Identifying and Modeling Botnet C&C Behaviors

Conference Paper · May 2014

DOI: 10.1145/2602945.2602949

Sebastián García
Czech Technical University in Prague

Identifying and Modeling Botnet C&C Behaviors

Sebastián García
Department of Computer
Science, Czech Technical
University in Prague
sebastian.garcia@agents
.fel.cvut.cz

Vojtěch Uhlíř
Department of Computer
Science, Czech Technical
University in Prague
vojtech.uhlir@agents
.fel.cvut.cz

Martin Rehak
Cisco Systems
Department of Computer
Science, Czech Technical
University in Prague
rehak@cisco.com

ABSTRACT

Through the analysis of a long-term botnet capture, we identified and modeled the behaviors of its C&C channels. They were found and characterized by periodicity analyses and statistical representations. The relationships found between the behaviors of the UDP, TCP and HTTP C&C channels allowed us to unify them in a general model of the botnet behavior. Our behavioral analysis of the C&C channels gives a new perspective on the modeling of malware behavior, helping to better understand botnets.

Categories and Subject Descriptors

H.4 [**Intrusion/anomaly detection and malware mitigation**]: Malware and its mitigation

General Terms

Experimentation, Security

Keywords

Malware, Botnet, Network Behavior, Network Security

1. INTRODUCTION

The Botnets are still the most important source of attacks on the Internet. While there are a lot of methods to detect them [4], there is yet a need to improve their detection performance. Since most of these methods use a behavioral model of the botnet on their algorithms and since most of them focus on the behavior of their Command and Control channels (C&C), we believe that having a better model of the behavior of these channels may help to create better detection algorithms.

We studied the traffic of a Zbot family Botnet during 57-days to extract the characteristics of its C&C channels and then create states models of their behavior. We hypothesize that only using a long capture it is possible to see the general behaviors of a botnet. The comparison of these C&C models allow us to also create a model of the botnet itself.

Our contributions can be summarized as follows:

- A deep analysis of the behavioral characteristics that distinguish the C&C channels. (Section 3).
- A state model of each C&C channel and the botnet. (Section 3).
- An analysis of the relationship between the channels and the botnet actions.
- A novel 57-days labeled and long botnet dataset. (Section 4)

The network topology used by Botnets has a strong relationship with multi-agents systems. Botnets typically consists of millions of interconnected computers that are controlled by one or more Botmasters, who in turn are following specific goals. The Botmasters send orders to each Bot, and the Bots reacts depending on the changing context. The individual Bots usually do not take complex decisions by themselves but they rely on the orders sent by the Botmaster. From this point of view, Botnets may possible by the largest implementation of a real-world distributed and efficient multi-agent system. Our work helps understand the details of the network behavior of each agent and ultimately helps to improve the agent-based security research area.

We conclude that it is possible to extract the behavioral patterns of a botnet C&C channels, to find hidden relationships between them, to correlate those patterns and to build state models of botnet behavior.

2. PREVIOUS WORK

The modeling of botnet behavior usually focus on their C&C channels since they are their most important characteristic. If they can be modeled, then they may be used to better detect botnets. An analysis of the periodicity of the C&C channels by studying its Power Spectral Density was presented by Basil et al. [13]. The difference with our work is that they use a simulated and controlled botnet and that they aggregate the packet count of traffic every 100 ms.

The behaviors of the P2P protocol depends on the type of algorithm used and the implementation of the botnet. The extensive analysis of the Nugache botnet done by Dittrich et al. in [3] shows that its P2P is encrypted, used TCP protocols and the port 8. Although extensive, the analysis does not include a behavioral analysis of the C&C channel.

Another common approach to analyze a C&C channel is to study its statistical features, such as the work presented by Kondo et al. [5]. Among the features analyzed are the packet sizes and the packets time interval. However, there is no information about which C&C it is or how these characteristics relate to the behavior of the channel. A more comprehensive work was done by Bilge et al. [2], where the features studied were the flow size, flows inter-arrival times, the established to attempted flows ratio and the flow volume as a function of time to find diurnal fluctuations. Their dataset was 40 days long, being one of the longest of the literature. The main difference with our work is that the authors focus on the detection algorithm and do not study the values of these features for the C&C channel.

Another behavioral analysis of a P2P C&C channel was presented by Zhao et al.. [9], where the features selected for detection are the variance of the length of the payload on each time interval, the number of packets exchanged on each time interval, the size of the first packet in the flow and the number of flows per address. The main difference with our work is that they focused on the detection algorithm and did not show an analysis of these features over time.

The idea of analyzing known malicious traffic to automatically generate detection signatures was analyzed and implemented by Rieck at al. [14]. The Botzilla program presented is trained with malware traffic and can be later implemented in a real network. The training is done by repetitively executing malware and extracting its invariant characteristics. They are computed from the tokens seen on the payload of packets. The main difference with our approach is that we focus on the time-based behavioral characteristics of malware instead on the statical content of packets.

An behavioral analysis of the stability of a P2P C&C channel was presented by Li et al. [6], where they aggregate the flows every one hour and compute the number of flows and the average packet size. The stability is computed by comparing these features in a moving time window of five minutes. The main difference is that is focused on the detection algorithms and does not analyze these features in the channel.

While most works focus on using flow features for detecting botnet C&C channels, we focus on a deep study of the features themselves to provide a solid background for future analyses.

