Chapter 16

# BotNet Detection: Enhancing Analysis by Using Data Mining Techniques

Erdem Alparslan, Adem Karahoca and Dilek Karahoca

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48804

## 1. Introduction

Recent years revealed that computers are not used only for scientific and business oriented purposes. Individuals of diverse ages, lifestyles, educations and psychologies are living more and more in a virtual reality. This virtual reality affects person's daily activities and habits. In the past, individuals have been used computers only to access knowledge. But nowadays they not only access knowledge, but also share their lives, make money, give or diffuse their opinions and act social. Computers are interfaces for individuals in their virtual social lives. The Internet is the living place of this virtual sociality with its opportunities, capabilities and facilities but also with threats. As the popularity of the Internet increases, the number of attackers who abuse the NET for their nefarious purposes also increases.

The increasing capability of detecting suspicious Internet activities oriented the attackers to a different and sophisticated attack methodology. Coordinated attacks are the attacks realized by more than one, related and co-influenced computer nodes. They make the attackers available to behave in an untraceable Internet activity. The untraceable feature of coordinated attacks is just what hackers/attackers demand to compromise a computer or a network for their illegal activities. Once an attack is initiated by a group of computer nodes having different locations controlled by a malicious individual or controller, it may be very hard to trace back to the origin due to the complexity of the Internet. For this reason, the growing size of events and threats against legitimate Internet activities such as information leakage, click fraud, denial of service (DoS) and attack, E-mail spam, etc., has become a very serious problem nowadays (Liu, Xiao, Ghaboosi, Hongmei, & Zhang, 2009).

The coordinated network attacks are realized by using infected victim computers. Those victims controlled by coordinated attackers are called zombies or bots which derives from the word “robot.” The term “bot” is the general terminology of the software applications running automated tasks over the Internet (“Wikipedia - Internet Bot,” n.d.). Botnet is a self-

propagating, self-organizing, and autonomous framework that is under a command and control (C2 or C&C) infrastructure. Generally, to compromise a series of systems, the botnet's master (also called as herder or perpetrator) will remotely control bots to install worms, Trojan horses, or backdoors on them. Majority of those zombie computers are running Microsoft Windows operating systems. The process of stealing host resources to form a Botnet is so called "scrumping"(Liu et al., 2009; "Wikipedia - Botnet," n.d.).

Because of their network-based, coordinated and controlled nature Botnets are one of the most dangerous species of the Internet attacks nowadays. Deriving their power both in their cumulative bandwidth and their access capabilities, botnets can cause severe network outages through massive distributed denial-of-service attacks. The threat of this outage can cost enterprises large amounts in extortion fees (Strayer, Lapsely, & Walsh, 2008). According to the recent Symantec's research report, botnets have become one of the biggest security threats. According to the report a large volume of malicious activities from distributed-denial-of-service (DDoS) attacks to spamming, phishing, identity theft and DNS server spoofing can be realized by using the distributed power of botnets (W. Lu, Rammidi, & Ghorbani, 2011). Also one of the largest spam filter companies, SpamHaus,2 has estimated that already in 2004, 70% of spam was sent out via such networks (NISCC, 2005) (Seewald & Gangsterer, 2010). According to the US FBI and public trackers, at least a million bots are known to exist like ShadowServer and the true number is likely to be much higher. The number of bots is also still growing at an exponential rate (Seewald & Gangsterer, 2010).

Botnets are as powerful as they can diffuse on new hosts by infecting them via well-known security holes. One can be infected by clicking a link on a website, opening an attachment of an e-mail or only viewing them, surfing on a website by a browser which has a security weakness. Secure computers are not exactly protected from botnet dissemination. 0-day vulnerabilities are used to attack secured / patched computers. There are some indications that botnet operators invest in R&D to find specific zero-day vulnerabilities, aiming at exploiting them at leisure.

## 2. Classification of Botnets

Because of their distributed architecture, botnets are quietly different from other types of malwares. Botnets can spread over millions of computers as worms can do. Unlike worms, zombie nodes in a botnet can work cooperated and be managed from a command and control center. Because of this distributed architecture, botnets cannot be classified as other malware types. Many works try to summarize the taxonomy of botnets. The main classification areas of botnets are the topology of C&C architecture used, the propagation mechanism, the exploitation strategy and available set of commands used by perpetrator.

### 2.1. Classification based on C&C topology

Command and control topologies of botnets are studied in various researches aiming to detect preventive measures for each kind of infrastructure. Detecting the organization of a malicious network may help preventing them.

IRC based botnets are the preliminary types of botnets which are still effective and usable for attackers. IRC is a text based instant messaging protocol over the Internet. It works on client-server architecture but it is also suitable for distributed environments. In most cases interconnected IRC servers communicate each other and each has own subscribers. Thus, a subscriber on an IRC server may communicate with others if IRC servers are interconnected and are on the same channel. This interconnection between the IRC servers is called multiple IRC (mIRC). IRC-based bots use this infrastructure for malicious purposes by managing access lists, moving malicious files, sharing clients, sharing channel information and so on. A typical IRC based botnet is shown in Fig. 1 Victim machine is the compromised internet host which runs the executable bot triggered by a specific command from IRC server. Once a bot is installed on a victim host, it will make a copy into a configurable directory and let the malicious program to start with the operating system. A secured channel set up by the attacker to manage all the bots is called control channel. IRC server may be a compromised machine or even a legitimate service provider. Attacker is the one controls botnet. As in Fig. 1 attacker opens a private IRC channel on an ordinary IRC server. After spreading malwares on victim computers attacker waits bots to subscribe his own private IRC channel. Then he gives commands and controls the botnet infrastructures for his malicious purposes (Puri, 2003).

