Multiclass Machine Learning Based Botnet Detection in Software Defined Networks

Farhan Tariq[1†] and Dr. Shamim Baig[2††]

[1]Electrical and computer engineering, Centre for advanced studies in engineering, Islamabad, Pakistan
[2]Computer Science and Engineering, HITEC University, Taxila, Pakistan

Summary
Continuously evolving nature of botnet by using innovative approaches and technologies derives the need for continuous improvement of botnet detection solutions. The state of the art network approaches in literature targeting network header level information only for behavioral-based detection. These techniques applying machine learning algorithms to automatically detect botnet patterns from network flows. The current work in flow-based approaches exploring SDNs to overcome traditional IP network complexities. The Software defined network technology platform with centralized visibility and control provide an opportunity to redesign these approaches. The current SDNs based proposed approaches apply binary classification to decide if the detected flow belongs to a botnet or not. This work proposed a multiclass machine learning based approach to address botnet problem in SDNs. The proposed scheme applies multiple binary classifiers each trained for a specific type of botnet class. These focused classifiers performed better in the detection of the specific type of botnet. The proposed approach uses the flow trace concept. The features are extracted for each detected flow trace and fed into these focused classifiers. These features are examined by all classifiers and detected label is added for each processed flow trace. These labels are aggregated in the second stage to decide if a flow trace belongs to any botnet class or not. This additional information of a class of the detected botnet trace is helpful during the incident response process. The experiments for evaluation of the proposed work are performed on real-world traffic traces and the result shows promising detection rate with the capability to detect unknown botnet.

Key words:
botnet detection, malware, Multiclass machine learning, NBA, SDN, TSDR, OpenFlow, Opendaylight, flows.

1. Introduction

The increasing number of malicious activities and attack sophistication posing a great threat to entire IT infrastructure. The evolving nature of malware introducing new attack vectors. This changing threat landscape becoming the biggest challenge to current defensive approaches. The signature-based approaches are not practical to cope up with a rapidly changing footprint of malicious activities. The encrypted command and control (C&C) communication of malicious activities also a bottleneck for signature-based approaches. This lead botnet researchers interest towards behavior based approaches. The latest network level proposal in literature focusing only on network header information to detect malicious call-backs.

The bot-malware, with networking capability and command and control mechanism, become one of the most powerful tools for malicious activities. A machine infected with bot-malware is recognized as a bot machine. The bot machines try to register with their C&C servers. This registration established a network of bot machines and their C&C servers. The network of bot machines controlled by bot-master is called botnet. The botnet is mainly categorized into centralized and decentralized botnet based on their underline command and communication network infrastructure. The IRC and HTTP botnet fall in the centralized botnet category as these connects back to their central C&C servers. The P2P botnet is decentralized as these bot machines first search the local network for the presence of similar bot machines and select a commander locally from these. This selected commander act as a C&C server for all bot machines on the local network. This local C&C server then connects back to central C&C servers for instructions. The botnet life-cycle mainly divided into three phases of recruitment/infection, C&C communication, and attack phase. The bot-master use recruitment phase to infect new machines with bot-malware. The newly infected bot machine connects back to its C&C server for registration during the C&C communication phase. The bot machines keep in touch with C&C server for instructions during this phase. The attack phase starts when bot machines receive instructions to perform some sort of malicious activities.

The detection techniques in botnet literature mainly focus on the C&C communication phase of a botnet. This phase of botnet life leaves a network footprint which helps detection technique to classify the same as a botnet. This phase also helps to detect both the bot machine and connected C&C server. The flow-based proposal for the classical network in literature suffers from flow collection complexities. The traditional IP network challenges moves

Manuscript received March 5, 2019
Manuscript revised March 20, 2019

the botnet research focus towards software defined networks (SDNs). The SDNs with centralized visibility and control provide researchers more flexibility to cope up with botnet problems. Botnet literature has few proposals to address the botnet problem in SDNs. These flows based proposed approaches apply the binary classifier to detect botnet patterns from monitored network flows [1] [3].