3. BEHAVIORAL ANALYSIS OF THE C&C CHANNELS

The variant of the Zbot (Zeus) Botnet analyzed in this paper executed several actions and communicated using different protocols. Each action generated a different behavior on the network traffic during the 57 days of execution. From these behaviors, we consider that the C&C channels are the most important, because they typically show very specific features that may be used to detect the botnet. The most relevant are that the C&C channels last for several days continuously, that they are present long before the attacks are started, and that they must be used by the Botmaster.

The C&C channels on this capture were manually detected and labeled, in order to later analyze them automatically. To identify them, we relied on the following assumptions: first, all the traffic comes from the botnet, so it is an attack or part of a communication system. Second, the C&C channels have a periodic behavior [13]. Third, they are usually encrypted [3] since if they are not their purpose can be easily identified. Moreover, their encryption algorithms are usually unidentifiable (unlike TLS). These assumptions along with our 3-tuple analysis described in Subsection 3.1.2, allow us to find out the C&C channels.

The behavioral analysis of botnet traffic deals with the actions, connections and patterns of botnets over time. However, these patterns can be so complex and interdependent that they may only by seen by analyzing a long-term capture [10]. Only a large botnet capture would give time to the patterns and behaviors to emerge.

The basic data structures that we used to analyze the botnet traffic are network flows. A flow is usually defined as all the packets that share the source IP address, the source port, the destination IP address, the destination port and the protocol. However, three more definitions are needed for separating flows: the timeout of the flow, the report time of the flow and the directionality of the flow. The timeout is the inter-packet time that must pass not to consider the next packet as part of the same flow anymore. The report time is how often should the processing application sends information about the flows in the network. The directionality is if one flow contains all the packets for both IP addresses or if one flow is used for each IP. We used the Argus 3.0.6.1 toolkit with its default timeout values and bidirectional flows. A description of the tools and topology is done in Appendix A.

Several different Command and Control (C&C) patterns were found during the analysis of the dataset. The following Subsections describe all of them: the UDP C&C channel, the TCP C&C channel and the HTTP C&C channel.

3.1 UDP C&C Channel

The UDP C&C channel consisted only of UDP flows. These flows had a duration up to several days, but they were sent in groups. This means that instead of being uniformly scattered, they were only sent in groups with a separation of thirty minutes. These groups had between 1 and 175 flows and most of them lasted between one and two minutes. Thirty minutes after one group ended, another one starts, continuing with the same flows. If a flow lasted more than thirty minutes, it appeared in several groups.

Based on this information, two analyses were done. First, an analysis of the changes in the groups during time, and second, an analysis of each flow from its start to its end through multiple groups.

One important characteristic the UDP C&C channel regarding the executable file was that it included a static list of IP addresses to start the channel. It was found because the bot contacted these IP addresses without any prior DNS request.

Another important characteristic of the UDP C&C channel was that its traffic was not plain text. Therefore, we performed an analysis to evaluate if its was encrypted. The verification method that seemed more appropriate was the one implemented by Lyda et al. [11], that uses entropy to evaluate the type of content in malware binaries. The authors worked with malware executable files instead of traffic, but since the traffic can be stored in pcap files, we could extended their work. The program used was ent[1], which computes the entropy, chi-square test, arithmetic mean, the monte carlo value for Pi and the serial correlation coefficient. We compute those values on four set of packets: the packets on the UDP C&C channel, the non-encrypted HTTP packets for the Google requests described in Section 4, the DNS packets and the TCP Actions traffic packet described in Section 3.2.1. The DNS traffic had an entropy of 5.299455 bits per byte, the Google HTTP traffic had an entropy of 6.517970 bits per byte, the UDP traffic an entropy of 7.705928

[1] https://www.fourmilab.ch/random/

bits per byte and the TCP actions had an entropy of 7.961638 bits per byte. According to Lyda et al. the entropy of our UDP and TCP connections corresponds exactly with the entropy of encrypted traffic. At first glance, the behavior of this channel looks similar to a P2P protocol because it consisted of small UDP flows that sent probably encrypted data using the same source port.

The Subsection 3.1.1 describes the group behavior of the flows while Subsection 3.1.2 describes the behavior of each flow individually.

3.1.1 Behavior of the groups of flows

The botnet sent a total of 2,674 groups of UDP flows. To analyze the patterns and relationships on these groups we separate and aggregate the traffic every thirty minutes. For each of these thirty minutes groups of flows, four features were extracted: first, the amount of UDP flows that received a response and therefore are considered *established*. Second, the amount of UDP flows that did not receive a response and therefore are considered *attempts*. Third, the amount of *established* flows that had new IP addresses compared to previous groups. And fourth, the amount of *attempted* flows that had new IP addresses compared to previous groups. Figure 1 describe the relationship between these values. It shows the amount of flows every thirty minutes for each of the fourth features.

The analysis of Figure 1 shows the patterns on the UDP C&C channel. The most visible behavior is the amount of *established* flows, which starts around fifty flows and keeps decreasing until twenty flows approximately. Another behavior is that the amount of *established* and *attempted* flows on each group differs. The groups mostly have a median of 29 *established* flows and 1 *attempted* flow, probably because the botnet had stable computers on the P2P network. However, some special groups had more *attempted* flows than *established*. These special groups are very important to understand the update mechanism of the C&C channel.