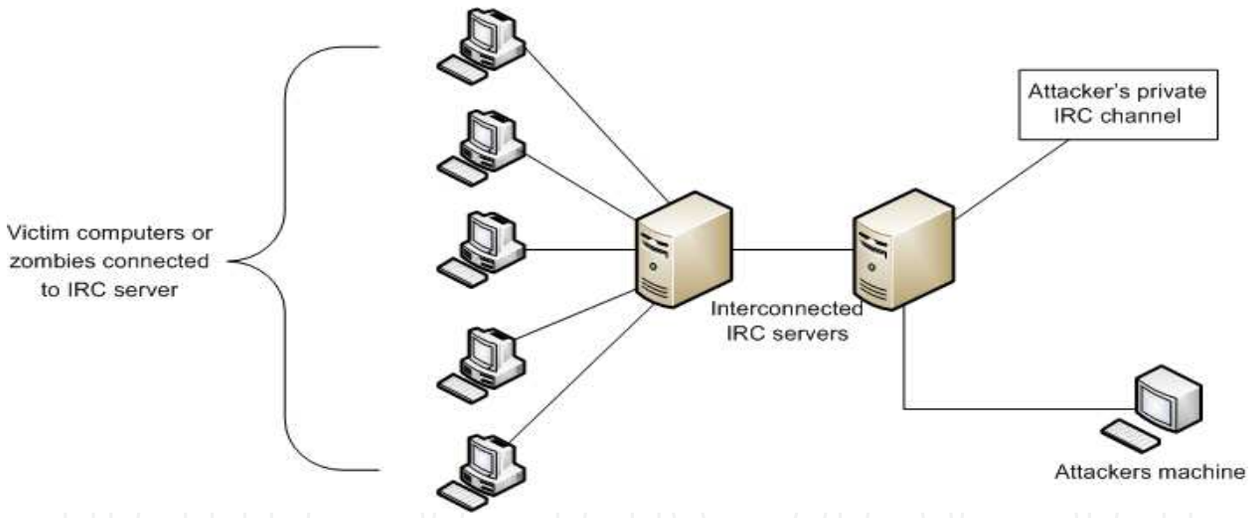


**Figure 1.** IRC-based botnets (Feily, Shahrestani, & Ramadass, 2009)

However the majority of botnet studies focus on IRC based C&C architecture, P2P based C&C architecture can spread easier and hide itself from intrusion detection techniques. In fact, using P2P networks to control victim hosts is not a novel technique. A P2P spreading worm named Slapper, infected Linux system by DoS attack in 2002. One year after, another

P2P-based bot, Dubbed Sinit appeared. In 2004, Phatbot was using P2P system to send commands to the other compromised hosts. Currently, Storm Worm (Holz, M, & Dahl, 2008) may be the most wide-spread P2P bot over the Internet. Many P2P networks have a central server or a list of peers who can be contacted to add a new peer. Centralized nature of this kind of P2P networks requires a bootstrap procedure which presents a weakness for P2P

networks. To overcome this problem authors in (P. Wang, Sparks, & Cou, 2008) presented specific hybrid P2P botnet architecture. Hybrid P2P botnet architecture has servant and client bots who behave as clients and as servers in a traditional P2P file sharing network. Servant bots are connected to each other and form the backbone structure of the botnet. An attacker or botmaster can inject his commands into any hosts of the botnet. Each bot knows only its directed neighbors and transmits the command to its neighbors. If one bot is detected by intrusion detection systems only its neighbors are affected. The hybrid architecture for P2P botnets delivers some new capabilities: (1) it requires no bootstrap procedure; (2) only a limited number of bots nearby the captured one can be exposed; (3) an attacker can easily manage the entire botnet by issuing a single command (Liu et al., 2009).

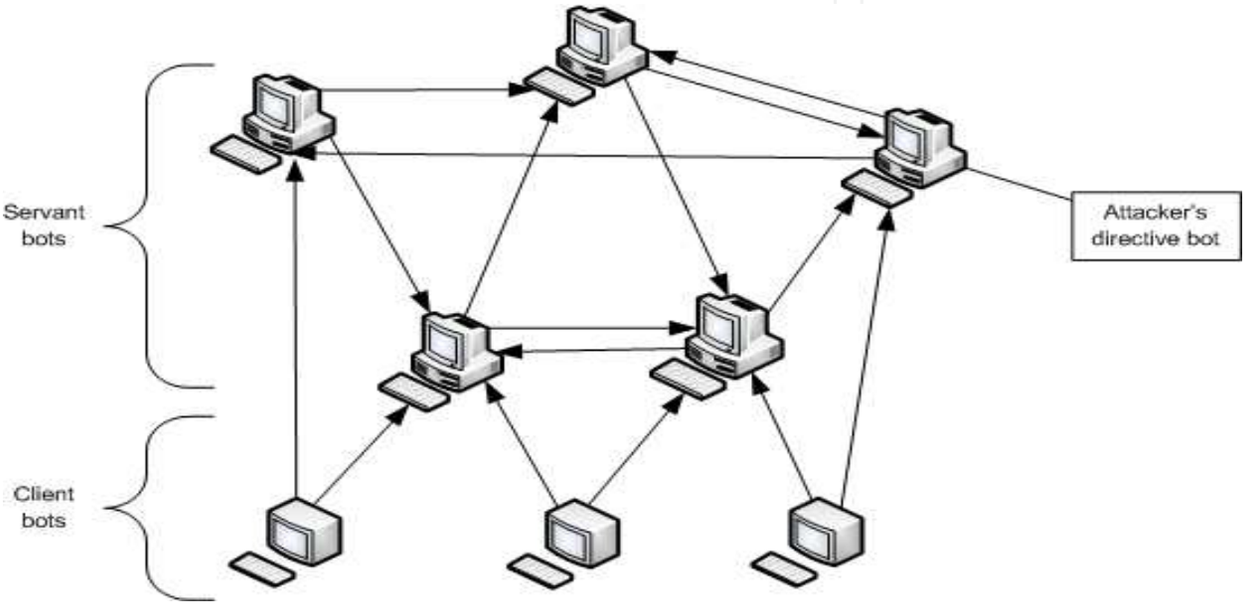


**Figure 2.** P2P-based botnets(P. Wang et al., 2008)

Another type of C&C mechanism widely used is http-based botnets. In http-based botnets, bots and C&C center communicate each other by using http protocol in an encrypted communication channel. In Chiang&Lloyd (2007), an http-based spam bot module in Rustock rootkit is analyzed by using a well-known analysis tool IDA Pro to find the encryption key. The paper summarizes that a typical routine for the spam bot to send a spam is as following:

i. The bot asks the controller for local processes/files to kill and delete.
ii. The controller sends back system information.
iii. The bot asks for SMTP servers.
iv. The bot gets failure responses from the SMTP servers.
v. The bot gets spam message
vi. The bot gets target email addresses.

In Nazario (2007), an HTTP-based DDoS bot, BlackEnergy is analyzed. The bot is only used for DDoS attacks. However, the bot does not have any exploit activities, so it cannot be captured by Honeynet.

Clickbot, a low-noise click fraud bot is discussed in (Daswani & Stoppelman, 2007). Clickbot propogates its client side malware by e-mail attachments. The bot also use http protocol for command and control (Zhu et al., 2008).

