Botnet C&C Traffic and Flow Lifespans Using Survival Analysis

Vaclav Oujezsky, Tomas Horvath and Vladislav Skorpil

***Abstract*—This paper addresses the issue of detecting unwanted traffic in data networks, namely the detection of botnet networks. In this paper, we focused on a time behavioral analysis, more specifically said – lifespans of a simulated botnet network traffic, collected and discovered from NetFlow messages, and also of real botnet communication of a malware.**

As a method we chose survival analysis and for rigorous testing of differences Mantel–Cox test. Lifespans of those referred traffics are discovered and calculated by lifelines using Python language.

Based on our research we have figured out a possibility to distinguish the individual lifespans of C&C communications that are identical to each other by using survival projection curves, although it occurred in a different time course.

***Keywords*—Botnet, Lifespans, Modeling, NetFlow, Survival Analysis,**

I. Introduction to the Problem

Nowadays, rapid networks demand developing of sophisticated method to uncover an unusual behavior of network traffic. Progressively, new techniques are being developed to predict or detect network behavior. These techniques collect traffic information from devices as Test Access Point (TAP), they use port mirroring techniques or they perform an analysis of NetFlow messages [1].

Our statistics survey shows that the most analysis approach of network anomalies detection is based on ”store and backmine” data to analyze it from a database or systems use regular expressions. Anomalies can be marked as intentional and unintentional. Intentional could be botnet networks, distributed denial of services (DDoS) attacks etc. Unintentional anomalies are errors in networks, for example.

This is an extended paper, previously published by International Conference on Telecommunications and Signal Processing (TSP) [2], and the intention of our research is to define and expand a method how to analyze malevolent communication among clients and servers which communicate over a wide transport network. The certain groups of devices can play a basic role in botnet. The word botnet is a combination of the words robot and network. It is very difficult and complex issue. More types of botnet networks and their behavior are distinguished [3]. The basic is with Internet Relay Chat (IRC) communication approach. The another types of botnet are: with the Hypertext Transfer Protocol (HTTP), Point to Point (P2P) and HTTP2P traffic combined, centralized and decentralized.

There are several types of traffic types than described above. Generally, a botnet network can be controlled and commanded with any type of access to the root privileges of devices. Such access can be deployed by Secure Shell (SSH) connectivity or, as is an different example, with e–mail access. Such technique is also used in the project GCAT [4], it uses a gmail account to create Command and Control (C&C) channels.

C&C channels represent receiving and sending commands and informations between botmaster (C&C server) and infected clients (C&C clients), as is shown in Fig. 1. Botmaster can affect and control many clients in short time period. With this control, clients can do a wholesale attack.

Given the above, we distinguish many types of existing approaches and methods how to create a botnet. What is essential to say that the process of taking control of network devices is not firmly defined and can be done basically by any individual approach, also by creating a custom solution and this solution would be for others unknown and unpredictable.

The detection of botnet is generally based on methodology behavior or signature and C&C infrastructure. Since a communication in botnet also demands to carry information, it can be utilized for the detection. The main issue is, that this communication or traffic is mixed with others. Then, the behavior of this malicious traffic can be similar as a normal traffic. If an intrusion detection system is used for the detection, there is again problem with the traffic encryption used between C&C machines and rule signatures cannot be applied.

From our previous research, our decision and goal is to

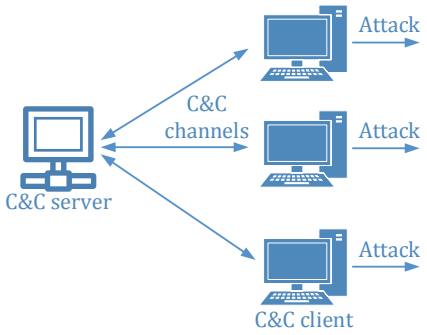

Fig. 1. Simple centralized botnet network

Manuscript received October 26, 2017, revised March 8, 2017.

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

V. Oujezsky, T. Horvath and V. Skorpil are with the Faculty of Electrical Engineering and Communication Brno University of Technology, Technicka 3058/10, BRNO 616 00 CZ. Corresponding author to provide (e-mail: vaclav.oujezsky@phd.feec.vutbr.cz).

survey the possibilities of detection these networks based on graph theory and time's persistence. What is very important, it is not sure if the botnet behavior is ergodic, stationary normalized or both or not. The final proof of it do not still exists regards to our discussion on teleconferences with other professionals. The focus has to be also on its stochastic.

