Intersection Features for Android Botnet Classification

Najiahtul Syafiqah Ismail, Robiah Yusof , Halizah Saad, Mohd Faizal Abdollah

Abstract*: The evolution of the Internet of things (IoT) has made a significant impact and availed opportunities for mobile device usage on human life. Many of IoT devices will be supposedly controlled through a mobile, giving application (apps) developers great opportunities in the development of new applications. However, hackers are continuously developing malicious applications especially Android botnet to steal private information, causing financial losses and breach user privacy. This paper proposed an enhancement approach for Android botnet classification based on features selection and classification algorithms. The proposed approach used requested permissions in the Android app and API function as features to differentiate between the Android botnet apps and benign apps. The Chi Square was used to select the most significant permissions, then the classification algorithms like Naïve Bayes and Decision Tree were used to classify the Android apps as botnet or benign apps. The results showed that Decision Tree with Chi-Square feature selection achieved the highest detection accuracy of 98.6% which was higher than other classifiers.

***Keywords* : Mobile Malware, Android Botnet; IoT, Malware Classification**

I. INTRODUCTION

Internet of Things (IoT) is a word expression used to indicate several appliances, low-level devices, and machines that have been connected to the Internet which allows for manageability and remote monitoring. In a manner of everything that is associated to the Internet, however, these devices are filled with countless of bad and good exposures, security threats, and software matters that expose them theoretically to the hackers. According to [1] the IoT visualizes the future where each individual or entity has a locatable, addressable, and readable counterpart on the Internet. Many current studies adapt traditional IoT protection mechanisms to overcome current threat attacks as attacks could involve various layers of device infrastructures. Supposedly, many of IoT devices will be controlled through a mobile application (apps) which risk them with threat attack that exposed them with potential privacy or intrusive behaviors. One of the threat attacks is applications running on mobile. According to [2], since user can get the mobile at affordable price with its high and current technology, Android operating system has turn out to be the most popular operating system in many countries. The consequences of Android operating system fame and practicable, mobile becomes an ideal target for cyber-criminal attacks. Mobiles have features of mobile phone and computer through call or text, e-mail, web browsing, GPS and online banking. Unlike computers, users always keep their mobiles with them and they rarely turn them off or disconnect them from internet access. Thus, users need to be alert for the kind of information that can be collected by various entities through their mobile.

The author [3] has stated that compared with other mobile operating systems (OS), Android OS is the most targeted platform by attackers. One of the attack methods is through malicious application (malware) installation. Mobile users are supposed to download and install Android application (app) from Google official applications market, Google Play. However, users can also download applications from the third party sources for these unofficial sources provide free or non-paying apps, drawing more users to the unofficial market and this action perhaps expose mobile users to download and install malicious applications. According to a report [1], usually attacker will use notable legitimate applications downloaded from Google Play and repackaged them with modified codes to create third party apps.

The compromised mobiles are at risk to threats such as credentials theft, stack based buffer overflow causing in arbitrary code execution, triggered and ran unknown services in the device without users' consent, service attacks denial and others. The attacks on Android mobile come in various forms such as viruses, trojans, worms and botnet. However, Android botnets are more dangerous as they pose serious threats. In current times, attacks and threats of Android botnets have been on the rise [2-5].

According to [20], Android botnet possess as a real threat for mobile compared to other Android malwares. The purpose of Android botnets likely the same to those of existing traditional botnets such as DoS, DDoS and spam distribution attacking [21]. Currently, only few research analyze Android botnet in terms of detection and implication. Hence, this paper filled the gaps. In brief, the major contributions of this paper are as follows:

- Applying Intersection Features method that can be used to effectively identify Android botnet stealth behavior.
- Proposing Intersection Features for Android botnet detection based on *Permission* and *API Call* using feature selection method that achieve over 98.6% accuracy

Revised Manuscript Received on November 15, 2019

Najiahtul Syafiqah Ismail*, Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM), Melaka, Malaysia.

Robiah Yusof, Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM), Melaka, Malaysia.

Mohd Faizal Abdollah, Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM), Melaka, Malaysia.

Halizah Saad, Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM), Melaka, Malaysia.

of detection.

