An efficient flow-based botnet detection using supervised machine learning

Matija Stevanovic and Jens Myrup Pedersen
Networking and Security Section, Department of Electronic Systems
Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg, Denmark
Email: {mst, jens}@es.aau.dk

***Abstract*—Botnet detection represents one of the most crucial prerequisites of successful botnet neutralization. This paper explores how accurate and timely detection can be achieved by using supervised machine learning as the tool of inferring about malicious botnet traffic. In order to do so, the paper introduces a novel flow-based detection system that relies on supervised machine learning for identifying botnet network traffic. For use in the system we consider eight highly regarded machine learning algorithms, indicating the best performing one. Furthermore, the paper evaluates how much traffic needs to be observed per flow in order to capture the patterns of malicious traffic. The proposed system has been tested through the series of experiments using traffic traces originating from two well-known P2P botnets and diverse non-malicious applications. The results of experiments indicate that the system is able to accurately and timely detect botnet traffic using purely flow-based traffic analysis and supervised machine learning. Additionally, the results show that in order to achieve accurate detection traffic flows need to be monitored for only a limited time period and number of packets per flow. This indicates a strong potential of using the proposed approach within a future on-line detection framework.**

***Keywords*—Botnet, Botnet detection, Traffic analysis, Traffic classification, Machine learning**

I. Introduction

Botnets are one of the most serious threats to the Internet security and one of the most challenging topics within the fields of computer and network security today. Botnets represent a usually large collections of computers compromised with a sophisticated bot malware, that puts them under the control of a remote attacker [1]. The compromised computers are often referred to as "zombies" or "bots" while attackers is referred to as the "botmaster". Contrary to other more conventional malware types, such as viruses, trojans and worms, bot malware has an advantage of being able to communicate with an attacker through a specially deployed Command and Control (C&C) communication channel. Botnets deploy C&C channel using a variety of communication protocols, such as: IRC, HTTP / HTTPS and in the most recent botnets, P2P protocols. Additionally, modern botnets use many resilience techniques that make C&C channel more resilient to detection such as encryption, protocol obfuscation, Fast-flux and DGA (Domain Generation Algorithm) [2]. Using the C&C channel, the botmaster can remotely control the behavior of the bot malware, making the operation of bots more flexible and adaptable to the miscreant's needs. Controlled and coordinated by the botmaster, botnets represent a collaborative and highly distributed platform for the implementation of a wide range of malicious and illegal activities, such as: sending SPAM e-mails, Distributed Denial of Service (DDoS) attacks, malware distribution, click fraud, information theft and attacks on industrial control systems and other critical infrastructure.

In order to successfully neutralize botnet threats an efficient detection approach is needed, that could provide high accuracy of detection and low time and computational requirements of operation. This paper explores how accurate and timely detection can be achieved using flow-based network traffic analysis and supervised machine learning as a tool for identifying malicious botnet traffic. The contribution of the paper is three-fold. First, the paper proposes a novel botnet detection system that relies on flow-based traffic analysis and supervised machine learning as a tool for identifying botnet traffic. Second, the paper evaluates performances of eight highly regarded supervised Machine Learning Algorithms (MLAs) for the task of classifying botnet traffic within the proposed system. Third, the paper explores how much traffic needs to be observed per flow so the malicious traffic patterns could be captured and accurate classification of traffic could be achieved. As a result the proposed approach obtains detection performances comparable to the contemporary approaches, for simpler traffic features and limited amount of traffic per flow. Furthermore, as the system achieves high accuracy of traffic classification for limited amount of traffic per flow, it has the potential of being used in more adaptable on-line setup.

The rest of the paper is organized as follows. Section II presents an overview of related work. Section III introduces a novel flow-based botnet detection system by elaborating on the employed principles of flow-based traffic analysis. Section IV presents the results of experimenting with the proposed system, analyzing the performances of the system for eight contemporary supervised MLAs. This chapter also presents how detection performances are dependent on the flow duration and the number of packets observed per flow. Section V discusses presented results and possibilities for further improvements. Finally, Section VI concludes the paper by summarizing the findings and outlining future work.

II. Related work

One of the most prominent classes of botnet detection methods is detection based on the identification of botnet traffic (both C&C and attack) using machine learning algorithms [1]. The main assumption of the machine learning-based approaches is that botnets create distinguishable traffic patterns that could be efficiently detected using MLAs. These approaches have a number of advantages comparing to conventional signature-based and anomaly-based detection

methods [2]. They promise a flexible detection that does not require any prior knowledge of botnet traffic characteristics and they are independent from the communication technology and resilience techniques employed by botnets. Supervised machine learning, as a class of well-defined machine learning algorithms capable of efficiently implementing various classification tasks, is often seen as a tool for accurate identification of botnet traffic. Several detection methods [2]–[6] have been developed using diverse principles of traffic analysis and an array of supervised MLAs over the last couple of years.

