Chapter 8
Botnet Forensics

Learning Objectives

- Understanding botnets and the importance of botnet forensics
- Study the botnet forensic process models
- Understanding of various botnet forensic frameworks
- Knowledge of standard tools available for botnet forensics

8.1 Introduction

A botnet is a network of compromised computers controlled by attackers from remote location via C&C channels [1]. The compromised computers are called drones, and the attacker controlling the botnet is called botmaster.

The attackers infect large numbers of vulnerable computers via any primary infection mechanism and guide them to communicate to C&C servers. Further, these infected computers get the secondary payload and other instructions from the C&C servers. In this way attackers remotely control the botnet army and use them for many illegal activities even without owners' knowledge.

These botnets are designed to propagate and communicate in a very covert manner and avoid detection by traditional security tools installed on the network terminals. Botnets run autonomously in a very covert manner and the attacker forwards commands to the bots' army via the randomly compromised hosts in the Internet. This mechanism puts the attacker in the far background and makes it very difficult to trace to the botmaster.

The various botnets discovered in the wild have caused huge financial losses to enterprises, governments, Internet Service Providers, educational systems, and even home users. In June 2014, FBI estimated GameOver Zeus botnets was responsible for more than $100 million loss to many banking and online services

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_8

before its takedown [5]. In the report FBI estimated that 500 million computers are compromised annually, incurring global losses of approximately $110 billion globally [2].

The botnets provide large distributed platforms to perform various malicious activities. Attackers use botnets for a variety of nefarious applications such as launching DDoS attacks, spamming, phishing, spying, click fraud, mining bitcoins, brute force password attacks, and many other malicious activities [3].

The detection of these botnets and traceback to the attackers are very difficult. Moreover, even if attacker location is traced or the C&C servers are located, the cross border presence makes the law enforcement issues very challenging. Therefore, botnet forensics is required to thoroughly analyze the botnets to improve security tools and techniques. The forensic investigation is also needed to collect the evidences to be used to seek permissions to remove the C&C servers and any law enforcement.

The use of computing devices like smartphones, tablets, personal computers, workstations, and other high-end servers connected with the high-speed Internet pave the way for attackers to scan, probe, infect, and highjack these computers to grow their botnet army. Moreover, the malicious content in various forms attract the users to download rogue software, malicious free games, and files and click phishing mails to get them infected.

The botnets run autonomously using multilayer architectures. They communicate using many covert techniques thus changing to track the attackers. The botnets have constantly evolved to more sophisticated and complex structures since the start in 1993. The first-generation botnets used IRC as their C&C protocol. During last few years, botnets using Web servers as C&C channels and HTTP as communication protocol have been discovered. These traditional centralized botnets exhibit C&C traffic leading to be detected, and thus suffer from a single point of failure [4]. Further, the malware authors developed decentralized botnets based on peer to peer (P2P) networks to overcome the weakness of single point of failure. Even after, the hybrid polymorphic botnets are also discovered in the wild. Understanding of the botnet features is important to detect, measure, and compare botnets.

In this chapter we explain the botnet threats to the Internet world. The botnet architectures, protocols, and life cycle are explained for the basic understanding of the threat. We explain the standard botnet forensic process and investigation techniques. Further, we discuss the usefulness of such forensic investigation process against the botnets. Finally, the identified research challenges are explained to serve as the future research direction.

8.2 Botnets Forensics

The first botnet, Eggdrop, was discovered in 1993 being used for sharing user information to protect channels. The botnet was based on IRC protocol. Afterward botnets have constantly evolved to more sophisticated and complex structures. Many small to large malicious botnets have been developed with multiple features and functionalities. Further botnets adopted different architectures and continuously evolved with diverse C&C protocols.

Very few botnets are developed from scratch, while most of the botnets discovered are modified versions of some earlier ones. The first-generation botnets used IRC as their C&C channels. The second-generation botnets evolved using Web servers as C&C and HTTP as communication protocol. Figure 8.1a shows the

