Characterizing Roles and Spatio-Temporal Relations of C&C Servers in Large-Scale Networks

Romain Fontugne
IIJ Research Lab.

Johan Mazel
National Institute of Informatics / JFLI

Kensuke Fukuda
National Institute of Informatics / Sokendai

ABSTRACT

Botnets are accountable for numerous cybersecurity threats. A lot of efforts have been dedicated to botnet intelligence, but botnets versatility and rapid adaptation make them particularly difficult to outwit. Prompt countermeasures require effective tools to monitor the evolution of botnets. Therefore, in this paper we analyze 5 months of traffic from different botnet families, and propose an unsupervised clustering technique to identify the different roles assigned to C&C servers. This technique allows us to classify servers with similar behavior and effectively identify bots contacting several servers. We also present a temporal analysis method that uncovers synchronously activated servers. Our results characterize 6 C&C server roles that are common to various botnet families. In the monitored traffic we found that servers are usually involved in a specific role, and we observed a significant number of C&C servers scanning the Internet.

Keywords

botnet, C&C server, traffic monitoring, Internet traffic

1. INTRODUCTION

Serious cybersecurity threats are often attributed to large networks of infected hosts controlled by criminal organizations, and commonly referred as botnets. The numerous compromised hosts rallying these networks empower criminals to carry out extensive harmful actions, including Distributed Denial-of-Service attacks (DDoS), spam campaigns, click frauds, and data thefts.

In reaction to the severe threats posed by botnets, security software companies, governmental agencies, and the research community have dedicated a lot of effort into botnet intelligence trying to anticipate imminent threats and take countermeasures to neutralize them. In return botnets have been increasingly sophisticated, evading introspection and becoming more resilient to disruptions of key botnet components. This endless cat-and-mouse game between security experts and cyber-criminals has led to an abundant scientific literature, advanced security tools but also very complex and constantly evolving botnets. Continuously monitoring botnets is hence increasingly necessary to survey new mechanisms devised by botmasters and assist defenders for prompt responses to new threats.

WTMC'16, May 30 2016, Xi'an, China

DOI: http://dx.doi.org/10.1145/2903185.2903192

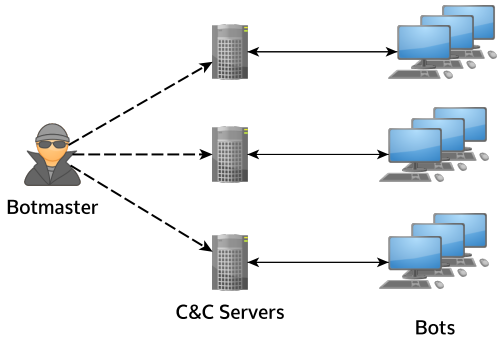


Figure 1: Overview of botnets: the botmaster is indirectly sending orders (e.g. using proxy servers) to the C&C servers that are relayed to the bots when they are connected.

The structure of botnets is typically dissected in three key components (see Figure 1), the botmaster, the Command and Control servers (C&C servers) and the bots. This structure stems from the fundamental mechanisms needed to create and operate botnets. These mechanisms include four stages that are inherent to botnets life-cycle (see [27] for more details on botnet life-cycle): (1) *Conception*: The botmaster designs the botnet regarding its needs and implements corresponding malware. (2) *Recruitment*: The botmaster usually requires substantial resources to execute preeminent attacks. Consequently, the implemented malware infects as many hosts as possible by exploiting vulnerabilities, or deceiving Internet users. (3) *Interaction*: As a consequence of the infection, bots acquire access to the botnet communication channel. This channel is maintained by C&C servers, and it allows bots to signal their presence in the botnet and receive orders from the botmaster. (4) *Attack*: The last stage is the primary goal of the botnet. Depending on the botmaster motivations the bots could perform, for example, DDoS attacks, spam distribution, or click frauds.

In this work, we monitor botnet traffic to study the different types of communications initiated by C&C servers and

their roles in the botnet. The analyzed traffic is captured at multiple measurement points including edge networks, backbone links and Internet exchange points for 5 months. This extensive dataset presents exceptional benefits for the study of botnet behaviors. Indeed, the captured traffic encompasses communications from numerous infected hosts from various botnet families, therefore, providing a wide range of possible botnet behaviors. Nonetheless, monitoring traffic in backbone networks raises certain challenges, the main one being the partial coverage because of high packet sampling rate and routing asymmetry.

The goal of this work is to leverage the potential of data captured on large-scale networks. We devise robust and unsupervised techniques to infer roles of C&C servers, and uncover their spatio-temporal characteristics, namely, C&C servers with similar peers or synchronously operating. The roles of C&C servers are deducted from traffic characteristics that are not bound to a specific protocol or application, hence, also suitable for unknown botnet families. Using C&C roles, we determine when servers are effectively communicating to bots and uncover servers sharing common peers. Finally, we propose a simple correlation technique to identify C&C servers that are activated at the same time.

Overall, our examination of botnet traffic exposes key characteristics of C&C operations. (1) The traffic of servers features 6 distinguishable behaviors that exhibit the roles of C&C servers in botnets. (2) A C&C server is rarely involved in many roles, as different tasks are usually performed by different servers. (3) A large fraction of the C&C servers reported by popular blacklists are scanning Internet hosts, which is to be taken into consideration when estimating botnets size from monitored traffic. (4) Distributed C&C infrastructures are identifiable using the servers spatial and temporal correlations, however, we observe asynchronous bots communications which may be detrimental for botnet detectors assuming bots synchronous behavior.

The remainder of this paper is structured as follows, Section 2 provides details on the collected traffic and C&C blacklists analyzed in this study, and Section 3 exhibits a macroscopic analysis of this dataset. The three following sections expose three analyses that reveal different aspects of botnets life-cycle: Section 4 presents the role identification method and describes C&C roles identified in captured traffic, then, Section 5 depicts uses of the identified C&C roles to investigate servers with common peers, and, Section 6 proposes a correlation technique to cluster C&C servers with similar activities. Section 7 and 8 state the related work and conclude this paper.

2. DATASET

Our analysis relies on two types of datasets. Firstly, botnets are identified using blacklists of C&C servers, then, the botnets behaviors are derived from passively measured data traffic.

2.1 Blacklists

Botnet detection has received a lot of attention in the past. Researchers have proposed numerous techniques to identify infected hosts, ranging from web browser infections [4, 8] and binaries introspection [21, 38, 17] to connection pattern analysis in network traffic [12, 11, 14].

In this article we leverage results obtained from certain of these techniques to monitor botnet infrastructures. Namely, we obtain blacklists of C&C servers from three different organizations: Abuse.ch, Cybercrimetracker, and Spamhaus. An evaluation of most of the analyzed blacklists is presented in [22].

Abuse.ch [1] is a Swiss security site that maintains blacklists for three different types of C&C servers. These blacklists, also known as trackers, report the network activities of malicious binaries executed in a controlled environment. The most active tracker is dedicated to the infamous Zeus crimeware toolkit. Zeus is a trojan horse malware that enables hackers to infect and control hosts connected to the Internet [3]. Originally designed for credentials-stealing, the original Zeus code base have been extensively revamped by numerous threat actors to achieve diverse malicious activities such as DDoS attack, malware dropping, or Bitcoin theft [1, 31]. The malware spreads mainly via spam emails and phishing, and Symantec estimates the number of infected hosts around 4 millions in 2014 [32].

Zeus botnets have been severely disrupted by several coordinated takedown actions from governmental organizations, including the F.B.I. and law enforcement counterparts in several countries [10]. The impact of these takedowns is, however, mitigated by the broad variety of botnets and the constant adaptation of malwares to circumvent detections mechanisms. Thereby, major Zeus botnet takedowns have been subsequently followed by the emergence of new Zeus variants. The family of Zeus malware is considered as the most commonly used financial trojans in 2014 [32], and the abuse.ch Zeus tracker allows us to monitor four well known variants: Zeus, Ice IX, Citadel, and KINS.

