Towards real-time profiling of human attackers and bot detection

Avgoustinos Filippoupolitis and George Loukas

Stelios Kapetanakis

I. Abstract

Characterising the person behind a cyber attack can be highly useful. At a practical security and forensic level, it can help profile adversaries during and after an attack, and at a theoretical level it can allow us to build improved threat models. This is, however, a challenging problem, as relevant data cannot easily be found. They are not often released publicly and may be the result of criminal investigation. Moreover, the identity of an attacker is rarely revealed in an attack. Here, we attempt a rather unusual approach. We attempt to classify the adversary as a type of human user, arguing that if it does not fit in any realistic profile of a human user, then it is probably a bot. Hence, we are working towards a system that is both a human attacker profiler and an anomaly-based bot detector. For this, we first need to build a technical system that collects relevant data in real-time. As no such information exists, we experimented with several different measurable input data and human profile characteristics, evaluating the usefulness of the former in determining the latter. We then present a case-based reasoning approach that classifies an attacker based on the values of these metrics. For this, we use experimental data that we have previously collected and are the result of a set of cyber-attack scenarios carried out by 87 users. As a practical application, we have developed an automated profiling tool demonstrating the potential real-time use of the proposed system in a quasi-realistic setting. We discuss this approach's ability for an adversary that has already gained access to a target system. The profile identified should tell us the characteristics of the adversary if it is human. If no profile can be identified, we argue that this is a good indication it is a bot.

II. Introduction

The behaviour of different cyber attackers may differ depending on their skills, knowledge, experience, mode of operation, tactics, education, target and many other parameters. As a result, researchers have always looked at the different profiles of cyber attackers from a social sciences perspective. Computer scientists instead have focused on developing security tools that help detect attack profiles while they are in progress, and forensic tools that help analyse attacks after they have occurred. There is also a class of real-time forensic tools that help a human user analyse attacks while they are in progress, for example by presenting real-time network information in a user-friendly manner [1]. Here, we set a different challenge. We try to determine the approximate human profile of an attacker, as defined by a set of characteristics, based on real-time measurements and without the involvement of a human operator. In addition, we argue that if a new attacker detected in a victim machine does not fit any realistic human profile, then this attacker is probably a bot. For this first attempt to provide a corresponding detection tool that is usable in real-time, we use the results of 87 experiments with human attackers and use machine learning to develop a set of rules that are incorporated in this tool. Machine learning has been used extensively in finance, business and health for its ability to provide answers based on association modelling, prediction and forecasting [19] and has recently been employed for scenario-based cyber incident notification [22] and to model human behaviour to anticipate insider cyber attacks based on personnel records and logs [20]. Here, we use machine learning to profile the cyber attacker based solely on events monitored automatically during an actual attack and without any prior information.

The rest of the paper is structured as follows: In Section III we give an overview of the literature related to our problem domain. The methodology we followed to conduct our analysis is presented in Section IV while the description of the algorithms used for profiling is given in Section V. Finally, Section VI presents how we used our approach to detect a bot attack. We conclude our investigation with Section VII which suggests directions for future work.

III. Related work

Bot detection is usually carried out at the level of a whole botnet. The defender needs to have access to multiple nodes across a network, so as to statistically analyse significant amounts of incoming traffic and determine commonalities. In this sense, botnet detection shares a lot of characteristics with distributed denial of service detection [2], [3], [4], where rates and similarities in large numbers of inbound data packets reveal information about the attacker. That is because the usual botnet life-cycle includes multiple phases, such as communication over a command and control channel, that involve the generation of non-negligible network traffic. For example, botsniffer uses correlation analysis al-

This work has been carried out with the support of EU ISEC project E-Centre (HOME/2011/ISEC/AG/INT/400000226 "ECENTRE: Cybecrime Centre of Excellence Network for Training, Research and Education")

gorithms to exploit the inherent synchronisation of multiple bots that belong to the same botnet [5]. BotMiner extends this with advanced data mining techniques [6]. Of course, the level of analysis that is required can only be performed offline, after collecting all data from the various distributed sources. A notable exception is the work of Ramachandran et al., whose heuristics identify botnets by passively analysing DNS-based Black-hole List lookup traffic [7]. As, in this paper, we aim to detect bot behaviour through access to a single machine, without network or other distributed information that can be statistically analysed, our approach is to try to first define what a human behaviour is, or in other words what the typical profile a human hacker is.