The rest of this paper is structured as follows. The next section II presents a description of the related works. After section III, the basic methods, techniques and principles used are explained. The following section describes the methods of testing and the results, then the discussion within a conclusion is presented.

II. Related Work

Until now, there have been several approaches used for the detection of botnet or malicious traffic. These approaches can be divided into the following categories, from the viewpoint of focus:

- Host-based detection.
- Network-level-based detection.
- Graph-Theory-based detection.

We are generally concerned in Network–level and Graph–Theory combined based detection. In both, numerous works have been addressed to these topics.

Host-based detection mixed with the Network-level based was chosen in paper [5], where the authors performed Behavioral Classification. Sérgio S.C. Silva et al. presented in [3] a comprehensive view on the issue of botnet networks. We take over principles of botnet behavior. Graph theory and cyber–thread infrastructure is covered by an article [6]. The authors also present "badness scores for domains", IP addresses. It leads us to the idea to do the comparative score for our NetFlow duration.

The Network-level-based approach has been presented in [7], where the authors used flow data collected from a backbone network to detect e-mail spammers. The similarity with our work is in how we gathered data to our analysis from a backbone devices.

Detailed research in this topic is presented by Stebastián García et al. The authors are interested in Botnet C&C Behaviors. Their work is concerned in the time-based behavioral characteristics. In article [8] the identification of the User Datagram Protocol (UDP), Transmission Control Protocol (TCP) and HTTP C&C channels and its analysis is presented. The second article [9], from which we were inspired, provides a comprehensive view of the comparison of the ways and possibilities to botnet detection. We adapted the method of "Aggregate the NetFlows by source Internet Protocol (IP) address" and the motivation to do the aggregation of NetFlow. Here are the properties of botnet (citation):

- Each bot communicates with the C&C server periodically.
- Several bots may communicate at the same time with the same C&C servers.
- Several bots attack at the same time the same target.

Our different approach is in the NetFlow aggregation and in the method of post-processing. From this research it is obvious that the communication among clients and servers has a certain periodicity and they communicate with a readable frequencies. A closer look at all the approaches currently used, due to the nature of the botnet network, it seems perspective to continue to deal with the combined model behavior and data collection.

To sum it up, many articles have been written about botnet and behavior. Not much has been written about the life-cycle of the botnet. The main question for us is given: what is the average lifespans of botnet, what about life-cycle? How to determine it with using existing network devices in a converged transport network.

III. Methods, Techniques and Principles Used

A. NetFlow and its use

NetFlow has been involved and implemented by company Cisco Systems, Inc in their network products [10]. This method is very popular and widely deployed by manufacturers of network elements of different brands. Latest successor of it is Internet Protocol Flow Information Export (IPFIX). These messages are used to send information about passing traffic from network devices to an analyzer (collector).

An example is Scrutinizer [1] – used for aggregating NetFlow protocol messages sent from individual network devices. If the network devices are configured appropriately, they periodically send NetFlow protocol informations about which nodes (IP addresses) communicate. It sends information about TCP and UDP ports and also information about the duration of a connection. If the information about entire network are aggregated in one place, than can be fairly accurately identified certain types of attacks against the network typically volumetric DoS / DDoS attacks, attempts to SYN flooding etc. It can also identify certain types of attempts of illegal penetration into the network and other incidents. This all depends on the quality of the collector, its ability statistical processing of individual NetFlow reporting.

Another use of NetFlow is a collection of "traffic data" named Data Retention (DR), which requires not only the Czech legislation. In the Czech Republic (CR), it is a law of "electronic communications" (no. 127/2005), specifically 97 paragraph 3. This law was based originally on a directive of the European Parliament and Council Directive 2006/24/EC. This directive was invalidated in 2013. It is therefore likely that a substantial number of European Union countries have legislation which is similar in intent to this CR law. The essence of the use of NetFlow within the DR is storing NetFlow reports and export them to the legitimate applicants in the original – unchanged – form.

NetFlow is also used by administrators of large corporate networks. It is primarily for monitoring of which node (computer / server) is communicating at a given time.

