevsky, Y. The Bagle Botnet. SECURELIST-Information about Viruses, Hackers and Spam. 22 April 2005. Available online: http://securelist.com/analysis/36046/the-bagle-botnet/ (accessed on 26 April 2015).
29. Cuevas, A. *Botnets: Zombies, Spam, and Attacks*; Sites At Penn State: State College, PA, USA, 2015; Available online: http://sites.psu.edu/psucybersecuritycuevas/2015/02/18/botnets-zombies-spam-and-attacks/ (accessed on 26 April 2015).
30. Miller, C. Researchers Hijack Control of Torpig Botnet. *IT Security News and Security Product Reviews-SC Magazine*. 2 May 2009. Available online: http://www.scmagazine.com/researchers-hijack-control-of-torpig-botnet/article/136207/ (accessed on 26 April 2015).
31. Gupta, B.B.; Joshi, R.C.; Misra, M. Distributed Denial of Service Prevention Techniques. *Int. J. Comput. Electr. Eng.* **2010**, *2*, 268–276. [CrossRef]
32. SOPHOS. *Security Threat Report 2014*; SOPHOS: Oxford, UK, 2014.
33. Stackpole, B. Is Your Firm Resting on its Security Laurals? Symantec Blog. 28 November 2017. Available online: https://www.symantec.com/blogs/feature-stories/your-firm-resting-its-security-laurels?es_p=5721813 (accessed on 1 January 2018).
34. KPMG. *Cybercrime Survey Report 2014*; KPMG: Mumbai, India, 2014; Available online: https://www.kpmg.com/IN/en/IssuesAndInsights/ArticlesPublications/Documents/KPMG_Cyber_Crime_survey_report_2014.pdf (accessed on 26 April 2015).
35. DeSimone, A.; Horton, N. *Sony's Nightmare before Christmas: The 2014 North Korean Cyber Attack on Sony and Lessons for US Government Actions in Cyberspace*; National Security Report; The Johns Hopkins University Applied Physics Laboratory LLC: Laurel, MD, USA, 2017; Available online: https://www.jhuapl.edu/Content/documents/SonyNightmareBeforeChristmas.pdf (accessed on 25 October 2019).
36. Böck, L.; Vasilomanolakis, E.; Wolf, J.H.; Mühlhäuser, M. Autonomously detecting sensors in fully distributed botnets. *Comput. Secur.* **2019**, *83*, 1–13. [CrossRef]
37. Salamatian, S.; Huleihel, W.; Beirami, A.; Cohen, A.; Médard, M. Why botnets work: Distributed brute-force attacks need no synchronisation. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2288–2299. [CrossRef]
38. Marupally, P.R.; Paruchuri, V. Comparative Analysis and Evaluation of Botnet Command and Control Models. In Proceedings of the 24th IEEE International Conference of Advanced Information Networking and Applications (AINA), Washington, DC, USA, 20–23 April 2010; pp. 82–89.
39. Rossow, C.; Andriesse, D.; Werner, T.; Stone-Gross, B.; Plohmann, D.; Dietrich, C.J.; Bos, H. Sok: P2pwned-modeling and evaluating the resilience of peer-to-peer botnets. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 19–22 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 97–111.
40. Xiao-Nan, L.; Yang, L.; Hua, Z. Peer-to-Peer botnets: Analysis and defense. In Proceedings of the 3rd IEEE International Conference on Communication Software and Networks (ICCSN), Xi'an, China, 27–29 May 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 140–143.
41. Xiang, C.; Lihua, Y.; Shuyuan, J.; Zhiyu, H.; Shuhao, L. Botnet Spoofing: Fighting Botnet with Itself. *Secur. Commun. Netw.* **2015**, *8*, 80–89. [CrossRef]
42. Vormayr, G.; Zseby, T.; Fabini, J. Botnet Communication Patterns. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2768–2796. [CrossRef]
43. Ogu, E.C.; Vrakas, N.; Ogu, C.; Ajose-Ismail, B.M. On the Internal Workings of Botnets: A Review. *Int. J. Comput. Appl.* **2016**, *138*, 39–43. [CrossRef]
44. Peng, T.; Leckie, C.; Ramamohanarao, K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput. Surv.* **2007**, *39*. [CrossRef]
45. Beek, C. *Necurs Botnet Leads the World in Sending Spam Traffic*. Available online: https://securingtomorrow.mcafee.com/mcafee-labs/necurs-botnet-leads-the-world-in-sending-spam-traffic/ (accessed on 24 June 2018).