## 3. Botnet attacks

Botnets are often used for DDoS attacks. Because of their distributed and hard to detect nature, denial of service attacks can be impressively applied by using botnets. Besides, botnets are also used to perform spamming, malware spreading, sensitive information leakage, identity fraud, click fraud. They are very valuable instruments of getting Advanced Persistent Threats (APT) for critical organizations.

"Denial of Service" (DoS) attacks are very powerful threats for organizations. They are inevitable when performed by a distributed environment, so called Distributed DoS or DDoS. Botnets are often used for DDoS attacks to consume network bandwidth of victim system from wide range of IP addresses. The victim system cannot add source IP addresses to the blacklist, because they act as a regular end-user. Evidence reveals that most commonly implemented by botnets are TCP SYN and UDP flooding attacks (Freiling, Holz, & Wicherski, 2005). Exploring the bots in a managed honeypot is one of the most effective prevention mechanisms, which will be discussed in the following chapters.

The internet security industry mostly concern spamming activities. According to recent researches %70 to %90 of the world's spam mailing traffic is caused by botnets. Researchers in (Pappas, 2008; Sroufe, Phithakkitnukkon, & Dantu, 2009) report that once the SOCKS v4/v5 proxy (TCP/IP RFC 1928) on a zombie computer is opened by a malicious bot, the bot can easily use this machine for its nefarious tasks, mostly spamming. On the other hand some types of bots can gather spamming e-mail delivery list from perpetrator. Therefore such a bot can be used for sending massive spam mails. In (Brodsky & Brodsky, 2007) a distributed content independent spam classification system, called Trinity, is proposed against spamming from botnets. According to the Trinity, one can assume that if a computer sends thousands of e-mails at the same time, this computer is probably hosting malicious bot software; so that any e-mail from this host can be considered as a spam. Many researches are performed to discover the aggregate behavior of botnet spamming. In Xie et al. (2008). have designed spam signature generation framework named AutoRE. Their analysis shows that botnet host sending patterns, such as the number of recipients per email, connection rates, and the frequency of sending to invalid users are clusterable and their sending times are synchronized.

Some bots may sniff not only the network traffic passing from victims IP interfaces but also the command data of operating system to retrieve sensitive information like usernames, passwords, and identities. According to the evidences, new generation bots are getting more sophisticated than the predecessors. They can quickly scan the entire system to retrieve corporate or financial data and send this sensitive information to the bot master. They rarely affect the performance of host machine, so that they are very hard to be caught. Keylogger

bots listen to the keyboard activities to gather such sensitive information like usernames, passwords, and identities (Feily et al., 2009).

Botnets are also can be used to generate and send phishing mails to the victim individuals. Phishing mail includes legitimate-like URLs and asks the receiver to submit personal or confidential information. This kind of attack is called identity theft. Ordinary mail servers can identify phishing mails by denoting sender IP address. By using botnet's distributed processing facility attacker may send e-mails from ordinary individual's computer which is not listed in mail server phishing blacklists (Erbacher, Marshall, Cutler, & Banerjee, 2008).

With the help of botnet, attackers or bot masters are able to install advertisement add-ons and browser helper objects (BHOs) for business purpose. Attackers may use botnets to click periodically on specific hyperlinks and thus promote the click-through rate (CTR) artificially.

## 4. Botnet analysis

Botnet researches are mostly performed to detect and prevent bot activities. Detecting a botnet often needs advanced analyzing capabilities which are related to the selected data for analysis track and the characteristics of issues performed. In this part, we will consider the types of analysis according to the characteristics and application data performed.

### 4.1. Classification based on behavior

#### *4.1.1. Active analysis*

Active approaches in botnet analysis cover all kinds of analysis techniques which makes bot master, directly or indirectly informed about botnet analysis / detection activity. Capturing bot malware and deactivating its malicious parts is a well-known active analysis type. Honeypots and honeynets are other active analysis methods performed in botnet detection and prevention. At first sight, while active approaches may seem useful, they have a big disadvantage of being easily detected. Once this happens, bot master will inevitably adapt and circumvent any actions taken against botnets (Zhu et al., 2008).

A good example of active analysis is the study of Dagon et al. (2006). The model that they proposed bases time zone information of victim computers. They assume that the individual users switch off or do not use their machines during the night. They use a DNS redirection technique to redirect known IRC Command & Control servers to IP addresses under their control. In six months they redirected approximately 50 botnets. Another work performed one year ago, was analyzing botnet connectivity structures and proposing botnet classification taxonomy based of their connectivity schemes.

#### *4.1.2. Passive analysis*

Passive approaches analyze traffic which the botnet generates without corrupting or modifying it. The analysis mainly focuses on secondary effects of botnet traffic such as

broken packets resulting from a distant DDoS attack. Darknets are good examples of passive analysis. They capture and analysis packages instead of using a machine which appears vulnerable (low interaction honeypot), or actually is vulnerable (high interaction honeypot) for attracting botnet attacks, malware, or spam. The bot master assumes that the IP simulated by darknet is empty; so that the TCP call is not responded by anyone. Both honeypot types, low or high interaction, can be detected by perpetrators. Low interaction honeypots are only basically simulates some services and give basic responses coming to specific service ports. The emulation is incomplete, so the perpetrator can easily detect the honeypot emulator by sending a little sophisticated command. On the other hand we know that high interaction honeypots can be detected by fingerprinting the operating system after successfully compromised it. Passive systems are more complex to implement but in the other hand they have the big advantage that they cannot be detected by intruder; because if perpetrator sends a message to a darknet, he will not get a SYN response. So a darknet is absolutely gives the same sense as an unused IP address to an intruder (Zhu et al., 2008; Seewald&Gangsterer, 2010).

Dhamankar and King propose a system detecting botnet by guessing protocol types without reference to the content transferred (Dhamankar & King, 2007). Guessing the protocol types can also detect encrypted botnet traffic for peer-to-peer networks. This approach is a good example of passive analysis, because this approach does not make any changes in original flow. It works only by mirroring the network flow data.

In another research, Collins et al. proposes a network quality metric based on spatiotemporal ratio of botnets. This means the proportion of bots among all IP address for a specific time and specific subnets. Their aim was predicting future bots according to the past botnet distribution (Seewald & Gangsterer, 2010).