This work aims to address the botnet problem in SDNs. The work proposes a multiclass machine learning based approach to analyze the network flows to detect C&C communication phase of a botnet in the monitored network. The work also evaluates the effect of multiple binary classifiers, trained to detect a specific type of botnet, on botnet detection capability. To do this, the real-world botnet traffic traces are collected to form a dataset of 9 botnet families. This dataset is comprised of 3 groups one from each specific type of botnet class of IRC, HTTP, and P2P. These three groups are used to train botnet type specific binary classifiers. These classifiers process network flows trace for botnet type detection. The results of these classifiers aggregated to decide if the processed flow trace belongs to a botnet or normal traffic. The conclusive results are compared with previously proposed binary classifiers-based approaches to evaluate the effect of type-specific classifiers. The contribution of this work is summarized as below.

- This work proposes a multiclass machine learning approach by splitting a generic botnet classification problem into three binary classifiers each trained to detect specific type of botnet class of IRC, HTTP, and P2P respectively.
- The output of these three classifiers are aggregated to make a final decision in binary form to make results directly comparable to the previous similar proposal.
- The proposed work tested on nine real-world botnet traces and shows improved detection rate.

The structure for the rest of the paper is as follows: Section 2 discussed the related work from SDNs and the traditional IP network. The proposed method is presented with all its detail in the section 3. The evaluation and results are discussed in section 4. The section 5 finally provides the conclusion of the paper and suggest future work.

2. Related Work

Botnet literature has a good coverage of network flow based botnet detection techniques addressing traditional IP network. This proposed work varies between approaches that are focusing to detect only a specific class of botnet and approaches that are designed to detect generic botnet independent of botnet underline C2C communication network structure. Flow based techniques extract statistical features from network flows of monitored network. These statistical features help to uniquely identify the botnet related network flows. The livadas et al. [12] introduced the use of machine learning algorithms to detect botnet. The machine learning algorithms help to automatically detect botnet using flows statistical feature set. Both supervised and unsupervised machine learning algorithms are applied to multiple state of the art flow-based botnet detection approaches. Software defined networks the emerging network architecture is not much explored by botnet researchers. The few proposed approaches provide only information if the detected flow belongs to botnet or not. This work applies the multiclass supervised machine learning approach to detect botnet related flows in SDNs. The detailed analysis of some of the stat of the art related approaches are discussed below.

The work proposed in Disclosure [9] apply supervised machine learning algorithms on NetFlow records, A Cisco proprietary protocol for the network flows information, to detect botnet presence in the tradition IP network. The approach introduces new statistical features of flow size based, client access pattern based, and temporal behavior based. This work focuses on C2C communication phase of the botnet and works only on servers related network flows. Three supervised machine learning algorithms namely C4.5, Random forest, and support vector machine are applied to automatically extract botnet behavior patterns. The approach shows a detection rate of 70 % in experimentation result. The generic botnet detection is proposed in this work. The detection approach moves to interval based analysis to detect the botnet in near real-time. The work proposed in [david zhao] investigate different time intervals to find out one that is enough to do flows analysis for botnet detection. This work divides the flow dataset into time-based intervals and analyzes one interval at a time. The statistical features are extracted from flows fall in the defined time interval. The reptree machine learning algorithm is applied to extract features to extract botnet patterns. The interval of 300 seconds is found to be best in experimental results of this work. The work addresses the generic botnet detection problem in traditional IP network.

The work proposed F. Haddadi in [4]. Investigate the structural differences in p2p and HTTP botnets. The work uses three machine learning algorithms including C4.5, SBB, and Bayesian networks to create detection models for p2p and HTTP botnets. The experimental results show that C4.5 outperforms SBB and Bayesian Networks for both p2p and HTTP botnets. The work also shows that the connection-based features have high information gain for p2p botnet whereas for HTTP botnet inter-arrival based features shows high information gain. The work in [2] proposed to detect P2P botnets using multiple binary classifiers in software defined networks. The proposed work used a binary classifier for each P2P traffic type including botnet traffic of Storm and Zeus, and normal P2P application including eMule, uTorrent, Skype. The experimental results show more than 95% average detection rate.