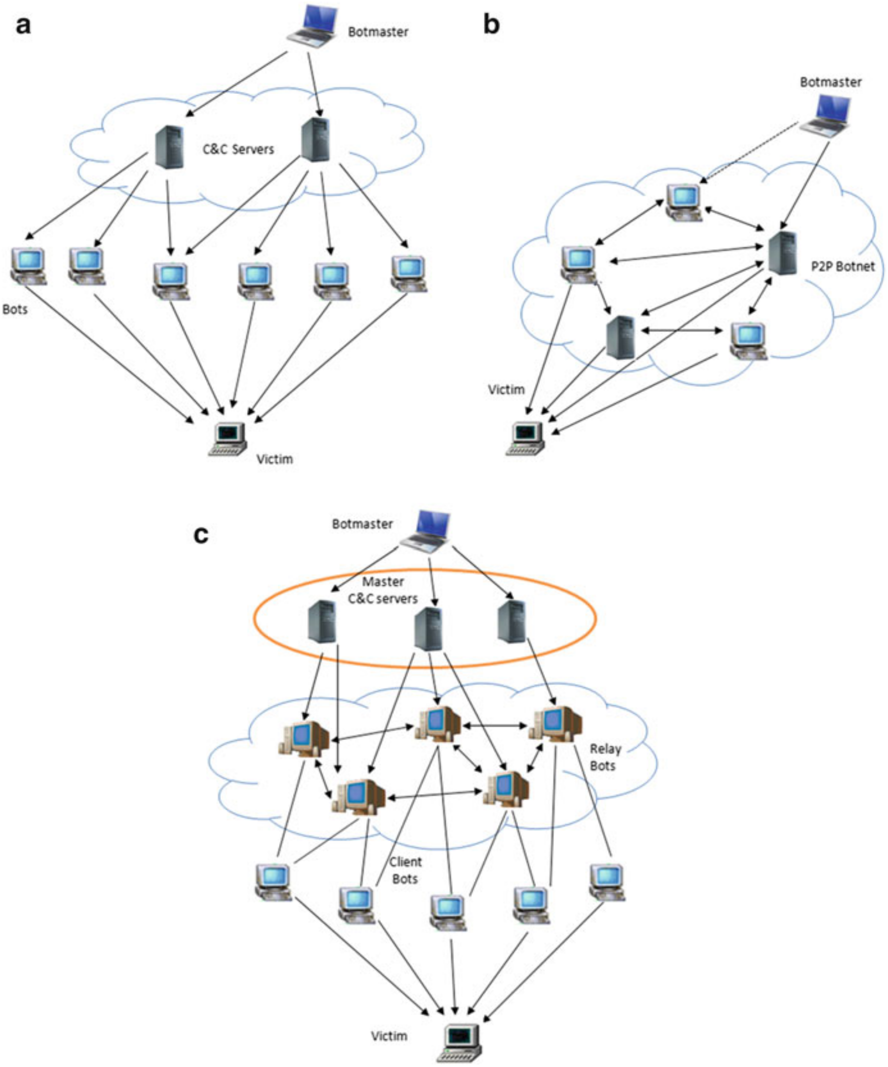

Fig. 8.1 The three types of botnet architectures. (**a**) Centralized (IRC/HTTP) Botnet Model (**b**) Decentralized P2P Botnet Model (**c**) Hybrid Botnet Model

architecture of the centralized botnets. These centralized botnets exhibit similarity in the C&C traffic leading to expose the servers, and thus suffer from a single point of failure [4]. The increased rate of botnet detections propelled the malware authors to develop decentralized botnets to overcome the weakness of single point of failure. Figure 8.1b shows topology of such decentralized P2P botnet model. These distributed botnets are based on the peer to peer (P2P) networks. The distributed P2P architecture leads to complicated network and non-efficient control. So, the hybrid botnets with the desirable features of both centralized and distributed architecture emerged. Figure 8.1c shows the topology of hybrid decentralized P2P botnet model with super peers.

The botnet uses various covert techniques during the different phases of its life cycle [5]. Botnets exploit many integral flaws, vulnerabilities, and social engineering techniques. Everyday numbers of vulnerabilities are exploited in different applications to infect the computers with various malware.

Botnets are differentiated by the C&C of the architecture. The C&C component of botnet architecture is used to control the bots from remote system. This forms the multi-tier architecture of botnets and differentiates them from other malwares. Botnet master compromises systems to set up the C&C servers to issue commands and get the results back from the *bots*. The botnets are usually classified according to their C&C architecture [6, 7]. Over the course of time bot authors have developed C&C channels based on different network structures. Cooke et al. [8] proposed three different botnet communication topologies: centralized (IRC based and HTTP based), decentralized (P2P based), and random.

The botnets employ multilayer architecture which keeps the attacker in the far background and thus makes it difficult to reach to him [9]. In the partially decentralized layered P2P botnets, some peers are controlled by the attacker to issue and disseminate commands/information to other peer bots. The P2P botnets take advantage of the flexible self-organizing network infrastructure, and the peer bots are easily added and replaceable with other hosts.

Botnets can propagate or communicate using either push-based methods or pull-based methods. (1) The push-based methods employ network scanning techniques to find the vulnerable hosts and infect them to turn into a bot. This method reflects the automatic self-propagating nature of the botnets. Conficker [9, 10] and Simda are the well-known examples of this kind of botnets. The push-based method is an active method as it automatically scans and infects new machines to grow the number of victims. (2) In pull-based methods bots propagate with the help of users or other methods, i.e., nonself-propagating methods. In this method bot masters compromise Web servers, upload the malicious codes, and drive users to download the malicious codes (social engineering). These methods also distribute the bot codes via already-infected machines using pay-per-install (PPI) scheme. Unlike the push-based methods, the pull-based methods are passive as these involve human actions or other malware to infect new machines. MegaD and Srizbi botnets employ pull-based method. Recent botnets use pull-based methodology to infect new machines [11]. This shift of botnet propagation from push to pull has made prevention and investigation more difficult. Such botnets are called downloader botnets. BredoLab and Botsniffer are some the examples such botnets.

Botnet forensics is a science that exhibits the process of getting bot clues in order to identify, acquire, analyze, attribute, and mitigate the botnet threats. Botnet forensics gets the root cause of security breaches. This may start from simple evidence for DDoS attack, phishing emails to other malicious information. The forensics provides important insight on how botnets behave in the wild. Further the information obtained may be useful to even identify new instances of botnets. The investigation may take place from the services run by host and suspected or vulnerable activities.

The botnet detection, investigation, and tracking require a forensic environment. The forensic environment uses software tools and techniques to collect digital evidence for understanding how botnets propagate, attack, or perform malice. The information obtained from the forensic investigation is useful for thorough understanding of botnet features and firepower. Further, the information may be used to detect C&C servers and track the attacker. Unfortunately, attackers use anti-forensic techniques and remain anonymous by using number of proxies.

The botnet forensic system is a security device with hardware and pre-installed software. There is a requirement of integrating existing knowledge of botnets and using the live forensic and tracking techniques. Live forensics is the process of imaging-infected machine, documenting the steps when it is running, and collecting all identified evidences without any changes.

Since botnets are the root cause of most of the cyber attacks and threats, the collaborative efforts are required between security firms, IPSs, DNS providers, research groups, and the law enforcement agencies. Further, educating users against social engineering tricks, improving system resilience, and hardening the computer systems are required to prevent and mitigate the effects of such threats. The centralized botnets employ a dedicated C&C infrastructure as a main source or sink of commands and data. Thus, it can be easily spotted and taken down. The decentralized P2P botnets distribute the command channel across multiple infected peers to be more resilient and thus preludes the traditional takedown techniques [12], [13].