This paper was to generate a new Android botnet classification based on *Permission* and *API Calls* features because not a lot work has been done in this field. The rest of this paper is arranged as follows: Section 2 describes the related work. Section 3 discusses the methodology used, Section 4 reviews the experimental results, and Section 5 presents the conclusion and future research.

II. LITERATURE REVIEW

Android botnet is a link of mobile devices that is designed and remotely controlled by an infected malware without user approval [5]. Furthermore, Android botnet was controlled by a BotMaster through a C&C network [6] However, Android botnet also can utilize other mobile technologies such Bluetooth, SMS and email as a C&C server. Android botnet work through a cellular network as a medium of communication. Attack happen when an attacker act out as a BotMaster and establishes communication with infected devices using a command and control (C&C) method [7].

Based on the Android botnet history, Eurograbber is one of the most notable and sophisticated attacks from Zeus malware family which occurred in 2012 where it infected more than 30,000 users and stole an estimated 36 million Euros [8-9]. The attack focused on Blackberry, Symbian, and Windows users. The first attack involving Android botnet was discovered in 2010 where the botnet used SMS as their C&C mechanism, aiming to obtain the victim's location [10]. 2011 saw the rise of Android botnet where many cases exposed such as Plankton, PjApps, DroidKungFu, PjApps and GoldDream. There are conveyed message with C&C servers using the HTTP protocol. Most of their motivation were to propagate possible malware, spy or access financial gain. Based on a report [11], the latest incident involving Android Botnet is WireX which occurred in August 2017 where the botnet includes mainly Android devices running malicious applications and is designed to create a denial-of-service (DDoS) attack. This attack causes thousands of users from 100 countries affected.

There are seven types of potential Android Botnet attacks like email interaction, short messaging/multimedia messaging system interaction, victim selection, spyware, privacy matters, mobile voting system and charity services [5].

III. METHODOLOGY

Boosting the performance of features used in Android botnet detection is a topic of recent interest as many previous detections have been unsuccessful in recognizing unknown attack patterns and producing high numbers of false alarms instead. Besides, it is an essential issue when selected feature sets are later analysed by domain experts to gain more insights into the modelled problem. Most Android botnet detections nowadays use one feature input which is inadequate because botnet can easily evade the detection. As a Result, this paper proposed a which apply two different feature input sets including *Permissions* and *API Call* functions. It defined a new theory known as *Intersection Feature*.

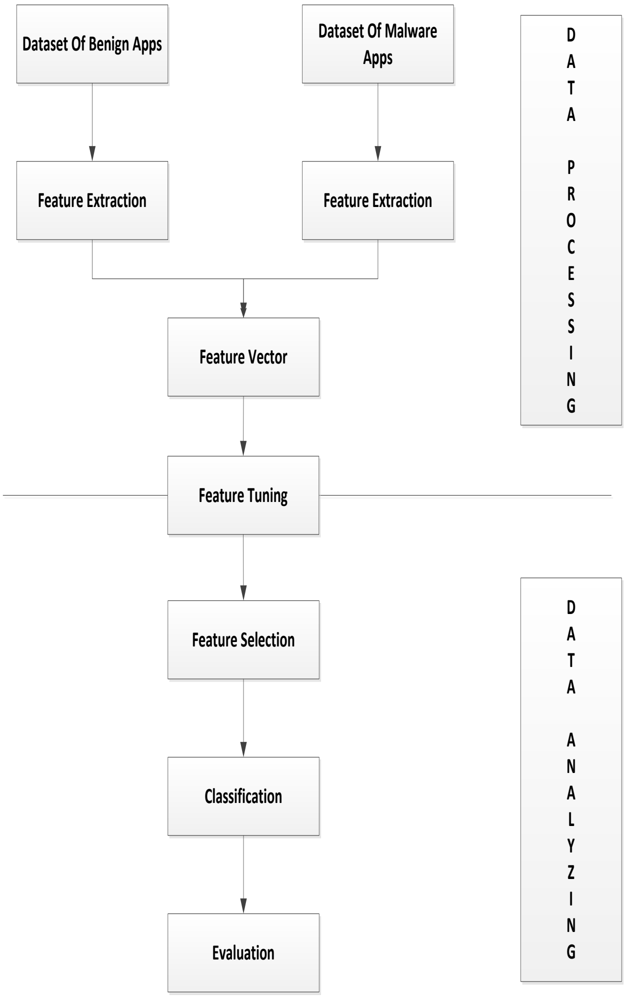

Fig. 1. Components of data processing and analyzing

This paper is part of an ongoing research, which is to improve the android mobile malware detection system. Based on Fig. 1, the whole system involved many components. However, for the purpose of this paper, only data processing and data analysing components were emphasized. The data processing is purposeful to alter raw data into an appropriate format, later to be used in the analysis module. The steps involved in this components included data collection, feature extraction and feature vector. Data analysis module in the other hand involved feature selection and classification. The feature tuning process occur between feature vector and feature selection process.

