SMS-Based Mobile Botnet Detection Framework Using Intelligent Agents

Abdullah J. Alzahrani[1] and Ali A. Ghorbani[2]

[1]*Assistant Professor at The College of Computer Science and Engineering (CCSE), University of Hail (UOH), Saudi Arabia*
[2]*Professor and Dean, Director, Canadian Institute for Cybersecurity, Canada Research Chair in Cybersecurity, Faculty of Computer Science, University of New Brunswick, Canada*
E-mail: aj.alzahrani@uoh.edu.sa; ghorbani@unb.ca

Received 2 July 2016; Accepted 9 January 2017;
Publication 27 January 2017

Abstract

Along with increasing security measures in Android platforms, the amount of Android malware that use remote exploits has grown significantly. Using mobile botnets, attackers concentrate on reliable attack vectors such as SMS messages. Short Message Service (SMS) has been increasingly targeted by a number of malicious applications ("apps") that have the ability to abuse SMS features in order to send spam, to transfer command and control (C&C) instructions, to distribute malicious applications via URLs embedded in text messages, to send text messages to premium-rate numbers, and to exploit smartphones.

In this paper, we propose an SMS-based botnet detection formwork that uses multi-agent technology based on observations of SMS and Android smartphone features. This formwork has the ability to detect SMS botnets and identify ways to block the attacks in order to prevent damage caused by botnet attacks. We developed an adaptive hybrid model of SMS botnet detectors by using a combination of signature-based and anomaly-based algorithms. These components utilize multi-agent technology to recognize malicious SMS and prevent users from opening these messages that infecting smartphones.

Journal of Cyber Security, Vol. 5_2, 47–74.
doi: 10.13052/jcsm2245-1439.523

This framework includes defence module that employed a more proactive approach that allows us to directly generate signatures and rules that can be used to protect Android smartphones from abuse by SMS botnets. The framework creates a user profile that is used to perform behavioural profiling analysis in order to identity malicious SMS and cut the C&C Channel.

Keywords: SMS, mobile botnet, intrusion detection, Android malware, multi-agent system.

1 Introduction

One of the most serious threats to Internet security is the proliferation of botnets. Recently, there has been a dramatic rise in the use of botnets. The etymological concept of botnet comes from the term "bot", which means that victims are controlled by an attacker. The attacker, also known as the bot master, has the capability of controlling large-scale networks of bots from various locations in order to carry out attacks. The characteristics of C&C channels have evolved from IRC-based to HTTP-based, FTP-based, DNS-based, Twitter-based, and SMS-based, and from the centralized structure to P2P and Fast Flux Network Services [15]. However, mobile botnets have become serious issue because of the increasing worldwide trend in the use of mobile devices. A mobile botnet is a network of compromised smartphones that share the same command and control (C&C) infrastructure, controlled by a bot master to perform a variety of malicious attacks [11].

One of the main components of a botnet is the C&C channel, which is used by attackers to carry out C&C communication. With the availability of SMS on smartphones, SMS messages are used to transfer C&C commands, send SMS spam, send premium-rate SMS messages without user knowledge [18] and distribute malware as propagation vectors. SMS based C&C presents some advantages for the bot master. The first advantage is that the attacker can communicate with the root node, given that communication utilizes a tree topology. The second advantage is that it makes detection of bot communications very difficult. On the other hand, SMS based C&C has disadvantages that pose challenges for bot masters. Initially, the bot master must assess whether the tree is still intact, and that there are no breaks in the tree or missing nodes that might compromise the communication process [16].

In this paper, we propose an SMS-based botnet detection framework using intelligent agents that are used to detect malicious SMS messages and monitor smartphone resources which are typically targeted by SMS botnet attacks. The framework is based on a multi-layer model which consists of three modules

and JADE agents namely, an SMS signature-based detection module, an anomaly-based detection module, and a defence module. An SMS signature-based detection module can be used to combat SMS botnets by applying pattern-matching detection approaches, and using rule-based techniques. An anomaly-based detection module employs unsupervised learning techniques to group SMS messages into four class labels and to classify reported text messages to one of those four classes. The module also uses profiling analysis to detect whether there are any correlations between classification results and alerts from profiling analysis and label SMS messages as either normal or malicious. A defence module can be used as a more proactive approach which directly generates signatures and rules in order to protect Android smartphones from abuse by SMS botnets. A multi-agent system that can be used to observe Android mobile devices and to interact with service provider agents in order to detect malicious applications and SMS botnet activities on Android mobile devices. We have developed an intelligent and proactive framework that scans incoming and outgoing text messages, monitors Android resources and observes user usage that includes user connectivity time. The framework creates a user profile that is used to perform behavioural profiling analysis in order to identity malicious SMS and cut the C&C Channel.