The special groups can be seen in Figure 1 as a lonely black circle in the upper part of the Figure. On those moments the botnet is sending the bot a new group of IP addresses to try and add to the P2P network. After these special groups that update the IP addresses, the bot had more *established* flows, but after some time these amount of *established* flows decreases. This can be seen in the Figure as a curve shape. This happens because not all the new IP addresses are active and because some of the active ones become unreachable after some days. The most probable explanation is that the peers in the network are also infected machines that are powered off or clean after some time. Two things can be learned from this behavior. First that the set of IP addresses is updated when the amount of *established* flow is approximately under 23 and second, that the median time for this refreshment of IP addresses is 36 hours.

These moments when the botnet refreshes the list of IP addresses in the C&C channel are characterized by a peak of the four previous features: the amount of *established* flows with new IP addresses becomes greater than 0 (with a median peak of 10), the amount of *attempted* flows with new IP addresses becomes greater than 1 (median of 23), the amount of *attempted* flows becomes grater that 15 (median of 77.06) and the amount of *established* flows becomes grater than 47 (median of 53.50). This threshold of 47 was needed to automatically find out which were the special groups and it was empirically selected by analyzing the histogram shown in Figure 2. This figure shows the amount of *established* UDP flows every 30 minutes. It can be seen that 47 is the first of the values that are more than two standard deviations above the mean ($\mu + 2\sigma$). On a normal distribution it is usually the top 2.4%. Although this is not a normal distribution, the threshold value was good enough to identify those moments. The groups of 30 minutes that have more that 47 flows, are considered top groups and they represent the more active groups in the UDP C&C channel. Later on, these top groups will be compared to the other channels.

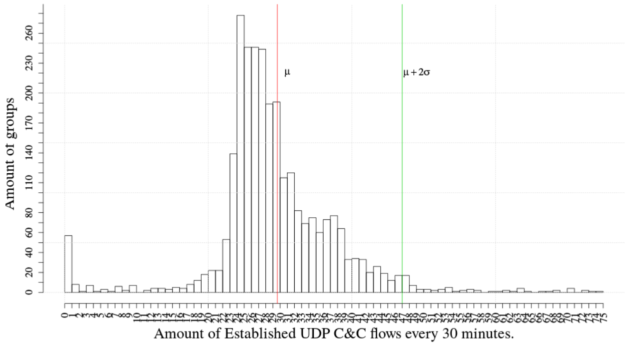

Figure 2: Histogram of the amount of *established* flows every 30 minutes in the UDP C&C channel. Above the $\mu + 2\sigma$ line are the top flows.

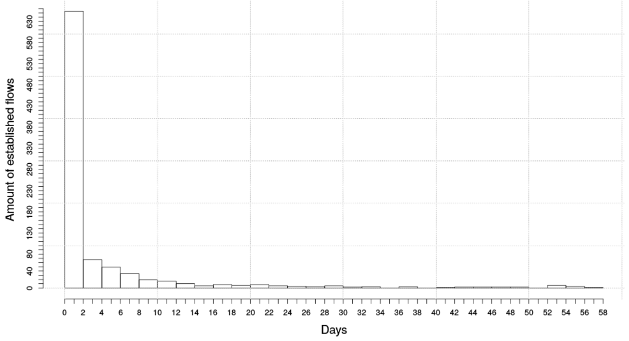

Figure 3: Histogram of the amount of flows that remained *established* at least X days in the UDP C&C channel.

Another characteristic of the UDP C&C channel is how long lasted each *established* flow. For example, the first group of flows contained 18 IP addresses that were hard coded in the binary file. From these 18 IP addresses, only 4 had *established* flows. These 4 IP addresses remained *established* for a minimum of 12 hours and a maximum of 35 days. In summary, from the total of 5,699 unique IP addresses contacted inside the UDP C&C channel, only 1,101 were part of *established* flows (19.31%). From these 1,101 IPs, 75 (6.81%) were active between 10 and 20 days, and 31 (2.81%) were active more than 30 days. Only 11 (0.99%) were active more than 50 days. The only flow that remained *established* during the 57 days was to IP address 85.100.41.9 and destination port 8835. Figure 3 shows a histogram of the amount of days that each flow lasted *established*.

3.1.2 Behavior of each flow individually

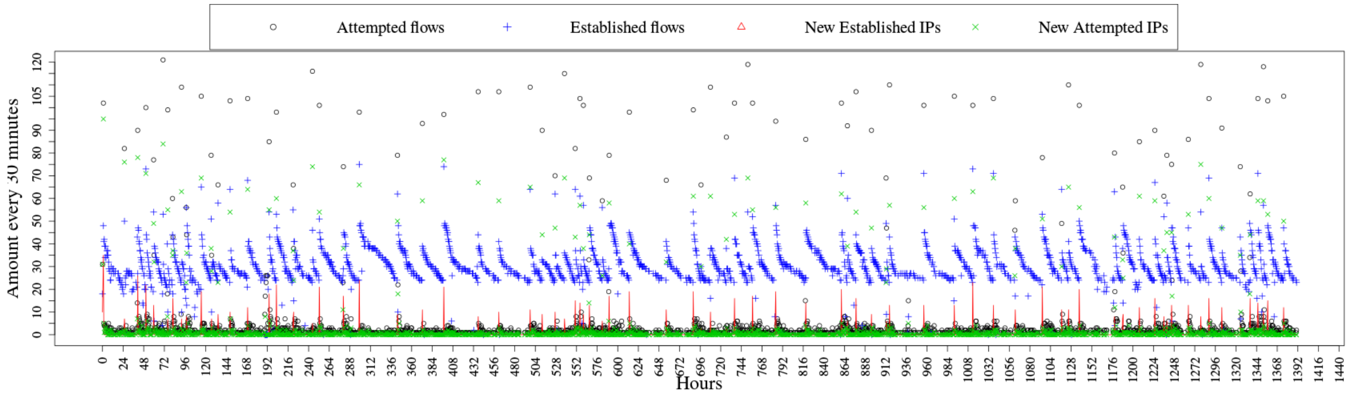

Figure 1: The amount of UDP *attempted* flows, UDP *established* flows, new UDP *established* IP addresses and new UDP *attempted* IP addresses in the UDP C&C channel. Aggregation every 30 minutes.