For daily use, the most important are algorithms, which are decisive. It is important that the collector only reports real

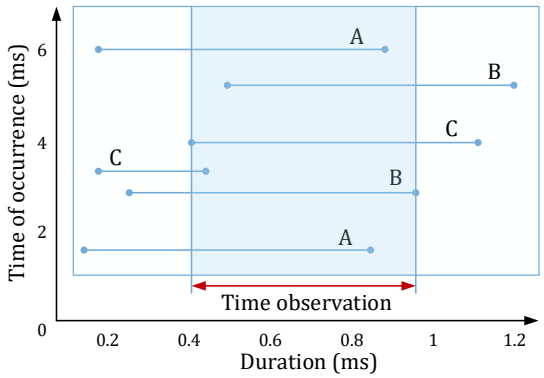

Fig. 2. Time of observed epoch

attacks and does not generate the number of false positives reactions.

B. Survival Analysis

Survival analysis was originally developed to measure lifespans of individuals. This analysis can be applied to any process duration. For example, it can be related to a web service, where the start of duration users are joining and the end is when the users leave the web service. Survival function is defined as:

$$S(t) = Pr(T > t) \tag{1}$$

where T represents random lifetime taken from a set of population and function $S(t)$ is defined as the probability of surviving until at least time t [11], equivalently, it defines the probability, that the death event of subject has not occurred yet at time t.

Survival function with the statement above has following properties, $0 \leq S(t) \leq 1$, $F_T(t) = 1 - S(t)$, where $F_T(t)$ is the CDF of T and $S(t)$ is non increasing function of t.

1) Censoring and truncation: Further, right and left censorship is defined. With the right-censored individuals we have information only about their current lifelines duration. On the other hand, with the left-censored individuals we do not know information about their birth (formation, start). The last type of censoring is the interval-censored. In this case we do not know exact time without event. We have partially observed events. The truncation happens, when the subjects have been at even before entering the study. Survival analysis is a very useful tool to understand duration.

Fig. 2 represents our example case of study with different combination of events and their start and end, thus duration. Letter *A* is a normal traffic, letter *B* and *C* would be C&C communications. Our aim is to find out the shape and result of survival function of each traffic for a protocol, port or IP address and compare them with one another. In our case, two survival functions of C&C traffic should be "almost" similar, even if they take place in a different time sequence, with regard to the conditions specified in [9] we set up behavior of botnet.

For the indication of the length of observation the random variable *T* is used. It expresses all captured NetFlows. δ is an indicator of the event. In case an event has occurred, δ is equal to 1 and if the individual was censored δ is equal to 0. For n studied subjects a plurality of pairs $\{(t_i, \delta_i), i = 1 \ldots n\}$ is received.

2) Estimating the Survival function: Kaplan–Meier estimator has been used within our test to estimate the survival function. It is a non–parametric method, therefore it does not require knowledge of the probability distribution that governs the survival of individual subjects. It is also called the product–limit method [11], [14]. Kaplan–Meier method gives an estimate of the survival function at every moment, in which there was a monitored event. The Kaplan–Meier Estimate is defined as:

$$\hat{S}(t) = \prod_{t_i < t} \frac{n_i - d_i}{n_i} \tag{2}$$

where d_i are the death events at time t_i and n_i are subjects at risk of death just prior to time t_i.

3) Compliance tests: Compliance tests of survival functions are used to compare two survival curves. There are many types of these tests, each of them has optimal properties for different situations. We could mention the Breslow test or the Tarone–Ware test. The most famous asymptotically valid tests include non–parametric Mantel–Cox test, named after Nathan Mantel and David Cox. Sometimes it is also named as "log rank test". The censoring process is independent here of the process that leads to the event. Mantel–Cox Chi–Squared is defined [14]:

$$\chi^2_{MC} = \frac{(O_1 - E_1)^2}{E_1} + \frac{(O_2 - E_2)^2}{E_2} \tag{3}$$

where the O_1 is the sum of occurrence of events for experimental group and the O_2 is the sum of occurrence of events for control group. The E means expected sum for each group.

Mantel–Cox test is also generalized to test more than two test's groups. Then, the equation 3 is just extended by $k - 1$ definitions $\chi^2_{MC} = k_1 + k_2 + k_n$. We compared with this test each traffic event and we observed its conformity, if two or more flows are similar or not.

C. Principle of use

From the base of botnet communication, it is very difficult to identify C&C traffic in transport network, where it is not possible to do a deep packet analysis. At first, it is not allowed by law, and the second reason, it is not effective to observe each packet and it is also time consuming. The situation is as difficult as decentralized this communication is. Therefore, we have focused on the traffic behavioral to find some stochastic in it.