Strayer et al. [3] were one of the first to demonstrate the use of supervised machine learning for identifying botnet traffic. The authors devised the detection approach that targets IRC botnets, by performing multi-phase traffic analysis, where the classification of TCP flows using supervised MLAs plays a key role. For the classification of traffic flows the authors considered three MLAs: Naive Bayesian, Bayesian network and C4.5 decision tree, providing relatively low false positive and false negative rates (under 3%). The main disadvantage of the approach is the fact that it is only modeling TCP traffic as the main carrier of IRC communication.

Masud et al. [4] proposed the flow-based detection system for identifying malicious botnet traffic. The system targets IRC botnets and TCP flows, by relying on host level forensic and deep packet inspection (DPI) for extracting some of the traffic features. The approach considers five different classifiers: Naive Bayesian, Bayesian networks, Support Vector Machine (SVM), C4.5 decision tree, and Boosted decision tree providing high accuracy of detection (over 98%) for the C4.5 classifier. The main disadvantage is the use of client level forensics and DPI for acquiring flow features.

Saad et al. [5] proposed a system for detecting P2P botnets using traffic analysis where both host-based and flow-based traffic features were observed. The authors considered five supervised MLAs: Nearest neighbors, Naive Bayesian, Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Gaussian-based classifier. As a result the system achieved 97% accuracy of detection for SVM. Also the authors assessed ability of providing timely detection by examining the time requirements of implementing classification task using different MLAs. However, the authors did not evaluate any of the capable tree classifier leaving the space for further work.

Bilge et al. [6] used the supervised MLAs for classifying wide-scale NetFlow traces as malicious or non-malicious. The method considers Support Vector Machines (SVM), C4.5 decision tree and Random forest classifier, where Random forest provided the best performances with very low false positive rate (under 0.5%) and a detection rate of over 70%. The main disadvantage of the Bilge et al. approach is that it relies on NetFlow records that are usually sampled over the observed traffic causing the possible loss of information about fine patterns of botnet traffic.

Although providing the fairly good detection performances on specific datasets the contemporary methods [2]–[6] do not provide a comprehensive insight on performances of different MLAs, neither have they explored how much traffic is needed to be observed per flow so the malicious and non-malicious traffic can be successfully modeled. This paper addresses these shortcomings through a novel flow-based detection system that uses supervised MLAs for identifying botnet traffic. The system does not make any assumptions about the botnet traffic observing both C&C and attack traffic. Furthermore, the system is based on purely flow-based features extracted from the packet headers preserving the privacy of communication. We also evaluate the most extensive set of MLAs so far, and we provide an insight on how much traffic per flow is needed to obtain accurate detection. Finally, as our study uses the same botnet dataset as one in [5], the comparison of obtained detection performances is possible.

III. Flow-based botnet detection using supervised MLAs

In order to identify botnets we propose a novel flow-based botnet detection system that classifies network traffic as malicious or non-malicious using supervised MLAs, as illustrated in Figure 1. The system consists of two main components: the Pre-processing entity and the Classifier entity. The first processes traffic so a set of statistical features is extracted and selected for every traffic flow, while the second is in charge of building the model of malicious / non-malicious traffic and classifying the traffic flows. The system operates in two phases i.e. Training and Test phase. During the training phase the traffic model is trained by using a labeled training data while in the test phase the model is tested by an unlabeled test data. As a result the system outputs the classification results indicating if the traffic can be considered malicious.

A. The Pre-processing entity: the principles of traffic analysis

The Pre-processing entity analyses network traffic on flow-level where flows are defined as traffic on specific five-tuple: *source IP address, destination IP address, source port, destination port and protocol identifier*. The system observes TCP, UDP and ICMP protocols as they are regarded as the main carriers of botnet network activity. However, it should be noted that the definition of flows used here differs from the conventional definition of application flows, in order for such flows to capture the characteristics of the observed traffic. The crucial aspect of flow-based traffic analysis is choosing a set of flow features that captures targeted botnet heuristics. In order to achieve this, we extract 39 different statistical features for every traffic flow, as illustrated by Table I. The features are purely flow-based and they do not rely on IP addresses as host identifiers, avoiding optimistic detection performances for IP-biased datasets, i.e. datasets where IP addresses determine the classification results. All presented features are extracted from the flow level while only 4 features (36.-39.) require the awareness of flow pair i.e. flow in the opposite direction. We intentionally avoid more sophisticated features in order to avoid possible over-fitting of the model and any additional pre-processing requirements.