As early as in 1985, Landreth, an accomplished hacker himself [10], categorised hackers into novices, students, tourists, crashers and thieves. Since then, there have been several efforts to profile hackers in terms of their motivation and personal characteristics, almost exclusively in social sciences. A 2003 study of 457 self-identified hackers within the Russian-speaking community was the basis for a flow-based model of hacker motivation in relation to challenges and skills [14]. More recently, the Hacker Profiling Project provided the first steps towards applying the science of criminal profiling to hacking [16]. Its researchers studied the background and behaviour of 570 hackers through questionnaires and identified several useful patterns with regards to demographics, age, attitudes and personal traits.

Kjaerland [12] used a 2000-2002 sample of reported incidents to CERT/CC to classify incidents based among others on the method of operation of the attacker. Using smallest space analysis, the author determined the factors that were most likely to happen together. The conclusions that were reached at the time may not be applicable today, but the approach probably is. Shaw et al. [11] have tackled the profiling specifically of malicious cyber insiders from a psychological point of view, focusing on history of negative social and personal experiences, lack of social skills, sense of entitlement and ethical flexibility. Watters et al. have provided an ethnographic study of cyber attacks with the aim to identify attacker profiles qualitatively [17]. Based on benchmark indicators of cyber crime within the Australian financial services system, their initial model relates frequency, distribution and impact of attacks with national indices, such as GDP, level of education, Internet penetration and perceived corruption. Such models can be used in the long-term to predict broader cyber attack trends, but cannot be used in real-time or to profile specific attackers. Kilger et al. [15] have provided an analysis of the social structure of the hacker community through content analysis of the words and phrases used. They provided a classification of motivations, including money, entertainment, ego, cause/ideology, entrance to a social group, and status. They also demonstrated a case study of how monitoring of IRC channels can help profile and often identify specific hackers, but did not provide any systematic framework for this.

Although very interesting, most of these personal characteristics of hackers cannot be measured in real-time and as such cannot be of use during an actual cyber attack. Their practical purpose has been to raise awareness and inform training, policy and business processes [21]. Our purpose instead is to develop a technical system that provides an approximate profile of an attack's perpetrator during the attack, using data that can be captured in real-time on the target computer. Thus, we have focused on characteristics that, we argue, can be observed in real-time and in an automated fashion (see Section IV-A).

IV. Methodology

In this section we give a description of the methodology we adopted in order to carry out our analysis. We describe the list of attacker features we identified, their classification based on their observability and give the details of the experimental data collection.

A. Identifying Cyber-attacker features

We have identified a range of features related to the properties of a potential cyber-attacker. These are related to cultural, professional and demographical characteristics of the person behind the attack. The following list gives an overview of the features in question.

- **Skill:** This feature captures the competence of the attacker in terms of IT skills.
- **Education:** The education level of the attacker is described by this feature.
- **Risk:** A person trying to avoid risks is very likely to behave in a different way compared to one that is risk prone. This feature tries to capture this aspect of the attacker's profile.
- **Gender:** Being able to state whether the person the attack is a male or a female can significantly help a forensics investigation.
- **Goal:** This features describes what was the reason for the attacker to target the computer system in question. A person that attacks a system in order to achieve financial gains needs to be treated separately compared to one that does it out of curiosity or in order to state his political beliefs.
- **Speed:** This feature measures the speed of the cyber attacker in commands per second. It is directly related to the attacker's IT skill level.
- **Mistakes:** A highly skilled attacker would make less mistakes compared to an inexperienced one. This feature captures the number of mistaken commands issued by the attacker in the course of the cyber attack.
- **Anti-forensics:** One of the most important elements of a forensic investigation is the analysis of potential tracks the attacker has left behind. This feature discriminates between attackers that tried to cover their tracks

Table I
CLASSIFICATION OF FEATURES WITH RESPECT TO OBSERVABILITY

Feature	Observable
Skill	No
Education	No
Risk	No
Gender	No
Goal	No
Speed	Yes
Mistakes	Yes
Anti-forensics	Yes
Success	Yes

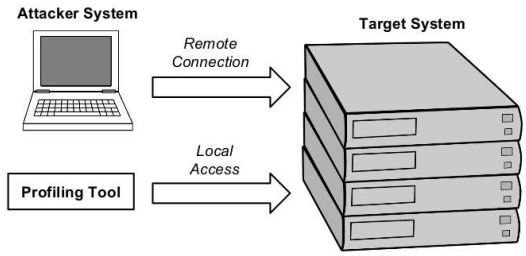

Figure 1. Experimental setup for profiling data collection

(e.g. by deleting log files) and the ones that did not take any such actions.