Abuse.ch is also providing a tracker for Feodo, a banking trojan that emerged in 2010. This tracker monitors different variants of the malware known as Cridex, Bugat, Geodo, or Dridex. Recent surges of the latest variant, Dridex, have been predominantly targeting corporate accounting services [7], and have been ranked by Symantec as the third most common financial trojan in 2014 [32].

The abuse.ch Palevo tracker monitors an older malware, first appeared in 2008, that is mainly spreading through P2P networks, instant messaging and removable drives. This malware is also known as Rimecud, Butterfly bot and Pilleuz.

Cybercrimetracker[2] is another security site that tracks malware activities, and reports the C&C IP addresses for various malwares and their variants, including, Zeus, Citadel, Kraken, Pony and Solar.

The **Spamhaus**[3] Botnet Controller List (BCL) is a block list service that reports the C&C servers detected by Spamhaus. This list does not advertise the malware family associated with the reported IP addresses, however, Spamhaus reported in the past that BCL monitors numerous malwares, including most of the ones reported by abuse.ch and cybercrimetracker [29].

We collected all IP addresses reported by abuse.ch, cybercrimetracker, and Spamhaus BCL from November 1^{st} 2014 to March 31^{st} 2015. Table 1 summarizes the number of reported IP addresses for each blacklist (see the *Interaction* row in Table 1).

In addition to these C&C blacklists, we also retrieved two blacklists reporting Internet abuses and scans (see the *Re-*

[1] https://www.abuse.ch/
[2] http://cybercrime-tracker.net/
[3] https://www.spamhaus.org/bcl/

Table 1: Blacklists used in this article. Phase is the corresponding stage in botnet life-cycle. #Reported is the total number of IP addresses reported in the blacklists. #Detected is the number of IP addresses from blacklists that are identified in traffic traces and #Peers is the number of IP addresses communicating with blacklisted hosts. Blacklists from abuse.ch are labelled with *(ch)*.

Phase	Blacklist	#Reported	#Detected	#Peers
Recruit.	OpenBL	12448	8583	151344
	Honeypot	12265	8620	312237
Interaction	Feodo (ch)	443	136	40798
	Palevo (ch)	69	15	4249
	Zeus (ch)	1491	349	98359
	Pony	142	110	24769
	Mailer	31	24	36313
	Backdoor	32	20	111
	Kraken	11	4	452
	Phase	12	4	2
	ZeuS	227	141	46357
	Citadel	60	29	17761
	Solar	54	40	12778
	Stealer	16	13	45
	Betabot	27	18	13656
	WSO	11	9	1973
	Spamhaus	1845	442	111767

cruit. row of Table 1). These two extra lists contain IP addresses of scanners looking for vulnerable hosts or trying to remotely log in Internet hosts. Since botnets behave similarly in their recruitment phase, we take advantage of these two blacklists to validate our study on C&C servers behaviors.

The **OpenBL**[4] project monitors Internet abuses from 39 locations around the world. The resulting blacklist contains IP addresses of hosts attempting bruteforce attacks and scans on certain well-known services, such as, email protocols (e.g. SMTP, POP3, IMAP), remote login (e.g. SSH, Telnet) and web services (e.g HTTP, HTTPS).

The second blacklist reporting abuses is compiled from the observations of a private **Honeypot**, thus, provide malicious IP addresses opening suspicious connections or trying to propagate malwares.

2.2 Data Traffic

We capture Internet traffic to investigate the connection patterns of C&C servers and characterize their behaviors. The analyzed traffic datasets consist of NetFlow and sFlow data captured at multiple vantage points in Japan. Table 2 depicts all the considered measurement points along with the sampling rate applied, the total number of bytes accounted in IP headers, the number of captured packets and the duration of the traces in days.

The vantage points are scattered at different locations in the Internet infrastructure, so we can observe detailed traffic of edge networks and more coarse-grained traffic at the core of the network. Namely, we monitor traffic between a major Japanese university campus and the Internet, and,

[4]https://www.openbl.org/

Table 2: Characteristics of measured traffic. *Type* of link where the traffic is captured, *sampling* rate at which packets are captured, total *#Bytes* reported by captured IP headers, *#Packets* collected, and *#Days* of the capture.

Name	Type	Sampling	#Bytes	#Packets	#Days
Uni1	Access	1/512	11.8TiB	13361.3M	126
Cloud1	Access	1/2048	80.9GiB	96.1M	142
Cloud2	Access	1/2048	138.3GiB	125.6M	141
BB1	Backbone	1/8192	370.5GiB	563.5M	139
BB2	Backbone	1/8192	198.7GiB	354.7M	141
BB3	Transit	1/4096	596.2GiB	2678.7M	130
IXP1	Exchange	1/8192	159.0GiB	224.0M	70
IXP2	Exchange	1/32768	739.1GiB	821.2M	117

the two Internet access links of a research cloud, hereafter respectively referred as Uni1, Cloud1 and Cloud2. The monitored core infrastructures consist of two Internet Exchange Points, IXP1 and IXP2, two backbone links in an academic network and one transit link between the same network and a commercial ISP, hereafter referred as BB1, BB2, and BB3.

Overwhelmed by the amount of data transmitted through the monitored infrastructure, we can only capture a fraction of the traffic. Consequently, our collectors are set to capture only one out of N packets transmitted on the network interface. The sampling rate, $1/N$, differs from one vantage point to another, for instance, traffic collected at edge networks is sampled with rates varying from 1/512 to 1/2048, while sampling rates for backbone links and IXPs range from 1/4096 to 1/32768 (see Table 2). These settings allow us to thoroughly monitor the three edge networks, Uni1, Cloud1 and Cloud2, and obtain coarse observations of botnets in backbone networks.

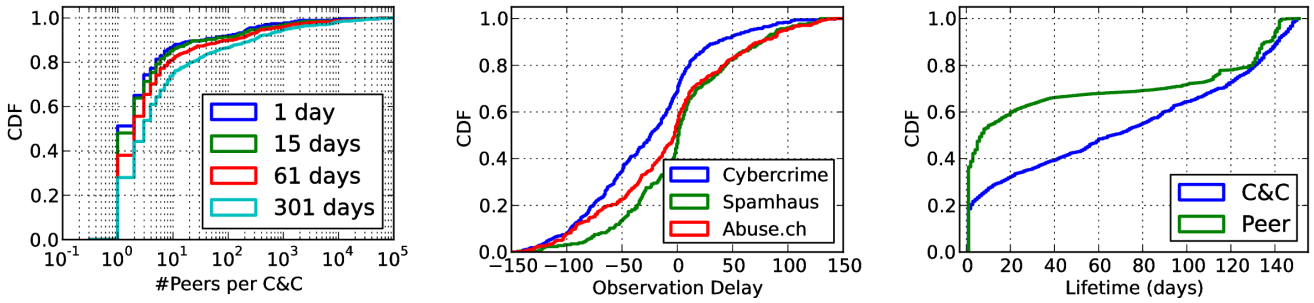
For all vantage points, we intended to continually capture traffic from November 1^{st} 2014 to March 31^{st} 2015, but due to the vast amount of data and hardware arbitrary issues, data loss is inevitable. On average we collected 126 days of traffic for each vantage points which accounts for more than 14TiB of traffic in total.

3. MACROSCOPIC OBSERVATIONS

Our analysis starts with an overview of the characteristics of the blacklisted IP addresses found in the monitored traffic. These observations aim to answer essential questions, such as: How many blacklisted IP addresses appear in the monitored traffic? How many hosts are communicating with these addresses? Are blacklisted IP addresses promptly reported? What is the average lifetime of blacklisted hosts? To answer these questions, we extract every flow corresponding to the IP addresses reported in the blacklists and inspect basic features of these flows.