In the following section we implement two scenarios of traffic analysis. First is the "batch" analysis where all flows are monitored from the start until the end of the trace and the flow features are calculated over the whole duration of flows i.e. from the first to the last packet on a specific 5-tuple. The second scenario is the "limited" analysis where traffic flows are observed for a specific time interval and a specific maximum number of packets starting from the first packet of the flow. The limited analysis helps us understand how much traffic needs to

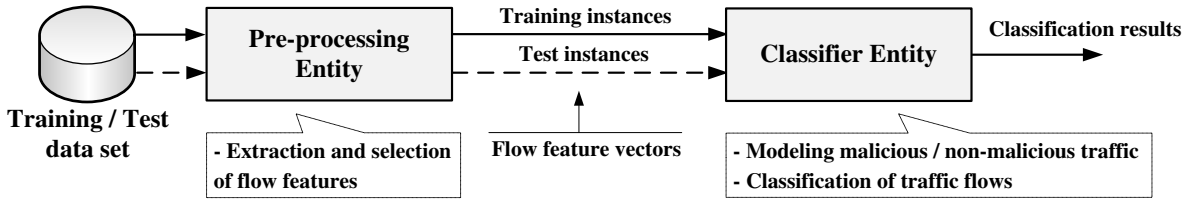

Fig. 1. Flow-based botnet detection system using supervised machine learning

TABLE I. LIST OF TRAFFIC FEATURES EXTRACTED FOR EVERY FLOW

	Feature	Type
1.	Source port	Numerical
2.	Destination port	Numerical
3.	L3 / L4 Protocol identifier	Categorical
4.	L7 Protocol identifier	Categorical
5.	Total number of packets	Numerical
6.	Total number of Bytes (B)	Numerical
7.	Total payload length (in B)	Numerical
8.	Mean of number of B per packet	Numerical
9.	Median of number of B per packet	Numerical
10.	Std of number of B per packet	Numerical
11.	Mean of payload B per packet	Numerical
12.	Median of payload B per packet	Numerical
13.	Std of payload B per packet	Numerical
14.	Percentage of packets < 128B	Numerical
15.	Percentage of packets in (128B,1024B]	Numerical
16.	Percentage of packets in (1024B,1518B]	Numerical
17.	Percentage of packets > 1518B	Numerical
18.	Number of packets per second	Numerical
19.	Number of B per second	Numerical
20.-26.	Counters for TCP flags	Numerical
27.-31.	Counters for ICMP flags	Numerical
32.	Flow duration	Numerical
33.	Mean of inter-arrival time (IAT)	Numerical
34.	Median IAT	Numerical
35.	Std of IAT	Numerical
36.	Bidirectional indicator	Binary
37.	Ratio of number of packets OUT/IN	Numerical
38.	Ratio of number of B OUT/IN	Numerical
39.	Ratio of IAT OUT/IN	Numerical

be observed per flow in order to successfully model malicious traffic. It also fits the scenario of on-the-fly detection, where the traffic classification is implemented periodically using only a limited amount of traffic per flow.

B. The Classifier entity: classification by supervised machine learning algorithms

The Classifier entity uses supervised MLAs to classify traffic flows as malicious or non-malicious. Supervised MLAs [7] are chosen as they represent the class of well-defined machine learning algorithms that provide generally good performances for diverse classification problems. The supervised MLAs generate a function (i.e., model) that maps inputs to desired outputs. The model is trained by inputs and their corresponding outputs, and it is then used to predict output for some future inputs. However, as there is no universally best MLA but only the optimal for the specific learning problem, we consider eight highly regarded supervised MLAs: Naive Bayesian classifier (NB), Bayesian Network classifier (BNet), Logistic Regression (LR), Artificial Neural Networks (ANN), Support Vector Machines with linear kernel (SVMLin), C4.5 decision tree (C4.5), Random Tree classifier (RTree) and Random Forest classifier (RForest). The MLAs have been optimized to provide optimal performances for the used dataset.