We could effectively use existing devices on borders in a large transport network to send flow messages to a database collector and analyze timestamps, UNIX time and type of ports of regarded traffic. The idea is that we compare each time duration of a traffic given in UNIX time and we are looking for lifespans of it. After it, it is possible to compare survival curves with the log rank test. Because C&C messages have to be send at once by server to do an attack, we can observe and

find time reciprocity in a network to estimate that it is non demanded traffic.

D. Test's environment

Fig. 3 shows an involvement of component in the laboratory. We developed GDP-1.0.0 NetFlow collector (GDP) [12], which is an application in Python language used to collect NetFlow messages of version 1 and 5.

For the modeling traffic lifespans we have used NetFlow messages of version 5. This application has a database (sqlite3 database file) and collects information from NetFlow messages as is: source IP address, destination IP address, source port, destination port, protocol number, timestamps, first time, last time and system up-time. This application runs on virtual Windows 10 and is developed for Python version 3.

Two Cisco routers were configured and used to send NetFlow messages version 5 to the GDP application. Both of them have different subnet assigned and the passing traffic is mixed with laboratory traffic.

We followed the idea to create own C&C server and we have programmed Python code on different virtual PC as the C&C server. C&C channel feature SSH connectivity to C&C clients. The command channels are provided using the `fabric` module [13], [15] of Python. Finally, the algorithm used periodically connects and sends demands toward clients.

In this case, we simulated a periodic operation of the server and the clients regardless of the type of traffic. In real traffic also a certain frequency of C&C occurs and is primarily the possibility to detect a relationship between these frequencies, whether they appear at any time, and different subnet.

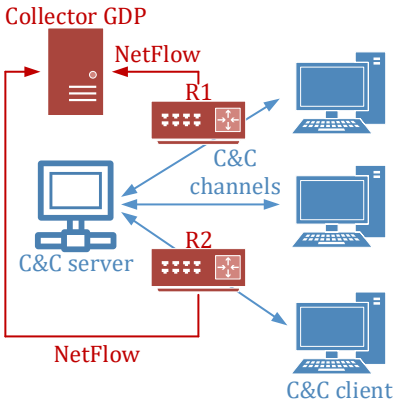

Fig. 3. Involvement of components in the laboratory

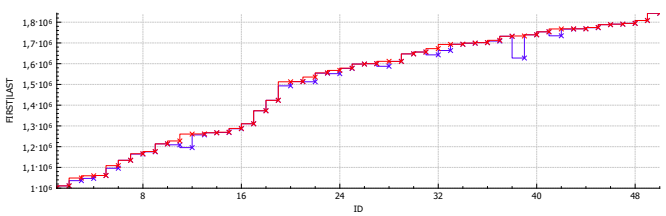

Fig. 4. FIRST/LAST time graph of captured flows I.

IV. Testing and Results

We observed similar duration and frequency in mixed traffic. Lifelines module [16] of Python as a implementation of survival analysis has been used for this purpose. It offers all of basic principles used in Survival Analysis. One limitation is with the other types of censorship. Interval-censored censorship is not implemented in lifelines yet. But the right-censored, left-censored and left-truncated censorship are included.

A. Initial Functional Test

At the beginning of the test, we examined the functionality of the program and the possibility of selection of network traffic and to display the lifelines.

We have compiled a test network over the public Internet using μtorrent clients. One of these clients was a server (sender) and the second was a receiver. Repeatedly, in a different timespan we transferred a reference file with the size of 100 MB.

These data were headed and a data–frame with the traffic information has been created and extracted in format as shown partly below:

	IP_SOURCE	IP_DEST	t	delta
0	192.168.1.11	192.168.1.1	7993	1
1	192.168.1.46	192.168.1.255	1	1
2	185.76.11.73	192.168.1.66	957	1
3	192.168.1.11	192.168.1.1	8089	1
4	192.168.1.11	192.168.1.1	7989	1
5	192.168.1.66	93.153.104.0	161721	1
6	192.168.1.46	192.168.1.11	161721	1
7	192.168.1.11	192.168.1.46	161741	1

The first three columns show IP source, IP destination and the t which is the duration of communication, taken from the value UNIX time of NetFlow. This value is calculated from `SysUptime` at the time the last packet of the flow was received minus `SysUptime` at start of flow. The fourth column shows a value $delta$. It represents the censorship. The value of the $delta$ is 1 due to the communication which ended during the reference period. In this case, all of the traffic has been terminated in the selected window time.