46. Xie, Y.; Yu, F.; Achan, K.; Panigrahy, R.; Hulten, G.; Osipkov, I. Spamming botnets: Signatures and characteristics. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 171–182. [CrossRef]

47. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the Mirai Botnet. In Proceedings of the 26th USENIX Security Symposium, Vancouver, BC, Canada, 16–18 August 2017; USENIX: Berkeley, CA, USA, 2017; pp. 1093–1110.
48. Newman, L. What We Know about Friday's Massive East CoastInternet Outage. *WIRED*. 21 October 2016. Available online: https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/ (accessed on 24 June 2018).
49. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and other botnets. *Computer* **2017**, *50*, 80–84. [CrossRef]
50. Kambourakis, G.; Kolias, C.; Stavrou, A. The Mirai botnet and the IoT zombie armies. In Proceedings of the MILCOM 2017–2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 267–272. [CrossRef]
51. Cimpanu, C. The Satori Botnet Is Mass-Scanning for Exposed Ethereum Mining Rigs. BLEEPINGCOMPUTER. 18 May 2018. Available online: https://www.bleepingcomputer.com/news/security/the-satori-botnet-is-mass-scanning-for-exposed-ethereum-mining-rigs/ (accessed on 24 June 2018).
52. Ragan, S. *ZBot Data Dump Discovered with over 74,000 FTP Credentials*; The Tech Herald: Mumbai, India, 2009; Available online: http://www.thetechherald.com/article.php/200927/3960/ZBot-data-dump-discovered-with-over-74-000-FTP-credentials (accessed on 17 November 2009).
53. Zetter, K. Hacker Lexicon: Botnets, the Zombie Computer Armies That Earn Hackers Millions. *WIRED*. 15 December 2015. Available online: https://www.wired.com/2015/12/hacker-lexicon-botnets-the-zombie-computer-armies-that-earn-hackers-millions/ (accessed on 24 June 2018).
54. BBC. *Botnet Steals 'Millions of Dollars from Advertisers'*; BBC: London, UK, 2013; Available online: http://www.bbc.com/news/technology-21860360 (accessed on 25 June 2018).
55. Plohmann, D.; Gerhards-Padilla, E.; Leder, F. *Botnets: Detection, Measurement, Disinfection & Defence*; The European Network and Information Security Agency (ENISA): Heraklion, Greece, 2011.
56. Kambourakis, G.; Anagnostopoulos, M.; Meng, W.; Zhou, P. *Botnets: Architectures, Countermeasures, and Challenges*; CRC Press: Boca Raton, FL, USA, 2019.
57. Stone-Gross, B.; Cova, M.; Cavallaro, L.; Gilbert, B.; Szydlowski, M.; Kemmerer, R.; Kruegel, C.; Vigna, G. Your botnet is my botnet: Analysis of a botnet takeover. In Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009; ACM: New York, NY, USA, 2009; pp. 635–647.
58. Ianelli, N.; Hackworth, A. Botnets as a vehicle for online crime. *Forensic Comput. Sci. IJoFCS* **2005**, *2*, 19–39.
59. Honeynet Project and Research Alliance. *Know Your Enemy: Tracking Botnets*. Available online: http://www.honeynet.org/papers/bots/ (accessed on 21 Ocotober 2019).
60. Luo, H.; Chen, Z.; Li, J.; Vasilakos, A.V. Preventing distributed denial-of-service flooding attacks with dynamic path identifiers. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1801–1815. [CrossRef]
61. Alsaleem, S.; Manickam, S.; Anbar, M.; Alnajjar, A.; Saleh, E. A Rule-Based Mechanism for Detecting HTTP Denial of Service Attacks During Flash Crowd Event. *Adv. Sci. Lett.* **2017**, *23*, 5423–5425. [CrossRef]
62. Saad, R.M.; Anbar, M.; Manickam, S. Rule-based detection technique for ICMPv6 anomalous behaviour. *Neural Comput. Appl.* **2017**, *30*, 1–10. [CrossRef]
63. Jazi, H.H.; Gonzalez, H.; Stakhanova, N.; Ghorbani, A.A. Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. *Comput. Netw.* **2017**, *121*, 25–36. [CrossRef]
64. Behal, S.; Kumar, K.; Sachdeva, M. Characterizing DDoS attacks and flash events: Review, research gaps and future directions. *Comput. Sci. Rev.* **2017**, *25*, 101–114. [CrossRef]
65. Lonea, A.M.; Popescu, D.E.; Tianfield, H. Detecting DDoS attacks in cloud computing environment. *Int. J. Comput. Commun. Control* **2013**, *8*, 70–78. [CrossRef]
66. D'Cruze, H.; Wang, P.; Sbeit, R.O.; Ray, A. A Software-Defined Networking (SDN) Approach to Mitigating DDoS Attacks. In *Information Technology-New Generations*; Springer: Cham, Switzerland, 2018; pp. 141–145.