The work proposed by F. Tariq in [1] proposed to detect generic botnets in software defined networks. The proposed approach works on a flow trace regardless of working on individual flows. Flow trace is an ordered sequence of flows between two endpoints. A flow trace with 10 flows considers as a trace of interest. Statistical features are extracted for each trace of interest from current network activity of the trace and historical flows for the source and destination IPs of the trace. This rich feature set helps supervised decision tree-based classification algorithm to uniquely identify botnet generated flow traces. The experimental result shows high detection rate with overall system accuracy of 94.8%.

3. Proposed Detection Model

This work proposed a network flow-based botnet detection mechanism to extract botnet behavior patterns with help of supervised machine learning algorithm. The proposed approach aims to address botnet detection problem in software defined networks and evaluate the effect of ensemble multiple botnet type specific classifiers. The proposed work detects C&C communication patterns from network flows of monitored network. The approach works on network flow traces as introduced in [1] rather than working on individual flows of a communication between two network endpoints. The global feature set is extracted for each detect flow trace. This feature set is then used during the training and testing phase of the proposed method. The figure1 shows the classification process of the proposed scheme. The flow stream collection, flow trace detection and feature extraction for each detected flow trace are common in both training and testing phase of the proposed method. The training phase is used to create a botnet type-specific detection model to detect IRC, HTTP, and P2P botnet respectively. These models are then used to detect botnet presence of botnet in the monitored network during the testing phase.

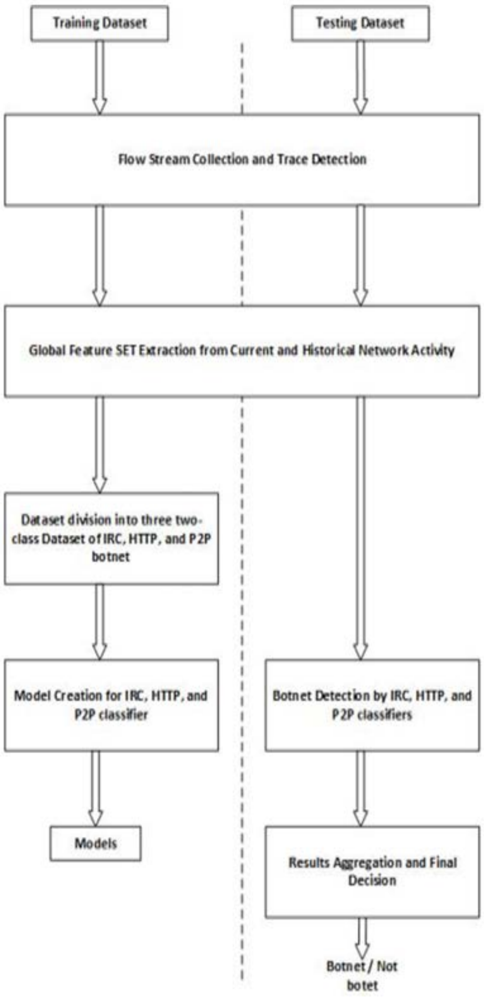
Training Dataset
Testing Dataset
Flow Stream Collection and Trace Detection
Global Feature SET Extraction from Current and Historical Network Activity
Dataset division into three two-class Dataset of IRC, HTTP, and P2P botnet
Model Creation for IRC, HTTP, and P2P classifier
Botnet Detection by IRC, HTTP, and P2P classifiers
Models
Results Aggregation and Final Decision
Botnet / Not botet

Fig. 1 Classification process of proposed method

The idea of using multiple classifiers tuned for a specific class of botnet conceived from the statistics of botnet literature where type specific techniques perform better than generic approaches. The detail of training phase and testing phase discussed in section 3.1 and 3.2 respectively.