The botnet structures have grown manifold in the past years, further, using advanced packing and encryption techniques; therefore the analysis is very challenging. Furthermore, even after detection the takedown operations/mechanisms involve different parties, e.g., registrars or hosting providers. Moreover, the global spread of the botnet needs the permission of various law enforcement agencies, who need correct evidences about the botnet for the required takedown and mitigation operation. Therefore, though botnet forensics is very difficult it is a pressing need against the botnets disrupting the Internet-based economy and the related privacy concerns.

8.3 Acquisition

The data acquisition is the first and important phase in botnet defense. Data acquisition is also part of the botnet detection. Acquisition can be performed at the host and network level. The host-level data acquisition captures the data, logs, and other details from the operating system. The network-level data acquisition may be performed at network interface card or routers level.

The botnet acquisition may involve data collection from the malicious operations and the botnet binary collection for further analysis and investigation. Honeypots are generally used to collect the malware binaries and log the operations. Further, the captured malware samples can be run in the sandboxes to analyze the botnet behavior, tools, and techniques.

Holz et al. [14] introduced a methodology to track botnets. The authors emphasized that in order to track the botnets, some information is required to be gathered by the honeypots. Further, based on this work Cremonini and Riccardi designed a Dorothy framework [15] to monitor the activities of the botnet named as siwa. The authors infiltrated and monitored a botnet to collect the information about the structure, communication, command language, distribution, and functions of the botnet.

The dynamic botnet analysis approaches also employ the well-established honeypot techniques. Honeypot is defined, "An information system resource whose value lies in the illicit use of the resource [16]." In this technique system vulnerabilities are exposed to the attackers and let the systems get infected/compromised and then capture the botnet binary, monitor the operations the malware performs, and log the traffic or information generated. This method also protects actual production systems by sinkholing the attacks/compromise.

The effective deployment of the honeypots forms a collection known as the "honeynet." The honeynet systems collect the suspicious scanning or probing traffic. The deployed honeypots may have low- or high-interaction level to the botnet. The low-interaction honeypot: HoneyD [17]. The high-interaction honeypot is a separate computer system that logs any suspicious activity and also mimics to perform as directed by the botnet master monitoring the applications running on the host and the ports they are communicating. Further, examining the TCP stream of any suspicious connections may help to discover the IP address of the C&C server. The content of the communication can be analyzed to get the details.

Cavalca and Goldoni [18] proposed Honeynet Infrastructure in Virtualized Environment (HIVE), an automated malware collection and analysis architecture. The authors used the architecture to collect various botnet malware to form the botnet code database which is useful/requirement to investigation in the botnet forensics. Further, they suggested possibilities of using the analysis services from the external providers.

Honeypots are successfully deployed in botnet defense and investigation system. Unfortunately, attackers counter all defense measures and develop tools and techniques to evade detection and stay hidden to continue the operations. The advanced botnets have the features to stay stealthy, hidden, and even detect any virtual environment.

Zou and Cunningham [19] presented the honeypot detection method based on the assumption that honeypots deployed for security operation cannot be allowed to participate in the actual malice operation. Attackers detect the honeypot based on the trial whether the compromised host can successfully relay attack commands. These honeypot-aware botnets detect the virtual environments like honeypots and sandboxes before they perform any malice activities. If such environment is detected, botnets exit and/or provide fake information.

Sandbox The execution of the botnet code can present information about the botnet infiltration into the system and the malicious activities like disk read/write and the network communication to get commands, download updates, or other binaries and spamming, etc. The botnet binaries can be run in a controlled environment and monitored in order to log the actions performed by the botnet. The commands executed and the external hosts contacted are monitored, and further analysis is done go get more insight of botnet.

8.4 Analysis

The botnet forensics includes the analysis of various components of the botnets including C&C servers/channels and compromised host. Further, forensic analysis process includes analysis of bots' functionality, C&C servers/traffic, botnet attack, and botnet design.

The previous attacking nature and behavior of botnet helps to identify the intention and method of the attacker. For this purpose analysis is required to obtain the exact facts from the evidence, so that it can mitigate. Analysis classifies and correlates the whole incident into different groups according to their behavior and nomenclature either for mitigating the effect of crime or permanently sorting out the crime. This analysis can be done through the different data mining soft computing tools or different machine learning techniques. TCPDump, Wireshark, TCPFlow, TCPTrace, OllyDbg, IDA Pro, NetFlow, TCPXtract, Snort, etc., are the different tools for supporting the botnet attack analysis.

- Analysis by modeling the botnets
- Analysis of the real botnets captured in the wild

The various botnet forensic analysis techniques can be studied into two broad categories: static and dynamic.

Static Analysis It is the method of understanding the malware behavior without executing it. The method includes analysis of log files, analysis of file systems, and the suspects of malware presence. To gain more insight of the malware, internal structure reverse engineering is performed and has no potential threats to the environment.

Dynamic Analysis This approach is performed by executing the botnet binary in a controlled environment, e.g., a sandbox. This approach studies and monitors the external view, thus, analog to the black-box testing approach. It mainly monitors the behavior and operations of the botnet. This is complementary to the static analysis and performance. The objectives of the approach are to understand the functions and features of the botnet. Often, this is performed in a virtual environment and controls the various parameters to study the behavior at different circumstances. Unfortunately, some botnets are even coded to recognize the virtual environment and thus exit immediately and clear the logs or provide false information.

Botnets can be detected by analyzing their flow characteristics. Several strategies have been proposed to dynamically analyze and defeat botnets. Barford and Yegneswaran [20] performed an in-depth analysis of the malicious bot source programs. The authors found that the botnet architecture and implementation are complex. The authors discovered that the Agobot employs test for debuggers and VMware, kills antiviruses, or provides false information.

