An analysis of network traffic characteristics for Botnet detection

Maria Jose Erquiaga[1], Carlos Catania[1] and Carlos García Garino[1,2]

[1] Instituto para las Tecnologías de la Información y las Comunicaciones (ITIC)
[2] Facultad de Ingeniería
Universidad Nacional de Cuyo, Mendoza, Argentina mariajoseerquiaga@gmail.com
{ccatania,cgarcia}@itu.uncu.edu.ar

Abstract. The fast evolution of Botnet malware made it extremely difficult to detect. Despite it can be just considered as a tool, nowadays it has become one of the most dangerous threats for system administrators. Botnets are used as the starting point for different kind of attacks, such as SPAM, Denegation of Service, key logging and traffic sniffing, among others. In this paper we analyze some of the most relevant network traffic characteristics used for Botnet recognition. We have reviewed the most important works in the field of Botnet detection and have carried out an analysis in order to establish which are more appropriate to describe the Botnet behavior. Our final goal is to provide to network administrators the bases for building tools that can help them in their daily fight against this security threat.

1 Introduction

Malware is a kind of software that damage computers and compromise the information inside them. The problem increases due to the accelerate development in the informatics field. With the important growth of malware along with the progress made in Artificial intelligence (AI) have resulted in the arising of Botnets.

A Botnet can be defined as net of bots, that is malware software installed on compromised computer with the ability of automatically propagating itself to new computers hosts. All the infected computers conform a *Net of Bots* with synchronized behavior. Botnet operation can be remotely controlled by a *Botmaster* to perform different malicious activities. Since a Botnet can be view as nothing more than a tool, there are several potential criminal uses for them.

Botnets have the capacity of continuously changing their behavior. This evolving capacity made them more difficult to detect. However, it is possible to recognize their presence in the network. For detecting Botnet activity on local networks it is important to know how Botnets work and which are the common characteristics of these nets. By studying Botnet behavior, we might be able to detect and mitigate their effects.

In this work we intend to analyze different network traffic features capable of recognizing the presence of Botnets. We analyze the most relevant Botnet detection methods [4,6,5,11,1] in order to find out the relation between the network traffic features they use and their relation with Botnets behavior. The goal is to establish which are the attributes with the higher discriminate power. Network administrators may follow the behavior of features in the network, by using tools such as MRTG [8] or RRDTOOL [10]. Eventually, they can find some anomalies in the traffic, and it might indicate that there is an attack, or a bot in the local network. The contribution of our work is to analyze certain attributes, in order to describe the relation between them and Botnets behavior. In that sense, it will be helpful to identify anomalies in traffic that leads us to detect Botnets and possible network attacks.

This work is organized as follows: in section 2, we present a background, it defines Botnets and gives a short explanation of how do they work. Then, we introduce the Intrusion Detection Systems (IDS), follow by the concept of network traffic feature and their classification. In section 4, we present the Botnets detection features, according to the given classification. Finally, in section 5, we present a conclusion.

2 Background

2.1 Botnets

As the word suggests, Botnet, is a net composed of bots. The word bot comes from RoBot, bots are "smart" programs that can be automatically executed and can perform different actions according to the orders given by someone remote controlling them [3]. No human intervention is required by the bots to perform their tasks. These nets composed of bots, have the purpose of attacking and taking new hosts in order to conform the botnet. Making this process automatically, the botnet grows. Thus, a botnet, also known as zombie army, is a net of bots.

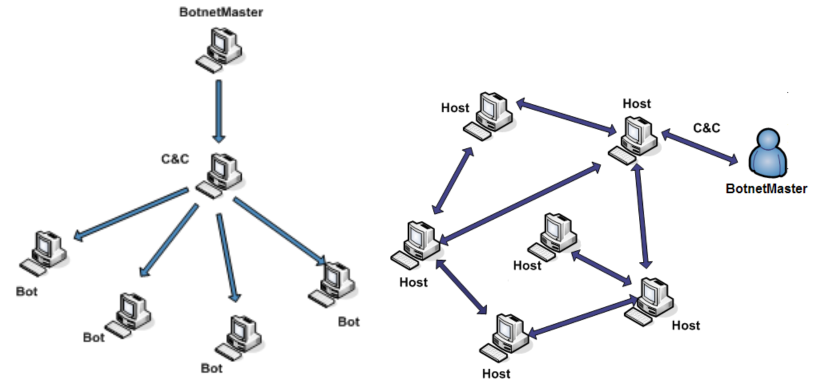
A Botnetmaster is the person who control Botnets trough remote control. They use their zombie army with different purposes, for example Distributed Denegation of Service (DDoS) attacks or spamming among others.