3.1 Peer Inference

Out of the 4101 unique blacklisted IP addresses, we found 864 of them in the traffic data. In Table 1, the *#Detected* column summarizes the number of C&C servers found for each blacklist. The next column, *#Peers*, represents the number of IP addresses communicating with the identified blacklisted IP addresses, in total we found 136407 unique peers for blacklists corresponding to the Interaction phase. A large fraction of the detected C&C servers are from the Zeus malware family and its variants (i.e. Citadel), hence,



(a) Distribution of the number of peers per C&C server.

(b) Distribution of the observation delay per C&C server. Namely, the interval of time between the first monitored traffic timestamp and C&C first blacklisted date.

(c) Distribution of the active time period of C&C servers and contacted peers.

Figure 2: Overview of C&C servers and peers identified in blacklists and monitored traffic.

confirming that this malware is still very active. These are also the C&C servers that have the most peers, followed by the Feodo malware family.

We found around 200 peers contacting numerous C&C servers across different malware families. Using reverse DNS we confirmed that these hosts are part of two research projects crawling websites or scanning Internet hosts in the whole IP space. As the traffic emitted by these hosts is unrelated with the malicious activity of C&C servers, we remove these hosts from our dataset thus they are not affecting the following results.

3.2 Number of Peers

Figure 2a depicts the distribution of the number of peers for each C&C server. The number of peers is calculated using a time window centered on the C&C server reported dates. These distributions are computed using different time windows. Thereby, using 1-day time window the number of peers is the total number of peers contacted during the dates reported by the blacklists. A time window of 15 days gives the number of peers contacted within a week before or after the blacklisted dates, and a time window of 301 days gives the total number of peers for the monitoring time period. With a 1-day time window we observe 269 C&C servers including 138 servers (i.e. 51%) with only one peer. Using larger time windows permits to capture more C&C servers and more peers. With the 301-day long time window, we observe 790 C&C servers of which 221 have only one peer.

The number of C&C servers with more than 100 peers increases from 21 to 106 using a time window ranging from 1 day to 301 days. Consequently, analyzing traffic only during the blacklists reported dates would miss valuable traffic from C&C servers. We have not found a characteristic time window length that could identify most peers without covering the entire measurement time period, so in the rest of the paper we employ the maximum time window (i.e. 301 days) to identify peers. In these experiments the server with the maximum number of peers contacted 83812 unique IP addresses.

3.3 Observation Delay

Prompt reports are essential for efficient blacklists. Reporting a C&C server that have already contacted all its bots is of little help to block botnet activities. However, the previous subsection reveals that C&C servers converse with Internet hosts during periods of time that are not reported in the blacklists. To measure this temporal aspect of blacklists we define the observation delay of a blacklisted IP address as the time difference between the timestamp of the first captured flow including this IP address and its first reported date. Let $obsDates(a)$ be the sequence of timestamps when flows to, or from, the IP address a are observed and $blDates(a)$ the sequence of timestamps when a is blacklisted, hence, the observation delay for a is defined as:

$$\delta(a) = min(obsDates(a)) - min(blDates(a)).$$

Figure 2b depicts the distribution of the observation delay for the three organizations providing blacklists. The median delay for both Spamhaus and Abuse.ch is around 0, meaning that half of the IP addresses are reported on the same day or before we observe the corresponding traffic. Overall, the mean observation delay for Spamhaus (3.2 days) is higher than the one for Abuse.ch (-6.9 days) and Cybercrime (-27.1 days), hence, Spamhaus is better-suited for prompt actions against C&C servers. Cybercrime, however, is reporting IP addresses with a substantial lag. These results are in accordance with the *reaction time* results of the blacklist evaluation study presented in [22].

3.4 C&C and Peers Lifetime

Another key temporal aspect of botnets is the lifetime of the different botnet elements. In this study, the lifetime of the C&C servers and bots is obtained from the monitored traffic between peers and C&C servers. The lifetime of a C&C server is defined as the time difference between the server first and last observed flow to any peer. The lifetime of a peer is defined as the period of time that has passed between the peer first and last connection to any C&C server. Figure 2c depicts the lifetime distributions of C&C servers and peers. Both distributions feature a bimodal shape, where the first mode is less than one day and the second mode is around 140 days, meaning that C&C servers and peers are either very short or very long lived. For instance, 50% of the peers lifetime is less than a week

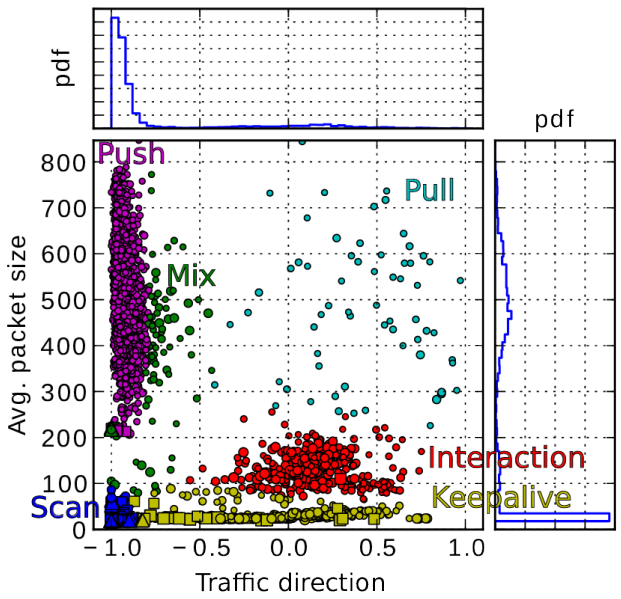


Figure 3: Traffic direction and average packet size computed every hour for the monitored C&C servers. Point shapes represent the number of peers contacted during the corresponding hour. Circles mean less than 100 peers, squares are between 100 to 1000 peers and triangles stand for more than 1000 peers. Point colors indicate the cluster identified with the Gaussian mixture model.

and 20% of them have a lifetime greater than 4 months. Intuitively the sampling rates used to capture the traffic (see Table 2) is a bias against long-lived hosts. Section 4.2 also reveals that C&C servers scanning the IP space is a main cause for these short-lived peers.

The average C&C lifetime (i.e. 68 days) observed in this study is fairly close to the average C&C *uptime* values reported by Gañán et al. in [10]. Their study employs similar C&C blacklists (Abuse.ch and Cybercrime trackers) but defines server uptime as the period of time between the C&C detection time and the time it is taken down. That approach relies only on the dates provided by the blacklists and is thus orthogonal to our definition of C&C lifetime which relies only on the network flows timestamps, but the two approaches yield consistent results.

Summary: As reported in [22] using sandboxes, we found significant delays between the time IP addresses are blacklisted and the time we observe them in the traffic. Therefore, bots are reaching C&C servers in an asynchronous manner, and we cannot rely on the blacklisted dates to monitor bots traffic. In the traffic we also observe two characteristic lifetimes for bots and C&C servers, they are either short or long-lived.

4. C&C ROLES

We pursue our analysis of botnets traffic by inspecting two discriminative quantities that reveal the distinct actions taken by C&C servers:

Average Packet Size (APS) is simply the mean packet size for all flows sent or received by a C&C server. Formally, the average packet size of a C&C server x is defined as:

$$APS(x) = \frac{RX_{byte}(x) + TX_{byte}(x)}{RX_{pkt}(x) + TX_{pkt}(x)}$$

where $RX_{byte}(x)$ and $TX_{byte}(x)$ (resp. RX_{pkt} and TX_{pkt}) are the number of received and transmitted bytes (resp. packets) by the C&C server x. Traffic with large APS highlights data transfers, whereas small APS values are the evidence of signaling traffic.

Traffic Direction (TD) reveals the course of bytes exchanged between C&C servers and peers. Namely, the traffic direction of a server x is defined as:

$$TD(x) = \frac{RX_{byte}(x) - TX_{byte}(x)}{RX_{byte}(x) + TX_{byte}(x)},$$

This metric ranges from 1 to -1. Values close to 1 mean that data is sent from the peers to the server, whereas values close to -1 mean that the data is sent from the server to the peers.