Botnet analysis is to enumerate the bots in the botnet. Rossow et al. [13] proposed a graph-based P2P botnet model to capture the properties and vulnerabilities of the botnet. The authors also analyzed the resilience of the botnets to the takedown efforts. Further, they proposed two P2P node enumeration techniques, crawling and sensor injection, for the botnet size measurement. The P2P botnets are susceptive to command injection attacks. Sality botnet employs a peer reputation scheme. The current P2P botnets are quite resilient to disruption attacks.

There is some possibilities through reverse engineering the domain generation algorithm (DGA) for registering the domain prior in which the bot can communicate at any future point which helps to ignore accessing to bot herder, and under their own control, it requests. It gives the different perspective among infected host and botmaster.

Holz et al. [14] proposed the concept for analyzing and monitoring the bot through honeypot technologies. This work gave the immense motivation to the development of Dorothy framework [15]. This framework provides very automated features to analyze, track, and visualize a botnet. Such framework was also made for the requirement of finance-based botnet investigation [21].

Botnet analysis includes two major ways to analyze the malware: examining the code and behavior. The botnet always tries to evade detection and also makes the bots code analysis harder.

(a) *Static Code Analysis:* There are many tools available for the code analysis. Some tools are simple, while others require significant efforts by the investigator, e.g., Hex editor WinHex (a popular tool for static analysis). The investigators require a copy of the malware code for analysis. The malware code may be in the form of a script or compiled binary code. The analysis of the simple script files is easier, while binary compiled files need binary decompiler routine to understand/determine the features and functioning. Unfortunately, uses of some decompilers require strong program understanding, and further, some botnet binaries prevent the use of decompilers. The botnet authors also use packers to minimize the size of the bot binary code, obfuscate binary data, and limit the function of the decompiler. Further, the combination of packing utility with encryption makes reverse engineering more difficult, as well as the effectiveness of the unpacking or decryption tools. The use of uncommon packing utilities even makes the unpacking task more difficult.

The static analysis tools identify the run time errors and security vulnerabilities. The tools also provide the valuable insight and information such as symbol tables, parse trees, and call graphs valuable for botnet analysis.

(b) *Run Time Code Analysis:* Static code analysis is not a complete solution to get botnet evidence. Sometimes static examination of bot binary is unable to analyze bot. To evade from the detection of existence, binary itself uses the different packets and encryption methodology. it may be necessary to execute the malware to monitor the actions that take place on the victim system, such as file system changes, registry changes, and network activity for gathering additional information of malware behavior. Run time code analysis is particularly helpful to identify such botnet network information where is the location of the bot connection to receive commands, as well as any usernames, handles, IRC channels, and passwords.

An important step of forensic investigation is preserving the evidence of any cybercrime for prosecution. The malware analysis is a considered to be the main activity of digital forensics. The analysis can further classified into following ways:

Spam-Based Analysis

Compromised system is that which used to send spam messages. The advantage of performing spam using botnet has reliance as even if we can identify a bot sending and are able to block it, still there will be other bots that will still be performing spam. For example, Rustock botnet was used for spamming and was sending 25000 messages per host per hour. So the magnitude of spam is huge when using a botnet.

Pathak et al. [22] did a comprehensive study of content agnostic characteristics of spam campaign. During the collection and analysis, non-proxy spamming domains were observed to exhibit spamming duration far longer than a five-day period whose effect was studied on spam campaign signature generation. Further analysis revealed workload distribution, sending patterns, and coordination among the spamming machines.

Pitsillidis et al. [23] described the mechanism for better filtering of spam by analyzing the vantage points of a spammer. By monitoring botnet host, we are able to identify new spam as it is created, and later we can create proper strategies to deal with such spams. This technique gave precise decisions with no false positive.

Distributed Denial of Service (DDoS)-Based Analysis

Freiling et al. [24] describe a prevention mechanism, which operates by infiltrating and analyzing mechanism of the remote controlling the bots. The method can be used over the Internet and infiltrate mostly IRC bots which is the most common type of botnet architecture used by botmasters.

Thomas et al. [25] explore the Koobface zombie infrastructure and analyzes its effect. It was discovered that despite domain blacklisting service by social network, over 213000 users were compromised, generating over 157000 clicks.

Provos et al. [16] presented a state of *malware and Web and emphasized* on the installation of malware which can be easily installed by analyzing the vulnerabilities of the host once it clicks on the malicious link.

Fast Flux-Based Analysis

Passerini et al. [26] developed a system named FluXOR to detect and monitor fast flux service networks. The detection totally relies on the analysis of a set of features observable from a view of a victim of a scam. Nazario et al. [27] established the fact that active lifetime of fast flux botnet varies from less than one day to months. The domains used for fast flux are registered months before they are used and kept as dormant.

The authors in [28] performed detailed technical analysis of the Festi botnet and discovered the distinguishing features of the botnet. The botnet implement object-oriented architecture into the kernel-mode driver to make it portable. The botnet also exhibits strong resistance to forensic analysis and has the ability to bypass IDS/IPS software tools.

Traffic-Based Analysis

In the Internet, we receive the data in the form of packets. The number of packets we send and receive from the Internet is called network traffic. Network traffic exhibits the packets we receive and sends to the destination. Broadly the traffic can be categorized in two ways: simple traffic and active traffic.

Simple Traffic In simple traffic, the timely delivery and quality of services are confirmed and given priority. The source of the traffic has expectation to deliver the packets in time. The traffic is also known as sensitive traffic. The examples include VoIP, video conferencing, online gaming and Web browsing, etc.

Active Traffic In this category the traffic ensures the quality of services with the speed. Active traffic is not sensitive to quality of service metrics such as jitter, packet loss, latency, etc. Sometimes we receive unwanted traffic which carries worms, botnet, or malicious activities. The Internet is full of malicious activities such as phishing, Denial of Services, click fraud, spamming, *etc., so therefore we* need to analyze the botnet traffic.