## 4.2. Classification based on used data

### *4.2.1. Analysis Based on IDS Data*

Krugel et al. define intrusion detection as "the process of identifying and responding to malicious activities targeted at computing and network resources". An intrusion attempt, also named as attack, denotes the sequence of actions to gain control of the system. Intrusion Detection System (IDS) discriminates intrusion attempts from normal system usage.

Intrusion detection systems are basically classified into two categories: Misuse-based IDS and Anomaly-based IDS. A misuse-based IDS, also known as signature-based or knowledge-based IDS, detects malicious traffic by comparing new data with a knowledge base or signatures of known attacks. The system delivers an alarm if a previously known intrusion pattern is detected. Misuse-base systems like Snort use pattern matching algorithms in packet payload analysis. It is obvious that misuse-based systems analyze not only the traffic flow of the network; they also analyze payload data of the flow. Misuse-based intrusion detection systems are highly accurate systems. But they need to pay attention on up to date the signature base of the system. They are also ineffective for

detecting new intrusion types and zero day threats. On the other hand anomaly-based IDS, also known as behavior-based IDS, compare input data with the expected behavior of the system. However behavior based systems can detect unknown attacks because of their anomaly based nature; they may give false positive alarms. For example flash crowd situation is not a malicious situation but it can be considered as a denial of service attack by an anomaly / behavior based system.

### *4.2.2. Analysis based on flow data*

The growing number of attacks and the rapid extension rates in network bandwidth are very important challenges for intrusion detection systems. IDS researchers assess the payload-based IDSs processing capability to lie between 100 Mbps and 200 Mbps when commodity hardware is used, and close to 1 Gbps when dedicated hardware is employed (Feily et al., 2009; Zhu et al., 2008). Famous tools like Snort and Bro consume high resource when they deal with huge amount of payload data of in today high speed networks. Besides, encrypted traffic is another challenge for payload based detection systems.

Given these problems above, flow based solutions are more comfortable than intrusion detection systems. Flows are monitored by specialized accounting modules usually placed in network routers. Flow-based solutions will analyze these flows to detect attacks. They analyze markedly lower amount of data than payload based intrusion detection systems. Netflow ("Cisco Netflow," n.d.) data is approximately %0.1 to %0.5 of overall data consuming on the network. Flow information tells about the following attributes:

- Source address: The originator of the traffic
- Destination address: The receiver of the traffic
- Ports: Characterizing the application of the traffic
- Class of service (COS): Examining the priority of the traffic
- Interfaces: Defining the usage of the traffic by the network device
- Tallied packets and bytes: To calculate packet and byte characteristics of the traffic

| Date flow start | Duration | Proto | Src IP Addr:Port | | Dst IP Addr:Port | Flags | Tos | Packets | Bytes | pps | bps | Bpp | Flows |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2011-12-27 14:59:51.000 | 0.000 | TCP | SOH | -> | SOH | .A...F | 0 | 1 | 54 | 0 | 0 | 54 | 1 |
| 2011-12-27 14:58:33.000 | 59.000 | TCP | SOH | -> | SOH | .AP... | 0 | 6 | 2236 | 0 | 303 | 372 | 1 |
| 2011-12-27 14:59:44.000 | 5.000 | TCP | SOH | -> | SOH | .AP... | 164 | 13 | 13144 | 2 | 21030 | 1011 | 1 |
| 2011-12-27 14:59:52.000 | 0.000 | TCP | SOH | -> | SOH | .AP... | 0 | 1 | 200 | 0 | 0 | 200 | 1 |
| 2011-12-27 14:59:57.000 | 0.000 | TCP | SOH | -> | SOH | .AP... | 0 | 1 | 522 | 0 | 0 | 522 | 1 |
| 2011-12-27 14:56:04.000 | 208.000 | TCP | SOH | -> | SOH | .A.... | 164 | 475 | 719150 | 2 | 27659 | 1514 | 1 |
| 2011-12-27 14:59:54.000 | 0.000 | TCP | SOH | -> | SOH | .A.... | 0 | 1 | 1514 | 0 | 0 | 1514 | 1 |
| 2011-12-27 14:59:32.000 | 0.000 | UDP | SOH | -> | SOH | ..... | 0 | 1 | 89 | 0 | 0 | 89 | 1 |

**Figure 3.** Sample netflow data*("Cisco Netflow," n.d.)*

The sample flow data in Figure 3 is captured within Cisco Netflow procedures. Data consist of date information of the flow, duration, protocol used, source IP and port, destination IP and port, some denoting TCP / UDP flags like S for SYN or A for ACK, type of service value (TOS) in the interval 0-255, number of packets in the flow, total amount of bytes transferred

by the flow, packets per seconds of the flow, transferred bits per second, average bytes per each packet of the flow.

As seen in Figure 3, network flow data does not care about payload information of the communication. In other words, network flow analysis is getting only meta-information of the communication as an input. Thus, it is obvious that flow-based analysis is therefore a logical choice for high-speed and intense networks.

Some researches claim that flow-based analysis may be insufficient in comparison to IDS based or signature based analysis. Flow measurements are aggregated information directly comfortable for data mining algorithms. Therefore they cannot give the chance of detecting malicious activities wrapped in the payload of the communication (Abdullah, Lee, Conti, & Copeland, 2005; Lu, Tavallaee, & Ghorbani, 2009). However the sustainability of the network can be monitored in real time by flow-based analysis. Besides some algorithms like time series can be used to get normal profiles of the inspected network and can detect any inharmonious anomaly activities for the detected profiles.

## 5. Botnet detection

In recent years, network security researchers are struggling with botnet detection and tracking as a major research topic. Different solutions have been proposed which can be classified under mainly two topics. The first approach basically uses honeypots and honeynets which can be considered as an active analysis. While the solutions in (Valeur, Vigna, Kruegel, & Kemmerer, 2004) have been initial honeynet-based solutions, many papers discussed detecting and tracking botnets for different honeynet configurations. The second approach, based on passive network monitoring and analysis, can be classified as signature-based, DNS-based, anomaly-based and mining-based (Feily et al., 2009; Seewald & Gangsterer, 2010). These two approaches and sub classifications are detailed below.

### 5.1. Honeypots and honeynets

A honeypot can be defined as an "environment where vulnerabilities have been deliberately introduced to observe attacks and intrusions"(Pouget & Dacier, 2004). They have a strong ability to detect security threats, to collect malware signatures and to understand the motivation and technique behind the threat used by perpetrator. In a wide-scale network, different size of honeypots form honeynet. Usually, honeynets based on Linux operating systems are preferred because of their ability richness and of toolbox contents.