The hourly average packet size and traffic direction for each C&C server observed in our dataset are displayed in Figure 3. Each point in this figure represents one hour of traffic for one C&C server. Inactivity periods are not displayed as the average packet size and traffic direction are undefined if a server receives or sends no data.

Prominent clusters are visually identifiable in Figure 3. For example, a large number of points are aggregated around $TD = -1$ and $APS > 300$, this cluster emphasizes data sent from the C&C servers to the peers, this could be either commands or binary updates sent to the bots. Another interesting group of points is the horizontal cluster along $APS = 40$, which highlights signalling traffic between the server and the peers that could be the botnets heart-beat traffic. Because the visual identification of these clusters is tedious and error-prone, we devise a systematic approach to identify them and provide an interpretation for each identified cluster.

4.1 Roles Identification

The visual clusters of Figure 3 reveal different roles assumed by the monitored C&C servers. We systematically identify these roles using the two proposed metrics (average packet size and traffic direction) and a Gaussian mixture model. The collected traffic is split in 1-hour time bin, and the two proposed metrics are computed for each C&C server, hence we obtain a sequence of APS and TD values for each server. These values are analyzed by means of a Gaussian mixture model, meaning that each component (i.e. cluster) is represented by a centroid and a full covariance matrix that are determined with the Expectation-Maximization (EM) algorithm [16].

A crucial parameter for the Gaussian mixture model is the number of components present in the data. To correctly set this parameter, we try the Gaussian mixture model with different parameter values and find the one that best fits our dataset. The quality of the resulting models is evaluated using the Bayesian Information Criterion (BIC). Models producing low BIC statistics are preferred as they feature a better data fitting. Figure 4 depicts the BIC values for various models with a distinct number of components. Models with less than 6 components produce high BIC values compared to those with a number of components ranging from 6 to 11. The best BIC score is obtained with 9 components, nonetheless, the improvement over the models with 6 to 11

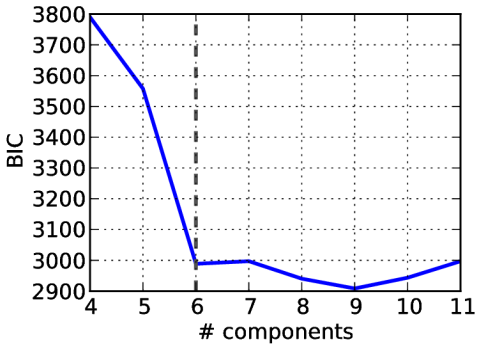


Figure 4: Identification of C&C roles: Evaluation of the Gaussian mixture model with a different number of components. The Bayesian Information Criterion (BIC) is used to estimate the models fit. The dashed line indicates the selected parameter.

components is negligible. The result with 6 components is particularly attractive as it gives a satisfactory description of the data while keeping the model fairly simple. Consequently, we define 6 C&C roles based on these results.

Colors in Figure 3 highlight the different components identified by the Gaussian mixture model. Our interpretation of these components is based on their centroid position, the number of peers for each C&C server and port information retrieved from the C&C traffic. The labels and interpretations of the 6 roles are as follows:

Scan: The component containing the largest number of C&C servers is represented in Figure 3 by blue points. This is a very dense component (see the probability density function in Figure 3), and the position of its centroid ($TD = -0.99$ and $APS = 23$) exhibits peculiar traffic features. The corresponding traffic is strongly asymmetric, indeed, packets are sent solely by C&C servers, and the traffic is exclusively composed of packets with no payload. Furthermore, we found that 94% of the C&C servers assigned to this component contacted more than 1000 peers within one hour, with the maximum observed value being 23113 peers in one hour. Considering our monitoring sampling rate, these C&C servers are undoubtedly contacting a very large number of peers but not transferring data to them. These observations are strong evidences of probing traffic found in the recruitment phase of botnet life-cycle, hence this component hereafter refers to scanning activities. Investigating corresponding traffic reveals that scanners are targeting a vast number of services, but we found that a lot of the scans are probing proxy servers (port 3128 and 8080).

Keepalive: The other component representing very small packets is, on the contrary, spanning through various TD values (see the yellow points in Figure 3). The centroid of this component ($TD = -0.16$ and $APS = 30$) is close to a null traffic direction, meaning that the traffic is equally sent from servers and bots. This is typical of signaling traffic employed by the keepalive (or heartbeat) mechanisms found in the interaction phase of botnet life-cycle.

Interaction: The red component of Figure 3 also exhibits balanced traffic direction, but the average packet size for this component is notably higher. The centroid ($TD = 0.15$ and $APS = 130$) indicates that packets contain an average payload around 100 bytes and this data is equally sent from servers and peers using usually SSH or HTTP (port 22 and 80). This exchange of small messages is associated to the interaction phase present in botnet life-cycle where servers send commands and maintenance operations to peers.

Pull: The sparse cyan component on the top right hand side of Figure 3 is the component with the highest TD values. Its centroid ($TD = 0.40$ and $APS = 459$) stands for traffic composed of significantly large packets and is sent primarily from peers to C&C servers. These observations evoke servers retrieving sensitive data from infected hosts which is part of the attack execution phase of botnet life-cycle. Traffic corresponding to this component is exclusively sent through HTTPS (port 443). Hence, these connections are encrypted and usually able to pass through firewalls.

Push: The other component standing for data transfer is the magenta component of Figure 3, located close to the left-top corner (centroid, $TD = -0.92$ and $APS = 507$). Here the data is transferred from the servers to the peers, and the wide range of APS values observed in this component suggests that servers send both small and very large files. Assuming the regular Ethernet maximum transmissible unit (MTU) of 1500 bytes and empty TCP acknowledgment packets (20 bytes), points around $APS = \frac{1500+20}{2} = 760$ reveal maximum-size packets sent from servers to peers. This role can be observed in both the recruitment and interaction phases of botnet life-cycle, for example in the case of new infections or binary updates. The port information of corresponding packets indicates that this traffic is solely sent on port 80.

Mix: The last component identified by the mixture model, green on Figure 3 is located at the intersection of the pull, push, interaction and scan components (centroid $TD = -0.81$ and $APS = 361$). The role underlying this component is unclear as it seems to be a mixture of the different roles. The corresponding traffic, however, is apparently composed of SIP packets (destination port 5060). Further investigations indicate that this results is biased by a few large SIP scans and the rest of the points in this component stand for various types of traffic.

4.2 C&C Role-based Clustering

The six roles described above reflects hourly activities of C&C servers, we now investigate the enrollment of a single server in different roles and identify servers with similar role changes over time. We devise a hierarchical clustering technique to group C&C servers playing similar roles and uncover common patterns across different servers.

The various roles associated to a server are summarized in a 6-dimensional feature vector where each dimension represents the ratio of time spent for a certain role. Thereby, all dimensions range in $[0, 1]$, 0 means that the server never played the corresponding role and 1 means that we observed the server playing only the corresponding role. Each server behavior is thus described by a 6-dimensional vector and the dissimilarity of servers is measured using the Euclidean distance. The proposed approach is an agglomerative hierarchical method that sets apart servers in their own cluster then recursively merges similar clusters as long as the determined linkage criterion is satisfied. Namely, we implement the Ward's minimum variance criterion to control clusters coherence at each merging step. Figure 5 depicts for each

Table 3: Results of the C&C role-based hierarchical clustering. The eight partitions are listed along with the number of corresponding C&C servers, the average total number of unique peers per C&C, the average observed time in hours, and the ratio of played roles. Each mean value is reported with the corresponding standard deviation.