Botnet traffic is a process of generating, recording, reviewing, and analyzing the botnet traffic for the purpose of acquiring, identifying, detecting, and mitigating the botnet attacks. This is the process of using manual and automated technique to review cluster detail and statistics within botnet traffic. Botnet traffic analysis is done through bandwidth monitoring software tools. Traffic statistics refer and help in following:

- Understanding the botnet and utility
- Evaluating the botnet
- Downloading and uploading the speeds
- The content, size, source, and destination of the information
- Identifying malicious and suspicious packets, etc.

(i) *Command-and-control-based traffic analysis*

Masud et al. [29] propose a temporal correlation technique for C&C traffic detection. Using the temporal correlation of two host-based log files, the author-illustrated bots react faster than the human operators. The authors applied this technique in log files for detecting the bot activity in a system using TCPDump and exedump. This tool records inflow and outflow network traffic packet and the start time of the application execution at the host machine, respectively. The authors apply data mining to extract relevant features from the log files and detect the C&C traffic.

(ii) *P2P-based traffic analysis*

By analyzing Waledac botnet, Dae-il et al. [30] proposed their study on infected HTTP2P botnet and also facilitated their detection. In order to breach the network security, infected botnet changed the protocol. In the beginning botnet used only IRC protocol. The botnet suggested by Dae-il et al. utilized multiple protocols which include TCP,UDP, HTTP, and so on. The infected bots utilize combination of protocols for instance in case of HTTP2P, i.e., P2P over HTTP. In case of HTTP2P, HTTP protocol enjoys the merits being firewall friendly, whereas P2P protocol is helpful in evading a client and server architecture. This prompted the author for proceeding further toward analyzing the characteristics of the phases of botnet behavior communicating and utilizing the HTTP2P protocol. For the study the author classified Waledac botnet into two categories, they were proxybot and workbot. The study results facilitated the detection of HTTP2P botnet in the network traffic.

Dafan et al. [31] studied the different phases in between the general peer to peer protocol and advance peer to peer protocol. The attacker hardcodes a search key in their bot program, which looks for the order command for future attack with the search key on regular time intervals. On the grounds of unstructured peer to peer protocols, the author has designed a very upgraded hybrid peer to peer botnet. He mentioned the different requirements for the peer to peer protocol and showed general peer to peer protocol does not require global information.

(iii) *IRC-based traffic analysis*

Mazzariello et al. [32] addressed centralized botnet detection. C&C structure provides the simplicity to create attacking scenario by the bot herder. Once C&C channel is identified, the whole botnet can be dismantled. He experimented and found that the known bots are characterized by the propagation mechanism. This tendency comes after inheriting the same strategies and characterizing by the next bot from the popular bot.

Karasaridis et al. [33] designed to measure the distance between monitored flow data and predefined IRC traffic flow.

(iv) *Flow-based traffic*

Shahrestani et al. [34] analyze the current network intrusion detection method. This method depends upon anomaly detection and passed from the flow-based botnet detection system to check trustfulness. Through visualization, it is then aggregated to reveal malicious traffic. Finally this information is forwarded for validation.

(v) *DNS network traffic*

Thomas et al. [35] designed for DNS-based detection. He described and analyzed the tracking and analysis for P2P version-2. His experiment captured result based on DNS and data hash list size. After maintaining large hash lists, the results explained the ability of TRAPP-2 to detect traffic under a saturated network load. They analyzed the DNS traffic to identify the malware family without the need for obtaining malware sample. He made the cluster of DNS traffic through different infected machine.

8.5 Attribution

To understand the functioning of botnet is important for attribution of the botnet. A typical botnet has several components and follows a different file cycle. Botnet forensics is forced to show the different communication paths between an infected system and initial point of attack origin. Botnet forensics uses incident response, prosecution, and the tough measurement of the botnet to identify the attacker. The attacker uses different techniques such as IP spoofing and stepping stone attack to hide himself. For this purpose attribution is required to find out the evidence and the kind of attack.

Attribution shows collection of the comprehensive information of botnet samples, further deploys honeypots, and traces the details about the developers.

Botnet attribution is done for various operations and purposes such as botnet size measurement, nodes classification and locating C&C servers, and even tracing botmasters. Rossow et al. [13] investigated botnet for its size measurement. The botnet investigation may start with collection of samples or passive monitoring of bots behavior.

The botnet attribution processes are the study of propagation, C&C structure, communications protocols, attacks, and victim investigation. Botmasters may also periodically perform the query to check DNS Blacklists (DNSBL) to check if their bots are listed in the blacklist. Ramachandran et al. [36] proposed a passive analysis of the DNS monitoring activity of the attackers to detect the bot nodes.

The botnet attribution can be performed in two major ways. (a) Capture the botnet binaries, collect the real botnets traffic in the wild, and perform various analysis methods to understand the botnet. (b) Run the known/unknown botnet binary in a controlled environment to monitor the behavior and log its traffic. The investigator is able to control all the variables in a controlled environment, but, this leads to a trade-off between the level of control and the realistic behavior of the malware. Such experiments help to dissect the behavior of the botnet.

An investigation of BredoLab botnet by National High Tech Crime Unit of the Netherlands' Police Agency (NHTCU) in 2010 estimated the three million infected machines and then on October 25, 2010 got access to hosting server in the Netherlands and successfully took over the botnet.

Cusack (2014) [11] concluded that botnet investigation is a complex process, and controlling the cost of botnet investigation is critical. Therefore, technical processes are required to be automated and to control the time and cost resource. He emphasized that the success of the integrated investigation framework will depend on the comprehensive centralized database maintained by the stakeholders.

Botnet attribution is a complex operation and requires significant time and efforts. Therefore, expert knowledge, automation, collaboration, and sharing tools and techniques are required for effective investigation and analysis.