3.1 Training Phase

As discussed above, the proposed approach applies multiclass machine learning. The purpose of the training phase is to create detection models that can identify the class of detected botnet. To handle multiclass classification this work applies binary decomposition strategy. The common decomposition strategies for the multiclass problem are one-vs-one (OVO) and one-vs-all (OVA). The OVO approach is more complex as compare to OVA as it requires a binary classifier for each possible combination of a pair of classes. whereas OVA approach divides the problem into the number equal to a total number of classes. This approach requires training a classifier for each class of botnet. The OVA binarization approach is applied in this work. The training dataset is divided into three one-class labeled datasets for botnet class of IRC, HTTP, and P2P respectively. These one-class datasets are used to train C4.5 binary classifiers and a botnet class focused detection models are created.

3.2 Testing Phase

The decision tree-based detection model created during training phase used to detect a corresponding class of botnet from unknown data in the testing phase. The feature set extracted for each detected flow trace fed into all of three detection models in parallel for processing. The detection mechanism of testing phase works in a layered approach. The first layer comprised of detection model of IRC botnet, HTTP botnet, and P2P botnet. These model process incoming stream of flow trace feature set and assign a label to each processed flow trace. The output of the first layer has class information with the detected botnet. The second layer of the testing phase aggregate the result of the first layer and provide a final decision if the processed flow trace belongs to a botnet or normal traffic. The figure2 shows the working of the testing phase. The aggregation function considers a combination as positive if one and only one of the classifier provide the positive output. The all other combinations are considered as negative. The goal of the second layer is to discard any positive output with low detection confidence from the first layer. Table 1 translates the working of the aggregation function.

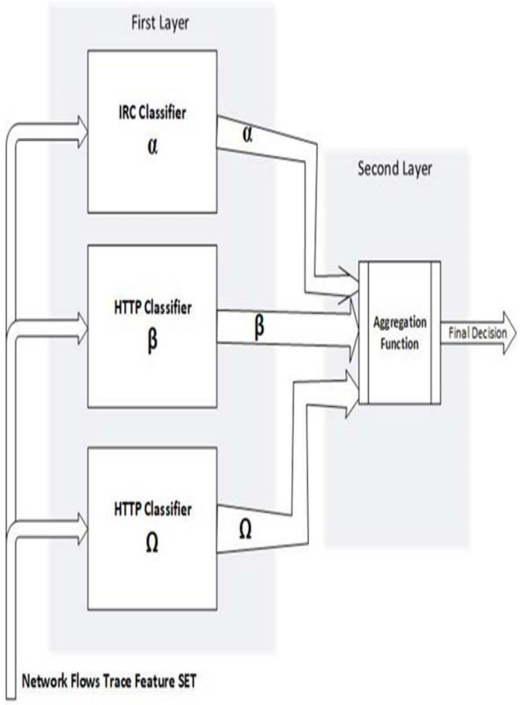

Fig. 2 Testing phase of proposed method

Table 1: Aggregation Function

IRC	HTTP	P2P	Decimal Representation	Final Decesion
0	0	0	0	Not Botnet
0	0	1	1	P2P
0	1	0	2	HTTP
0	1	1	3	Not Botnet
1	0	0	4	IRC
1	0	1	5	Not Botnet
1	1	0	6	Not Botnet
1	1	1	7	Not Botnet

3.3 Flows Collection and Trace Detection

The proposed approach process network flows as an input and group together flows of same network communication between two endpoints. The network flows are collected at every five-minute interval centrally from SDN controller. Each collected batch of flow processes to identify and construct flow traces using a trace key <Source IP, Destination IP, Destination Port, Protocol> as used in proposed work of [1]. The trace detection function of this work counts all the flows of a trace from a collected batch of flows rather than restarting the flow counter after ten flows as proposed in [1]. All the detected flow traces having 10 or more flows are marked as a trace of interest (TOI).