Furthermore, botnetmasters manage their Botnets using different control mechanisms. Those mechanisms include the protocols and the commands used by them to control his Botnet. There are two kinds of network architectures: centralized and decentralized ones. As shown in Fig. 1 (a), centralized nets usually have a central server where the bots are connected to. Decentralize Botnets have a server as well, but they differ in that they use peer-to-peer communication (P2P), this is shown in Fig. 1(b).

Moreover, there are different kinds of servers, the ones who use the IRC protocol, Fig. 2(a) and the others: web servers, that use the HTTP protocol Fig. 2(b).

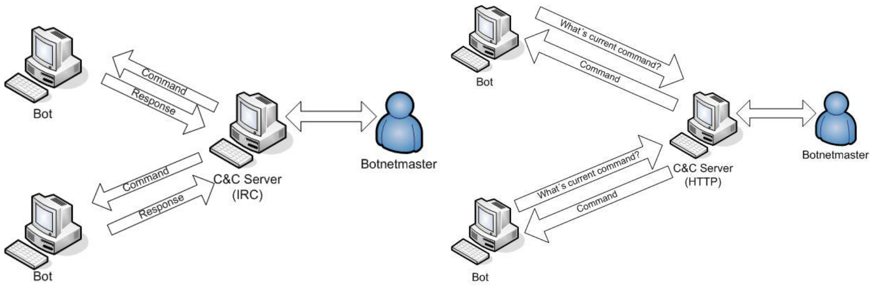
2.2 Intrusion Detection

An NIDS (Network Intrusion Detection System) is a software that detects unauthorized access to a network by sniffers and analyzing the network traffic. The



(a) Centralized Botnet (b) Decentralized Botnet

Fig. 1: Botnet Topologies



(a) IRC Protocol (b) HTTP Protocol

Fig. 2: Botnet underlying protocols

goal of this kind of systems is to detect intrusions in a network. Hence, a NIDS analyses the activity of a certain network, looking for possible intrusions and threats.

There are two main intrusion detection methods: anomaly-based and misuse-based [2]. The first method defines a model of normal traffic and compares it with the network traffic to be analyzed. If there are any differences with the normal traffic, it is considered an anomaly and it might be a threat or a possible attack. The normal traffic model is built by following certain parameters, for example, the number of connections, packet distribution according to protocol, etc. In opposition, the second method defines a model of intrusions, and waits for them to occur. These methods use different techniques. For example, anomaly-based may use statistics, machine learning or data mining. On the other hand, misuse-based methods usually use pattern recognition, implication rules or data mining [2].

2.3 Features

The term attributes or features is usually related to data mining or machine learning process [12]. However, for the purpose of our study, features are defined as certain characteristics of a set of data that can be obtained from network traffic captures. The analysis of network traffic features are the bases for network-based IDS.

There are two attributes classifications, one related to the computational resources needed to be obtained and the other one to the network traffic source [2]. In the first classification, there are two cases: *low-level* features that can be acquired from raw traffic captures (as IP headers, or protocol) and *high-level* features that are the results of the traffic capture processing. For instance: Bytes per packet, packets per second, etc. These features might be obtained from low-level attributes.

The second classification mentioned, has three possibilities: packet, flow and payload features. These are obtained from packet headers, the information of network connections and packet payload (application layer) respectively. To clarify this classification, we present a diagram in Fig. 3.