- **Success:** This feature describes weather the attacker was successful in carrying out the cyber-attack.

B. Classification of cyber-attacker features

A natural question arises when we think of the features we identified within the context of a cyber attack in progress: how would one monitor the computer system in question in order to identify these features? Motivated by this observation, we have categorized the cyber-attacker features in terms of observability. The term observability describes our ability to define the value of a feature based on measurements that have taken place on the machine being attacked. Table I illustrates this classification of cyber-attacker features.

C. Profiling Data Collection

Performing the actual measurement of an observable feature is not a straightforward task. There are many methods that could be adopted, depending on the software tools available and the architecture of the computer system being attacked. The approach we adopted was based on a software tool we developed, running on the computer system being attacked. The software is responsible for monitoring and recording the values of the observable features we have identified.

Figure 1 illustrates the experimental setup for the collection of profiling data. Our data consist of 87 participants being asked to conduct a cyber attack against a computer system. Each participant is given the IP address of the target machine and is asked to attack it by scanning for potential vulnerabilities and use the respective exploitation methods to gain access. When the attacker manages to connect to compromise the target machine, the profiling tool detects this event and starts recording the changes in the values of the observable features. These include the speed at which the attacker is issuing commands, the potential mistakes he makes, his attempts at covering his tracks by deleting log files and any actions that qualify as a successful attack. We should note that the latter depends on the computer system in question and can vary from accessing a specific folder to deleting user data. When configuring the profiling tool, we must define which actions qualify as a successful attack. Regarding the collection of non observable data, each participant was asked to fill in a questionnaire.

V. DESCRIPTION OF THE CYBER-PROFILING SYSTEM

For the needs of the data analysis and classification, machine learning was applied in the form of predictive modelling. Our target was to be able to identify underlying rules and data patterns, associate attributes and extract knowledge regarding the attacker's profile by identifying a prediction function for the target fields. As input variables were regarded the four observable features from the server side (Speed, Mistakes, Anti-forensics, Success). The latter were used as predictors for the six attributes of Education, Risk, Gender, Goal, Age and Skill in an attempt to illustrate the intruder's profile.

A. Balancing

Machine learning was imposed, following the CRISP framework [24] in terms of cleaning, cleansing, auditing and qualifying the data quality of the 87 user cases. In order to build a qualitative, scalable model and avoid data overfitting, any biased data were excluded from the provided dataset. This was achieved through balancing. For example, the dataset contained 84.93% male ("m") and 15.07% female cases ("f"). Balancing data with approximately equal numbers of "f" and "m" seemed to build a more accurate model and successfully discovered patterns within data. The algorithms seemed to work successfully with variations of for example 65% male versus 35% female users and not strictly between a 50-50% distribution.

B. Modelling

For the modelling part, data transformation was applied; data were balanced with sampling and transformation methods were used (e.g. dimension reduction) to minimise explicitly the chances of over fitting. The selected machine learning algorithms were justified based on their ability to identify and generate rules based on variable interassociations. Decision tree learning was selected as the most appropriate predictive approach to classify the attribute associations and project them via a visual representation. This assists significantly to both the indented rule generation as well as providing an unambiguous way of interpreting

the algorithmic results. For the needs of the classification four algorithms were selected and applied based on their characteristics. Those were the: C5.0, improved version of C4.5 [25], CHAID [26], Classification and regression trees (CART) [27] and QUEST [28]. Based on a preliminary performance analysis, it seemed that CHAID and QUEST were not sufficient for the modelling part. That was due to their inability to process the investigated variables (over-fitting), their stopping criteria were not met and they failed to recognise patterns on any provided data. This was probably due to the size and range of the provided data. However, C5.0 and CART seemed efficient in terms of generating decision trees and rule association with certain confidence.

C. Generated Rules and Results

For the needs of these experiments a 10-fold validation was used and its results were averaged to project predictions. In order to test and evaluate the proposed model the dataset was separated in train and test sets with 70-30% distribution. For balancing nodes the distribution was 75-25% for training-test sets respectively due to the lack of rare events and homogeneity of the sets. Both C5.0 and CART were applied on balanced and unbalanced datasets in a pursuit to depict any association rules among variables. In order to standardise the extracted rules each algorithm was applied 10 times on any investigated dataset and its most prevalent rules were recorded. These rules were validated by applying them 10 times to randomly selected (test) sets. Figure 2 shows the extracted rules from C5.0 for the target field education whereas Figure 3 shows the extracted decision tree from CART for the same attribute.