Besides the behaviors analyzed previously, each IP address also had its own behavior. To find them, the flows were aggregated by the source IP, destination IP and destination port. This 3-tuple ignores the source port to focus on the *service* accessed by the bot. In this way, if the bot creates a new connection to the same service using another source port, the 3-tuple is still able to group them.

The features stored for each 3-tuple are the mean, stdev and median of: the *size* of the flow, the *duration* of the flow and the *time difference* to the previous flow. This last feature was used to find out the periodicity. Using these features it was possible to create the behavioral characteristics of each 3-tuple. The periodicity, size and duration of a 3-tuple with a large amount of flows are the main features that identify it as part of the UDP C&C channel.

The features values of the *established* and *attempted* 3-tuples, and therefore their behavior, was different. Table 1 shows that the *established* 3-tuples have a median periodicity of 30.6 minutes that is stable (stdev of 45.16). On the other hand, the *attempted* 3-tuples have a median periodicity of 310.9m. This means that apart from the groups analysed on the previous Subsection, each *established* 3-tuple also sent a flow every thirty minutes. The table shows that the median duration of the *established* 3-tuples was 37.6 times larger that the *attempted* 3-tuples, because the *attempted* flows did not get an answer. The table also shows that the median size of the *established* 3-tuples doubles that of the *attempted* 3-tuples, because of the amount of responses. Finally, the stdev of the size of the *attempted* flows is low, showing that despite having more than five times the amount of 3-tuples, its size was almost always the same.

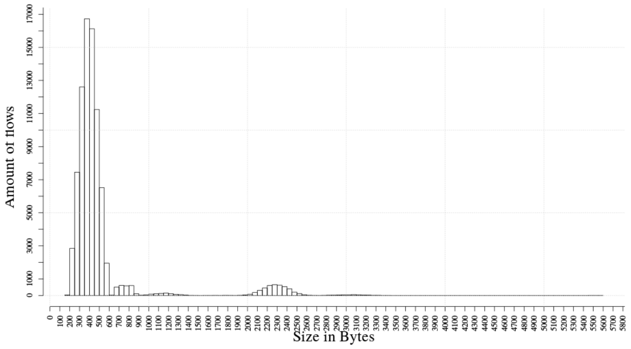

Figure 4: Histogram of the size in bytes of the *established* flows in the UDP C&C channel.

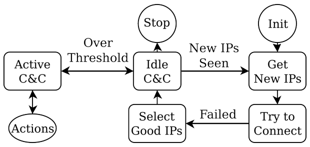

Figure 5: State Model for the UDP C&C channel.

A deeper analysis showed that the *established* flows sometimes had a different behavior. The first difference found was that the stdev of the periodicities was low but not zero. This created a variance of the time differences between 4 and 200 seconds (3.3 minutes) that may be characteristic of the C&C. The second difference found was that the periodicity of the *established* 3-tuples was sometimes broken by a flow with a five seconds time difference. The third difference was that, from time to time, some *established* 3-tuples sent larger flows. These larger flows were related to a change in the state of the C&C from idle to active. Larger flows mean more information going through the channel. For example, one 3-tuple sent 117 flows during 3 days with a size of 350 bytes. Then, it sent 32 flows with a size of 2,500 bytes during 16 hours. The difference can be seen as a small distribution of values around 2,300 bytes in the histogram of the sizes of the *established* flows in Figure 4.

3.1.3 UDP C&C Channel States Model

The behavior of the UDP C&C channel shown in Figure 1 can be modeled as states. The goal was to identify a simple set of states that can represent the transitions found in the traffic. Figure 5 shows the state model created. The boxes represent states and the circles actions. The important states are the process to get new IP address in the P2P and the change to an active C&C. The threshold to go from the Idle state to the Active state is the previously defined value of 47 *established* flows. This model is important to understand the basic behaviors of the channel and it can be used to create better detection algorithms.

3.2 TCP C&C Channel

At the same time that the UDP C&C channel started, the botnet also started connecting using the TCP protocol. When their 3-tuples

UDP Type	#3-tuples	Duration			Size			Periodicity		
		Mean	Stdev	Median	Mean	Stdev	Median	Mean	Stdev	Median
Established	1132	0.282s	0.229	**0.226s**	528.4b	296.07	**431.4b**	40.3m	45.16	**30.6m**
Attempt	5906	0.018s	0.031	0.006s	204.1b	**57.18**	200.5b	411.7m	384.8	310.9m

Table 1: Differences in the duration (seconds), size (bytes) and periodicity (minutes) between the *established* 3-tuples and the *attempted* 3-tuples in the TCP C&C channel.