An *Intersection Feature* is created when there are intersection between two or more features from separate feature sets which undergo the same process. For instance, two features including *sendTextMessage* API function, and *SEND_SMS Permission* undergo the same process, intersecting with each other, thus, an *Intersection Feature* comprising these two features is created. Features in the *Intersection Feature* can improve the detection system. Table I shows the concept of Intersection Feature.

Table I. Intersection Feature concept

#	API	Permission
1	*getUriForDownloadedFile*	*INTERNET*
2	*restartPackage*	*KILL_BACKGROUND_PROCESSES*
3	*getLongitude*	*ACCESS_COARSE_LOCATION*
4	*sendTextMessage*	*SEND_SMS*

Features include in the same Intersection Features might change the feature value of each other. For instance, botnet can evade the *Permission-based* detection mechanism, by did not request *SEND_SMS Permission*. Therefore, the value of this permission in the feature vector of malware was 0. Still, if malware called *sendTextMessage API Call* function the value of *SEND_SMS Permission* was set as 1. The idea is illustrated in Fig. 2.

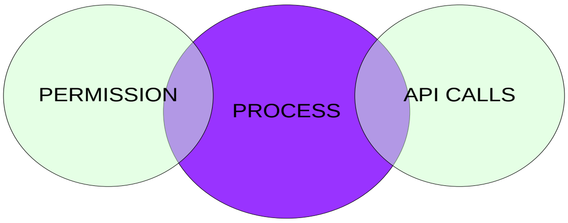

Fig. 2. Illustrated of feature intersects.

Fig. 2 shows the illustrated idea of *Intersection Feature* approach. It shows two feature sets; *Permission* and *API Call* intersects.

The summary of the work flow for *Intersection Feature* approach is shown in Fig. 3.

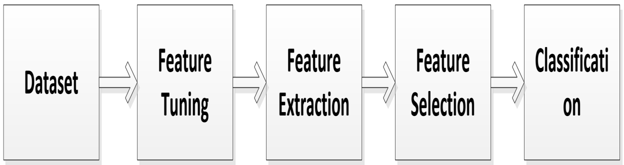

Fig. 3. Work flow of *Intersection Feature* approach

Fig. 3 shows the illustrated idea of *Intersection Feature* approach. It shows that two feature sets intersected; *Permission* and *API Call*. The proposed approach was conducted in five phases; *dataset*, *feature tuning*, *feature extraction*, *feature selection*, and *benign/malware classification.*

A.DATASETS

Two datasets were used for this study which are training and testing dataset. Training dataset was used to construct a detection model, while a testing dataset was to validate the model. The training dataset for this research was taken from University of New Brunswick [12]. The dataset consisted of 1929 Android botnet samples in 14 different Android botnet families. These samples covered the majority of existing Android botnets from 2010 (the first appearance of Android botnet) to 2014. For the purpose of this study, 1505 Android botnet samples were randomly selected and analyzed. The testing dataset was a benign application downloaded from Google Play, an official market that hosted Android application. A total of 850 benign apps were downloaded and used for this study. The highly crucial architectures for Android botnet were the use of command and control (C&C) channel and the network controlled by the botmasters through http, SMS & email. Five samples were taken from each of the ten malware families that are categorized into botnet types; AnserverBot, Bmaster, DroidDream, Geinimi, MisoSMS, NickySpy, NotCompatible, PjApps, Tigerbot, and Zitmo. All the Botnet apps were categorized accordingly based on their own categories such as Games, Music, Entertainment and others at the preliminary stage.

- Permissions: Application activity started immediately after permissions requested by it. Android architecture provided a well framed permission mechanism to provide security.
- API calls: The application programming interface calls were invoked at the execution time to perform some specific tasks.