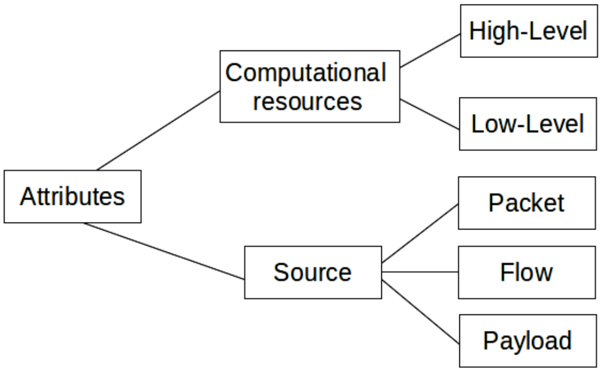


Fig. 3: Attributes classification

3 Network traffic features used in Botnet detection

In this section we present the features for Botnet detection, as suggested by the most relevant authors in the Botnet detection field [4,5,6,11,1] All these authors also implement different algorithms in order to detect Botnets. However, our analysis is only focused on Botnets features. We will describe those attributes following the classification related to the network traffic source (packet, flow, payload). It should be noted that some features are included in more than one category, for instance an attribute might be packet and flow based.

3.1 Packet based features

Some of the features for Botnet detection are included in this category. In particular, a number of authors such as Gu [4], Karasaridis [5], Strayer [11] and Livadas [6] have worked with the Bytes-per-packet (Bpp) feature, and Strayer [11] mentioned that *Bpp* is the attribute that present the most discriminatory power. Moreover, Gu [4] utilized the features: average number of bytes per packets (Bpp) and number of packets-per-flow (ppf). This last one, can be considered a flow feature as well, because it takes into account the information of the flow.

Binkley [1] proposed the metric TCP work weight, which is obtained by using the following formula: $w = (S_s + F_s + R_r)/T_{sr}$, where S_s are the SYN's plus SYN-ACKs sent, F_s the FIN's sent, R_s the RESETS and T_{sr} are the total number of TCP packets. This information is obtained from the TCP packets. The value of this metrics is expressed as a percentage. If this value is closer to 100% (percent), there are more possibilities for some kind of anomaly to occur. Another packet feature considered by [1] is IP source. This information is obtained from the packet header. Thus, it is consider a low-level attribute.

Furthermore, Karasadiris [5] considers the feature packets-per-flow (ppf), which is also a flow feature, and bytes-per-packet (Bpp). Likewise, this last feature is also suggested in [11].

In addition, Livadas informs that the features with most discriminatory power are: the percentage of packets that are pushed (PctPktsPushed), and the variance in the Bytes-per-packet (varBpp), besides Bpp.

3.2 Flow based features

The flow features suggested by Gu [4] are: the number of packets per flow (ppf), the number of flows per hour (fph) and the average number of bytes per second (bps). Karasadiris [5] besides using packets-per-flow (ppf) also uses flows-per-address (fpa). This feature is obtained making an association between a local IP address and their local port with several remote IP addresses and remote ports. Then, [11] expresses that the features with high discriminatory values are: duration, role (it means, who started the flow: the client or the server), average bits per second (bps), and average packets per second (pps).

3.3 Payload based features

Binkley [1], selected several payload features using two lists: Channel Name (is the name of the IRC channel), Joins (to the channel), Private Messages, Hits (the joins plus the private messages), number of IPs in the channel and the list of the IP numbers. All those features are related to a channel, that is why they are called *Channel List*. The other list is the *Node List* related to the metrics obtained from any IP address in any IRC channel. The flow features in this list are: Total Messages, Joins, Pings, Pongs, Private Messages, Channels and Server Hits (the number of messages sent to/from a host).

3.4 Analysis

In Table 1, we summarize the information discussed in previous sections. The table shows information regarding the authors, the features utilized by them, the kind of attributes, and the detection method selected by each.