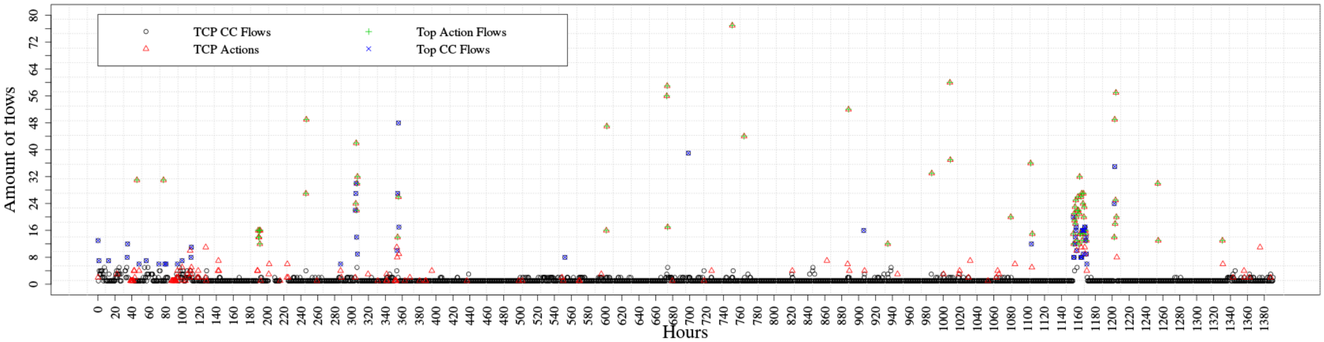

Figure 6: Comparison of the amount of flows between the TCP C&C channel and the TCP action flows (Flows that look like the TCP C&C flows but are not periodic). Aggregation every 30 minutes.

were created, they were identified as a new TCP C&C channel.

The TCP C&C channel was composed of 22 unique *established* 3-tuples (unique IP addresses). They are few compared to the 5,699 IPs of the UDP C&C channel. These 3-tuples lasted from 1 to 56 days, with a median of 28.5 days. Figure 6 shows the behavior of this channel. The flows inside each 3-tuple changed their state several times. That is, instead of being *established* all the time, it was usual that the flows become *attempted* or they simply time out. In contrast with the UDP C&C channel, these changes made the TCP C&C channel more unstable and erratic. For example, the 3-tuple to IP address 155.230.189.121 and destination port 6758 had 63 flows during 1 day, but only 14 (22.2%) were *established* (with a periodicity of 30 minutes). The rest of the flows were not answered and become *attempted*. These changes may indicate that a TCP C&C channel might be more prone to being blocked, filtered or taken down.

The most stable 3-tuple was to IP address 84.59.151.27 and destination port 3285, which lasted 35 days. It had 1058 flows and 789 (74.5%) had a periodicity of 30 minutes.

The TCP C&C was active during the whole capture, with a median of 1 flows every 30 minutes. However, on special moments there were up to 48 flows every 30 minutes. These increases may indicate a more intense communication. It is worth noting that a botnet C&C channel is periodic only when it is idle. When it needs to send or retrieve information, the periodicity is lost in favor of performance.

3.2.1 TCP Actions

The main difference between the UDP and TCP channels is that the TCP channel does not show a strong group behavior. However, we found a new type of TCP flows that we called TCP Action flows. Their 3-tuples are similar to the TCP C&C 3-tuples except that they are not periodic and they usually contain a small amount of flows that transfer a large amount of bytes. These flows are only used on special moments that seems to be related with botnet actions.

To compare of the TCP C&C channel with the rest of the channels we separated and aggregated the traffic in thirty minutes groups. Figure 6 shows a comparison between the amount of TCP C&C flows and TCP Action flows in each of these thirty minutes groups.

Just like in the UDP C&C channel, we identified the top groups. If we consider the distribution of the amount of flows per group, the top groups are the groups that have an amount of flows that is more than two standard deviations from the mean of the distribution. On a normal distribution it is usually the top 2.4%. Although this is not a normal distribution, this ad-hoc limit allow us to focus on the groups that were probably representing actions in the botnet. The Figure also shows these top groups.

From all the 2,670 groups that had at least 1 TCP C&C flow, 1,246 (46.66%) had exactly 1 TCP C&C flow, 333 (12.47%) had more than 1 TCP C&C flow, 142 (5.31%) more than 2 TCP C&C flows. Finally there were 53 (1.98%) top groups because they had more than 5 flows in them.

From the 191 groups that had at least 1 Action flow, 70 (36.64%) had more than 11 and were top groups. On 26 (0.93%) groups, both the TCP C&C and Action flows had top flows, meaning that they could represent moments of important changes in the botnet.

3.2.2 TCP C&C State Model

The TCP C&C channel presented some behaviors that can be modeled as states. Figure 7 shows this state transition model. The boxes represent states and the circles actions. The most important changes occur when the C&C channel becomes active, and when the TCP Actions are sent. The "More flows in group" threshold is the 5 flows threshold defined previously. Other actions are related with this channel, but they are discussed in Section 4.

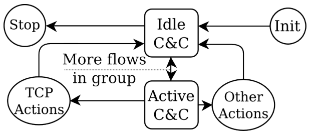

Figure 7: TCP C&C channel state model.

3.3 HTTP C&C Channel

The botnet started a new type of C&C channel after 32 days of operation. It was a HTTP protocol with encrypted payloads that were verified like in the UDP channel. Five IP addresses were used. Table 2 shows a comparison of the characteristics of their 3-tuples. It shows that the IPs had a large amount of periodic flows, in contrast with the TCP and UDP C&C channels, were it was more common to loose the periodicity because of ports being filtered. This was the only C&C channel that had a periodicity different to 30 minutes.