2 Related Works

Considering the advances and evolution of mobile botnets, botnets have become an effective malware-launching platform through which a new "worm" or virus may be sent out instantaneously by numerous bots. These malicious applications exfiltrate SMS messages in a unique way. Some botnets forward the content of the user SMS messages to unintended recipients, using specific phone numbers that are provided by the attacker. Other SMS botnet activities include sending SMS spam, transferring C&C instructions, launching denial-of-service (DoS) attacks to send premium-rate SMS messages without user permission, and propagating malware via URLs sent within SMS messages.

In terms of combating the use of SMS as a propagation vector, Hua et al. [12] have designed and developed security managers based on mobile characteristics; this is a host-based solution. Another solution proposed by Kok et al. [13] is use of anti-botnets that consist of four different modules: analysis, detection, mitigation and prevention. However, they only describe their model in general terms without giving any details on how to spot and respond to mobile botnets. In contrast, Vural et al. [19] discuss the potential threat of botnets based on mobile networks by differentiating between human

and bot activities with respect to the delay, volume, and median of weekly outgoing and incoming text messages. They propose an anomaly detection approach that uses computational intelligence techniques to detect botnets.

The use of a multi-agent approach in smartphone security with a focus on the Android platform is a new area of research. Several multi-agent platforms have been developed for smartphones. Agüero et al. [2] present the Agent Platform Independent Model (APIM) for mobile devices. Their proposed approach has been implemented and tested on the Android platform with some limitations. Alam et al. [3] present a context-aware multi-agent based framework for securing Android devices by collecting Android data which is then analyzed in the central server. Cheng [8] proposes a multi-agent security system for the Android platform that uses agents to collect data from Android devices, and then send it to agent service providers to make a decision. The study shows that multi-agent systems can be adapted to the Android platform with its inherent resource limitations.

3 Proposed Framework

The architecture of the proposed detection system consists of four different components: a multi-agent system, a SMS signature-based detection, an anomaly-based detection module, and a defence module. These components are divided into two tiers: Android mobile devices and a service provider. Figure 1 shows the proposed model design that will serve as a comprehensive SMS botnet-detection mechanism.

3.1 Multi-Agent System

The framework incorporates two components: Android mobile devices and a server that offers services. A multi-agent system has different agents with related responsibilities and goals to achieve. Figure 1 shows our complete framework design that functions as a comprehensive SMS botnet detection mechanism, and illustrates the interaction between agents and other modules.

Our proposed framework requires a multi-agent system with extensive knowledge about distributed systems and required agent interactions in order to observe, monitor, and handle the data exchange. One of the most well-known multi-agent system frameworks is the JADE platform [5].

The proposed framework includes the following task functions:

1. An Android user can install the SMS botnet detection application in order to protect his smartphone against SMS botnets.