The graphs, plotted in Fig. 4 and 5 show all captured flows (ID) and the last time (red line) and the first time (blue line) of the two transmissions of the 100 MB reference file. It also includes the normal traffic. As seen, in this type of graph representation it is not possible to find any dependencies. Also, the amount of captured flows is not the same.

Furthermore, we separated the communication of the μtorrent's server and client. In this test case, we know the

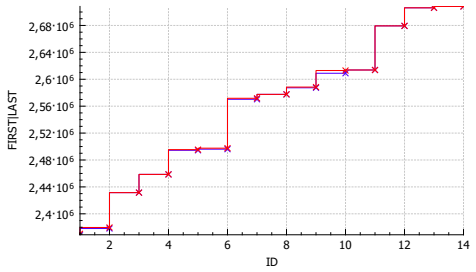

Fig. 5. FIRST/LAST time graph of captured flows II.

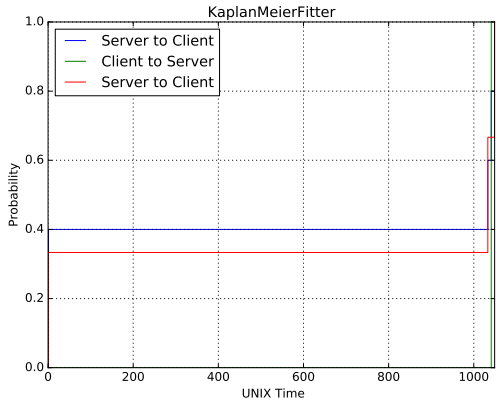

Fig. 6. Survival curve

IP addresses of these devices. This communication has undergone an analysis of Kaplan–Meier estimator with the left-censorship. The result is shown in Fig. 6. The curves of server–client communication in dependence on time and the probability is approaching to close identity even if they were taken at different time period.

This result was subjected to log rank test. This test is used to test the null hypothesis that there is no difference between the probability of an event at any time point.

Each group of server–client communication has been taken in different time scope. In group 1 the proper IP address was 5 times observed and in the group 2 the proper IP address was observed 3 times. So, the score is 5:3. The following listing is an extract from the log rank test performed:

```
Results
null distribution: chi squared
df: 1
t 0: -1
test: logrank
alpha: 0.05

_p-value|test statistic|_test result _|is significant
0.80408 |        0.062 |  Reject Null |        True
```

A p-value of less than 0.05 based on the log rank test indicates a difference between the two survival curves. In our case it is the value of $\cong 0.8$, which represents a value close to the maximum consensus of the two survival curves of server–client communication. The p-value of normal traffic comparing

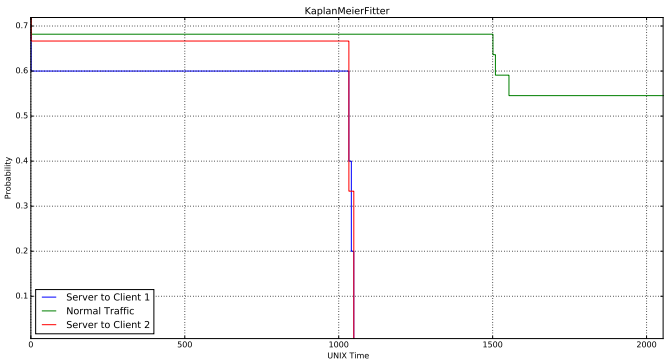

Fig. 7. KM estimate right censored - dependence test

server–clients traffic was 0.00735.

The initial functional test was successful and we obtained the values and survival curves for tested communication. The next provided test was a dependence test of C&C channels.

B. C&C Dependence Test

The test of C&C channels was assembled as is described above. C&C server periodically followed up SSH communication to the C&C clients placed in different subnetworks. This communication had been mixed by normal traffic. Captured communication using NetFlow reporting was again transferred to the Python `panda` data–frame by the GDP application.