Table 1: Botnet Detection Feature analysis

Author	Features	Type of feature	Detection technique
Guofrei Gu [4]	number of flows per hour (fph)	high level, flow	Anomaly-based and misuse-based
	number of packets per flow (ppf)	high level, flow, packet	
	average number of bytes per packets (bpp)	high level, packet	
	average number of bytes per second (bps)	high level, flow	
Karasaridis [5]	flows-per-address (fpa)	high level, flow	Anomaly-based
	packets-per-flow (ppf)	high level, flow, packet	
	bytes-per-packet (bpp)	high level, packet	
Strayer [11], Livadas[6]	duration	high level, flow	Misused-based
	role	high level, flow	
	average bytes per packet (Bpp)	high level, packet	
	average bits per second (bps)	high level, flow	
	average packets per second (pps)	high level, packet	
	% of packets that are pushed (PctPktsPushed)	high level, packet	
	variance in the Bytes-per-packet (varBpp)	high level, packet	
Binkley[1]	TCP work weight	high level, packet	Anomaly-based
	Chanel Name (CHANNAME)	high level, payload	
	joins	high level, payload	
	hits	high level, payload	
	private messages (PRIVMSGS)	high level, payload	
	number of IPs in the channel (NOIPS)	high level, payload	
	list of the IPs (IP_LIST)	high level, payload	
	IP source (IPSCR)	low level, packet	
	total messages (TOTALMSG)	high level, payload	
	pings	high level,payload	
	pongs	high level ,payload	
	private messages (PRIVMSGS)	high level, payload	
	channels	high level, payload	
	server hits	high level, payload	

In order to understand the features presented by each author, it is important to know how do Botnets works. For example, we can say that Botnet behavior is synchronized [4,5]. In that sense, we present the features and analyze their relation with Botnet operation.

The feature Bpp, is selected for several authors for Botnet detection [4,6,5,11]. In the case of Botnets, the value of Bpp is low because the messages between the bots and the Botnetmaster are short commands. So, the size of the packets are approximately 1KB [7]. Furthermore, the duration of this communication is short, so it is another feature to be considered.

Moreover, the number of flows per hour (fph) suggested by Gu et al. [4] has a relation with Botnet behavior too. When the Botnetmaster send commands to their bots, he do it at the same time, so suddenly a lot of flows may appear. And the number of flows-per-address (fpa) will be high as well.

In the case of Botnets, the attribute packets-per-flow (ppf), does not reach a high value because there are a few packets that the botmaster sends to the bots with instructions. Bytes-per-second reaches a low number too as the messages sent are short. Regarding the role feature, considered by Strayer [11], the importance is on who initiates the communication, which is usually the server (in this case, the Botnetmaster).

Blinkley's [1] analysis focuses on finding IRC-based Botnets, by using two list: the channel list and the node list. The first one, is composed by the following features: *{CHANNAME, HITS, JOINS, PRIVMSGS, NOIPS, IP_LIST}*, which have already been described. The node list, mentioned in subsection 3.3, contains the features: *{IPSCR, TOTALMSG, JOINS, PINGS, PONGS, PRIVMSGS, SERVERHITS}*. To identify Botnets, the author proposed to find certain values or characteristics in these features. One of them, is to identify the evil channels that must be the ones with a lot of hosts with a high value of TCP work weight. This attribute, TCP Work Weight, is an indicator of a possible scanning or the presence of a worm.

As we have seen, the features selected have a relation with Botnet behavior. However, our goal is to find the most useful features for Botnet detection. These ones are not related to a particular kind of Botnet. For instance: IRC based, HTTP-based, centralized, etc. In that sense, the features selected by [1] would not be the best ones, because it intent to detect only IRC-based Botnets. Therefore, an important feature to consider is Bpp, because the communication between the bots and the Botnet master always include short messages, while data transfer packets in normal traffic are longer. Another attribute to highlight is the number of flows per hour (fph), as we have indicate, this feature have also a relation with Botnets operation. In general the messages are sent to all the bots simultaneously. Since Botnets are larger, commands are given to the entire net, and not to individuals bots [11]. Since communication between the Botnetmaster and their bots have a small interval of time, we consider that duration is another helpful attribute to detect Botnets as well.

4 Concluding remarks and future work

Today, Botnets are one of the most important threats in network security. They are originated by the combination of malware and IA. Since they are constantly evolving, it is really difficult to detect and mitigate them. In that sense, it is crucial to understand the behavior of Botnets, in order to find the network traffic features that can potentially help system administrators to detect them.