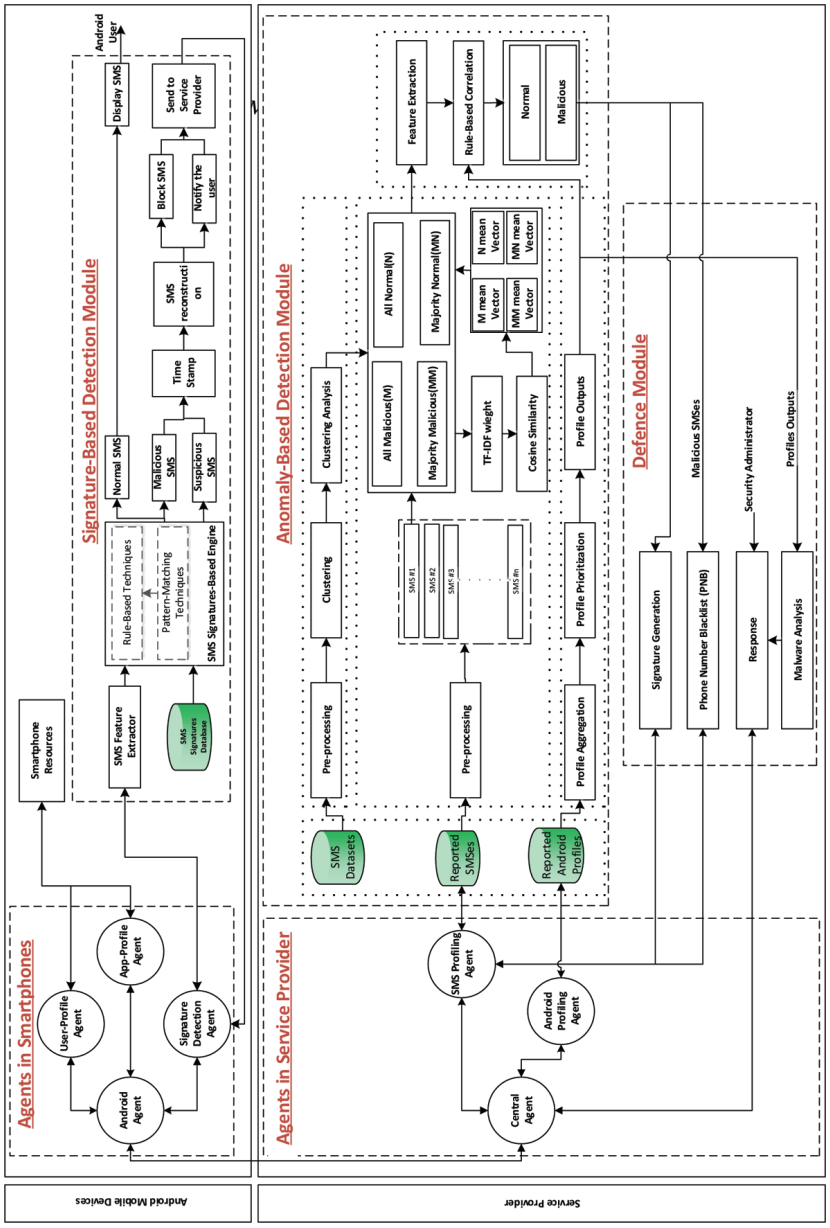

Figure 1 Overview of the proposed SMS botnet detection framework.

2. An Android user must register with a service provider that can provide all the services and keep a list of all registered devices.
3. The service provider is responsible for maintaining the list of Android devices and offering the services to the subscribed Android devices.
4. An Android mobile device must check its phone status, including Internet connection, battery level, and network status, and report them to the service provider.
5. An Android device runs a signature detection algorithm to check the incoming and outgoing text messages, and then reports the suspicious and malicious SMS messages to the service provider.
6. The service provider must frequently update the signatures and send the update to the subscribed Android device in order to detect malicious SMS messages.
7. An Android device updates its current profile and sends information to the service provider with the user's permission. The service provider keeps updated the user profiles for each Android device in order to perform further analysis.
8. The service provider has a detection module that will perform anomaly detection and profile analysis on the reported data. It also finds any correlations between the data, and accordingly makes recommendations to the Android device. The recommendations include but are not limited to updating the signature, maintaining the phone number blacklist, submitting the Android profile, removing a certain application, and reporting back information about a specific application. It is up to the mobile device user to take action based on the recommendations.

The Android profiling approach is an agent-based implementation of the SMS botnet detection framework. Each module of the SMS botnet detection framework is implemented in order to interact with agents using the JAVA/JADE [5] development environment. The architecture of the proposed multi-agent system is shown in Figure 1. In the service provider, there are three types of agents: the central, SMS profiling and Android profiling agents. For each Android smartphone, there are four different types of agent: the Android, signature detection, app-profile, and user-profile agents.