Both C5.0 and CART were used to extract any associated rules among predictors and the 6 target variables. Their efficiency varied, with C5.0 being most successful in terms of target attributes, reaching approximately an averaged 60% success over all attributes. CART performed slightly better, approximately 61%, excluding over-fitted age and goal attributes. Averaging predictions over all attributes, its results go beyond 50% which seems unsatisfactory for this experiment.

Figure 4 shows the overall classification performance for both C5.0 and CART.

As it can be seen C5.0 performs better compared to CART with performance range between approximately 55% and 80% for Education, Risk, Gender; 68% for Age and 58% for Skill. CART has equivalent performance for Education, Risk and Gender, however Goal and Age over-fits. Skill is predicted with 52% success which may be acceptable, still though less compared to C5.0.

VI. Cyber-Profiling Tool and Bot Detection

The software tool we used in Section IV-C for monitoring and collecting the observable cyber-attacker features can also be applied for the detection of a bot attack against

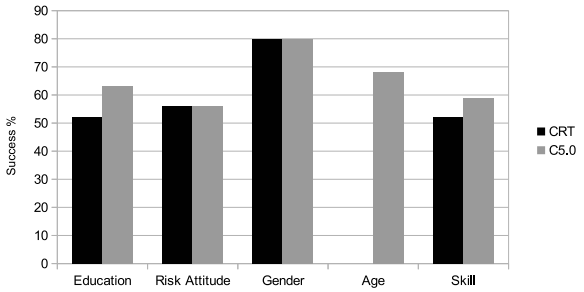

Figure 4. Comparison between C5.0 and CRT

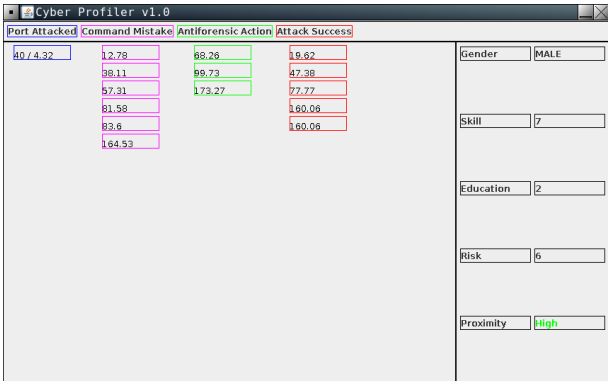

Figure 5. The interface of the cyber-profiling tool

a computer system. Before we explain how this can be achieved, let us give a detailed description of the tool's functionality.

At its current form the tool can distinguish among the four observable features we described in Section IV-B. Whenever a new attack event has been detected, the tool classifies it based on its type and also logs the time at which the event occurred. Figure 5 illustrates the graphical user interface of the tool. The left panel shows the attack events that have been detected, along with their respective timestamps. The right panel depicts the output of the Machine Learning algorithm that runs in the background and uses the observed features as input for predicting the attacker's profile characteristics.

In order to evaluate our tool with respect to botnet detection, we developed a bot prototype which replicates the initial infection phase of a botnet's life-cycle [23]. During this phase, the bot scans potential victim machines for known vulnerabilities and uses various exploitation methods to infect them. Our prototype reproduces this behaviour by targeting the victim machine. Using our cyber profiling software, we monitored the values of the observable features for the case of the bot attack. In Figure 6 we illustrate the observed values for speed and command mistakes for the case of human and bot attackers.