In Fig. 7 the mutual comparison of survival's curves can be seen. The red line and the blue line are curves of C&C communication and the green line expresses the plot of normal traffic. The shortened example of format of event table obtained is following:

```
The duration of KM 1 is [1049 1041    1 1033    1]
          removed  observed  censored  entrance  at_risk
event_at
0               0         0         0         5        5
1               2         2         0         0        5
1033            1         1         0         0        3
1041            1         1         0         0        2
1049            1         1         0         0        1
```

We form the table in a modified form. This disposition of the table presents how many events were obtained for each duration. And the duration creates event time. The p-value, compare normal laboratory traffic and simulated C&C traffic was 0.17090 in comparison with the p-value of both of C&C was $\cong 0.75$. Again, we were able to find and to distinguish each communication channel.

V. Real Botnet Observation with Lifelines

We continued to observe data from real botnet network communication and apply proposed analysis to create its lifespans model. The .csv file of CTU-Malware-Capture-Botnet-1 [17] has been used as background information. The information of infected machine: Windows Name: Win8, IP: 10.0.2.22 (Label: Botnet-V1). All detailed information are given on author's website.

Database file "dataset" has been created by importing .csv file into it. For this testing purpose, we limited the number of inputs to 1,000,000. We isolated the communication of the machine x.22 and separately, we conquered the traffic to KaplanMeierFitter with the left censorship. The data have been headed and dataframe with traffic information created and extracted in the format as is shown partly below.

```
0                    SrcAddr           DstAddr         T C
1         00:00:00:00:00:00 00:00:00:00:00:00  0.000000 0
2                  0.0.0.0         10.0.2.22  2.003218 1
3                       :: ff02::1:ff01:e8a5  0.000000 0
4  fe80::705a:530f:1701:e8a5          ff02::2  4.003125 1
5  fe80::705a:530f:1701:e8a5         ff02::16  0.498083 1
6  fe80::705a:530f:1701:e8a5          ff02::2  0.000000 0
7  fe80::705a:530f:1701:e8a5        ff02::1:2  3.004039 1
8                 10.0.2.22          10.0.2.2  0.000096 1
...                     ...               ...       ... .
999970            10.0.2.22          8.8.8.8  0.010117 1
999971            10.0.2.22          8.8.8.8  0.010097 1
999972            10.0.2.22     94.100.176.20  0.000927 1
999973            10.0.2.22     74.125.142.26  0.001110 1
```

A. *Analysis of botnet traffic*

Once this set was formed, was subjected to analysis, see Fig. 8. Generally, the y-axis represents the probability communication is still around after the X time. In this case, the lifelines were observed for a particular service of the botnet as TCP, UDP, SPAM, Domain Name Service (DNS), and for background communication as well. The outputs are in one figure to make them easy read and comparability. It creates together their event table. Then, the median values, duration, and confidence intervals were calculated as is shown below.

```
The median of background is 0.000436
The median of background_arp is 1.898133
The median of botnet_1_UDP_Attempt is 2.193814
The median of botnet_1_TCP_Attempt is 3.003442
The median of botnet_1_TCP_Established is 0.353073
The median of botnet_1_UDP_Established] is 0.180717
The median of botnet_1_SPAM is 0.00136
The median of botnet_1_DNS is 0.010147
```

So far, we have formulated values for Win8. At the time-axis can, therefore, be selected certain time, for which we have defined probability. We assume, that this probability should be comparable independently of an occurrence. Now, we are able to compare background traffic with the each separate service of botnet traffic or whole botnet communication with the background traffic and observe the p-values and significance.

We have used the separate instance of Kaplan-Meier fitter to observe the difference between the botnet traffic and the background traffic and, it has been associated with one subplot, as shown in the Fig. 9. The result is significantly different only in the time line.

The same values we subjected the log rank test. The value p-value was less than the set limits, p-value < 0.000. The value of the test statistic was 249,388. Again, we have used chi squared null distribution and alpha 0.5.

We are also interested in the probability of distribution of represented protocols. The following Fig. 10 shows the difference in this probability after time n. The y-axis represents the probability a protocol is still around after t timeline, where t ms is on the x-axis. We also see that the shaded area of confidence interval is different for each protocol. The