3.1.1 Service provider agent responsibilities

As shown in Figure 1, the service provider has three agents and two modules that are used to process the data, in order to detect SMS botnets and malicious SMS, to make intelligent decisions, and to perform actions. The three major

agents that perform the majority of the activities of the detection system are the central agent, the Android profiling agent and the SMS profiling agent, as outlined below. Based on the results of profiling analysis, these agents provide service and offer further analysis to achieve a high detection rate and make intelligent decisions, in order to detect SMS botnet activities.

An Android mobile device must subscribe to the *central agent* in order to obtain all the defined services from the SMS profiling and android profiling agents. The central agent registers the Android agents and informs the service provider agents about each new subscription request, so that they may start offering services to all agents in the Android smartphone. The central agent also informs the Android agent that these agents have been added, so they can begin to protect Android smartphones. The central agent sends commands to the Android agents, detailing the decision that has been obtained and established in the defence module. It manages all the local agents situated in the service provider. The central agent also performs activities that are relevant to Android mobile device agents, such as managing, updating, blocking, deleting and controlling. The *SMS profiling agent* handles reported SMS messages that are considered malicious or suspicious, and maintains SMS logs. These received SMS data and logs are forwarded to the detection module to verify whether they are deemed to be botnet activities. This agent also obtains the generated signatures and rules that are reported by the defence module, which need to be forwarded to the signature detection agents. Once the profile updates are received by the *android profiling agent* in the service provider, this particular agent will then maintain and update the profile for all subscribed smartphones. Additionally, this agent updates the received changes from the detection module. It responds to detection module requests, which are findings and actions that need to be acted upon.

3.1.2 Android smartphone agents responsibilities

An Android mobile user must subscribe to the central agent in order to obtain all the defined services and to maintain the interaction between local agents. The agents monitor incoming and outgoing SMS messages and send them to the SMS signature detection module. They also observe smartphones behaviour and resources. The function of SMS signature detection module will be described extensively in the next section. There are four major agents within Android mobile devices, namely, an Android agent, an SMS signature detection agent, an app-profile agent, and a user-profile agent.

The *Android agent* establishes a connection with the central agent. It plays a critical role in the system, since it creates a channel that allows

communication from the actual phone to the service provider. It also manages and supervises the interactions between local agents, and sends requests to the app-profile and user-profile agents when malicious or suspicious SMS is detected. The *signature detection agent* monitors incoming and outgoing SMS messages. This agent obtains updates of the signatures and rules from the SMS profiling agents. It forwards all incoming and outgoing SMS messages to the SMS Signature module, which performs the detection and sends the results back to the agent. If the SMS message is labeled as normal, the agent will deliver it to the SMS default app, and if the SMS message is malicious or suspicious, the agent will send a command to the Android agent, which can then request the current profile update to be sent by the app-profile and user-profile agents. All local agents will forward the current profiles to Android profiling agent, while simultaneously sending reported SMS messages to the SMS profiling agent. The *app-profile agent* is responsible for creating app profiles by observing installed applications that include the granted permissions, and apps that access the browser or try to communicate and broadcast with other installed applications. This agent responds to the requests from the Android agent, and forwards the current profiles to the Android profiling agent. The *user-profile agent* builds a profile by monitoring user connectivity time, maintains the phone number blacklist, and reports daily usage of the mobile phone. This agent responds to the requests from the Android agent and forwards the current profiles to Android profiling agent.

3.2 SMS Signature-Based Detection

Focusing on incoming and outgoing SMS messages, the proposed design for Android mobile devices uses a signature-based detection algorithm to identify SMS botnets. As illustrated in Figure 1, SMS signatures are obtained and copied from an SMS signature database where signatures of known botnets and malware are stored.

We employ a content-based mechanism using a signature-based approach. Signatures are patterns or sets of rules that can uniquely identify an attack. Traditionally, the signature-based approach extracts the features from traffic and detects malicious activity by comparing incoming traffic to the signatures of attacks. The main disadvantage of signature detection is its inability to detect an unknown attack, for example an attack that has not been seen before, or an attack that does not have a corresponding signature. In order to address this issue, our approach labels an unknown attack as either suspicious or normal using rule-based techniques. We use a real-time content-based

signature detection to differentiate between normal, suspicious and malicious SMS based on the content of the SMS, and then display the result of the suspicious SMS to the Android user, who may then choose whether to remove the SMS or not.