3.4 Global Feature Set Extraction

The statistical features are extracted for each trace that is marked as TOI. Feature extraction process works in two steps. First step extract features from a batch of flows constructed during trace detection. Table2 list the extracted features.

Table 2: Extracted features – First step

Feature	Description
Duration {Min, Max, Mean}	Flow duration
Bytes {Min, Max, Mean}	Total bytes received
Packets {Min, Max, Mean}	Total packets received
BPS {Min, Max, Mean}	Bytes per second
PPS {Min, Max, Mean}	Packets per second
BPP {Min, Max, Mean}	Bytes per packet
Fit {Min, Max, Mean}	Flow interarrival time

In second step historical flows of the source and destination IPs of the processed trace are collected for last 60-minute window. This batch of flows helps to extract more enrich features as given in table 3.

4. Evaluation

The dataset selection is a key feature in performing experiments to evaluate a machine learning based detection approaches. This selection must satisfy generality, realism, and representativeness as identified in work proposed in [6]. The selection of the dataset for this work is based on following considerations.

Table 3: Extracted features – Last 60 minute

Feature	Description
Count of SP-SIP	Unique count of Source Port for Source IP
Count of DP-SIP	Unique count of Destination port for Source IP
Count of SP-DIP	Unique count of Source Port for Destination IP
Count of DP-DIP	Unique count of Destination Port for Destination IP
Count of DIP	Unique count of Destination IP for Source IP
Count of SIP	Unique count of Source IP for Destination IP
Flows-SIP	Total number of flows from Source IP
Flows-DIP	Total number of flows from Destination IP

a. The dataset must be generic and have botnet representation from each class of botnet.
b. The selected botnet traffic traces must be generated by real botnet samples.
c. The normal traffic traces collected from real-world traffic capture to represent real network environment.
d. The selected traffic traces are publicly available.

The real-world traffic traces for botnet and normal traffic are collected from two well-known publicly available botnet datasets [7] [11]. These network traffic traces are collected in the form of packet capture "pcap" files. The selected dataset comprises of nine real-world botnet traffic traces and normal traffic. These files are used to replay capture traffic traces to generate training and test data. The selected files and their collection sources are shown in table4.

4.1 Training Dataset

The aim of this paper is to improve detection rate by splitting a single generic botnet classifier into multiple botnet class focus classifiers. For this purpose, three structure focused classifiers use to detect IRC, HTTP, and P2P botnet, respectively. To address this multiclass problem binary decomposition is applied using OVA approach as discussed above. This binary decomposition requires three one-class datasets for IRC, HTTP, and P2P classifiers. The subsequent sections discussed the training dataset for each classifier.

Table 4: Selected Traffic Traces

traffic traces	Type	Collection Source
Normal	Mix Traffic	ISOT
Rbot	IRC	CVUT
Murlo	IRC	CVUT
Neris	IRC	CVUT
Virut	HTTP	CVUT
Soguo	HTTP	CVUT
Zeus	HTTP	CVUT
NSIS	P2P	CVUT
Sality	P2P	CVUT
ZeroAccess	P2P	CVUT

a. IRC Botnet Classifier

To generate the training dataset for IRC classifier, two of the IRC botnets including Neris and RBot selected to train the IRC classifier. The traffic traces of these two are labeled as "IRC". whereas the remaining traffic including HTTP botnet, P2P botnet, and normal traffic traces are labeled as "NOT IRC". The table5 shows the training dataset for IRC classifier.