Botnet attribution covers all the components of the botnet including the victim investigation, infected host investigation, network traffic, and C&C server.

The C&C server is the most important component in botnet which may serve in many ways in different botnets. The C&C server may host commands, spam templates, stolen email ids, etc.

Network-Based Attribution

Network-based attribution is primarily based on traffic monitoring, detecting C&C servers, phishing Web sites, and analysis of traffic. This attribution is the botnet attribution specific on IRC, HTTP, DNS, P2P, mobile, cloud, etc.

Host-Based Attribution

However, many researchers proposed botnet attribution at the network level, but botnet attribution at the infected host level is also important and has potential advantages. Practically, both approaches are relevant and in fact complementary to each other.

Law et al. [37] used the host-based botnet investigation approach, and further, the authors highlighted that the host-based approaches are easier than network-based approaches requiring large traffic log storage and flow monitoring as the required data collation is less.

Ard et al. [38] defined botnet attribution's two phases as: (1) The analysis of the malware shows the postmortem of binaries. This kind of attribution is also known as run time analysis for identifying the network and its information. (2) The second phase of attribution is tracking sources for identifying the DNS name registers, the IRC controllers, and servers. The authors presented the study of basic analysis techniques for reverse engineering botnets. Some P2P botnets also use custom protocols which make reverse engineering to get an insight of the botnet [13].

Cusack [11] proposed the botnet forensic techniques. The authors applied the various proposed investigation guidelines and concluded that it is possible and feasible to investigate botnet attacks, but controlling the cost of investigation is critical. The author further recommended quantifying botnet investigations into five levels of cost: based on time, complexity, and technical requirements.

Table 8.1 illustrates the botnet forensic tools used at the data collection and analysis phases.

Obstacles in Botnet Attribution

Botmaster performs recon and even anti-recon actions to continue their illegal services. They even may perform DNSBL reconnaissance queries to check if their services are blocked [36]. Botmasters also implement anti-recon strategies in response to the takedown attempts by reconnaissance. The anti-recon strategies may include attacks such as DDoS against recon nodes, automatic black listing nd reputation schemes [12].

Different security agencies such as FBI and Microsoft have taken step and experimented to reduce Zeus botnet family's threat.

Botnet forensics follows that the strategy of prevention is better than attack for its defensive approach. Botnet forensics provides and spends attacker's most of the time and energy to maintain their route through trapping network so that he can get the least time to launch the attack. These restrictions improve the security issues and reduce the crime rate. Thus criminal cannot harm or penetrate the real network very easily. In order to ensure the network security and to fix the accountability, the

Table 8.1 The tools used for botnet forensics

Process	Tools	Purpose
Malware collection	Dionaea	A low-interaction honeypot that collects a copy of the malware exploiting vulnerabilities
Virtualization	VMware workstation, Oracle Virtual Box	Tools for visualizing the computer system
Forensic Image	Helix Pro	A forensic tool for incident response
Memory analysis	Volatility framework	A forensic tool that can extract various types of information from a memory image
Initial virus scan	Virus total	A public service that analyzes suspicious files and URLs
Initial sandbox analysis	Anubis, CWSandbox	Public services that analyze the behavior of Windows PE executable with special focus on the analysis of malware
Packer detectors	PEiD v 0.94	A tool for detection of packers, cryptors, and compilers for Windows PE executable
String extractor	BinText v3.03	This tool search ASCII, Unicode, and resource strings in a file
Dissemblers and debuggers	IDA Pro, OllyDbg	Reverse engineering tool

Internet service providers are made responsible for the activities on their networks. Further in this direction one more significant step is taken, now it is mandatory for the firms engaged in e-commerce and online business to reveal their security breaches and disclosure of majors taken to ensure the network security as a part of statuary compliance so that the scale of harm, damage or loss it caused, share of bandwidth used, and traffic load on the network can be minimized.

Botnet measurement is useful to calculate the size of the botnet, estimate the growth and predict the future changes in the botnet model. The results of botnet measurement help to design better prediction or defense systems against present and future botnet threats.

The forensic attribution is very important to better understand the ever evolving botnet picture and threat. Botnet developers use continuously evolving technology to increase the stealth and covertness of the botnets. Therefore, united efforts are required to mitigate the botnets at the successive phases.

Botnet developers use the full potential/advantage of interconnectivity of the Internet to infect and undertake large numbers of computers from all over the world. Unfortunately, this is a major hurdle in botnet defense and investigation.

Botnet forensic is very essential to gather the information about botnets and analyze how they propagate, behave along protection and detection systems, and perform malicious actions. The thorough understating of botnets is further useful to develop the more effective protection systems against botnet threats.

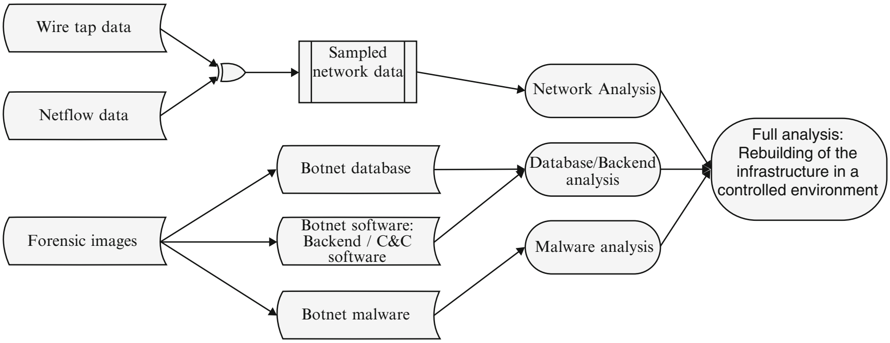
Wire tap data
Netflow data
Sampled network data
Network Analysis
Forensic images
Botnet database
Botnet software: Backend / C&C software
Botnet malware
Database/Backend analysis
Malware analysis
Full analysis: Rebuilding of the infrastructure in a controlled environment

Fig. 8.2 Botnet forensic attribution model [39]

Botnet forensic framework is inclusive of botnet identification, bot classification, bot analysis, and identification of potential bot attacks. The success of any botnet attack attribution requires the effective extraction of features, behavior, signature, and protocol from the different networks.