67. He, Z.; Zhang, T.; Lee, R.B. Machine Learning Based DDoS Attack Detection from Source Side in Cloud. In Proceedings of the IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 26–28 June 2017; IEEE: New York, NY, USA, 2017; pp. 114–120. [CrossRef]
68. Nidhi, M.V.; Prasad, K.M. Detection of Anomaly Based Application Layer DDoS Attacks Using Machine Learning Approaches. *i-Manag. J. Comput. Sci.* **2016**, *4*, 6–13.

69. Liu, B.; Bi, J.; Vasilakos, A.V. Toward incentivizing anti-spoofing deployment. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 436–450. [CrossRef]
70. Liu, B.; Bi, J. On the Deployability of Inter-AS Spoofing Defenses. *Network* **2015**, *29*, 82–87. [CrossRef]
71. Ying, W. Encrypted Botnet Detection Scheme. In Proceedings of the Ninth International Conference on P2P, Parallel, Grid, Cloud andInternet Computing (3PGCIC), Guangdong, China, 8–10 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 559–565.
72. Zhang, H.; Papadopoulos, C.; Massey, D. Detecting encrypted botnet traffic. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 163–168.
73. Zand, A.; Vigna, G.; Yan, X.; Kruegel, C. Extracting probable command and control signatures for detecting botnets. In Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, 24–28 March 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 1657–1662.
74. Bilge, L.; Balzarotti, D.; Robertson, W.; Kirda, E.; Kruegel, C. Disclosure: Detecting botnet command and control servers through large-scale netflow analysis. In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando, FL, USA, 3–7 December 2012; Association for Computing Machinery: New York, NY, USA, 2012; pp. 129–138.
75. Bhatia, J.S.; Sehgal, R.K.; Kumar, S. Honeynet based botnet detection using command signatures. In *Advances in Wireless, Mobile Networks and Applications*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 69–78.
76. Wang, K.; Huang, C.-Y.; Tsai, L.-Y.; Lin, Y.-D. Behaviour-based botnet detection in parallel. In *Security and Communication Networks*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2014.
77. Boukhtouta, A.; Lakhdari, N.E.; Mokhov, S.A.; Debbabi, M. Towards fingerprinting malicious traffic. *Procedia Comput. Sci.* **2013**, *19*, 548–555. Available online: http://www.sciencedirect.com/science/article/pii/S1877050913006819 (accessed on 21 Ocotober 2019). [CrossRef]
78. Zhao, D.; Traore, I.; Sayed, B.; Lu, W.; Saad, S.; Ghorbani, A.; Garant, D. Botnet detection based on traffic behavior analysis and flow intervals. *Comput. Secur.* **2013**, *39*, 2–16. [CrossRef]
79. Caglayan, A.; Toothaker, M.; Drapeau, D.; Burke, D.; Eaton, G. Behavioral analysis of botnets for threat intelligence. *Inf. Syst. E-Bus. Manag.* **2012**, *10*, 491–519. [CrossRef]
80. Jia, Y.; Chen, Y.; Dong, X.; Saxena, P.; Mao, J.; Liang, Z. Man-in-the-browser-cache: Persisting HTTPS attacks via browser cache poisoning. *Comput. Secur.* **2015**, *55*, 62–80. [CrossRef]
81. Mathew, S.E.; Ali, A.; Stephen, J. Genetic algorithm based layered detection and defense of HTTP botnet. *Int. J. Netw. Secur.* **2014**, *5*, 50–61.
82. Choi, J.; Choi, C.; Ko, B.; Kim, P. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. *Soft Comput.* **2014**, *18*, 1697–1703. [CrossRef]
83. Eslahi, M.; Hashim, H.; Tahir, N.M. An efficient false alarm reduction approach in HTTP-based botnet detection. In Proceedings of the 2013 IEEE Symposium on Computers Informatics (ISCI), Langkawi, Malaysia, 7–9 April 2013; pp. 201–205.