Honeypots are classified as high-interaction and low-interaction according to their emulation capacity. A high-interaction honeypot can simulate almost all aspects of a real operating system. It gives responses for known ports and protocols as in a real zombie computer. On the other hand, low-interaction honeypots simulate only important features of a real operating system. High-interaction honeypots allow intruders to gain full control to the operating system; however low-interaction honeypots do not. Honeypots are also classified according to their physical state. Physical honeypot is a real machine running a

real operating system. Virtual honeypot is an emulation of a real machine on a virtualization host.

The value of a honeypot is determined by the information obtained from it. Monitoring the network traffic on a honeypot lets us gather information that is not available to network intrusion detection systems (NIDS). For example, we can log the key strokes of an interactive session even if encryption is used to protect the network traffic. NIDS require signatures of known attacks to detect malicious behavior, and often fail to detect compromises that were unknown before deployment. On the other hand, honeypots can detect vulnerabilities that are not found yet. For example, we can detect compromise by observing network traffic on the honeypot even if the cause of the exploit has never been seen before (Pouget & Dacier, 2004).

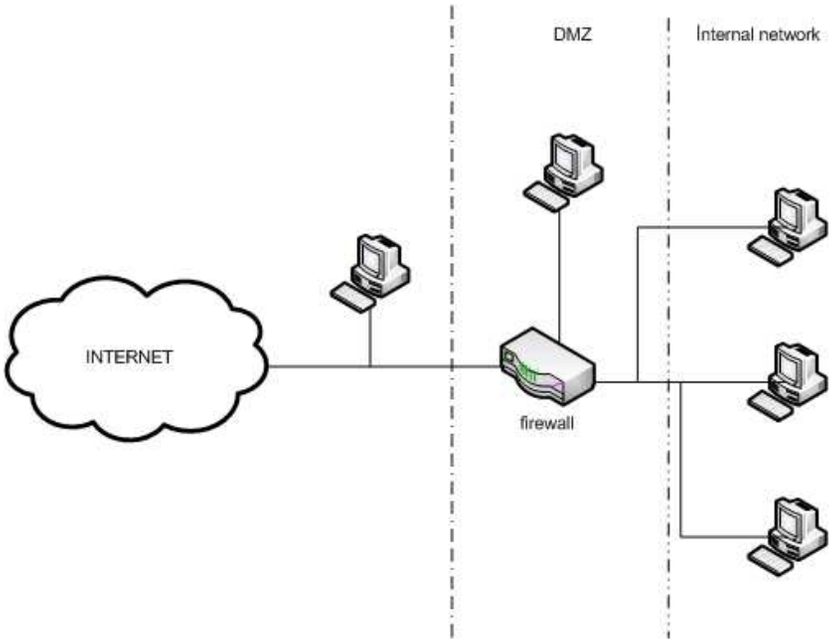


**Figure 4.** Potential Honeypot Traps *(Pouget & Dacier, 2004)*

Figure 4 depicts sample positioning styles of honeypots. Honeypots can be positioned as a computer in secure corporate network, as a computer in demilitarized zone or as a computer outside of the corporate network. Each position represents a different level of security. Internal network computers are hard to reach but after contamination of a malicious code, these computers may be very harmful for the corporation. Besides DMZ computers have some security restrictions rather than an outside computer but less than an internal computer. Outside computers are hard to reach an useful for DDoS, spamming and other types of attacks.

As honeypots and honeynets are very popular in detecting and preventing threats, intruders are seeking new ways of protecting honeypot traps. Some feasible techniques are used by intruders like detecting VMWare or other emulator virtual machines, detecting incoherent responses from bots. Gu et al. (2007), have successfully identified honeypots using intelligent probing. They used public internet threat report statistics. In

addition, Krawetz (2004) have presented a commercial spamming tool, called "Send-Safe's Honeypot Hunter", which is capable of anti-honeypot function. Zou and Cunninqham have proposed a system to detect and eliminate honeypot traps in P2P networks.

## 5.2. Signature based detection techniques

Malware executable signatures are widely used for detecting and classifying malware threats. Signatures based on known malwares have a discriminating power on classification of executables running on an operating system. Rule based intrusion detection systems like Snort are running by using known malware signatures. They monitor the network traffic and detect sign of intrusions. The detection may be according to the signatures of executable malwares or according to the signatures of malicious network traffic generated by malware. However, signature-based detection techniques can be used for detection of known botnets. Thus, this solution is not useful for unknown bots.

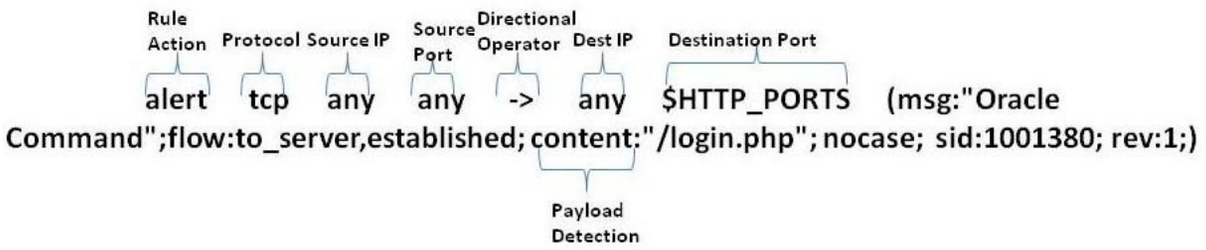


**Figure 5.** Example rule for Snort IDS *(Xie et al., 2008)*

In Figure 5 an example rule configuration for Snort IDS is given. It is obvious that payload information of network traffic is transformed and embedded into the signature or rule. The IDS detects malicious traffic fitting the communication parameters defined by the rule.

In a wide-scale network there may exists many kinds of intrusion detection systems, firewalls or other perimeter protection devices and systems. Each of these systems generates threat alerts. The alerts generated from diverse source of systems must be correlated to improve accuracy and avoid false positive alarms. Alert correlation is a process that analyzes the alerts produced by multiple intrusion detection systems and provides a more succinct and high-level view of intrusion attempts. Gu et al. (2007) propose a framework, "BotHunter", to correlate IDS based detection alerts. They use a network dialog correlation matrix. Each IDS dialog is inserted into the matrix after pruned or evaluated by BotHunter. The system is based on a weighted score threshold system. Each IDS dialog has a weight and after the correlation the total weight of correlated events is calculated by the system. The system then decides whether the correlated event is a malicious activity or not. Thus, false positive rates are lowered to an acceptable rate.