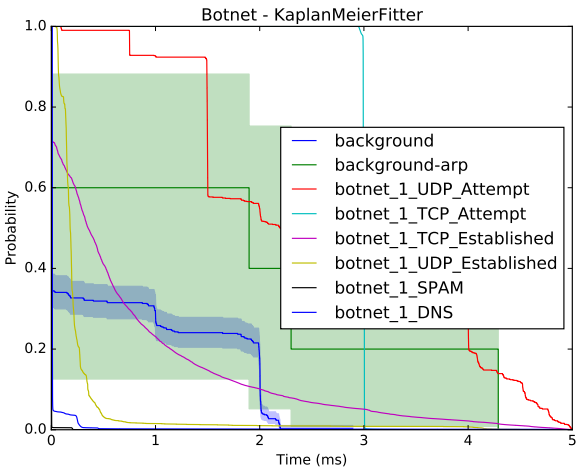

Fig. 8. KM estimate of Botnet of source 10.0.2.22

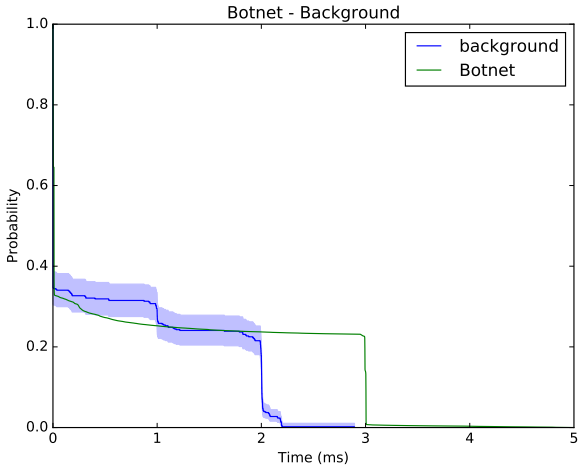

Fig. 9. Comparation of KM estimate of botnet and background traffic

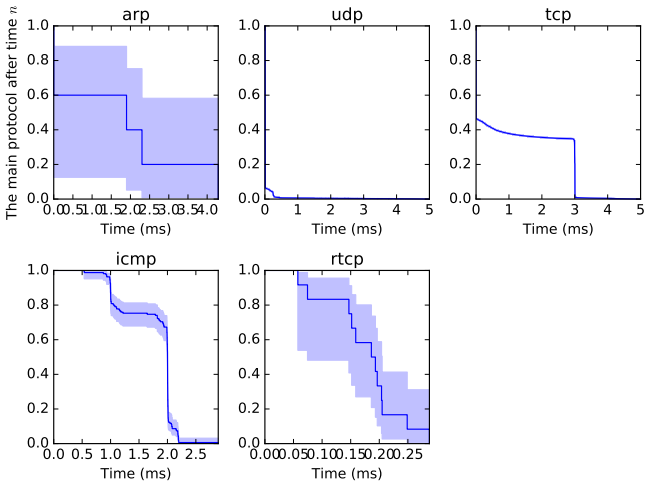

Fig. 10. KM estimation of protocols

confidence interval is the effect of sample size, methods of selection and population size, and it for the survival curve, whose reliability reaches values of $100(1-\alpha)$ for a given time t. From this point of view we have a reference map of the traffic.

VI. Concluding Remarks

The numerous methods are developed to detect botnet and others malicious traffic in heterogeneous networks. In this paper, we present a method for detecting command and control channels of a botnet. Our method is based on behavioral analysis and includes unique combination of the features as is the flow collector combined with survival analysis method. We created own application to simulate this C&C channel and also GDP-1.0.0 application to collect and store NetFlow messages.

We subtracted information form NetFlow messages, such as: duration of each communication in UNIX time, IP addresses and ports. We then created data–frames from this obtained information and we have subjected them to the survival analysis. This processing of NetFlow has shown the possibility of

detection of two traffic that are time-independent themselves, so the results demonstrate the possibility of detection by this method based on the time-duration.

By defining method above, we have used the .csv file of CTU-Malware-Capture-Botnet-1 to create a sample model of lifespan. The individual values and appearance of the function of lifespan were counted. These values are stored for future use as a pattern.

The benefit of this method is that it does not need to have a knowledge about proper time when a traffic occurs. It only needs to have knowledge about the duration of flows. Similarly, as in the case of traffic modeling using Markov chain, the appearance of communication is modeled. But with the difference, that traffic patterns are not generated, but two or more similar progressions are sought according to the previous conditions of behavior of botnet networks. This method accelerates a selection which traffic choose to the shortlist to decide whether the traffic is demanded or not.

In future work, we would like to extend our work to observe the survival values of Win12 of CTU-Malware-Capture-Botnet-1 and calculate multivariate log rank test. We would like to continue calculating values for each dataset. We plan to build own laboratory environment for capturing botnet communication and its evaluation method described above.