3.2.1 Signature-based detection engine

As shown in Figure 1, the first step to effectively spot malicious SMS is to extract SMS features that have the potential to distinguish the behaviour of SMS text messages. The names and descriptions of these features are given in Table 1. In order to develop an effective signature-based detection approach to combat malicious malware, we extract sender phone numbers and SMS content from our dataset; then, from the SMS content, we extract embedded URLs, commands, phone numbers, and phishing words as signatures for our approach. To process the SMS message, we have implemented an SMS feature extractor element to extract the selected features of the incoming and outgoing SMS messages, and then pass these features the signature detection engine. The signature-based detection engine initially compares the selected features of a given SMS message (FromPhone#, ToPhone#, and Content) with provided signatures, and, if there is a match, the SMS is blocked. However, if the selected features do not match any of the signatures, the algorithm goes deeper and analyzes the body of the SMS. We extracted URLs, phone numbers, and commands by finding token strings in each SMS text body and matching them against the defined signatures.

As some attackers use obfuscation techniques to avoid detection, we define three variable patterns to match obfuscated URLs, phone numbers, and commands using regular expression. If the SMS text has a URL, command, or phone number, the algorithm matches it against provided signatures. If there is a match, the SMS is blocked, but if there is no match, more evidence is sought to classify the SMS by applying rule-based techniques. A set of

Table 1 Features selected for SMS signature-based detection

Feature	Description
FromPhone#	sender phone number
ToPhone#	recipient phone number
URLs	links within SMS message
Command	specific words
Phones#	phone number in SMS content
Content	SMS text
Phishing Words	words used with malicious content

rules is then applied for unknown SMS. If the SMS matches a rule, it is classified as suspicious. Otherwise, it is considered normal. Finally, the output from the SMS signature-based detection algorithm labels the SMS as normal, suspicious, or malicious. If the SMS message is malicious, it will be deleted; however, if the SMS message is suspicious, it will be shown to the user with some information related to the detected SMS. If the SMS message is normal, a message to this effect will be displayed for the user.

3.3 Anomaly-Based Detection Module

The anomaly-Based approach requires that an application be installed on user smartphones, to perform real-time signature detection and monitor smartphones' behaviour, in order to build Android profiles.

The detection module consists of five components. The architecture of the detection module is shown in Figure 1. The first component is an SMS and profile collection, which is responsible for receiving, combining, storing and retrieving data. The second component is SMS clustering, which groups SMS messages based on their similarity. The third component is SMS classification that classifies an SMS text message to one of the class labels. The fourth component employs profile analysis on each of the reported Android profiles. The final component is SMS correlation that applies rule-based correlation techniques to identify whether there are correlations between outputs from the class labels and any abnormal activities in Android profiles.

The basic steps of our anomaly detection approach are as follows: The pre-process step takes the labelled datasets and applies the "stop words removal" and "stem words" functions. Afterward, the clustering step uses an X-Mean algorithm that assigns SMS messages to a number of clusters, such that each cluster has similar distances between the instances. The clustering analysis step then analyzes the output of the clusters, and groups them into four class labels. In the SMS classification step, only the reported SMS messages from the Android devices are input into the classifier. The classifier assigns each reported SMS message to one of the four class labels. In this step, we verify each class label to confirm that the messages are correctly classified. This step is repeated until all SMS messages have been classified. Next, the profile analysis step carries out profiling analysis on the reported profiles. Finally, the SMS correlation step is employed to draw rule-based correlations between the four class labels and profile outputs, in order to label the SMS messages as either normal or malicious and to identify SMS botnets.