IV. EXPERIMENTS AND DETECTION RESULTS

The system has been realized by the pre-processing entity implemented in Python and the classifier entity implemented by Weka machine learning toolbox [7]. Weka is chosen as it provides a stable framework that implements an array of different machine learning algorithms. For the experiments combined malicious and non-malicious traffic traces in a form of large pcap file were used, where pre-processing entity parses the pcap file into arff file suitable for use within the Weka. All experiments were done using an off-the-shelf computer with Intel Core i7 at 3.4 GHz and 16GB of RAM memory.

A. Dataset

Experiments were conducted using ISOP dataset [5] that represents the combination of four publicly available malicious and non-malicious datasets. The ISOP dataset includes two malicious traffic datasets obtained by the French chapter of the honeynet project involving the Storm and Waledac botnets traffic. Waledac is one of the most advanced P2P botnets and it is widely considered as the successor of the Storm botnet with a more decentralized communication protocol. Unlike Storm that uses overnet as a communication channel, Waledac utilizes HTTP communication and a fast-flux based DNS network exclusively. The non-malicious traffic is represented by two datasets, one from the network traffic laboratory at Ericsson Research in Hungary and the other from the Lawrence Berkeley National Lab (LBNL). The Ericsson lab dataset contains a large number of general traffic from a variety of applications, including web browsing, on-line gaming, and popular bit-torrent clients. LBNL contains traffic traces recorded at medium size enterprise network. The four traces have been mapped to the shared address space as illustrated in [5]. However, due to the limited number of IP addresses with malicious traffic, using the IP addresses as features for classification can lead to the optimistic performances of classification. Therefore, in the contrast to the work of Saad et al. [5] we are not using them as features within further analysis. As illustrated by the summary given in Table II the ISOP dataset offers an abundance of data with more than 1.6 million traffic flows. However, the dataset is skewed i.e. different classes are differently represented within the dataset, where malicious flows make 5.76% of total number of flows. This indicates a need for a classification algorithm that can perform well over unbalanced dataset.

TABLE II. THE SUMMARY OF THE ISOP TRAFFIC DATASET

Traffic class	Traffic trace	Number of flows	%
Non-malicious	LBNL and Ericsson lab	1588473	94.24%
Malicious	Storm	64856	3.85%
	Waledac	32118	1.91%
		96974	5.76%
Total		1685447	100%

B. Experiments set-up and evaluation procedure

The experiments are realized for the two scenarios of traffic analysis indicated in Section III. First, we perform the "batch" traffic analysis, where flow statistics is extracted over the whole traffic trace. The batch analysis is realized using the eight MLAs, indicating what MLA can provide the best performances in terms of accuracy and time requirements of classification. In the second, "limited" scenario of analysis we use the best performing MLA from the first scenario, and we vary the duration of flow and the number of packets per flow over which the statistics is taken, illustrating the effect of the two parameters on classification performances. The evaluation of the performances for both analysis scenarios is realized using stratified percentage split evaluation scheme where 66% of total data is used for training and 34% for testing. In order to minimize the influence of random choice of instances in training and test sets, the evaluation is repeated 100 times and the mean of the performances is taken over the repetitions.

The classification performances are expressed by performance metrics that describes both accuracy and time requirements of traffic classification. The accuracy is expressed by following metrics, capable of describing the performance of binary classification over the skewed data set:

1) *Precision (PRC)*: $precision = \frac{TP}{TP+FP}$
2) *Recall (RCL)*: $recall = \frac{TP}{TP+FN}$
3) *F-measure (FM)*: $F-measure = 2 * \frac{recall * precision}{recall+precision}$
4) *Matthews correlation coefficient (MCC)*:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Where TP, TN, FP, FN are number of true positives, true negatives, false positives and false negatives, respectively. In order to assess the time requirements of MLAs two performances are analyzed: CPU time used for training of the model and CPU time used for the actual classification.

C. Results of Experiments

The results of the experiments are presented by the Figure 2, Figure 3 and Table III. Figure 2 illustrates classification performances for malicious botnet traffic and the "batch" scenario of traffic analysis using the eight different supervised MLAs. The performances of detection are expressed by previously introduced performance metrics. Figure 3 illustrates the training and classification time required by different MLAs. The two figures give us the better understanding of the capabilities of MLAs to accurately and timely classify malicious traffic. Table III presents the results of "limited" flow-based analysis where flows were observed for a specific time interval and a specific maximum number of packets. The results of the limited analysis are generated by using Random Tree classifier that yielded the "optimal" results in the first part of analysis. As in the previous case, the presented results refer to the classification of malicious instances. We look at the flows for following maximum duration of 60, 600, 1800 and 3600 seconds and for each of the time intervals we consider several maximum number of packets i.e. 10, 50, 100, 1000, 10000 and 100000 packets. The table illustrates how much traffic need to be observed so the malicious flows could be accurately modeled. Due to the page limitation classification performances for non-malicious traffic are not presented, but it should be noted that all MLAs except Naive Bayesian provided nearly perfect classification of non-malicious instances.