REFERENCES

[1] Plixer: *Flow Analytics*. PLIXER INTERNATIONAL, Inc. Plixer-Malware Incident Response.

[2] V. Oujezsky, T. Horvath and V. Skorpil, "Modeling Botnet C& C Traffic Lifespans from NetFlow Using Survival Analysis," in *Proc. 39th International Conference on Telecommunication and Signal Processing, TSP 2016*. Vienna, Austria 2016. pp. 50–55, ISBN 9781509012879, ISSN 1805-5435.

[3] S.C.S. Silva, R.M.P. Silva, R.C.G. Pinto, and R.M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, pp. 378–403, February 2013.

[4] GCAT: A fully featured backdoor that uses Gmail as a C&C server, GitHub.

[5] J. McHugh, R. McLeod, and V. Nagaonkar, "Passive network forensics: behavioural classification of network hosts based on connection patterns," *ACM SIGOPS Operating Systems Review*, vol. 42, pp. 99–111, April 2008.

[6] A. Boukhtouta, D. Mouheb, M. Debbabi, O. Alfandi, F. Iqbal, and M.E. Barachi, "Graph-theoretic characterization of cyber-threat infrastructures," *Digital Forensics & Incident Response*, vol. 14, pp. S3–S15, August 2015.

[7] W.K. Ehrlich, A. Karasaridis, D. Liu, and D. Hoeflin, "Detection of spam hosts and spam bots using network flow traffic modeling," in *Proc. 3rd USENIX conference on Large-scale exploits and emergent threats LEET'10*, pp. 7-7, ©2010.

[8] S. Garcia, V. Uhlir, and M. Rehak, "Identifying and modeling botnet C&C behaviors," in *Proc. 1st International Workshop on Agents and CyberSecurity - ACySE 14*, pp. 1–8, 2014.

[9] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers and Security*, vol. 45, pp. 100–123, September 2014.

[10] *Introduction to Cisco IOS NetFlow - A Technical Overview*. CISCO SYSTEMS, Inc. CISCO, 2012.

[11] *Lifelines*, Cam Davidson-Pilon, Copyright 2014.

[12] *GDP - NetFlow Collector*. Network Security Research, ©2016.

[13] *Fabric: Pythonic remote execution*, ©2016.

[14] Norman, Geoffrey R. and David L. Streiner. *Biostatistics: the bare essentials*. 3rd ed. Shelton, Conn.: People's Medical Pub. House, 2008. ISBN 9781550093476

[15] *GitHub, Fabric: Simple, Pythonic remote execution and deployment*, GitHub, 2016.

[16] C.D. Cam, *Lifelines*, 2014.

[17] *Stratosphere IPS*, Dataset, ©2015.

Vaclav Oujezsky (MSc) was born in Brno, Czech Republic. Post graduate student at Brno University of Technology, Department of Telecommunications, Senior Network Engineer at T-Mobile CZ, and currently at IBM CZ. Working actively on projects of security and transport networks at laboratory SIX. His research interests include implementation of evolutionary algorithm, Cisco, Python, VHDL, and converged networks. His topic of dissertation thesis is Converged Networks and Traffic Tomography by Using Evolutionary Algorithms.

Tomas Horvath (MSc) was born in Havirov, Czech Republic. He received his MSc. degrees in Telecommunications from the Brno University of Technology, Brno, in 2013. His research interests include passive optical networks (xPON) and optoelectronics. Currently, he is post graduate student at Brno University of Technology, Department of Telecommunications. His topic of dissertation thesis is Optimization Services in FTTx Optical Access Networks.

Vladislav Skorpil Vladislav Skorpil was born in Brno in 1955. He received the MSc. and CSc. degrees in Brno University of Technology (BUT). From 1980 to 1982 he worked as a designer for the telecommunication design office. He again entered the Department of Telecommunications of BUT in 1982 as a university teacher and he has been working in this department since that time. Now he is an associate professor and a vice-head of this department. He takes a keen interest in modern telecommunication systems. He is the author of about 153 international scientific papers and some manuals. He has cooperated on telecommunication projects such as digital transmission and switching systems, telecommunication broadband networks, data networks LAN and MAN, neural networks, grammatical algorithms, Quality of Service, data bit rate compression, etc. He is a member of international organisations IEEE and WSEAS.