3.3.1 SMS and profiles collection

The SMS and profiles collection is where input data is stored. There are three types of input sources, namely, labelled SMS datasets, reported text messages, and reported Android profiles. The SMS profiles collection is responsible for collecting, combining, storing, retrieving and managing this data, to allow for more robust detection. There are no other existing tools used to capture SMS messages and smartphones' behaviour. We have implemented a real-time signature detection method in Section 3.2 to detect SMS botnets in smartphones, and to report SMS messages and profiles that need further analysis. A set of the features, as listed in Table 2, is each reported SMS messages, along with a time stamp. The third input source to our detection, Android profiles have a set of features as listed in Table 3. The Android profiles are collected in smartphone devices and are reported whenever a suspicious or malicious SMS is detected.

3.3.2 SMS clustering

The advantage of using the clustering technique is that it provides a logical summary of the collected data in terms of text-clusters [14]. It can be used

Table 2 Reported SMS features

Feature Name	Feature Description
TypeofSMS	label SMS as incoming and outgoing
FromPhone#	sender phone number
ToPhone#	receiver phone number
Text	SMS text
URLs	links within SMS message
Command	botnet C&C instructions
Phone#	phone number in SMS content
AgentID	identify an agent who delivered SMS
Timestamp	time when profile received
In contact list	is the FromPhone# in contact list

Table 3 Android user profiling features

Feature Name	Feature Description
Installed Applications	List of current installed applications.
Running Applications	List of running applications.
Granted Permissions	List of current application permissions.
Running Services	List of current running services.
Browser Accessibility	Keep track of browser usages.
Connectivity Times	Observe user connectivity time.

to offer a summary understanding of the complete content of the underlying dataset. The unsupervised clustering algorithm that is used for SMS botnet detection is X-means clustering.

Given a set of SMS messages, *S*, to be clustered into X number of groups and an N $\times$ N distance matrix, we must begin by randomly initializing the first cluster center that is selected from among the data points that are being clustered. This cluster center, called the centroid, assigns each SMS to the closest center cluster, and then follows an iteration of the following steps until it gets to its stable status: first, determine the centroid coordinate, then determine the distance of each object from the centroid, and finally, group the object based on minimum distance (i.e., find the closet centroid). After all input SMS messages have been assigned, the new mean value is updated by calculating each centroid and then the SMS are re-assigned to clusters. This reassigning and updating continues until each SMS message is in the correct cluster.

The data set that is used for the cluster is a combination of malicious and legitimate SMS messages. As an output of the clustering, a number of clusters will have different kinds of messages. We analyze the result of clusters and group them into four class labels. The first class is called *'All Malicious'*, which consist of only malicious SMS; the second is *'Majority Malicious'*, in which the majority of the text messages are malicious messages; the third is *'All Normal'*, that contain all legitimate text messages; and the fourth is *'Majority Normal'*, where the majority of text messages are legitimate.

3.3.3 SMS classification

Although clustering is fundamentally an unsupervised learning approach, the clustering technique can also be used to increase the classification accuracy of supervised detection [1]. In order to classify the reported SMS to one of the four class labels list in Section 3.3.2, the following method is used.

In this stage, when new suspicious SMS messages are reported to the detection module, we begin by pre-processing each one. After that, we take each text message and add it to all the class labels. For each class label, we calculate the TF-IDF weight, then apply the cosine similarity method to measure the similarity of the text message to each group by calculating the mean of each group. We find the minimum mean vector among the four class labels, and assign the text message to that class. After that, we remove the SMS message from other class labels, and update the class labels. We consider that all the SMS messages in 'All Malicious' and 'Majority Malicious' classes are malicious SMS, and we check the majority malicious class to look for any

misclassification. If no misclassification is found, we then give a reason why the message is labelled as malicious. In order to verify the SMS messages in 'All Normal' and 'Majority Normal' classes, a further analysis is required, with additional information, to make a decision about the reported messages. The four class labels will then be sent to SMS correlation components.

3.3.4 Android profiling analysis

To build an Android profile, we extracted the features that are related to SMS botnet behaviours. The profiles are collected in smartphones and then reported these profiles to service provider by the app-profile and the user-profile agents.