84. Seo, I.; Lee, H.; Han, S.C. Cylindrical Coordinates Security Visualisation for multiple domain command and control botnet detection. *Comput. Secur.* **2014**, *46*, 141–153. [CrossRef]
85. Futai, Z.; Siyu, Z.; Weixiong, R. Hybrid detection and tracking of fast-flux botnet on domain name system traffic. *Commun. China* **2013**, *10*, 81–94. [CrossRef]
86. Abdullah, R.S.; Mas' ud, M.Z.; Abdollah, M.F.; Sahib, S.; Yusof, R. Recognizing P2P botnets characteristic through TCP distinctive behaviour. *Int. J. Comput. Sci. Inf. Secur.* **2011**, *9*, 7–11.
87. Wang, J.; Paschalidis, I.C. Botnet detection using social graph analysis. In Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 27–30 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 393–400.
88. Zhang, J.; Xie, Y.; Yu, F.; Soukal, D.; Lee, W. Intention and Origination: An Inside Look at Large-Scale Bot Queries. In Proceedings of the 20th Annual Network & Distributed System Security Symposium, San Diego, CA, USA, 24–27 February 2013.
89. Raghava, N.S.; Sahgal, D.; Chandna, S. Classification of botnet detection based on botnet architechture. In Proceedings of the International Conference on Communication Systems and Network Technologies (CSNT), Rajkot, India, 11–13 May 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 569–572.
90. Spitzner, L. *Honeypots: Tracking Hackers*; Addison Wesley Professional: Boston, MA, USA, 2003; Volume 1.

91. Al-Hakbani, M.M.; Dahshan, M.H. Avoiding honeypot detection in peer-to-peer botnets. In Proceedings of the IEEE International Conference on Engineering and Technology (ICETECH), Liverpool, UK, 26–28 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–7.
92. Daniel, A.; Hongmei, C. An empirical study of botnets on university networks using low-interaction honeypots. In Proceedings of the 51st ACM Southeast Conference, Savannah, GA, USA, 4–6 April 2013; Association for Computing Machinery: Atlanta, GA, USA, 2013. [CrossRef]
93. Moon, Y.H.; Kim, E.; Hur, S.M.; Kim, H.K. Detection of botnets before activation: An enhanced honeypot system for intentional infection and behavioral observation of malware. *Secur. Commun. Netw.* **2012**, *5*, 1094–1101. [CrossRef]
94. Barfar, A.; Mohammadi, S. Honeypots: Intrusion deception. *Inf. Syst. Secur. Assoc. J.* **2015**, *48*, 15.
95. Landecki, G. Detecting Botnets, Issue 177. *Linux®Journal*. 1 January 2009. Available online: http://www.linuxjournal.com/magazine/detecting-botnets (accessed on 17 Januuary 2018).
96. Panimalar, P.; Rameshkumar, K. A review on taxonomy of botnet detection. In Proceedings of the International Conference on Advances in Engineering and Technology (ICAET), Singapore, 29–30 March 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–4.
97. Seewald, A.K.; Gansterer, W.N. On the detection and identification of botnets. *Comput. Secur.* **2010**, *29*, 45–58. [CrossRef]
98. Huang, C.Y. Effective bot host detection based on network failure models. *Comput. Netw.* **2013**, *57*, 514–525. [CrossRef]
99. Narang, P.; Ray, S.; Hota, C.; Venkatakrishnan, V. Peershark: Detecting peer-to-peer botnets by tracking conversations. In Proceedings of the 2014 IEEE Security and Privacy Workshops (SPW)), San Jose, CA, USA, 17–18 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 108–115.
100. Narang, P.; Hota, C.; Venkatakrishnan, V.N. PeerShark: Flow-clustering and conversation-generation for malicious peer-to-peer traffic identification. *EURASIP J. Inf. Secur.* **2014**, *1*, 1–12. [CrossRef]
101. Watkins, L.; Kawka, C.; Corbett, C.; Robinson, W.H. Fighting banking botnets by exploiting inherent command and control vulnerabilities. In Proceedings of the 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE), Fajardo, PR, USA, 28–30 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 93–100.