B.FEATURE EXTRACTION

Permissions and API Call features extracted in this data processing component were produced by 348 Android applications (botnet and benign). The extracted Permissions by each of 348 Android applications and API Call were compared with standard Permission and API Call release by Android system using string similarity method. If extracted Permission and API Call matched with the feature list that extracted features permission was noted as 1 to show its existence in the sample while 0 showed the nonexistence of the permission.

Let R be a vector containing set of 400 of Android features including features from Permission and API Call. For every ith application in the Android application dataset (botnet and benign),

Ri = {r1, r2, r3,…rj} and

$$r_j = \begin{cases} 1, & \textit{If feature } j_{th} \textit{ exist} \\ 0, & \textit{otherwise} \end{cases} \tag{1}$$

The result of this phase finding was used in feature vector phase.

C.FEATURE TUNING

The feature tuning purpose was to enhance the quality of features. This phase consisted

of two steps namely removing extra features where some of rarely used features were removed, and modifying feature values where the values of features within the same Intersection feature were modified. The feature tuning process was also based on the Android Botnet characteristic and behavior.

D. FEATURE VECTOR

In Feature Vector stage, result from Feature Extraction stage was transformed into a vector in comma separated value (CSV) file for each of 348 Android applications. The vector ended with value 1 or 0 to indicate the existence of either botnet or benign. Fig. 4 shows examples of permission vectors.

0, 1, 1, 0, 0, 0, 1, …….. 1, Botnet

Fig. 4. Examples of permission vector

E. FEATURE SELECTION

The feature selection stage selected a subset of relevant features. Feature selection was an es-sential concern in classification, as it might have a significant effect on accuracy of the classi-fier [13]. It reduced the sizes of the dataset, thus, the processor and memory usage decrease; the data become more coherent and easier to study on. Gain Ratio, Chi-square Information Gain are the most positively effective feature selection algorithms [14]. Besides, Information Gain gives stable performance for different number of features and Gain Ratio gives the best results for large number of sentimental features selection [15] whereas [16] stated that Chi Square provides more accuracy with the least number of features. Hence, the feature selec-tion algorithm used in the proposed method was Chi Square.

F. BENIGN MALWARE CLASSIFICATION

The classification stage used various machine learning techniques to classify applications. Decision Tree and Naïve Bayes were used in this experiment. In order to find the impact of feature tuning and feature selection, these algorithms were applied twice: before and after the feature tuning stage. The results indicated the improvement of achieved metrics.

IV. EXPERIMENTAL RESULT

The classification stage used various machine learning techniques to classify applications. Decision Tree and Naïve Bayes algorithm were used in this experiment. In order to find the impact of feature tuning and feature selection, these algorithms were applied twice: before and after the feature tuning stage. The results indicated the improvement of achieved metrics.

In this section, the evaluation metrics and the results are discussed.

A. EVALUATION METRICS

In order to evaluate the detection method, several evaluation metrics were used.

- True Positive (TP): the number of malicious applications which are correctly classified as positive.
- False Positive (FP): the number of benign applications which are incorrectly classified as positive.
- False Negative (FN): the number of malicious applications which are incorrectly classified as negative.
- True Negative (TN): the number of benign applications which are correctly classified as negative.

Additional metrics are defined based on TP, FP, TN and FN.

$$\text{Precision}=\frac{TP}{TP+FP}$$

$$\text{Accuracy}=\frac{TP+TN}{TN+FN+TP-FP}$$

$$\text{Recall}=\frac{TP}{FN+TP} \qquad (3)$$

B. EVALUATION RESULTS

In this section, the results of the implementation of the proposed approach are presented. In order to show the improvement achieved through using *Permission* and *API Call* functions, two feature vectors have been constructed. These feature vectors are as follows:

- Feature Vector 1 (FV1): It contains permissions only.
- Feature Vector 2 (FV2): It contains permissions and API functions.

The classification algorithms including Naïve Bayes and Decision Tree were applied on the constructed dataset. In order to find the impact of feature refinement and feature selection, these classification algorithms were used twice: before feature refinement and selection and after these phases. The results showed the improvement of achieved accuracy after feature refinement and feature selection.Table II shows the accuracy achieved by running 2 difference classification methods before and after applying feature tuning on Feature Vector 1.