V. DISCUSSION

Based on the Figure 2 and Figure 3 we can conclude that the system has the capability of efficiently identifying botnet traffic using simple feature set. However, the performances of the different MLAs are varying, where from the eight tested algorithms only tree classifiers (C4.5, RTree and RForest) have showed promising classification performances for botnet traffic. The tree classifiers performed well in the terms of accuracy of classification of malicious traffic, while the Random Forest classifier performed the best. However, as the Random Tree classifier stroke the balance between the accuracy of detection and time needed to perform classification, it is consider as the "optimal" for learning task in question. On the other hand the Naive Bayesian classifier underperformed for both malicious and non-malicious traffic, which can be explained by the fact that the independence assumption made by the approach is not fulfilled. The other four MLAs have performed well on the non-malicious traffic which leads us to the conclusion that they are not capable of performing well over the skewed classes.

As illustrated by Table III, the results of limiting length and the number of packets observed per flow show that the system using Random Tree classifier is able to provide steady performances over the various values of the time of observation and the number of packets per flow. As a matter of fact, the system is able to achieve the high performances of detection for monitoring only first 60 seconds and 10 packets per flow, while at 1000 packets per flow the performances peek and further increase of number of monitored packet do not improve the detection results. This can be explained upon a closer inspection of malicious botnet traffic where we found that the majority of malicious flows have less than 1000 packets, while a significant number of flows have length less than 10 packets. The is expected from P2P botnets such as Waledac and Storm that have relatively short C&C communication and that implement Spam campaigns as attack vectors.

The detection performances obtained by the "limited" analysis are comparable with ones reported by Saad et al. [5], where our system used much simpler feature set, and limited number of packets per flow. Additionally, our system does not rely on IP addresses as traffic features avoiding optimistic classification results on IP-biased datasets. The obtained results can be explained do to the use of tree classifiers that are more suitable for modeling malicious traffic than MLAs used in [5].

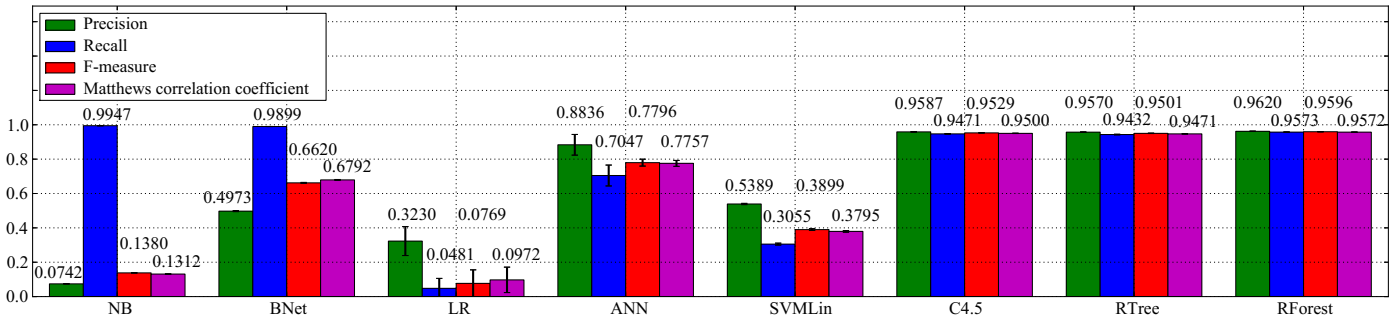

Fig. 2. Classification performances for botnet traffic using different supervised MLAs

TABLE III. CLASSIFICATION PERFORMANCES FOR BOTNET TRAFFIC USING RANDOM TREE CLASSIFIER AND VARYING DURATION OF FLOW AND NUMBER OF PACKETS PER FLOW