The first two IP addresses in the Table 2, i.e. 95.211.9.145 and 212.124.126.66, belonged to the same C&C sub-channel. The bot sent the same type of GET requests, the same type of responses and it alternately connect to them every few days. Figure 8 shows a comparison of the amount of flows every thirty minutes. It can be seen that the bot connected to the IP address 95.211.9.145 for the first time near the hour 765. After that, the bot switched to the IP address 212.124.126.66 during 9 days. All the responses contained the text "l0sM/OJnke52FQI=". Near the hour 940 the bot switched to the IP address 95.211.9.145. It sent the same GET requests every 5 minutes and got the same answer. After six days (hour 1,060) the bot switched to the IP address 212.124.126.66, this time receiving as answer the text "q+RkdjZAgJsfSj4=". Five days later (hour 1,162), the IP address 212.124.126.66 sent a new special answer of 255 characters to the bot. After this special answer the bot connected several web pages, indicating that the new answer was probably an order. This alternating behavior continued until the end of the capture. In total, the C&C sub-channel sent 58 different GET requests to both of these IP addresses and it received 21 different responses.

The last three IP addresses in Table 2 belong to another C&C sub-channel. They served similar php files and get similar requests. The bot sent POST requests to the IP address 194.28.87.64 for 6 days (hour 910). After those six days the IP stopped answering packets.

The HTTP connections to the IP address 97.74.144.110 were the firsts to use a host name. We could differentiate two different behaviors. First, the bot sent POST requests during 14 hours. After that, the IP address was not contacted for 15 days. Second, the bot tried to use the IP again, but the web page was not working anymore. The answers, then, alternated between a 404 "Not Found" web page (65% of total flows) and not answering at all (15.72% of total flows).

The IP address 62.149.140.209 was contacted for the first time near the hour 780. On that moment, the bot downloaded a binary file from "/cache/abuild.exe" and then it received a 404 "Not Found" web page as an answer for the next five days. After that (on hour 910), the bot started receiving a 500 "Internal Server Error" answer.

3.3.1 HTTP C&C State Model

The behavior of the HTTP C&C channel can be modeled based on the previous analysis. Figure 9 shows the model that represents the most important behaviors. The boxes represent states. The bot can remain on those states. The circles represent actions. The important parts of the model are the *Trying to connect* state and the *Idle* and *Active* states. The first one is responsible for most of the traffic in the channel, since it generated more flows. The last two make the difference between a working and a not working botnet.

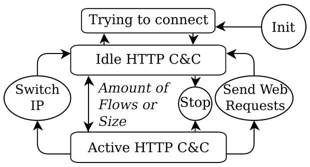

Figure 9: State model of the HTTP C&C channel.

4. COMPARISON OF C&C BEHAVIORS

This section compares all the previous C&C channels to find the behaviors of the botnet. Apart from the C&C channels, the botnet also sent a huge amount of flows to the domain google.com (with and without TLS), sent large amounts of DNS flows, accessed web sites and downloaded executable files. To analyze these actions, the flows were aggregated on thirty minutes groups, and the total size of the group was computed. The comparison between the C&C channels and actions can be seen in Figure 10. From all the C&C channels, only the UDP C&C channel is restricted to its top groups (the top 2.4% of the groups). The vertical lines indicate when a binary executable was downloaded from the web.

By correlating the information in Figure 10 we identified the most important behaviors. They are usually characterized by a peak in the size of most of the C&C channels. Larger flows that deviate from the mean size of a C&C channel indicate more data being transmitted and maybe something different happening. These important moments are shown as numbers in the bottom of Figure 10.

The first moment was at hour 0, when the first group had large UDP C&C, TCP C&C, TCP Actions and the DNS Established flows. From this moment on, the botnet connected to www.google.com every thirty minutes until the end of the capture, probably to know if it has Internet access. The second moment was near hour 45, when there were large flows for DNS requests, TCP Actions, TCP C&C and UDP C&C. There were no visible changes in the traffic, so it could be related with changes in the inner state of the botnet.

The third moment, near hour 120 is important because the UDP and TCP channels stopped sending large flows after five days. From this moment both channels become stable. The fourth moment had large TCP C&C flows, probable sending an order, followed by large DNS requests and finally large TCP Action flows. The fifth moment showed large TCP and UDP C&C flows, then DNS requests, large TCP Actions flows and finally a peak in the requests to www.google.com. The sixth moment showed the same pattern as the fifth one.

The seventh, eight and ninth moments come 10 days after the sixth one and their actions were very important. After a large group of TCP C&C flows, there were large DNS requests, large web access flows and for the first time, three binary download from web pages. The tenth moment was one of the most important ones. It was characterized by large TCP C&C flows, DNS requests, web access and binary downloads. The actions after the binary downloads

IP	# Flows	Periodic Flows	Periodicity	Dur.	URL
95.211.9.145	2,778	99.3%	5 min	26d	http://95.211.9.145/m/IbQCFVVjju9Ob3XaHeN(...)
212.124.126.66	4,398	99.70%	5 min	24d	http://212.124.126.66/m/IbQCFVVjjl9Ob3XaHe(...)
194.28.87.64	56,841	10%	5 min	25d	http://194.28.87.64/sdughwejaskvmomsdv/file.php
97.74.144.110	3,682	13%	4-5min	25d	http://site-serv.com/redir.php
62.149.140.209	24,529	79%	11min	25d	http://www.cam-spa.net/cache/check.php

Table 2: Comparison of the characteristics of the five IP addresses in the HTTP C&C channel. All connections are to port 80/TCP.