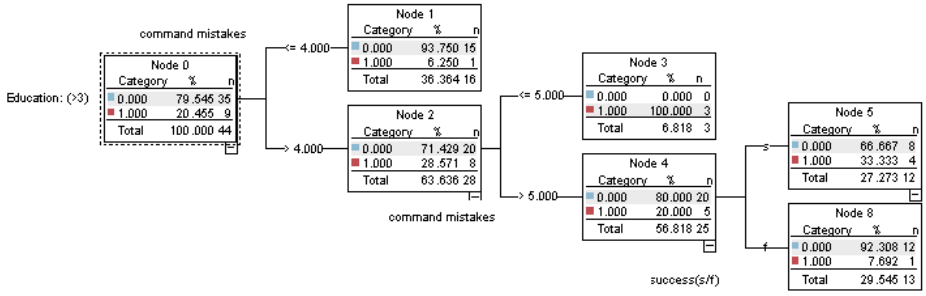

Figure 2. Extracted rules for Intruder's education using C5.0

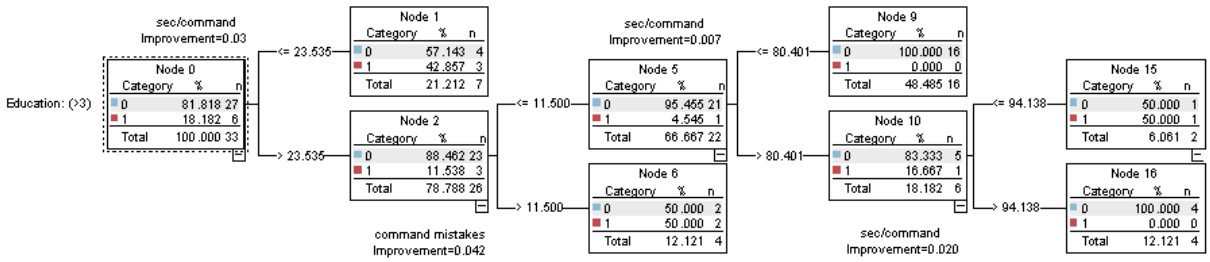

Figure 3. Extracted rules for Intruder's education using CART

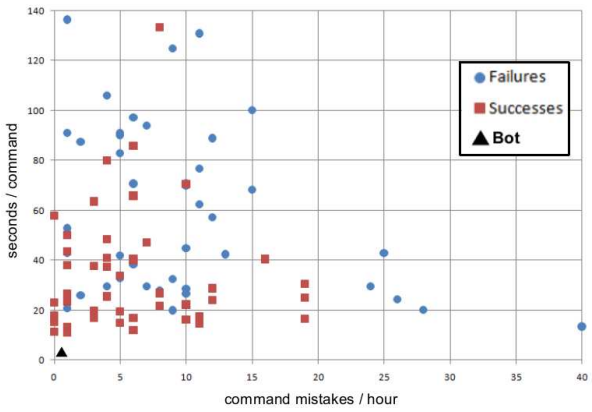

Figure 6. Bot detection using profile metrics

We can observe that the bot attack event clearly lies outside the cluster formed by the human attack events. The reason for this is twofold: firstly the speed at which a bot issues commands is far higher than that of a human attacker. Secondly, since a bot is essentially a computer software the number of mistaken commands is expected to be zero. On the contrary, a human attacker will most likely make mistakes when interacting with a computer system.

We should also note the two sub-clusters forming inside the cluster of human attacks. We can verify that the successful attacks are characterized by a high command issuing speed and a low number of command mistakes. This is a strong indication that our observable features can identify a more skillful attacker (i.e. fast and cautious) who is more successful in carrying out an attack.

VII. Conclusions and Future Work

In this paper we presented a system that aims to detect the characteristics of a human cyber adversary. We have identified several human profile features and have classified them based on their observability. We have used our previously collected experimental data in conjunction with a machine learning approach that classifies the attacker based on the values of the observed features. An automated profiling tool has also been developed. Our initial results indicate that by using the aforementioned features we are able to extract the characteristics of a human adversary and also give an indication of a bot attack.

In future work we will investigate the extraction of different cyber-profiling features and how it affects the performance of our system. Moreover, we will conduct further experiments with larger participant population and will also design alternative profiling algorithms to be used in the cyber-profiling tool.

References

[1] Krasser, S., Conti, G., Grizzard, J., Gribschaw, J., and Owen, H. Real-time and forensic network data analysis using animated and coordinated visualization. In Information Assur-

ance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC (pp. 42-49). IEEE, June 2005.

[2] G. Loukas and G. Öke. Protection against denial of service attacks: a survey. The Computer Journal, 53(7), pp. 1020-1037, 2010.

[3] G. Oke and G. Loukas. A denial of service detector based on maximum likelihood detection and the random neural network. The Computer Journal 50, no. 6, pp. 717-727, 2007

[4] G. Loukas, and G. Oke. Likelihood ratios and recurrent random neural networks in detection of denial of service attacks. In Proceedings of International Symposium of Computer and Telecommunication Systems, SPECTS (Vol. 7), 2007.