	#C&C	#peers	#hours	scan	keepalive	pull	push	mix	interaction
Partition 1	**113**	2481.58 ± 7378.05	$\mathbf{6.49 \pm 19.20}$	$\mathbf{1.00 \pm 0.01}$	0.00 ± 0.01	-	-	-	-
Partition 2	5	176.00 ± 145.05	40.40 ± 83.16	0.02 ± 0.04	$\mathbf{0.98 \pm 0.04}$	-	-	-	-
Partition 3	7	$\mathbf{18855.71 \pm 33010.80}$	57.14 ± 135.84	$\mathbf{0.50 \pm 0.05}$	$\mathbf{0.47 \pm 0.04}$	-	-	-	0.03 ± 0.08
Partition 4	8	8795.62 ± 13107.75	11.00 ± 10.36	$\mathbf{0.73 \pm 0.07}$	0.19 ± 0.12	-	0.02 ± 0.06	0.05 ± 0.12	0.01 ± 0.02
Partition 5	5	8915.00 ± 12093.26	$\mathbf{485.80 \pm 534.32}$	0.05 ± 0.11	-	-	$\mathbf{0.89 \pm 0.12}$	0.05 ± 0.05	-
Partition 6	4	13998.00 ± 11988.52	8.00 ± 4.97	-	-	0.06 ± 0.07	0.08 ± 0.10	$\mathbf{0.86 \pm 0.18}$	-
Partition 7	3	596.00 ± 995.93	19.67 ± 32.33	-	-	$\mathbf{0.95 \pm 0.09}$	-	0.02 ± 0.04	0.03 ± 0.05
Partition 8	4	235.00 ± 329.67	93.25 ± 143.75	0.02 ± 0.03	0.02 ± 0.04	0.14 ± 0.24	-	0.00 ± 0.01	$\mathbf{0.82 \pm 0.23}$

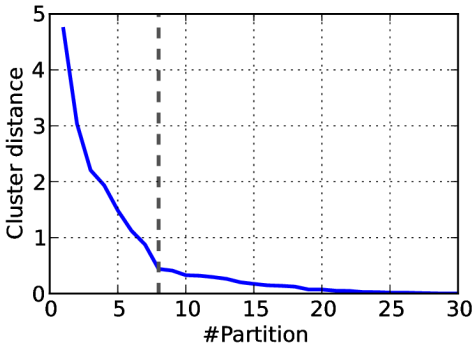


Figure 5: Role-based C&C hierarchical clustering: Cluster distance exhibit the coherence of partitions at different agglomeration levels. The dashed line indicates the selected threshold.

merging step the clusters distance in terms of the Ward's objective function. Lower cluster distance values emphasize a better partitioning of the C&C servers and the knee observed for 8 partitions represents the best trade off for a low number of coherent partitions.

Each identified partition exposes a set of roles that are commonly played by groups of C&C servers. The partitions are presented in Table 3 along with the roles, the number of C&C servers, and the number of peers they represent.

Partition 1: The largest partition in terms of number of C&C servers contains hosts exclusively enrolled in scanning activities. The average observation time for these servers is particularly short (6.49 hours) meaning that servers are performing a single scanning activity then are idle. Interestingly, most of these scans have a limited scope (average of 2481 peers) and seems to target specific hosts as only 1 of the 113 servers is reported by our Honeypot and none are identified in OpenBL.

Partition 2: This partition groups servers that are mainly associated to the keepalive role. These servers feature intermittent communications with a low number of peers. Figure 6a depicts the number of bytes and peers over time for a server assigned to this partition. Although constantly reported by Spamhaus over three months, we found in captured traffic that this server is sparingly communicating with a few peers. Notice that this type of traffic is particularly difficult to monitor in backbone networks due the employed sampling rates.

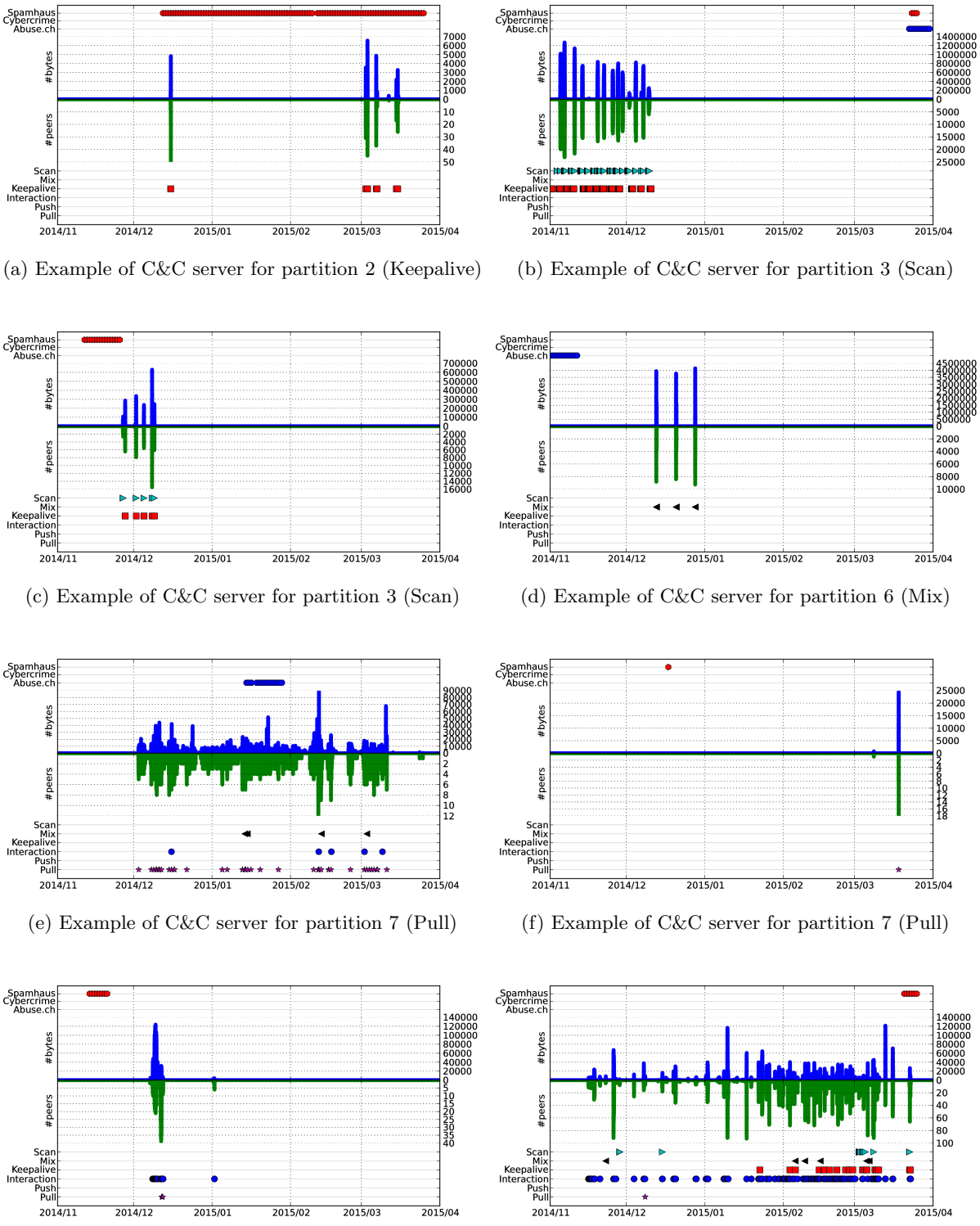
Partition 3: The servers identified in this partition are involved in large scale scans and 4 out of the 7 servers are also reported by OpenBL or the Honeypot blacklists. This partition includes the server with the maximum number of unique peers over the monitoring time period (i.e. 83812 unique peers) which is depicted in Figure 6b. The scans initiated by this server are observed in the traffic and reported by the Honeypot three months before it is reported as a C&C server by the blacklists. Meaning that this host was compromised several months before being included in the C&C infrastructure. Figure 6c illustrates the activity of another server from partition 3, which is on the contrary probing hosts just after being reported by Spamhaus. This suggests that the server was not taken down, but instead the attackers have changed the function of this server after being detected by Spamhaus. Although servers in this partition are assigned to both scan and keepalive roles, we found that in this partition the two roles always appear consecutively and the scans responses that fall in the next time bin are misclassified as keepalive.