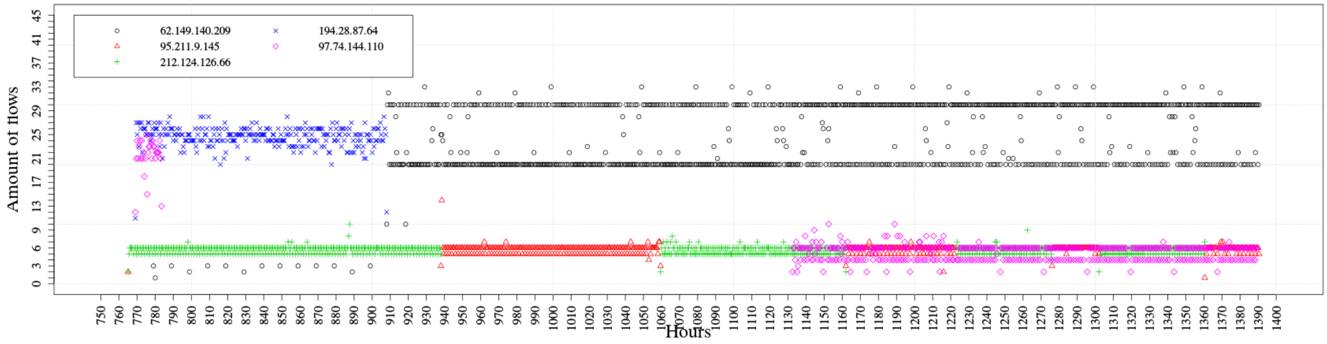

Figure 8: Comparison of the amount of flows for the five IP addresses in the HTTP C&C channel.

were significant since the new HTTP C&C channel was created. Those binaries therefore were a huge update to the behavior. After the new HTTP C&C started there were also huge amount of DNS requests. Interesting enough, the amount and size of the Google requests were also affected. However we don't know why.

At the eleventh moment there was a change in the HTTP C&C. The www.cam-spa.net domain started getting more flows and the HTTP C&C of IP 194.28.87.64 stopped working. That is why it can be seen a huge drop in the size of flows. The eleventh moment is significant because after some TCP C&C flows there are fewer and smaller flows to google.com. The thirteen moment came seven days later and its most important action was to start a new HTTP C&C to IP 97.74.144.110. This generated a lot of new DNS requests that can be seen at the bottom of the Figure. On the fourteen moment and during 5 hours the TCP and UDP C&C channels and the TCP Actions send large flows. The fifteen moment is very similar to the previous one, except that there is only a peak in the UDP C&C, the TCP Actions and the Google requests. The sixteen moment was characterized by a peak in the TCP C&C channel along with a binary web download.

The behaviors analyzed and compared are representative of botnet traffic. However, it is difficult to predict how they could match also normal traffic, because the definition of normal depends on the network analyzed. In our experience, we had not seen any normal application that matched these behaviors, but such a comparison was not in our goals.

4.1 Botnet state model

The analysis of all the C&C channels in the previous Section described new relationships between them. The moments were the C&C channels correlate were also characterized by actions in the network. It is not possible to decide which C&C channel was responsible for the actions, but it is possible to identify those moments. With this information, a general state model for the botnet was created. It is shown in Figure 11. This model state was designed to be simple and to capture the most general state transitions of a botnet. It means that the C&C channel can be Idle or can become active and start doing actions. This basic separation is a important advance in the modeling of botnet states.

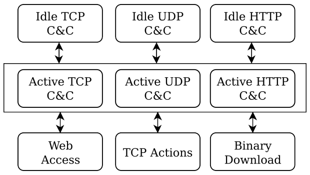

Figure 11: Botnet general state model.

5. CONCLUSION

We precisely identified the UDP, TCP and HTTP C&C channels in a new 57-day long botnet dataset. We analyzed the behavioral characteristics of each of them, including the amount of flows, size of the flows and periodicity to discover when the channels were idle or active, and which type of actions were related to them. The behaviors of the channels were compared to identify the complete botnet decisions. A basic state model was created for each C&C channel and for the complete botnet. We believe that this analysis is an incremental step in the modeling of malware behavior.

The obtained statistics and the state models may be used to create state machines to generate NetFlows similar to the botnet ones. Their characteristics may be controlled to produce new unseen but almost real malware traffic. Also, our analysis may be useful to create new automated action recognition and assisted forensics algorithms. We hope that our conclusions may be suitable for researchers in the A.I. and security fields to test their hypothesis in malware behaviors, to improve take down strategies [7] or even to