[5] G. Gu, J. Zhang, and W. Lee. Botsniffer: Detecting botnet command and control channels in network traffic, in Proceedinds of the 15th Annual Network and distributed System Security Symposium, 2008.

[6] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: Clustering analysis of network traffic for protocol- and structure independent botnet detection. In Proceedings of the 17th USENIX Security Symposium, 2008.

[7] N. F. A. Ramachandran and D. Dagon, Revealing botnet membership using DNSBL counter-intelligence, in Proceedings of 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet, 2006.

[8] B.J. Wood. An insider threat model for adversary simulation. Menlo Park, CA: SRI International, Cyber Defense

[9] G. Schudel and B. Wood. Modeling behavior of the cyber-terrorist, Washington, DC: Defense Advanced Research Projects Agency, Information Assurance Program, 2000.

[10] B. Landreth. Out of the Inner Circle: A Hacker's Guide to Computer Security, Microsoft Press, 1985.

[11] E.D. Shaw, J. Post and K. Ruby. Inside the mind of the insider. Security Management, pp. 34-44, December 1999.

[12] M. Kjaerland. A taxonomy and comparison of computer security incidents from the commercial and government sectors. Computers and Security, 25:522-538, 2006.

[13] E.D. Shaw and L. Fischer. Ten tales of betrayal: an analysis of attacks on corprorate infrastructure by information technology insiders, volume one. Monterrey, CA: Defense Personnel Security Research and Education Center, 2005.

[14] Alexander E. Voiskounsky and Olga V. Smyslova. Flow-based model of computer hacker's motivation. CyberPsychology & behavior 6.2, pp. 171-180, 2003.

[15] M. Kilger, O. Arkin and J. Stutzman. Profiling. The Honeynet Project (2nd Ed.), Know your enemy. Addison Wesley Professional, 2004.

[16] Raoul Chiesa, Stefania Ducci, and Silvio Ciappi. Profiling Hackers: The Science of Criminal Profiling as Applied to the World of Hacking. CRC Press, 2008.

[17] P. A. Watters, S. McCrombie, R. Layton, and J. Pieprzyk. Characterising and predicting cyber attacks using the Cyber Attacker Model Profile (CAMP), Journal of Money Laundering Control, Vol. 15 (4): 430-441, Emerald Group Publishing, 2012.

[18] S. Kapetanakis and M. Petridis. Evaluating a Case-Based Reasoning Architecture for the Intelligent Monitoring of Business Workflows. Successful Case-based Reasoning Applications-2. Springer Berlin Heidelberg, 2014. 43-54.

[19] T. Hastie, R. Tibshirani and J. Fridman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. The Mathematical Intelligencer, 27(2), pp. 83-85, 2009.

[20] H. Yu and D. Wang. Mass log data processing and mining based on Hadoop and cloud computing. In proceedings of the 7th International Conference on Computer Science & Education (ICCSE), pp. 197-202, 2012.

[21] E.D. Frangopoulos, M. M. Eloff, and L. M. Venter. Psychosocial risks: Can their effects on the security of information systems really be ignored?. Information Management & Computer Security 21.1 (2013): 53-65.

[22] S.M. Woskov, M.R. Grimaila, R.F. Mills, and M.W. Haas. Design considerations for a case-based reasoning engine for scenario-based cyber incident notification. In IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 84-91, IEEE, April 2011.

[23] M. Feily, A. Shahrestani and S. Ramadass. A survey of botnet and botnet detection. In Proceedings of the Third International Conference on Emerging Security Information, Systems and Technologies, pp. 268-273, IEEE, 2009.

[24] R. Wirth and J. Hipp. CRISP-DM: towards a standard process model for data mining. In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, pp. 29-39, 2000

[25] M. M. Mazid, A. Shawkat and K.S. Tickle. Improved C4.5 algorithm for rule based classification. In Proceedings of the 9th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases, pp. 296-301, 2010.

[26] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. Applied Statistics 29 (2), pp. 119-127, 1980.

[27] W. Y. Loh. Classification and regression tree methods. Encyclopedia of Statistics in Quality and Reliability, F. Ruggeri, R. Kenett, and F. W. Faltin (Eds.) Wiley, pp. 315-32, 2008.

[28] Y. Cheng and M. D. Shuster. Improvement to the Implementation of the QUEST Algorithm. Journal of Guidance, Control, and Dynamics, Vol. 37, No. 1, pp. 301-305, 2014