Partition 4: This partition is composed of scanners sharing features with those from Partition 1, their lifetime is particularly short and their number of peers can be significant, but differ in the other roles played. Servers in partition 4 are occasionally involved in other roles, we however found no common patterns in the sequence of roles played by these servers.

Partition 5: Servers in partition 5 are distinguished by their very long lifetime and are mainly assigned to the push role, meaning that they send data to peers. We also found that 3 out of these 5 servers are reported by the Phishtank website, hence evincing the threat of transferred binaries. The inspection of the mix roles intermittently observed with these servers reveals that, in some cases, peers are also sending data to the servers. The average number of peers for this partition is significantly affected by one server involved in large scanning activities, the average number of unique peers (1949 peers) for other servers is significantly lower.

Partition 6: The few servers primarily classified with the mix role are clustered in partition 6. Figure 6d depicts the activity of the prominent server found in this partition. The three observed peaks are exactly one week apart from each other, and the corresponding traffic consists only of UDP packets (port 5060, SIP) sent from the server ($TD > 0.99$). Furthermore, the large number of peers, 26179 unique peers in total and the average payload size ($APS = 220$) suggests that this server is also scanning the IP address space, but with UDP packets carrying a certain payload data.

Partition 7: This partition contains servers retrieving



(a) Example of C&C server for partition 2 (Keepalive)

(b) Example of C&C server for partition 3 (Scan)

(c) Example of C&C server for partition 3 (Scan)

(d) Example of C&C server for partition 6 (Mix)

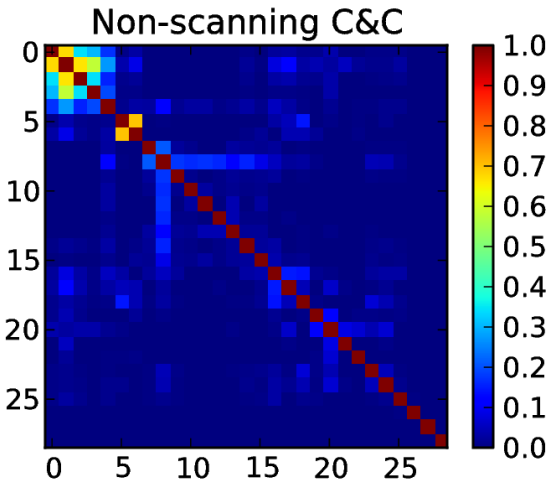
(e) Example of C&C server for partition 7 (Pull)

(f) Example of C&C server for partition 7 (Pull)

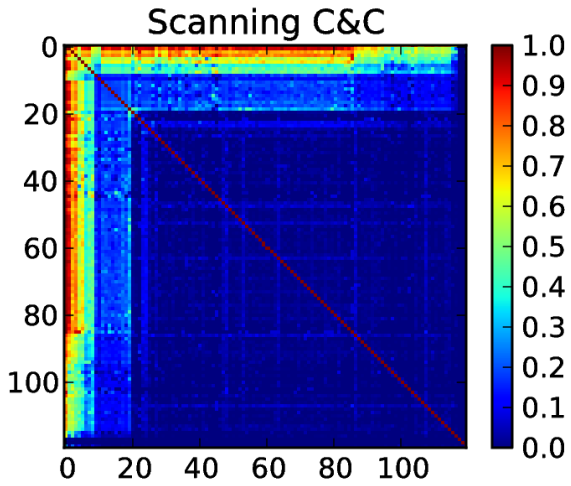
(g) Example of C&C server for partition 8 (Interaction)

(h) Example of C&C server for partition 8 (Interaction)

Figure 6: Examples of C&C activities. In each figure, points on the top three lines represent the time and blacklists when the server has been reported. The blue and green plots respectively indicate the number of transmitted and received bytes and the number of contacted peers. Both metrics are given for one hour time bins. The six bottom lines depicts the roles assigned to the server.



(a) Spatial overlap for non-scanning servers, i.e. Push, Pull, Interaction, Keepalive roles.



(b) Spatial overlap for servers involved in scanning activities, i.e. Scan role.

Figure 7: Pairwise comparisons of C&C servers in terms of spatial overlap. Non-scanning C&C roles (a) and scan role (b) are handled separately.

data from peers (i.e. pull role). The number of peers per hour for these servers is fairly low (see Figure 6e and 6f). Nevertheless, one of these servers accounts for a total of 1745 unique peers over the entire monitoring period (see Figure 6e). This server is continually receiving data from distinct peers over three months in the form of HTTPS traffic and is intermittently interacting with some hosts. The two other servers have a very short lifetime as all their peers are synchronously contacting the server within the same one hour time bin (see Figure 6f).

Partition 8: Servers mainly labelled with the interaction role are clustered in Partition 8. These servers usually feature a low number of peers but are active for an extended period of time. Figure 6g exhibits the activity of one of these servers, the server is constantly active for a week and then disappears. Other roles are usually played by these servers, for example, Figure 6h depicts a server involved primarily in interaction and keepalive roles.

Summary: Using simple metrics and a Gaussian mixture model, C&C servers traffic is clustered into 6 distinct behaviors. These behaviors reveal the roles of servers in botnets and are in accordance with previously reported botnet lifecycles [27, 20]. We found that servers are usually involved in a single role and the contacts with bots can span over long periods of time in an asynchronous fashion (e.g. Figure 6e and 6h).

5. SPATIAL OVERLAP

C&C servers maintained by the same botmaster are potentially contacting common peers over time as one server may be replaced or share its load. In our dataset this results in servers with common peers, but only for servers that are not scanning the IP space. Scanners inherently contact a large number of hosts but these peers are not necessarily infected, thus cannot account for botnet members. Consequently, we investigate peers overlap, hereafter referred as spatial overlap, for different servers while they are not involved in scanning activities.

Let P_x and P_y be the set of peers contacted by server x and y while they are neither assigned to the scan or mix roles, then their spatial overlap is defined as:

$$s(P_x, P_y) = \frac{|P_x \cap P_y|}{min(|P_x|, |P_x|)},$$

where $|A|$ is the cardinality of A, and $A \cap B$ is the intersection of the two sets. The spatial overlap ranges in [0,1], 0 means that the two servers have no peers in common, and 1 means that all peers of one server are a subset of the other server peers.

5.1 Non-scanning C&C

Figure 7a illustrates the spatial overlap computed pairwise for peers of non-scanning C&C servers, i.e., peers contacted when servers roles are classified push, pull, interaction or keepalive. If a server is involved in both scanning and non-scanning activities then only its peers from non-scanning activities are taken into account. The two prominent clusters at the top-left corner of Figure 7a exhibit two groups of servers with common peers.

The largest group contains 5 C&C servers with an average spatial overlap $\bar{s} = 0.37$. C&C server 1 has a central role in this cluster as its spatial overlap with other servers is significantly high. We found that peers usually contact this server and a different one on the same day. For instance, 82% of common peers for server 1 and 2 are observed for both servers on the same day. Therefore, the number of unique peers for server 1 during non-scanning activities (8758 peers) is fairly higher than the numbers for other servers (average of 2185 peers). Servers 0, 1, 2, and 4 are similarly assigned to the push role, consequently, the role-based clustering of Section 4.2 classified these four servers in partition 5. Server 3, however, is primarily retrieving data from peers (see Figure 6e), thus exhibiting a complementary behavior to the one

observed for the other servers. The apparent spatial overlap between these servers and their complementary roles highlights the close association of these servers.

The other group is composed of server 5 and 6 (Figure 7a), the activity of these servers is also depicted in Figure 6h and 6g, respectively. Server 5 is active during most of the measurement period whereas server 6 is only active for a week in December 2014. This week is of particular interest as no packet from server 5 is observed at these dates, suggesting that the server was unreachable. Thereby, server 5 operations seem to be relayed to server 6 during this period of time. Along with the spatial overlap, the port information from corresponding traffic also strengthen this evidence, 96% and 99% of the packets observed for, respectively, server 5 and 6 are transmitted on port 22 which is an uncommon port in the analyzed dataset.