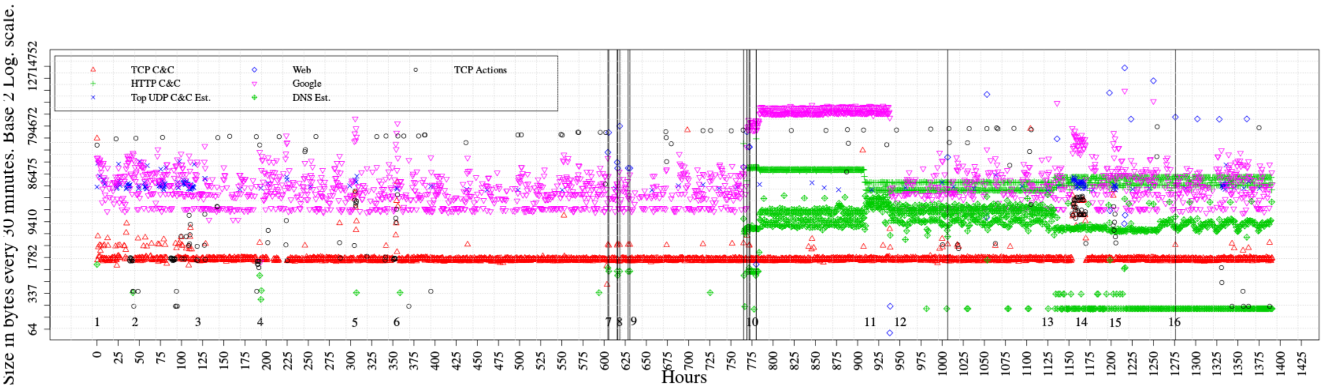

Figure 10: Comparison of the size of the flows for each botnet behavior. Size is in bytes and the scale is logarithmic in base 2. Aggregation time is 30 minutes.

help modeling new malware-based game-theoretical applications [8].

The statistics and thresholds shown in our work are extracted from only one botnet capture, and therefore it is difficult to extrapolate them. However, since we are working on analyzing other botnets captures, we believe that there are common characteristics on the behaviors that can lead to a more general state model of the botnets.

The most important limitations of our work are the fixed time windows of thirty minutes for the analysis of the statistical features, the analysis of only one botnet and the lack of formalization of the state model. Currently we are working on extending this work to other botnet captures and to compare all of them in search for common behaviors between different botnet families.

6. ACKNOWLEDGMENTS

This project was supported by the project of the Czech Ministry of Interior No. VG20122014079.

7. REFERENCES

[1] A. Basil, M. Jose L. David. Periodic Behavior in Botnet Command and Control Channels Traffic. Global Telecommunications Conference. GLOBECOM 2009. IEEE, 1–6. IEEE 2009.

[2] L. Bilge, D. BalzarottiW. Robertson. Disclosure: detecting botnet command and control servers through large-scale NetFlow analysis. ACSAC '12 Proceedings of the 28th Annual Computer Security Applications Conference 2012.

[3] D. DittrichS. Dietrich. P2P as botnet command and control: A deeper insight. MALWARE 2008. 3rd International Conference on malicious and Unwanted Software, 41–48, 2008.

[4] S. Garcia, A. ZuninoM. Campo. Survey on Network-based Botnet Detection Methods. Security and Communication Networks, John Wiley & Sons 2013.

[5] S. KondoN. Sato. Botnet Traffic Detection Techniques by C&C Session Classification Using SVM. Advances in Information and Computer Security. Lecture Notes in Computer Science, Volume 4752. 91–104. 2007.

[6] Z. Li, B. Wang, D. Li, H. Chen, F. LiuZ. Hu. The Aggregation and Stability Analysis of Network Traffic for Structured-P2P-based Botnet Detection. Journal of Networks Vol 5, Issue 5, 517–526, 2010.

[7] Y. Nadji, M. AntonakakisR. Perdisci. Beheading hydras: performing effective botnet takedowns. CCS13 Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 121–132. 2013.

[8] Z. Yin, M. Jain, B. Bošanský, M. Tambe, P. Michal. Game-theoretic Resource Allocation for Malicious Packet Detection in Computer Networks. Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, 905–912, 2012.

[9] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, D. Garant. Botnet detection based on traffic behavior analysis and flow intervals. Computers & Security Journal 39, 2–16, 2013.

[10] J. Jusko, M. Rehak. Identifying Peer-to-Peer Communities in the Network by Connection Graph Analysis. International Journal Of Network Management 39, 1–20, 2014.

[11] R. Lyda, J. Hamrock. Using entropy analysis to find encrypted and packed malware. IEEE Security & Privacy, 40–45, 2014.

[12] M. Rehak, M. Pechoucek, M. Grill, J. Stiborek. Adaptive multiagent system for network traffic monitoring. Intelligent Systems, IEEE, 16–25, 2009.

[13] B. AsSadhan, J.M.F. Moura, D. Lapsley. Periodic Behavior in Botnet Command and Control Channels Traffic. Global Telecommunications Conference,IEEE GLOBECOM, 1–6, 2009.

[14] R. Konrad, S. Guido, L. Tobias, H. Thorsten, L. Pavel. Botzilla: Detecting the ?Phoning Home? of Malicious Software. Proceedings of the 2010 ACM Symposium on Applied Computing (SAC 2010), 1978–1984, 2010.

APPENDIX

A. EXPERIMENT SETUP

The dataset used in this paper was created by the CVUT Malware Capture Facility Project [2] and can be downloaded with the name *CTU-Malware-Capture-Botnet-25*. The MD5 of the executable is e1090d7126dd88d0d1d39b68ea3aae11 and it is reported as a member of the Zbot botnet family[3]. More details can be found on the web page.

[2] `http://mcfp.felk.cvut.cz`

[3] https://www.virustotal.com/en/file/3fc6bef5eac0656be77
f8e96f2b7e08cadb418c11430e8c3d53b33788a93c86a/analysis/