Due to Botnets behavior changes, it is difficult to detect them, however in the past years, several complex Botnet detection methods have been proposed. These methods were based on machine learning, pattern recognition and clustering among others. Still, we believe a much simpler approach is possible. Given some traffic network features with high discrimination power, it is possible to utilize certain network analyzer tools as MRTG[8] or RRDTools [10] in order to help system administrator in the Botnet detection process. Moreover, it is also possible to use such features for writing a proper signature rules for using with a NIDS such as Snort [9]. Notice that we are aware this approach will be not enough for fully Botnets detection, but we think it can potentially provide good enough information for helping system administrators in their daily fight against intruders.

Therefore, regardless of the detection method utilized by the different authors, our goal with this work was to understand how the attributes are related with Botnet behavior. We analyzed network traffic attributes under two major aspects. First, from the point of view of computational resources requirements. Second, how the attributes are related with Botnet behavior

From our analysis came out that, since the most interesting attributes are the ones related to Botnets behavior. On the other hand, attributes such as the protocol used, or the connection ports are useless. Moreover, we consider that, the attributes with most discriminatory power can be, in the first place Bpp, because the kind of messages send by the bots and the Botnetmaster are particularly short. In second place we consider flows per hour (fph) as another network traffic feature with high potential for Botnet detection. Finally, connection duration, is also an important network traffic characteristic for recognizing Botnet behavior. These network traffic features provide a good trade off between their discriminative power and the computational resources they need. Therefore, we believe system administrators can exploit the benefits of these network traffic features in his daily battle against Botnet propagation.

In a future work, we intent to evaluate the performance of the features selected in this survey in order to find the optimal performance in Botnet detection.

Acknowledgments

The authors would like to thank the financial support received by the projects PAE-PICT 2312 granted by ANPCyT and 06/M036 granted by SECTYP, UNCuyo.

References

1. J. R. Binkley and S. Singh. An algorithm for anomaly-based botnet detection. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, SRUTI'06, pages 7–7, Berkeley, CA, USA, 2006. USENIX Association.

2. C. A. Catania and C. García Garino. Automatic network intrusion detection: Current techniques and open issues. *Computers and Electrical Engineering*, 2012. Accepted. In Press. DOI:10.1016/j.compeleceng.2012.05.013.
3. M. J. Erquiaga. Botnets: Mecanismos de control y de propagacion. In *CACIC 2011. XVII Congreso argentino de ciencias de la computacion*, pages 1076–1085, 2011.
4. G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th conference on Security symposium*, SS'08, pages 139–154, Berkeley, CA, USA, 2008. USENIX Association.
5. A. Karasaridis, B. Rexroad, and B Hoeflin. Wide-scale botnet detection and characterization. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, pages 7–7, Berkeley, CA, USA, 2007. USENIX Association.
6. C. Livadas, R. Walsh, D. Lapsley, and W.T. Strayer. Usilng machinelearning technliques to identify botnet traffic. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 967 –974, nov. 2006.
7. M.M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K.W. Hamlen. Flow-based identification of botnet traffic by mining multiple log files. In *Distributed Framework and Applications, 2008. DFmA 2008. First International Conference on*, pages 200 –206, oct. 2008.
8. T. Oetiker. MRTG - the multi router traffic grapher. In *Proceedings of the 12th USENIX conference on System administration*, LISA '98, pages 141–148, Berkeley, CA, USA, 1998. USENIX Association.
9. M. Roesch. SNORT - lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX conference on System administration*, LISA '99, pages 229–238, Berkeley, CA, USA, 1999. USENIX Association.
10. J. Sellens. RRDTool: Logging and graphing. In *USENIX Annual Technical Conference, General Track*. USENIX, 2006.
11. W.T. Strayer, R. Walsh, C. Livadas, and D. Lapsley. Detecting botnetswith tight command and control. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 195 –202, nov. 2006.
12. I. Witten and E. Frank. *Data Mining, Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2nd edition, 2005.