Valeur et al. (2004) have suggested a very comprehensive and detailed framework for intrusion detection alert correlation. Their system was based on the most complete set of components in the correlation process. In their suggested framework sensor alerts are normalized and pre-processed according to a Sensor Ontology Database. After the preparation tasks alerts are fused and verified. The connected alerts are consolidated in an attack session and a multistep alert correlation is performed on the consolidated attack session. After a prioritization, intrusion reports are delivered to the security administrator (Valeur et al., 2004).

The model proposed by Andersson, Fong, & Valdes (2002) and Valdes & Skinner (2000) present a correlation process in two phases. The first phase aggregates low-level events using the concept of attack threads. The second phase uses a similarity metric to fuse alerts into meta-alerts, in an attempt to provide a higher-level view of the security state of the system.

### 5.3. Anomaly based detection techniques

Exploring new botnet detection techniques based on network behavior is a considerable research area for botnet researchers. Anomaly based botnet detection, tries to detect bot activities based on several network behavior anomalies such as unexpected network latencies, network traffic on unusual and unused ports, high volumes of traffic for a mid-class network or unusual system behaviors that could indicate the existence of malicious parties in the network (Feily et al., 2009).

Karasaridis, Rexroad, & Hoeflin (2007) proposed an algorithm for detection and characterization of botnets using passive analysis. Their approach was based on flow data in transport layer. This algorithm can also detect encrypted botnet communications, because the algorithm that they used does not care about the encrypted payload data of network flow.

Binkley & Singh (2006) presented an algorithm based on statistical techniques for detection of on campus botnet servers. Proposed algorithm is derived from two experimental flow tuples that collect statistics based on four types of layer-7 IRC commands: PRIVMSG, JOIN, PING, and PONG.

Recently, Gu & Zhang (2008) proposed a system, BotSniffer, to detect botnet Command and Control channels by using both network behavior anomalies and network channel similarities. Their approach is very simple and useful: The bot clients have to reveal same network behavior anomalies and they also communicate with C&C server by the same network behavior characteristics, simultaneously. Hence, it employs several correlation analysis algorithms to detect spatial-temporal correlation in network traffic with a very low false positive rate.

Some other entropy based solutions to detect network behavior anomalies are also proposed by researchers. These approaches are not only to detect botnets and malicious network traffics. They are proposed for general purposes which also include security oriented ones like botnet detection.

### 5.4. DNS based detection techniques

As mentioned in the first section, a typical bot activity resumes by getting commands and execution parameters of commands from command and control center. Thus bots are bound to send DNS queries to know the IP address of the command and control center. C&C servers have generally a distributed nature in present botnets. Hence they have to use dynamic DNS (DDNS) entries with short time to live to hide them from intrusion detection/prevention systems. Thus, it is possible to detect botnet DNS traffic by monitoring the DNS activities and detecting unusual or unexpected DNS querying.

DNS based techniques are quietly similar to other anomaly based detection techniques. They are commonly based on detection of anomalous DNS network traffic generated by bot computers.

Dagon (2005) proposed an algorithm to identify botnet C&C server addresses by monitoring abnormally high or temporarily intense DDNS queries. This approach is nearly the same as Kristoff's approach (Kristoff, 2004) and both of them are usually useful. But sometimes many important web sites can use short time to live values. Because of naïve nature of this approach many important false positive cases may occur.

Kim et al. (Inhwan, Choi, & Lee, 2008) proposed a methodology for security advisers and administrators providing meaningful visual information do detect botnets. The proposed system is based on DNS traffic which is only a small piece of total network traffic. Hence, this methodology is also comfortable for real time analysis.

Choi, Lee, Lee, & Kim (2007) suggested a system monitoring DNS traffic to detect botnet sub-structures which form a group activity in DNS queries simultaneously. They have identified unique attributes of DNS traffic which help to form groups according to the relevance of these unique features for diverse nodes of network. Their anomaly based approach is more robust than the previous approaches because of detecting botnet flows regardless the type and hierarchy of the botnet structure.

In 2009, Manasrah et al. proposed a system to classify DNS queries and detect malicious DNS activities. The system is based on a simple mechanism which monitors the DNS traffic and detects the abnormal DNS traffic issued by the botnet. Their approach is based on the fact that botnets appear as a group of hosts periodically (Manasrah, Hasan, Abouabdalla, & Ramadass, 2009).

### 5.5. Data mining based detection techniques

Anomaly based techniques are mostly based on network behavior anomalies such as high network latency, activities on unused ports. However C&C traffic usually does not reveal anomalous behavior. It is mostly hard to differentiate C&C traffic from usual traffic behavior. At this point of view pattern recognition and machine learning based data mining techniques are very useful to extract unexpected network patterns.

Firstly it can be useful to introduce a research of preprocessing tasks of anomaly and data mining based botnet detection systems. Davis and Clark introduce a review of known

preprocessing tasks for anomaly based and mining based intrusion detection techniques (Davis & Clark, 2011).

Strayer et al. (2008) suggested a mechanism to detect botnet C&C traffic by a passive analysis applied on network flow information. Their approach is based on flow characteristics such as duration, bytes per packet, bits per second, TCP flags and pushed packets in the flow. The proposed system has a preprocessing phase for flows reducing the set by a factor of about 37, from 1.337.098 to 36.228. They used J48 decision trees, naïve Bayes and Bayesian net algorithms to classify network flows.

Masud, Gao, Khan, & Han (2008) proposed another mining based passive analysis to identify botnet traffic. Their approach is based on correlating multiple log files obtained from different points of the network. The system is not only to detect IRC-based botnet but also applicable for non-IRC botnets. The method is also effective because of its passive and regardless of payload nature. Hence, it is applicable for intense networks and also effective for encrypted communication.

Lu et al. (2011) proposed a system to detect botnet communication patterns based on n-gram feature selection analyzing both payload and flow. They first classify the network traffic into different applications by using traffic payload signatures. Secondly they perform a clustering for each application community to detect anomalous behavior based on the n-gram features extracted the content of network flows. Their approach is payload-aware and hard to execute on a large scale network.

Recently Wang, Huang, Lin, & Lin (2011) proposed a behavior-based botnet detection system based on fuzzy pattern recognition techniques. Their motivation is based on identifying bot-relevant domain names and IP addresses by inspecting the network traces. They used fuzzy pattern recognition techniques with 4 membership functions: (1) generating failed network connection; (2) generating failed DNS queries; (3) having similar DNS query intervals; (4) having similar payload sizes for network communications.