The risks and challenges are involved while executing the malware binaries in the controlled environment. In the system/process of the risk management, one key functionality is to protect the integrity of the data collected/evidence from the infected system or suspected traffic and to ensure the security of the investigator information system/tools.

Graaf et al. [39] proposed a botnet forensic investigation model to investigate and analyze botnet called BredoLab (first discovered in 2009). The authors identified that the traditional forensic investigation models were not much effective to investigate and analyze the large BredoLab botnet resources. BredoLab serves as large malware downloading platform and is also used for installing malware to third parties, i.e., used for pay-per-install (PPI). He performed the forensic investigation of the BredoLab botnet and discovered that its infrastructure consisted of C&C servers, a central proxy server, several proxies installed on bots, database server, a back-end server, a personal hacking server, and a VPN server (Fig. 8.2).

8.6 Research Challenges

The analysis of the botnets is not an easy task due to the hybrid and ever-changing nature of botnets. There are many challenges and issues with botnet forensic analysis and investigation. The identified challenges are:

- The botnets continuously advance in the propagation techniques including push based, pull based, and drive-by download to defeat to botnet prevention measures.
- Some bots are even preprogrammed to destroy them if suspected to be caught by defenders and clear the evidences. The bot authors also use many anti-forensic

techniques to challenge the analysis and investigation processes. The botnets even clear the logs once the task is over or suspect the presence of any virtual machine/honeypot.

- The various reports disclose that the botnets adapt in response to the defense deployed by the security fronts and further explore new technologies such as mobile computing, cloud computing, and even Internet of Things (IoT) and mobile botnets [40].
- A proper forensic environment is required to collect data and analyze it. Researchers and security persons use honeynets to attract the botnet compromise and log the tools and techniques used by the botnets. Unfortunately, bot authors use anti-honeypot technique to ignore the honeynets.
- The botnet authors use covert communication by hiding the traffic from the victim users and even the networks administrators.
- None of the studies purely focused on the botnet forensics and discovered the anti-forensic techniques. Therefore, the future research would also focus on developing research methods for botnet investigation which are more effective, direct, and systematic.

8.7 Conclusion

The botnets are the main source of large number of cyber threats and attacks. Botnets cause huge financial, social, and privacy damage all across the world. To combat the botnet threats, integrated and shared repositories are required to be built for standard references. Some countries and banking association are working to build the blacklist of infected IPs. The chapter explains the effectiveness of various botnet forensic acquisition, analysis, and attribution techniques.

Tackling botnets requires a collaborative effort between researchers, ISPs and DNS providers, law enforcer, and self-organized security communities such as shadow server. The chapter may be useful to design standard botnet forensic tools and conduct the botnet investigation research in a systematic way. Further, it provides guidelines for hardening the computing system, educating users, and for improving the resilience of the systems to further attack.

8.8 Questions

Multiple Choice Questions

1. The phase is not part of the botnet forensic framework.

    (a) Investigation
    (b) Identification
    (c) Analysis
    (d) None of the above

2. In botnet forensic attribution model cover.

   (a) Malware analysis
   (b) Attribution
   (c) Acquisition
   (d) None of the above

3. Analysis of botnet forensics not covered.

   (a) Static analysis
   (b) Run time code
   (c) Dynamic analysis
   (d) None of the above

4. Which of the following is not a part of the botnet life cycle?

   (a) Infection
   (b) Invocation
   (c) Communication
   (d) Attack

5. Which of the following statement is incorrect?

   (a) IRC-based botnets are prone to detection due to centralized server.
   (b) HTTP-based botnets are prone to detection due to centralized server.
   (c) P2P-based botnets are prone to detection due to centralized server.
   (d) None of the above

6. Fast-flux mechanism is used by botnets authors for ________

   (a) Ensure the availability botnet service
   (b) Make the botnets more resilient
   (c) Both A and B options
   (d) None of the above options

7. Which of the following is/are used for data acquisition?

   (a) Honeypots
   (b) Sandboxes
   (c) Both (A) and (B) options
   (d) None of these options

8. Which of the following is part of the botnet forensic framework?

   (a) Investigation
   (b) Identification
   (c) Analysis
   (d) All of the above

9. The botnet forensic attribution model covers ____________
   (a) Malware analysis
   (b) Attribution
   (c) Acquisition
   (d) None of the above
10. Analysis of botnet forensics do not cover ___________
   (a) Static analysis
   (b) Run time code
   (c) Dynamic analysis
   (d) None of the above

Short-Answer Type Questions

1. What is botnet?
2. What are the various threats posed by botnets?
3. Explain the significance of botnet forensics.
4. Elaborate different phases in investigation of botnet attacks.
5. What is difference between run time code and static code analysis?

Long-Answer Type Questions

1. Explain the different botnet architectures.
2. Discuss the requirement of botnet forensics.
3. Explain the process of data acquisition step of botnet forensics.
4. Discuss the botnet process model in details.
5. What are the various tools used botnet forensics. Explain in details.