In an Android platform, an SMS botnet wishing to infect a smartphone must trick an Android user into installing a malicious application that can receive commands from a C&C server through SMS text messages. To detect the SMS botnet, it is important to monitor all installed applications, running applications, granted permissions for running applications, and running services. We take an extra precaution by keeping track of browser usages, in order to prevent any communication with the bot through the browser. One botnet behaviour is to send out SMS messages at premium-rate without the user's knowledge. We are able to spot this behaviour by observing user connectivity times and SMS logs. In Android devices, permissions are used to notify the user of what activities will be carried out and which resources an application will gain access to. The basis of the profile analysis component is a form of profile aggregation which accepts profiles as input and produces high level attack scenarios as output, displaying the results in an Abnormal Profile Table (APT), as shown in Figure 2.

3.3.4.1 *Profile aggregation*

The main idea behind profile aggregation is to group all similar profiles together. Studies differ in their criteria for alert aggregation; for example in some studies, alerts are considered similar to each other if they have the same attributes, while in others similarity may be based on several attributes. Alert aggregation has been recognized to be highly effective in decreasing the alert volume. The aggregation of profiles takes into account the similarity between particular profile features. Similarity between values of each feature (e.g. Android_ID, FromPhone#, ToPhone#, sending_time, received_time, URLs, Command, Phones#, Content, Phishing Words, contact_list, dangerous permissions, services, connectivity time) has been well-defined based on the characteristics of each feature. What alert aggregation is looking for is any deviation that can be recognized as abnormal behaviour. This abnormal

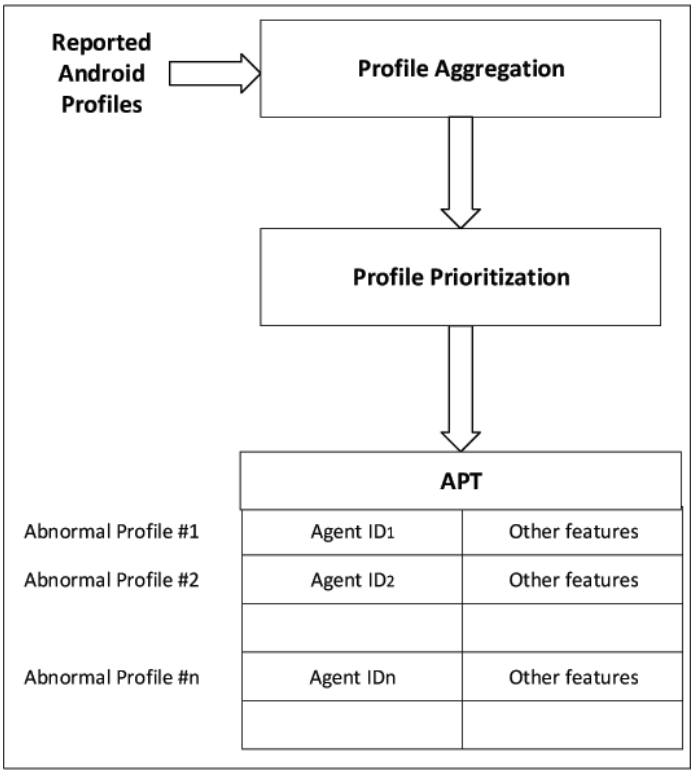

Figure 2 Profile analysis diagram and abnormal profiles table.

behaviour is referred to as malicious activity. For instance, one of the attributes is 'sender phone number' ("FromPhone#"). We would examine in connection with the text messages sent from that number, the percentage of devices that have reported the same suspicious phone number. Similarly, we compare user connectivity time with outgoing SMS time stamps by investigating the logs reported by SMS detection agents.

3.3.4.2 *Profile prioritization*

The next phase of profile analysis is to prioritize each profile based on the following two features: dangerous permissions, and user connectivity time. The objective is that, by means of the profile priority rank, an administrator can choose a high risk profile as the selected profile for further correlation and analysis. If the profile has dangerous permissions and connectivity time, it will be considered a high risk profile; otherwise, it will be considered low risk. The profile outputs will be stored in the abnormal profile table.

Android attackers enable malicious applications to send out SMS messages to premium-rate phone numbers without the user's knowledge, and to send out text messages while the phone is in sleep mode. The proposed framework is able to detect these malware activities by taking user connectivity time into account, along with dangerous permissions, when prioritizing profiles.