Table II. Classification result of Feature Vector 1

	PERMISSION (FV1)			
	Before Feature Tuning		After Feature Tuning	
	Naïve Bayes	Decision tree	Naïve Bayes	Decision tree
Accuracy	0.955	0.936	0.97	0.95
Precision	0.927	0.891	0.945	0.909
Recall	0.981	0.98	0.995	0.99

Although Feature Vector 1 produced high accuracy scores for both classification techniques, the accuracy increased about 0.15 for both classification technique after applying the tuning process. Table III shows the accuracy achieved by running 2 difference classification methods before and after applying feature tuning on Feature Vector 2.

Table III. Classification result of Feature Vector 2

	PERMISSION + API CALL (FV2)			
	Before Feature Tuning		After Feature Tuning	
	Naïve Bayes	Decision tree	Naïve Bayes	Decision tree
Accuracy	0.97	0.95	0.986	0.955
Precision	0.945	0.91	0.977	0.92
Recall	0.994	0.984	0.995	0.99

Feature Vector 2 produced high accuracy scores for both classification techniques. Besides, comparison of results with results from Table II showed that the improvement was achieved by adding new features from API Call sets.

Table IV. Result Comparisons Of Proposed Method With Previous Work.

The Comparison of The Proposed Method And Related Works			
Malware Detection	Precision	Recall	Accuracy
A two-layered permission-based android malware detection scheme [16]	0.898	0.550	0.986
DroidMat : Android Malware Detection through Manifest and API Calls Tracing [17]	0.967	0.873	0.978
Merging Permission and API Features for Android Malware Detection [18]	0.943	0.96	0.953
Proposed method	**0.977**	**0.995**	**0.986**

As shown in this Table IV, the proposed method with tuning process result with high accuracy results compared with other related works.

V. CONCLUSION

Android botnet are real threats for mobiles as they possess more serious threats compared with other Android malwares. In earlier works, there are many researchers implement Android malware detection systems without considering various scenarios of different feature sets. Besides, most detection systems only use general characteristics of Android malware without considering the special characteristics of Android botnets. As a result, most proposed methods are simply evaded by Android botnet. In this paper, Intersection Features approach is proposed by considering different features such as Permissions and API Call. The proposed approach modifies the values of some intersecting features. Therefore, the achieved precision increases to 97.7% and the achieved accuracy improves to 98.6%. For future research, features from dynamic analysis such as system call should be utilized to achieve a better detection accuracy.