5.2 Scanning C&C

The spatial overlap of servers involved in scanning activities does not provide direct insights into the C&C infrastructure but allows to better understand the scope of scans performed by botmasters.

Figure 7b depicts the spatial overlap for peers of C&C servers contacted during scanning activities. Notice that servers are ordered by similarities thus the indices in Figure 7a and 7b are unrelated. Servers labelled from 0 to 8 in Figure 7b have significant overlaps with all other servers indicating that these are very large scans that encompass a large fraction of the monitored IP space. The average number of unique peers for these servers is 39504. Servers 9 to 20 have a much lower average number of peers (i.e. 8546 peers). Nonetheless, their overlaps with all other servers is also noticeable, meaning that these servers are also scanning the entire monitored IP space but with a lower intensity. The rest of the servers, on the other hand, exhibit really low spatial overlap among themselves. These observations, as the one made for partition 1 in Section 4.2, illustrate that most of the scans have a very limited scope and probably targets specific sets of hosts.

Summary: Based on the roles identified in Section 4 the monitored spatial overlap helps to understand C&C infrastructures. It permits to effectively distinguish bots contacted by several C&C servers while ignoring peers probed during scanning activities. The spatial analysis of peers during scanning activities, on the other hand, provides details on the scope of scans.

6. TEMPORAL CORRELATION

Previous sections mainly focus on C&C traffic characteristics and spatial distribution of peers, we now investigate temporal aspects of C&C servers. The goal here is to identify servers that are synchronously operating, hence governed by the same entity. To find these synchronous activities we study the temporal correlation of C&C servers traffic.

We compute, for each C&C server, a signal compiling the number of bytes sent and received per hour (similar to the blue plots of Figure 6). All servers are compared pairwise with the following two-step method:

1. We perform a Pearson's chi-square test to check if the signals from servers x and y are statistically independent. The null hypothesis is that the two signals are uncorrelated and the test is done with a 95% level of confidence. If the null hypothesis is rejected, meaning that the signals are indeed dependent variables, then we compute the correlation coefficient for the two signals, $\rho_{x,y}$. Nonetheless, as servers have disparate activity periods, this correlation coefficient may be misleading in certain cases. As shown in Figure 2c, duration of traffic observed for each server varies from a few hours to several months. Comparing two servers activities that are completely disjoint in time as little meaning, therefore, in addition to the correlation coefficient we also consider the temporal overlap of servers activities.

2. We provide a second test to check if servers share activities in time. Let T_x and T_y be the two sets of hourly time bins where servers x and y are active (i.e. receiving or transmitting traffic), then the relative common activity of these servers is defined as:

$$rca(T_x, T_y) = \frac{T_x \cap T_y}{max(T_x, T_y)}.$$

If $rca \geq 0.5$ then the two servers are mostly active at the same time, and the computed correlation coefficient $\rho_{x,y}$ exhibits their interdependence. Otherwise, if $rca < 0.5$, the two servers are mostly asynchronous and we consider these two servers activities to be related only if the corresponding correlation coefficient $\rho_{x,y}$ is higher than a confidence threshold. In our experiments, we arbitrarily set this threshold to $\rho_{x,y} > 0.5$, thus pairs of servers that pass the independence test but fail this test (i.e. $rca < 0.5$ and $\rho_{x,y} < 0.5$) are said uncorrelated.

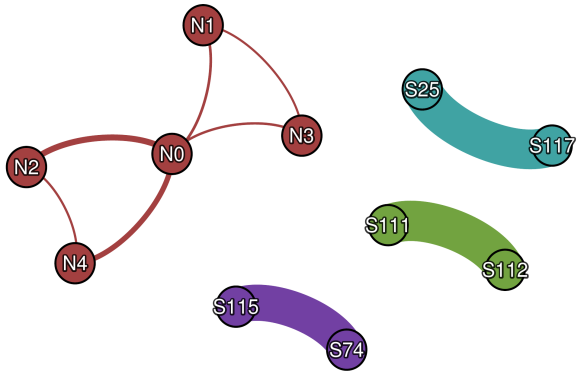


Figure 8: Temporal correlation of C&C servers, The node labels, $N0$ to $N4$, reflect the indices of the matrix in Figure 7a. Edges weight are indicated by the width of edges.

Pairs of servers that pass both tests are represented in a graph where nodes stand for servers, and two correlated servers, x and y, are connected by edges weighted with the corresponding correlation coefficient, $\rho_{x,y}$. Dense connected components in this graph indicate sets of synchronously operated C&C servers.

Figure 8 depicts the graph of correlated servers obtained with our dataset. The largest component is composed of the same servers as the prominent cluster identified with the spatial overlap in Section 5.1. Therefore, these servers exhibit both spatial overlap and temporal correlation which

reinforces evidences of the common management of these servers.

The other connected components from Figure 8 consist only of a pair of strongly correlated nodes (i.e. $\rho > 0.9$). All these nodes are primarily assigned to the scan role and are active only for a few hours. For example, both nodes $S111$ and $S112$ are only active during the same two hours, their spatial overlap is equal to 0 but both servers are targeting the same service (TCP port 80). The three pairs of nodes manifest the same synergy that we attribute to coordinated scans and reveals the common operations executed by several servers.

Summary: Spatial and Temporal correlations permit to identify the same C&C infrastructure composed of 5 servers. Synchronized scans are also emphasized with the temporal correlation method proposed in this section. Similarly to the observations of Section 4.2, we observe here a limited number of synchronized servers, these asynchronous communications can be substantially prejudicial for botnet detection methods based on bots' synchronous behavior [14].

7. RELATED WORK

Botnets have received a lot of attention from the research community, and have been studied from different perspectives. Research on bot and C&C channel detection has been particularly active. Several studies detect botnet communications by looking at peculiar usages of conventional protocols. IRC [25, 34, 13] and HTTP [23, 6] are typical examples of such protocols employed by early botnets. Researchers have also proposed more general approaches that rely either on fundamental characteristics of botnet traffic, or by relating datasets of different natures. For example, some works identify botnets through their periodic communications [2] or typical behaviors [35, 24, 11, 15, 39]. Others are investigating multiple datasets, for example, host and network level information [37, 28], Honeypots [26] or DNS traffic [18, 36]. Most of these techniques are able to identify a wide variety of botnets as they make no assumption on the communication protocol, hence, are also effective if botnets employ custom or encrypted protocols. The clustering methods employed in previous work, however, are particularly difficult to implement in the case of backbone networks. For instance, Botminer [11] relies on deep packet inspection and CoCoSpot [9] requires non-sampled traffic which is unpractical for our study case.

We refer the reader to [27, 20] for comprehensive surveys on botnet detection. Detection techniques are complementary to the analysis presented in this paper, as our analysis relies on C&C blacklists summarizing results from botnet detection algorithms.

Botnet infiltration is an effective approach to monitor prominent botnets and measure their distinctive characteristics. Therefore, by taking control of C&C servers, researchers have investigated the operations of the Torpig botnet [30]. Controlled infection in sandbox [19] or bot simulation [33] also permits to infiltrate botnets and obtain relevant information on them. These approaches are appropriate to inspect specific botnets but are difficult to generalize to any botnet.

Closer to the work presented in this paper, several studies rely on blacklists and external datasets to infer the condition of botnets. For example, with blacklists reporting Zeus C&C servers and by scanning the reported hosts, researchers have derived the Zeus C&C lifetime and factors affecting the longevity of the Zeus infrastructure [10]. An evaluation study of blacklists [22] classifies reported entries as parked domains, unregistered domains, and sinkholes, using DNS records and sandbox results. A recent study is also monitoring Internet traffic to classify botnets based on their size, and to uncover botnets collaborations [5].

Our study of botnet traffic supplements the vast literature on botnets by proposing generic tools to monitor the different roles played by C&C servers and their relationships.