102. Yan, Z.; Kantola, R.; Shen, Y. Unwanted traffic control via hybrid trust management. In Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Liverpool, UK, 25–27 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 666–673.
103. Hangxia, Z. Mitigating Peer-to-Peer Botnets by Sybil Attacks. In Proceedings of the International Conference on Innovative Computing & Communication, 2010 and Information Technology & Ocean Engineering and 2010 Asia-Pacific Conference on (CICC-ITOE), Macao, China, 30–31 January 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 241–243.
104. Lin, S.C.; Chen, P.S.; Chang, C.C. A novel method of mining network flow to detect P2P botnets. *Peer Peer Netw. Appl.* **2014**, *7*, 645–654. [CrossRef]
105. Eskandari, M.; Raesi, H. Frequent sub-graph mining for intelligent malware detection. *Secur. Commun. Netw.* **2014**, *7*, 1872–1886. [CrossRef]
106. Tsuruta, H.; Shoudai, T. Structure-based Data Mining and Screening for Network Traffic Data. In Proceedings of the IIAI International Conference on Advanced Applied Informatics (IIAIAAI), Matsue, Japan, 31 August–4 September 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 152–157.
107. Garant, D.; Lu, W. Mining Botnet Behaviors on the Large-Scale Web Application Community. In Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Barcelona, Spain, 25–28 March 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 185–190.
108. Ohrui, M.; Kikuchi, H.; Terada, M.; Rosyid, N.R. Apriori-PrefixSpan Hybrid Approach for Automated Detection of Botnet Coordinated Attacks. In Proceedings of the 14th International Conference on Network-Based Information Systems (NBiS), Tirana, Albania, 7–9 September 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 92–97.
109. Monshizadeh, M.; Yan, Z. Security Related Data Mining. In Proceedings of the IEEE International Conference on Computer and Information Technology (CIT), Xi'an, China, 11–13 Septeberm 2014; pp. 775–782.

110. So-In, C.; Mongkonchai, N.; Aimtongkham, P.; Wijitsopon, K.; Rujirakul, K. An evaluation of data mining classification models for network intrusion detection. In Proceedings of the Fourth International Conference on Digital Information and Communication Technology and It's Applications (DICTAP), Bangkok, Thailand, 6–8 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 90–94.
111. Mtibaa, A.; Harras, K.A.; Alnuweiri, H. From botnets to MobiBots: A novel malicious communication paradigm for mobile botnets. *IEEE Commun. Mag.* **2015**, *53*, 61–67. [CrossRef]
112. Anagnostopoulos, M.; Kambourakis, G.; Gritzalis, S. New facets of mobile botnet: Architecture and evaluation. *Int. J. Inf. Secur.* **2016**, *15*, 455–473. [CrossRef]
113. Anagnostopoulos, M.; Kambourakis, G.; Drakatos, P.; Karavolos, M.; Kotsilitis, S.; Yau, D.K. Botnet Command and Control Architectures Revisited: Tor Hidden Services and Fluxing. In Proceedings of the International Conference on Web Information Systems Engineering, Moscow, Russia, 7–11 October 2017; Springer: Cham, Switzerland, 2017; pp. 517–527.
114. Conti, M.; Mancini, L.V.; Spolaor, R.; Verde, N.V. Analyzing Android Encrypted Network Traffic to Identify User Actions. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 114–125. [CrossRef]
115. Kadir, A.F.; Stakhanova, N.; Ghorbani, A.A. *Android Botnets: What URLs are Telling Us. Network and System Security*; Springer: Cham, Switzerland, 2015; pp. 78–91.
116. Farina, P.; Cambiaso, E.; Papaleo, G.; Aiello, M. Understanding DDoS Attacks from Mobile Devices. In Proceedings of the 3rd International Conference on FutureInternet of Things and Cloud (FiCloud), Rome, Italy, 24–26 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 614–619.
117. Alzahrani, A.J.; Ghorbani, A.A. Real-time signature-based detection approach for SMS botnet. In Proceedings of the 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, 21–23 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 157–164.
118. Natarajan, V.; Sheen, S.; Anitha, R. Multilevel Analysis to Detect Covert Social Botnet in Multimedia Social Networks. *Comput. J.* **2015**, *58*, 679–687. [CrossRef]
119. Liao, Q.; Li, Z. Portfolio optimisation of computer and mobile botnets. *Int. J. Inf. Secur.* **2014**, *13*, 1–14. [CrossRef]
120. Eslahi, M.; Rostami, M.R.; Hashim, H.; Tahir, N.M.; Naseri, M.V. A data collection approach for Mobile Botnet analysis and detection. In Proceedings of the IEEE Symposium on Wireless Technology and Applications (ISWTA), Kota Kinabalu, Malaysia, 28 September–1 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 199–204.