REFERENCES

1. Symantec Corporation, “Internet Security Threat Report,” Symantec 2013 Trends, vol. 19, no. April, p. 97, 2014.
2. H. Pieterse and M. S. Olivier, ‘Android Botnets on the Rise : Trends and Characteristics’, in Information Security for South Africa, 2012, pp. 1–5.
3. S. Anwar, M. F. Zolkipli, Z. Inayat, J. Odili, M. Ali, and J. M. Zain, “Android botnets: A serious threat to android devices,” Pertanika J. Sci. Technol., vol. 26, no. 1, pp. 37–70, 2018.
4. W. Hijawi, J. Alqatawna, and H. Faris, “Toward a Detection Framework for Android Botnet,” 2017 Int. Conf. New Trends Comput. Sci., no. October, pp. 197–202, 2017.
5. A. Karim, S. A. A. Shah, R. Bin Salleh, M. Arif, R. Md Noor, and S. Shamshirband, “Mobile botnet attacks ??? an emerging threat: Classification, review and open issues,” KSII Trans. Internet Inf. Syst., vol. 9, no. 4, pp. 1471–1492, 2015.
6. G. Geng, G. Xu, M. Zhang, Y. Yang, and G. Yang, “An improved SMS based heterogeneous mobile botnet model,” 2011 IEEE Int. Conf. Inf. Autom. ICIA 2011, no. June, pp. 198–202, 2011.
7. M. Yusof, M. M. Saudi, and F. Ridzuan, “A new mobile botnet classification based on permission and API calls,” Proc. - 2017 7th Int. Conf. Emerg. Secur. Technol. EST 2017, pp. 122–127, 2017.
8. N. Etaher, G. R. S. Weir, and M. Alazab, “From ZeuS to zitmo: Trends in banking malware,” Proc. - 14th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2015, vol. 1, pp. 1386–1391, 2015.
9. F. Tchakounte, “A Malware Detection System for Android,” Universitat Bremen, 2015.
10. R. Nigam, ‘A Timeline of Mobile Botnets’, in The Botnet Fighting Conference 3rd Edition BOTCONF, 2014, pp. 1–23.
11. Cochran J, The WireX Botnet: How Industry Collaboration Disrupted a DDoS Attack. In: Cloudflare Blog. https://blog.cloudflare.com/the-wirex-botnet/. Accessed 25 Mar 2018
12. A. F. Abdul Kadir, N. Stakhanova, and A. A. Ghorbani, “Android Botnets: What URLs are telling us,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9408, pp. 78–91, 2015.
13. E. M. Karabulut, S. A. Özel, and T. İbrikçi, “A comparative study on the effect of feature selection on classification accuracy,” Procedia Technol., vol. 1, pp. 323–327, 2012.
14. A. Sharma and S. Dey, “Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis,” Int. J. Comput. Appl., no. June, pp. 15–20, 2012.
15. R. Kaur and M. Sachdeva, ‘Study and Comparison of Feature Selection Approaches for Intrusion Detection’, in Proceedings on International Conference on Advances in Emerging Technology, 2016, vol. 2, pp. 1–7.
16. X. Liu and J. Liu, “A two-layered permission-based android malware detection scheme,” Proc. - 2nd IEEE Int. Conf. Mob. Cloud Comput. Serv. Eng. MobileCloud 2014, pp. 142–148, 2014.
17. D. Wu, C. Mao, T. Wei, H. Lee, and K. Wu, “DroidMat : Android Malware Detection through Manifest and API Calls Tracing,” Proc. 7th Asia Jt. Conf. Inf. Secur. (Asia JCIS 2012), pp. 62–69, 2012.
18. M. Qiao, A. H. Sung, and Q. Liu, “Merging Permission and API Features for Android Malware Detection,” 2016 5th IIAI Int. Congr. Adv. Appl. Informatics, pp. 566–571, 2016.
19. J. van der Geer, J.A.J. Hanraads, R.A. Lupton, “The art of writing a scientific article,” Journal of Science Communication, Vol. 16, Issue 3, pp. 51-59, 2000.
20. Z. Abdullah, M. M. Saudi, and N. B. Anuar, ‘ABC: Android botnet classification using feature selection and classification algorithms’, Adv. Sci. Lett., vol. 23, no. 5, pp. 4417–4420, 2017.
21. S. Anwar, M. F. Zolkipli, Z. Inayat, J. Odili, M. Ali, and J. M. Zain, Android botnets: A serious threat to android devices’, Pertanika J. Sci. Technol., vol. 26, no. 1, pp. 37–70, 2018.

AUTHORS PROFILE

Najiahtul Syafiqah Ismail, currently pursuing her PhD degree in the Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM). She received a BSc (Hons) in Computer Science and the MSc in Information Technology from University Technical Malaysia Melaka (UTeM). Her research interests include computer networking, computer security and mobile security.

Robiah Yusof, currently a Senior Lecturer in the Universiti Teknikal Malaysia Melaka (UTeM). She received the BSc (Hons) of Computer Studies and Master of Information Technology from Liverpool John Moore’s University, UK and Universiti Kebangsaan Malaysia, respectively. She obtained the Doctor of Philosophy, Network Security from Universiti Teknikal Malaysia Melaka (UTeM). Her research interests include network security, computer system security, and network design.

Mohd Faizal Abdollah currently working as a Senior Lecturer in Department of Computer and Communication System, Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM). He received his first degree and Master degree from University Utara Malaysia and University Kebangsaan Malaysia. He obtained his PhD from University Technical Malaysia Melaka in Computer and Network Security. Previously, he worked as a MIS Executive at EON Berhad, Selangor and as a System Engineer in Multimedia University, Melaka for six years. His interest is mainly in network and wireless technology & network and wireless security.

Halizah Saad, currently pursuing her PhD degree in the Faculty of Information and Communication Technology, University Technical Malaysia Melaka (UTeM). She received a BSc (Hons) in Computer Science and the MSc in Information Technology from University Technical Malaysia Melaka (UTeM). Her research interests include computer networking, computer security and mobile security.