Table 5: Training Data for IRC Classifier

Botnet	Type	Percentage of flows
Rbot	IRC	12.8%
Neris	IRC	15.3%
Virut	NOT IRC	8.5%
Soguo	NOT IRC	5.3%
NSIS	NOT IRC	4.7%
Sality	NOT IRC	6.5%
Normal	NOT IRC	46.9%

b. HTTP Botnet Classifier

The Virut and Zeus botnet from HTTP class of botnet is used to train this classifier. The traffic traces of these two botnets are labeled as "HTTP", and all other traffic traces including botnet samples from other classes and normal traffic are labeled as "NOT HTTP". The distribution of training data for HTTP classifier is shown in table 6.

Table 6: Training Data for HTTP Classifier

Botnet	Type	Percentage of flows
Rbot	NOT HTTP	8.7%
Neris	NOT HTTP	10.1%
Virut	HTTP	18.5%
Zeus	HTTP	14.3%
NSIS	NOT HTTP	7.5%
Sality	NOT HTTP	8.1%
Normal	NOT HTTP	32.8%

c. P2P Botnet Classifier

Two P2P botnets including Sality and ZeroAccess used to prepare the training data for the P2P classifier. The traffic traces replayed from these two botnets are labeled as "P2P". The botnet traces from other classes and normal traffic files are also used to generate a representative traffic that is labeled as "NOT P2P". Table 7 shows the traffic distribution for P2P botnet classifier.

4.2 Test Dataset

To assess the performance of created botnet detection models of IRC, HTTP and P2P a generic test data is simulated in a test environment. The three new botnet samples one from each of botnet class are added into test data to introduce diversity. This help to analyze if the proposed approach can detect unknown botnets. Table 8 shows the distribution of traffic traces of test data.

Table 7: Training Data for P2P Classifier

Botnet	Type	Percentage of flows
Rbot	NOT P2P	8.8%
Neris	NOT P2P	9.6%
Virut	NOT P2P	9.2%
Soguo	NOT P2P	6.8%
Sality	P2P	11.8%
ZeroAccess	P2P	16.1%
Normal	NOT P2P	37.7%

Table 8: Test Data

Botnet	Type	Percentage of flows
Rbot	IRC	8.8%
Neris	IRC	9.6%
Murlo	IRC	5.9%
Virut	HTTP	9.2%
Soguo	HTTP	6.8%
Zeus	HTTP	10.2%
NSIS	P2P	4.3%
Sality	P2P	9.8%
ZeroAccess	P2P	11.1%
Normal	Normal	24.3%

4.3 Results

This work proposes a multiclass botnet detection method by splitting a generic botnet classifier into three structure focus classifiers. The experiments are performed using real-world botnet traffic traces to evaluate the performance of these classifiers. This section compiles the results to analyze the performance of individual classifier and the overall botnet detection capability of the proposed method.

The proposed class specific classifier performs well individually as shown in table 9. The IRC classifier achieve the highest precision rate of 98.4%, following by the P2P classifier with precision of 97.7%. The HTTP classifier stand at third by achieving precision of 97.4%. These results are directly comparable to the results shown in previously proposed approaches [1] [2] [5].

The results of individual classifiers are recompiled into a binary form to compare the overall accuracy of the system with previously proposed approaches. To access performance of the proposed method the classifiers results for both botnet and normal traffic traces are compiled to form a confusion matrix. Table 10 shows the confusion matrix to represent the combined result of individual binary classifiers.

Table 9: Result Summary of Each Classifier

Classifier	TOI	Correctly classified	Incorrectly classified	Detection
IRC	63	62	1	98.4%
HTTP	78	76	2	97.4%
P2P	45	44	1	97.7%

The accuracy and the precision of the proposed system are calculated using equation (1) and (2) respectively from the confusion matrix.

$$Accuracy = \frac{182+357}{182+357+15+4} \quad (1)$$

$$Precision = \frac{182}{182+4} \quad (2)$$

The proposed system achieves the accuracy is 96.6% with a precision of 97.8% which is higher than the reported results of work proposed in [1], [3], [6], and [9].