	60 seconds				600 seconds				1800 seconds				3600 seconds			
	PRC	RCL	FM	MCC	PRC	RCL	FM	MCC	PRC	RCL	FM	MCC	PRC	RCL	FM	MCC
10 packets	**0.9561**	**0.9406**	**0.9483**	**0.9452**	0.9558	0.9408	0.9482	0.9451	0.9558	0.9415	0.9486	0.9455	0.9561	0.9417	0.9489	0.9458
50 packets	0.9562	0.9408	0.9485	0.9454	0.9559	0.9411	0.9484	0.9453	0.9561	0.9417	0.9489	0.9458	0.9564	0.9420	0.9491	0.9461
100 packets	0.9561	0.9411	0.9486	0.9455	0.9558	0.9410	0.9484	0.9453	0.9561	0.9416	0.9488	0.9458	0.9564	0.9414	0.9489	0.9458
1000 packets	**0.9563**	**0.9410**	**0.9486**	**0.9455**	**0.9558**	**0.9412**	**0.9485**	**0.9454**	**0.9563**	**0.9418**	**0.9490**	**0.9459**	**0.9567**	**0.9419**	**0.9492**	**0.9462**
10000 packets	0.9563	0.9409	0.9485	0.9455	0.9557	0.9412	0.9484	0.9453	0.9561	0.9416	0.9488	0.9457	0.9564	0.9417	0.9490	0.9460
100000 packets	0.9563	0.9409	0.9485	0.9455	0.9557	0.9412	0.9484	0.9453	0.9562	0.9417	0.9489	0.9458	0.9564	0.9418	0.9491	0.9460

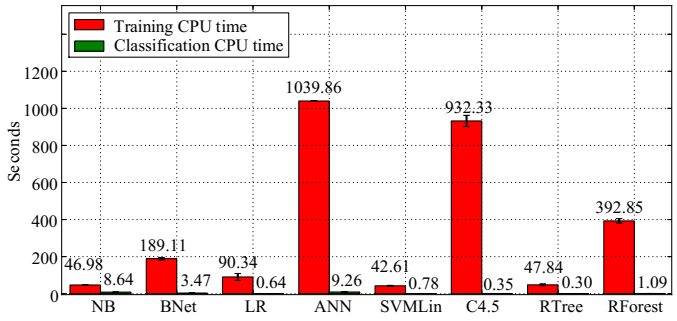

Fig. 3. Training and classification CPU time using different supervised MLAs

The findings give us a positive indication that the proposed detection system can provide both accurate and timely detection using only a limited amount of traffic per flow, thus promising detection in less time and expense. As the system is able to detect the malicious flows for a small number of packets we argue that even if the attack is ongoing it could be detected in the early phase before having a greater impact. Furthermore, the system could be used within a future on-line detection framework where the system would be able to operate in a time window being periodically retrained. The future work will be directed at the further evaluation of the system with additional botnet traces as well as feature optimization so the patterns of malicious traffic could be modeled even better. Furthermore, we will consider using the detection principles presented here for the development of future on-line detection system.

VI. CONCLUSION

Botnets, as one of the most serious cyber security threats require efficient detection in order to be effectively neutralized. This paper explores how flow-based traffic analysis and supervised machine learning can be used to provide that. We developed a novel botnet detection system that relies on flow-level network traffic analysis and supervised MLAs for capturing patterns of malicious botnet traffic. For the realization of the system an array of contemporary MLAs have been considered. As a result, the new detection system has proved to be accurate in detecting botnet traffic using simple flow features and Random Tree classifier. Additionally, the experiments showed that in order to provide a high accuracy of detection the traffic flows need to be monitored for only a limited duration of time and a limited number of packets per flow. The proposed system achieved accurate detection of the botnet traffic for only 10 packets and 60 seconds of monitoring per flow. The results indicate possibilities of using the presented approach in a more adaptive set-up that could provide on-line detection. The future work will be devoted to the optimization of traffic analysis and deployment of the presented detection approach in on-line fashion.

REFERENCES

[1] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378 – 403, 2013.

[2] M. Stevanovic and J. Pedersen, "Machine learning for identifying botnet network traffic," Aalborg University, Tech. Rep., 2013.

[3] W. T. Strayer, D. Lapsely, R. Walsh, and C. Livadas, "Botnet detection based on network behaviour," in *Botnet Detection*, ser. Advances in Information Security, W. Lee, C. Wang, and D. Dagon, Eds. Springer, 2008, vol. 36, pp. 1–24.

[4] M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. Hamlen, "Flow-based identification of botnet traffic by mining multiple log files," in *Distributed Framework and Applications, 2008. DFmA 2008. First International Conference on*, oct. 2008, pp. 200 –206.

[5] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, and P. Hakimian, "Detecting p2p botnets through network behavior analysis and machine learning," in *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, july 2011, pp. 174 –180.

[6] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 129–138.

[7] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.