BotMiner (Gu, Perdisci, Zhang, & Lee, 2008), an improvement of BotSniffer (Gu & Zhang, 2008), is a recent and successful solution to detect bot activities. The proposed technique is based on clustering similar communication traffic and similar malicious traffic. After clustering normal and abnormal activity patterns, it correlates these two cross clusters to identify the host that share the same communication pattern and malicious activity pattern. Thus it can be possible to identify botnet structures embedded in the network. BotMiner can detect real-world botnets including IRC-based, HTTP-based, and P2P botnets with a very low false positive rate.

Gu et al. proposed BotHunter (Gu et al., 2007) to detect malware infection by using correlation of intrusion detection dialogs. The system monitors both inbound and outbound network traffic and correlates anomalous flow and unexpected payload information. BotHunter not only uses data mining techniques. Rule based engines and statistical engines are also embedded in BotHunter.

Additionally some graph based solutions are performed to detect botnet sub-graph structures. BotGrep (Nagaraja, Mittal, Hong, Caesar, & Borisov, 2010) is a recent and

effective solution to detect bots by using structured graph analysis. For modern bot structures, because of their distributed C&C architecture, detecting sub-graph network is a very useful and convenient way of intrusion detection.

## 6. Conclusion

This survey aimed to discover the recent state of botnet research in academia. A progressive survey technique is followed which starts with botnet definition, proceeds with attack types caused by botnets and well-known botnet classifications and ends with diverse types of detection techniques.

According to the orientation of recent studies on botnet detection, we can assert that data mining and machine learning based approaches may have an important contribution on detecting malicious bot structures on a wide-scale network. Botnet detection techniques are classified into two classes according to the types of information they use: network flow information and payload information. Increasing network speeds, growing sizes of payload information streaming on the network complicates payload-base analysis in wide-scale networks. Hence, flow based methods are more convenient as they only attend to discover network flow information which can be understood as meta-information of a network flow without payload.

Data mining and machine learning techniques are easily applicable on network flow information. Flow data have a structured and related nature, which do not require massive preprocessing tasks. Besides, flow information implies patterns inside, which makes data mining algorithms convenient and effective for analysis.

## Author details

Erdem Alparslan
*Bahçeşehir University Software Engineering Department, Turkey*
*Center of Research for Advanced Technologies of Informatics and Information Security, Turkey*

Adem Karahoca
*Bahçeşehir University Software Engineering Department, Turkey*

Dilek Karahoca
*Bahçeşehir University Software Engineering Department, Turkey*

## 7. References

Abdullah, K., Lee, C., Conti, G., & Copeland, J. A. (2005). Visualizing Network Data for Intrusion Detection. the Sixth Annual IEEE SMC (pp. 100-108).

Andersson, D., Fong, M., & Valdes, A. (2002). Heterogeneous Sensor Correlation: A Case Study of Live Traffic Analysis. IEEE Information Assurance Workshop.

Bethencourt, J., Franklin, J., & Vernon, M. (2005). Mapping internet sensors with probe response attacks. 14th Conference on USENIX Security Symposium (pp. 193-208).

Binkley, J. R. (2006). Anomaly-based Botnet Server Detection. FloCon 2006.

Binkley, J. R., & Singh, S. (2006). An algorithm for anomaly-based botnet detection. SRUTI 2006. Retrieved from http://static.usenix.org/events/sruti06/tech/full_papers/binkley/binkley_html

Brodsky, A., & Brodsky, D. (2007). A distributed content independent method for spam detection. 1 stWorkshop on Hot Topics in Understanding Botnets.

Chiang, K., & Lloyd, L. (2007). A case study of the rustock rootkit and spam bot. First workshop on hot topics in understanding botnets.

Choi, H., & Lee, H. (2012). Identifying botnets by capturing group activities in DNS traffic. Computer Networks, (56), 20-33.

Choi, H., Lee, H., Lee, H., & Kim, H. (2007). Botnet Detection by Monitoring Group Activities in DNS Traffic. 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 715-720. Ieee. doi:10.1109/CIT.2007.90

Cisco Netflow. (n.d.). Retrieved from http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

Dagon, D. (2005). Botnet Detection and Response, The Network is the Infection. OARC Workshop.

Daswani, N., & Stoppelman, M. (2007). the Google Click Quality, and S. Teams. The anatomy of clickbot. First workshop on hot topics in understanding botnets.

Davis, J. J., & Clark, A. J. (2011). Data preprocessing for anomaly based network intrusion detection: A review. Computers & Security, 30(6-7), 353-375. Elsevier Ltd. doi:10.1016/j.cose.2011.05.008

Dhamankar, R., & King, R. (2007). Protocol identification via statistical analysis (PISA). Black Hat.

Dietrich, C., Rossow, C., Bos, H., & Steen, M. V. (2008). On Botnets that use DNS for Command and Control. kerstin.christian-rossow.de. Retrieved from http://kerstin.christian-rossow.de/publications/dnscnc2011.pdf

Erbacher, R. F., Marshall, J., Cutler, A., & Banerjee, P. (2008). A Multi-Layered Approach to Botnet Detection. Sensors (Peterborough, NH), 42.

Feily, M., Shahrestani, A., & Ramadass, S. (2009). A Survey of Botnet and Botnet Detection. 2009 Third International Conference on Emerging Security Information, Systems and Technologies, 268-273. Ieee. doi:10.1109/SECURWARE.2009.48

François, J., Wang, S., Bronzi, W., State, R., & Engel, T. (2011). BotCloud : Detecting Botnets Using MapReduce. International Workshop on Information Forensics and Security (WIFS), 2011 IEEE (pp. 1-6).

Freiling, F., Holz, T., & Wicherski, G. (2005). Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks. th European Symposium on Research in Computer Security (ESORICS '05).

Gu, G., & Zhang, J. (2008). BotSniffer: Detecting botnet command and control channels in network traffic. 15th Annual Network and Distributed System Security Symposium (NDSS'08). Retrieved from http://users.csc.tntech.edu/~weberle/Fall2008/CSC6910/Papers/17_botsniffer_detecting_botnet.pdf

Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008). BotMiner: Clustering Analysis of Network Traffic for Protocol and Structure Independent Botnet Detection. SS'08 Proceedings of the 17th conference on Security symposium.