Table 10: Confusion Matrix

		Actual	
		Botnet	Normal
Detected	Botnet	182	4
	Normal	15	357

5. Conclusions

This paper proposes to split generic botnet classification process from single classifier to multiple botnet class specific classifiers. The botnet families can be divided into three classes of IRC, HTTP, and P2P based on underline command and control communication network architecture. To do this one-vs-all binary decomposition approach of multiclass machine learning is applied to train three binary classifiers for IRC, HTTP, and P2P botnet detection. The contribution of this work is twofold. First, the proposed work additionally provides the class of each detected botnet flow. This additional information of class with each detected botnet flow has potential to direct incident response process towards more focus way. Second, the experimental results proved that proposed scheme improved the botnet detection performance as compared to previously proposed approaches.

This study provides a network flow based method for generic botnet detection in software defined networks. The output of this botnet classification scheme is directly consumable in security operation center of the monitored network as an alert. The potential future extension of the proposed scheme is to use botnet class information to design automated mitigation strategies.

References

[1] Tariq, F., & Baig, S. (2017, November). Machine learning based botnet detection in software defined networks. International Journal of Security and Its Applications, 11(11), 1-12.

[2] Su, S. C., Chen, Y. R., Tsai, S. C., & Lin, Y. B. (2017). Detecting P2P Botnet in Software Defined Networks. Security and Communication Networks.

[3] Tariq, F., & Baig, S. (2016). Botnet classification using centralized collection of network flow counters in software defined networks. International Journal of Computer Science and Information Security, 14(8), 1075.

[4] da Silva, A. S., Machado, C. C., Bisol, R. V., Granville, L. Z., & Schaeffer-Filho, A. (2015, September). Identification and selection of flow features for accurate traffic classification in sdn. In Network Computing and Applications (NCA), 2015 IEEE 14th International Symposium on (pp. 134-141). IEEE.

[5] Haddadi, F., & Zincir-Heywood, A. N. (2015, October). A Closer Look at the HTTP and P2P Based Botnets from a Detector's Perspective. In International Symposium on Foundations and Practice of Security (pp. 212-228). Springer, Cham.

[6] Beigi, E. B., Jazi, H. H., Stakhanova, N., & Ghorbani, A. A. (2014, October). Towards effective feature selection in machine learning-based botnet detection approaches. In Communications and Network Security (CNS), 2014 IEEE Conference on (pp. 247-255). IEEE.

[7] Garcia, S., Grill, M., Stiborek, J., & Zunino, A. (2014). An empirical comparison of botnet detection methods. computers & security, 45, 100-123.

[8] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., & Garant, D. (2013). Botnet detection based on traffic behaviour analysis and flow intervals. Computers & Security, 39, 2-16.

[9] Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., & Kruegel, C. (2012, December). Disclosure: detecting botnet command and control servers through large-scale NetFlow analysis. In Proceedings of the 28th Annual Computer Security Applications Conference (pp. 129-138). ACM.

Farhan Tariq received his Master of Computer Engineering degree with first class honors from CASE Pakistan in 2011. He is currently working towards a Ph.D. degree at Center for Advanced Studies in Engineering. His research interests include network monitoring and security. Specifically, network behavioral monitoring to detect the presence of malicious call-backs.

Dr. M Shamim Baig is Ph.D. in Computer Science from George Washington University Washington DC, USA; MS in Industrial Electronic from Cranfield Institute of Technology UK. He has more than 40 years of Academic, Research & Engineering Management experience in the field of Supercomputing, Digital System Design, Networking & Information Security. He has been Air Vice Marshal in Pakistan Air Force, Principal Scientific Officer at A.Q. Khan Research Labs & Director General / Dean “Centre of Excellence for Cyber Security” at National University of Science & Technology Islamabad. He is currently a Professor/ Director Advanced Studies & Research at Center for Advanced Studies in Engineering Islamabad. He has published more than35 Int’l Journal/ conference papers. He has been Chair IEEE education activities & Keynote/ invited speaker at multiple Seminars