References

1. Wang P et al (2010) Honeypot detection in advanced botnet attacks. Int J Inf Comput Secur (IJICS) 4(1):30–51
2. Stevenson A (2014) Botnets infecting 18 systems per second, warns FBI. July 16, 2014 [cited 2015 9 March 2015]; Available from: http://www.v3.co.uk/v3-uk/news/2355596/botnets-infecting-18-systems-per-second-warns-fbi, 31 Mar 2016
3. Rajab MA et al (2006) A multifaceted approach to understanding the botnet phenomenon. In: Proceedings of the 6th ACM SIGCOMM conference on internet measurement (IMC'06), ACM, Rio de Janeiro, Brazil
4. Grizzard JB et al (2007) Peer-to-peer botnets: overview and case study. In: Proceedings of first workshop on hot topics in understanding botnets (HotBots'07), USENIX Association, Cambridge, MA, pp 1–8
5. Rodríguez-Gómez RA, Maciá-Fernández G, García-Teodoro P (2013) Survey and taxonomy of botnet research through life-cycle. ACM Comput Surv (CSUR) 45(4):45
6. Zhu Z et al (2008) Botnet Research Survey. In: 32nd annual IEEE international computer software and applications (COMPSAC'08)
7. Feily M, Shahrestani A, Ramadass S (2009) A survey of botnet and botnet detection. In: Third international conference on emerging security information, systems and technologies (SECURWARE'09). IEEE

8. Cooke E, Jahanian F, McPherson D (2005) The Zombie roundup: understanding, detecting, and disrupting botnets. In: Proceedings of the USENIX workshop on steps to reducing unwanted traffic on the internet (SRUTI '05). Boston: USENIX Association, Berkeley, CA
9. Seungwon S et al (2012) A large-scale empirical study of conficker. IEEE Trans Inf Forensics Secur 7(2):676–690
10. Fitzgibbon N, Wood M (2009) Conficker. C: a technical analysis. SophosLabs, Sophon Inc
11. Cusack B (2014) Botnet forensic investigation techniques and cost evaluation. In: Proceedings of the conference on digital forensics, security and law
12. Andriesse D, Rossow C, Bos H (2015) Reliable Recon in adversarial peer-to-peer botnets
13. Rossow C et al (2013) SoK: P2PWNED – modeling and evaluating the resilience of peer-to-peer botnets. In: IEEE symposium on security and privacy (SP)
14. Bacher P et al (2005) Know your enemy: tracking botnets. In: The Honeynet Project & Research Alliance
15. Cremonini M, Riccardi M (2009) The Dorothy project: an open botnet analysis framework for automatic tracking and activity visualization. In: European conference on computer network defense (EC2ND)
16. Provos N, Holz T (2007) Virtual honeypots: from botnet tracking to intrusion detection. Addison-Wesley Professional
17. Provos N (2003) Honeyd-a virtual honeypot daemon. In: 10th DFN-CERT workshop, Hamburg, Germany
18. An open architecture for distributed malware collection and analysis. (2010)
19. Zou CC, Cunningham R (2006) Honeypot-Aware advanced botnet construction and maintenance. In: International conference on dependable systems and networks (DSN '06)
20. Barford P, Yegneswaran V (2007) An inside look at botnets. In: Christodorescu M et al (eds) Malware detection- advances in information security. Springer US, pp 171–191
21. Riccardi M et al (2010) A framework for financial botnet analysis. In: eCrime Researchers Summit (eCrime), 2010
22. Pathak A et al (2009) Botnet spam campaigns can be long lasting: evidence, implications, and analysis. ACM
23. Pitsillidis A et al. Botnet judo: fighting spam with itself
24. Freiling F, Holz T, Wicherski G (2005) Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks. Computer Securityâ€"ESORICS 2005, pp 319–335
25. Thomas K, Nicol DM. The Koobface botnet and the rise of social malware. IEEE
26. Passerini E et al (2008) Fluxor: detecting and monitoring fast-flux service networks. In: Detection of intrusions and Malware, and vulnerability assessment (DIMVA'08), Lecture Notes in Computer Science
27. Nazario J, Holz T (2008) As the net churns: fast-flux botnet observations. In: 3rd international conference on Malicious and unwanted software (MALWARE '08), Alexandria, VA
28. Matrosov A, Rodionov E (2011) Festi botnet analysis & investigation
29. Masud MM et al (2008) Flow-based identification of botnet traffic by mining multiple log files. IEEE.
30. Dae-il J et al (2009) Analysis of HTTP2P botnet: case study waledac. In: Communications (MICC), 2009 IEEE 9th Malaysia International conference on
31. Dafan D et al (2008) Deep analysis of intending peer-to-peer botnet. In: Grid and cooperative computing, 2008. GCC '08. Seventh international conference on
32. Mazzariello C (2008) IRC traffic analysis for botnet detection. Ieee
33. Karasaridis A, Rexroad B, Hoeflin D (2007) Wide-scale botnet detection and characterization. In: Proceedings of the first conference on first workshop on hot topics in understanding botnets. Cambridge, MA
34. Shahrestani A et al (2009) Architecture for applying data mining and visualization on network flow for botnet traffic detection. In: Computer technology and development, 2009. ICCTD '09. International conference on

35. Thomas B et al (2011) An FPGA system for detecting malicious DNS network traffic advances in digital forensics VII. Springer, Boston, pp 195–207
36. Ramachandran A, Feamster N, Dagon D (2006) Revealing botnet membership using DNSBL counter-intelligence. In: Proceedings of the 2nd workshop on steps to reducing unwanted traffic on the internet (SRUTI'06), San Jose, California, USA
37. Law FYW et al (2010) A host-based approach to BotNet investigation? In: Goel S et al (eds) Digital forensics and cyber crime. Springer, Berlin/Heidelberg, pp 161–170
38. Ard C (2007) Botnet analysis. Int J Forensic Comput Sci 2(1):65–74
39. de Graaf D, Shosha A, Gladyshev P (2013) BREDOLAB: shopping in the cybercrime underworld. In: Rogers M, Seigfried-Spellar K (eds) Digital forensics and cyber crime. Springer, Berlin/Heidelberg, pp 302–313
40. Vural I et al (2010) Mobile botnet detection using network forensics. In: Future internet – FIS. Springer, Berlin/Heidelberg, pp 57–67