Gu, G., Porras, P., Yegneswaran, V., Fong, M., & Lee, W. (2007). BotHunter : Detecting Malware Infection Through IDS-Driven Dialog Correlation. SS'07 Proceedings of 16th USENIX Security Symposium.

Holz, T., M, S., & Dahl, F. (2008). Measurement and mitigation of peer-to-peer-based botnets: a case study on storm worm. 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (pp. 1-9).

Inhwan, K., Choi, H., & Lee, H. (2008). Botnet Visualization using DNS Traffic. WISA 08.

Karasaridis, A., Rexroad, B., & Hoeflin, D. (2007). Wide-scale Botnet Detection and Characterization. USENIX Workshop on Hot Topics in Understanding Botnets.

Krawetz, N. (2004). Anti-Honeypot technology. IEEE Security and Privacy, 2, 76-79.

Kristoff, J. (2004). Botnets. 32nd Meeting of the North American Network Operators Group.

Liu, J., Xiao, Y., Ghaboosi, K., Hongmei, D., & Zhang, J. (2009). Botnet: Classification, Attacks, Detection, Tracing and Preventing Measures. Journal on Wireless Communication and Networking.

Lu, W., Rammidi, G., & Ghorbani, A. A. (2011). Clustering botnet communication traffic based on n-gram feature selection. Computer Communications, 34, 502-514.

Lu, W., Tavallaee, M., & Ghorbani, A. A. (2009). Automatic Discovery of Botnet Communities on Large-Scale Communication Networks. 4th International Symposium on Information, Computer, and Communications Security (pp. 1-10).

Manasrah, A. M., Hasan, A., Abouabdalla, O. A., & Ramadass, S. (2009). Detecting Botnet Activities Based on Abnormal DNS traffic. Journal of Computer Science and Information Security, 6(1), 97-104.

Masud, Mohammad, Al-khateeb, T., & Khan, L. (2008). Flow-based identification of botnet traffic by mining multiple log files. , 2008. DFmA 2008., 200-206. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4784437

Masud, MM, Gao, J., Khan, L., & Han, J. (2008). Peer to peer botnet detection for cyber-security: a data mining approach. CSIIRW 08. Retrieved from http://dl.acm.org/citation.cfm?id=1413185

Nagaraja, S., Mittal, P., Hong, C.-yao, Caesar, M., & Borisov, N. (2010). BotGrep : Finding Bots with Structured Graph Analysis. 19th USENIX Security Symposium (pp. 1-24).

Nazario, J. (2007). Blackenergy ddos bot analysis.

Pappas, K. (2008). Back to basics to fight botnets. Communications News, 45, 5-12.

Pouget, F., & Dacier, M. (2004). Honeypot based forensics. Asia Pacific Information technology Security Conference (AusCERT '04).

Puri, R. (2003). Bots and botnets: an overview.

Roschke, S., Cheng, F., & Meinel, C. (2010). A Flexible and Efficient Alert Correlation Platform for distributed IDS. Network and System Security.

Seewald, A., & Gangsterer, W. (2010). On the detection and identification of botnets. Computers & Security, 29, 45-58.

Sperotto, A., & Pras, A. (2010, May). Flow-based intrusion detection. 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops. Ieee. Retrieved from

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5990529

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., & Stiller, B. (2010). An Overview of IP Flow-Based Intrusion Detection. IEEE Communications Surveys, 12(3), 343-356. doi:10.1109/SURV.2010.032210.00054

Sroufe, P., Phithakkitnukkon, S., & Dantu, P. (2009). Email shape analysis for spam botnet detection. 6th IEEE Consumer Communications and Networking Conference (CCNC '09) (pp. 1-2).

Strayer, W., Lapsely, D., & Walsh, R. (2008). Botnet detection based on network behavior. Botnet Detection (pp. 1-24). Springer. Retrieved from http://www.springerlink.com/index/N77M076734522777.pdf

Valdes, A., & Skinner, K. (2000). An Approach to Sensor Correlation. Recent Advances in Intrusion Detection.

Valeur, F., Vigna, G., Kruegel, C., & Kemmerer, R. (2004). Comprehensive approach to intrusion detection alert correlation. IEEE Transactions on Dependable and Secure Computing, 1(3), 146-169. doi:10.1109/TDSC.2004.21

Villamarín-salomón, R., & Brustoloni, J. C. (2008). Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic. Communications Society, (1), 476-481.

Villamarín-salomón, R., & Brustoloni, J. C. (2009). Bayesian bot detection based on DNS traffic similarity. 2009 ACM symposium on Applied Computing (pp. 2035-2041). Retrieved from http://dl.acm.org/citation.cfm?id=1529734

Wang, K., Huang, C.-Y., Lin, S.-J., & Lin, Y.-D. (2011). A fuzzy pattern-based filtering algorithm for botnet detection. Computer Networks, (55), 3275-3286.

Wang, P., Sparks, S., & Cou, C. (2008). An advanced hybrid peerto- peer botnet. 1st Workshop on Hot Topics in Understanding Botnets (p. 2).

Wikipedia - Botnet. (n.d.). Retrieved from

Wikipedia - Internet Bot. (n.d.). Retrieved from http://en.wikipedia.org/wiki/Internet%0Abot

Wurzinger, P., Bilge, L., Holz, T., Goebel, J., Kruegel, C., & Kirda, E. (2009). Automatically Generating Models for Botnet Detection. ESORICS 09 (pp. 232-249).

Xie, Y., Yu, F., Achan, K., & Panigraghy, R. (2008). Spamming botnets: signatures and characteristics. ACM SIGCOMM Conference on Data Communication (SIGCOMM '08).

Yen, T.-F., & Reiter, M. (2008). Traffic Aggregation for Malware Detection. 5th international conference on Detection of Intrusions and Malware.

Zeidanloo, H. R., & Manaf, A. A. (2010). Botnet Detection by Monitoring Similar Communication Patterns. Journal of Computer Science, 7(3), 36-45.

Zeidanloo, H. R., Zadeh, H. M. J., M, S., & Zamani, M. (2010). A Taxonomy of Botnet Detection Techniques. Industrial Engineering, 158-162. Retrieved from http://www.mendeley.com/research/a-taxonomy-of-botnet-detection-techniques/

Zhu, Z., Lu, G., Chen, Y., Fu, Z. J., Roberts, P., & Han, K. (2008). Botnet Research Survey. 2008 32nd Annual IEEE International Computer Software and Applications Conference, 967-972. Ieee. doi:10.1109/COMPSAC.2008.205