8. CONCLUSIONS

This paper investigates different botnet families traffic collected at Internet exchange points, backbone and edge networks. A clustering technique is devised to identify six different functions of C&C servers. Using these C&C roles, we classify servers with similar behavior and found that servers rarely perform multiple roles. We also proposed techniques to effectively identify C&C servers with common bots and servers that are synchronously activated. Our observations with five months of traffic reveal a large amount of C&C servers dedicated only to scans. This is particularly important to take into account when inferring bots or estimating the size of a botnet from traffic data. Although measuring traffic at core routers can potentially expose a large fraction of botnet resources, we found that in practice the significant sampling rate imposed by the large amount of transmitted traffic on backbone network complicate this type of analysis and should be taken into consideration when designing similar traffic analytical methods.

Acknowledgments

This research has been supported by the Strategic International Collaborative R&D Promotion Project of the Ministry of Internal Affairs and Communication in Japan (MIC) and by the European Union Seventh Framework Programme (FP7 / 2007- 2013) under grant agreement No. 608533 (NECOMA). The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the MIC or of the European Commission.

9. REFERENCES

[1] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos. Highly resilient peer-to-peer botnets are here: An analysis of gameover zeus. In *Malicious and Unwanted Software:" The Americas"(MALWARE), 2013 8th International Conference on*, pages 116–123. IEEE, 2013.

[2] B. AsSadhan, J. M. Moura, D. Lapsley, C. Jones, and W. T. Strayer. Detecting botnets using command and control traffic. In *Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on*, pages 156–162. IEEE, 2009.

[3] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang. On the analysis of the zeus botnet crimeware toolkit. In *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*, pages 31–38. IEEE, 2010.

[4] A. Buescher, F. Leder, and T. Siebert. Banksafe information stealer detection inside the web browser. In *Recent Advances in Intrusion Detection*, pages 262–280. Springer, 2011.

[5] W. Chang, A. Mohaisen, A. Wang, and S. Chen. Measuring botnets in the wild: Some new trends. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '15, pages 645–650. ACM, 2015.

[6] C.-M. Chen, Y.-H. Ou, and Y.-C. Tsai. Web botnet detection based on flow information. In *Computer Symposium (ICS), 2010 International*, pages 381–384. IEEE, 2010.

[7] Cisco. Dridex attacks target corporate accounting. http://blogs.cisco.com/security/dridex-attacks-target-corporate-accounting, March 2015. Accessed: 2015/07/14.

[8] C. Criscione, F. Bosatelli, S. Zanero, and F. Maggi. Zarathustra: Extracting webinject signatures from banking trojans. In *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*, pages 139–148. IEEE, 2014.

[9] C. J. Dietrich, C. Rossow, and N. Pohlmann. Cocospot: Clustering and recognizing botnet command and control channels using traffic analysis. *Computer Networks*, 57(2):475–486, 2013.

[10] C. Gañán, O. Cetin, and M. van Eeten. An Empirical Analysis of ZeuS C&C Lifetime. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '15, pages 97–108, New York, NY, USA, 2015. ACM.

[11] 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.

[12] G. Gu, P. A. Porras, V. Yegneswaran, M. W. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *Usenix Security*, volume 7, pages 1–16, 2007.

[13] G. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee. Active botnet probing to identify obscure command and control channels. In *Computer Security Applications Conference, 2009. ACSAC'09. Annual*, pages 241–253. IEEE, 2009.

[14] G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic. 2008.

[15] H. Hang, X. Wei, M. Faloutsos, and T. Eliassi-Rad. Entelecheia: Detecting p2p botnets in their waiting stage. In *IFIP Networking Conference, 2013*, pages 1–9. IEEE, 2013.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer series in statistics. Springer, 2 edition, 2009.

[17] G. Jacob, R. Hund, C. Kruegel, and T. Holz. Jackstraws: Picking command and control connections from bot traffic. In *USENIX Security Symposium*, volume 2011. San Francisco, CA, USA, 2011.

[18] N. Jiang, J. Cao, Y. Jin, L. E. Li, and Z.-L. Zhang. Identifying suspicious activities through dns failure graph analysis. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 144–153. IEEE, 2010.

[19] J. P. John, A. Moshchuk, S. D. Gribble, and A. Krishnamurthy. Studying spamming botnets using botlab. In *NSDI*, volume 9, pages 291–306, 2009.

[20] S. Khattak, N. Ramay, K. Khan, A. Syed, and S. Khayam. A taxonomy of botnet behavior, detection, and defense. *Communications Surveys Tutorials, IEEE*, 16(2):898–924, Second 2014.

[21] C. Kolbitsch, P. M. Comparetti, C. Kruegel, E. Kirda, X.-y. Zhou, and X. Wang. Effective and efficient malware detection at the end host. In *USENIX security symposium*, pages 351–366, 2009.

[22] M. Kührer, C. Rossow, and T. Holz. Paint It Black: Evaluating the Effectiveness of Malware Blacklists. In *Research in Attacks, Intrusions and Defenses*, volume 8688 of *Lecture Notes in Computer Science*, pages 1–21, 2014.

[23] J.-S. Lee, H. Jeong, J.-H. Park, M. Kim, and B.-N. Noh. The activity analysis of malicious http-based botnets using degree of periodic repeatability. In *Security Technology, 2008. SECTECH'08. International Conference on*, pages 83–86. IEEE, 2008.

[24] W. Lu, M. Tavallaee, and A. A. Ghorbani. Automatic discovery of botnet communities on large-scale communication networks. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 1–10, New York, NY, USA, 2009. ACM.

[25] C. Mazzariello. Irc traffic analysis for botnet detection. In *Information Assurance and Security, 2008. ISIAS'08. Fourth International Conference on*, pages 318–323. Ieee, 2008.

[26] V.-H. Pham and M. Dacier. Honeypot trace forensics: The observation viewpoint matters. *Future Generation Computer Systems*, 27(5):539–546, 2011.

[27] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro. Survey and taxonomy of botnet research through life-cycle. *ACM Comput. Surv.*, 45(4):45:1–45:33, Aug. 2013.

[28] S. Shin, Z. Xu, and G. Gu. Effort: Efficient and effective bot malware detection. In *INFOCOM, 2012 Proceedings IEEE*, pages 2846–2850. IEEE, 2012.

[29] Spamhaus. Celebrating the first birthday of the spamhaus bgpf. http://www.spamhaus.org/news/article/699/celebrating-the-first-birthday-of-the-spamhaus-bgpf, June 2013. Accessed: 2015/07/14.

[30] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: analysis of a botnet takeover. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 635–647. ACM, 2009.

[31] Symantec. Internet security threat report. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v19_21291018.en-us.pdf, 2014. Accessed: 2015/07/14.

[32] Symantec. The state of financial trojans 2014. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-state-of-financial-trojans-2014.pdf, March 2015. Accessed: 2015/07/14.

[33] K. Thomas and D. M. Nicol. The koobface botnet and the rise of social malware. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pages 63–70. IEEE, 2010.

[34] W. Wang, B. Fang, Z. Zhang, and C. Li. A novel approach to detect irc-based botnets. In *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC'09. International Conference on*, volume 1, pages 408–411. IEEE, 2009.

[35] P. Wurzinger, L. Bilge, T. Holz, J. Goebel, C. Kruegel, and E. Kirda. Automatically generating models for botnet detection. In *Computer Security–ESORICS 2009*, pages 232–249. Springer, 2009.

[36] S. Yadav, A. K. K. Reddy, S. Ranjan, et al. Detecting algorithmically generated domain-flux attacks with dns traffic analysis. *Networking, IEEE/ACM Transactions on*, 20(5):1663–1677, 2012.

[37] Y. Zeng, X. Hu, and K. G. Shin. Detection of botnets using combined host-and network-level information. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 291–300. IEEE, 2010.

[38] Y. Zhang, M. Yang, B. Xu, Z. Yang, G. Gu, P. Ning, X. S. Wang, and B. Zang. Vetting undesirable behaviors in android apps with permission use analysis. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 611–622. ACM, 2013.

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