121. Mtibaa, A.; Alnuweiri, H.; Harras, K. Mobibots: Risk Assessment Of Collaborative Mobile-to-mobile Malicious Communication. In Proceedings of the Qatar Foundation Annual Research Conference, Doha, Qatar, 18–19 November 2014; p. ITPP1085.
122. Abdullah, Z.; Saudi, M.M.; Anuar, N.B. Mobile botnet detection: Proof of concept. In Proceedings of the 5th IEEE Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 4–5 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 257–262.
123. Hamon, V. Android botnets for multi-targeted attacks. *J. Comput. Virol. Hacking Tech.* **2014**, 1–10. [CrossRef]
124. Mtibaa, A. MobiBots: Towards detecting distributed mobile botnets. In Proceedings of the Qatar Foundation Annual Research Conference, Doha, Qatar, 24–25 November 2013; p. ICTO-05.
125. Choi, B.; Choi, S.K.; Cho, K. Detection of mobile botnet using vpn. In Proceedings of the Seventh International Conference on Innovative Mobile andInternet Services in Ubiquitous Computing (IMIS), Taichung, Taiwan, 3–5 July 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 142–148.
126. Apvrille, A. Symbian worm Yxes: Towards mobile botnets? *J. Comput. Virol.* **2012**, *8*, 117–131. [CrossRef]
127. Mtibaa, A.; Harras, K.; Alnuweiri, H. Malicious attacks in Mobile Device Clouds: A data driven risk assessment. In Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN), Shanghai, China, 4–7 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–8.
128. Li, Q.; Larsen, C.; van der Horst, T. IPv6: A Catalyst and Evasion Tool for Botnets and Malware Delivery Networks. *Computer* **2013**, *46*, 76–82. [CrossRef]
129. Zhao, S.; Lee, P.P.; Lui, J.; Guan, X.; Ma, X.; Tao, J. Cloud-based push-styled mobile botnets: A case study of exploiting the cloud to device messaging service. In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando, FL, USA, 3–7 December 2012; Association for Computing Machinery (ACM): New York, NY, USA, 2012; pp. 119–128.

130. Badis, H.; Doyen, G.; Khatoun, R. A collaborative approach for a source based detection of botclouds. In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 906–909.
131. Kebande, V.R.; Venter, H.S. A cognitive approach for botnet detection using Artificial Immune System in the cloud. In Proceedings of the Third International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), Beirut, Lebanon, 29 April–1 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 52–57.
132. Alosaimi, W.; Zak, M.; Al-Begain, K.; Alroobaea, R.; Masud, M. Mitigation of Distributed Denial of Service Attacks in the Cloud. *Cybern. Inf. Technol.* **2017**, *17*, 32–51. [CrossRef]
133. Wahab, O.A.; Bentahar, J.; Otrok, H.; Mourad, A. I Know You Are Watching Me: Stackelberg-Based Adaptive Intrusion Detection Strategy for Insider Attacks in the Cloud. In Proceedings of the IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 25–30 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 728–735.
134. Daffu, P.; Kaur, A. Mitigation of DDoS attacks in cloud computing. In Proceedings of the 5th International Conference on Wireless Networks and Embedded Systems (WECON), Rajpura, India, 19–20 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5. [CrossRef]
135. Chang, W.; Wang, A.; Mohaisen, A.; Chen, S. Characterizing botnets-as-a-service. In Proceedings of the 2014 ACM Conference on SIGCOMM, Chicago, IL, USA, 17–22 August 2014; Association for Computing Machinery (ACM): New York, NY, USA, 2014; pp. 585–586.
136. Bottazzi, G.; Me, G. The Botnet Revenue Model. In Proceedings of the 7th International Conference on Security of Information and Networks, Glasgow, UK, 9–11 November 2014; ACM: New York, NY, USA, 2014; pp. 459–465.
137. Vasilomanolakis, E.; Wolf, J.H.; Böck, L.; Karuppayah, S.; Mühlhäuser, M. I Trust my Zombies: A Trust-enabled Botnet. *arXiv* **2017**, arXiv:1712.03713.