3.3.4.3 *Abnormal Profiles Table (APT)*

Figure 2 contains an illustration of an APT. APTs maintain records of all reported Android profiles. In the detection module, the SMS profiling and Android profiling agents decide about an SMS message and its profile on receipt. The APT divides profiles into two categories. Normal profiles, which consist of Android profiles with no indication of suspicious activity; Abnormal profiles, which are recorded in the APT of the profile analysis, and require a suitable response.

The Android profiling analysis engine combines those profiles that have the same attributes, except "Android_ID", "sending_time", and "received_time", and then flags these as combined profiles. The "Android_ID" is the unique name of the reported device. The algorithm then applies profile prioritization to prioritize the profile outputs based on risk. The suspicious profiles will be stored in the APT that can then be used by the SMS correlation component and the Security Administrator.

3.3.5 SMS correlation

SMS messages have additional attributes that are noted in the detection module. These attributes are used to create profiles. We attach the features to find any correlation between outputs from profiles and the text messages. To reconstruct attack scenarios based on the profiles and the reported SMS in each class label, we use SMS correlation to identify the relationship between the outputs of the profiles and each detected SMS message. The idea behind SMS correlation is to provide insight into attacks by analyzing raw profile outputs and SMS messages. SMS correlation is too difficult to be attempted in a single stage.

3.3.5.1 *Rule-based correlation*

In an Android platform, any application that sends or uses SMS service features must have permissions to access the service. An attacker has to define the permissions it needs before starting an attack. The important fields that need to be considered are in rule-based correlation as follows: attack prerequisites, or logical conditions that ensure the success of the attack; attack consequences,

or logical conditions that identify the influence of the attack when this attack succeeds; and scenarios that describe the combination of events which are necessary to detect an occurrence of the attack [9]. In this approach, alerts are basically a set of logical facts about how Android platforms and SMS botnets work. In order to correlate two alerts directly using rule-based correlation, one predicate in the consequences condition of the fist alert should be connected with one predicate in the prerequisites condition of the second alert. We have studied the characteristics of Android SMS botnets and extracted some features that can aid in the detection of SMS botnets. After these alerts are discovered, correlation rules are applied, which explain the conditions under which an alert may occur, and preparation is made for the second alert, in order to correlate them directly if applicable.

3.3.5.2 *SMS correlation engine*

At first we apply feature extraction to each SMS message. The results of the feature extraction are called alerts. In the second step, we will correlate each SMS messages to its profile outputs. If the message has an alert, we apply the correlation rules explained in Table 4. If any match of the rules is found, the SMS message will either be labelled as malicious or will require further analysis by an administrator. However, if there is no match, the algorithm will apply the next correlation rules. Take for example the scenario of an SMS that has "has_URLS" as one of the extracted features. That SMS will be compared with corresponding profile outputs that have the features, "Blacklist" (F1), "In contact list" (F2), "Dangerous permissions" (F3), "SMS sent in sleep mode" (F4), and "The percentage of same SMS reported by Android devices" (F5), and with the profile that have at least one of these feature, as shown in Table 4.

3.4 Defence Module

Typically, an SMS defence module begins by gaining insight into unknown SMS botnets and then generates signatures and rules. The defence module described in this paper attempts to protect Android smartphones by introducing a proactive approach to generate signatures and rules. The defence module consists of four components, namely, signature generation, phone number blacklist (PNBL), malicious application analysis, and response action. The signatures identify known SMS botnets and the rules that are used to spot unknown SMS botnets. The defence module uses the output received from a detection module to make logical decisions based on a set of policies that have been established by a human-network manager.

Table 4 Correlation rules

Extracted Features	F1	F2	F3	F4	F5	Action
Sender_num	Yes					Malicious
	No	No	Yes	No	0%	Malicious
	No	No	No	No	5%↑	Malicious
	No	No	No	No	5%↓	Check URL
Has_URL	No	Yes	Yes	No	0%	Malicious
	No	Yes	No	No	20%↑	Malicious
	No	Yes	No	No	20%↓	Check URL
	No	No	No	No	5%↓	Malicious
	No	Yes	No	No	10%↓	Malicious
	No	No	Yes	No	0%	Malicious
	No	No	No	No	5%↑	Malicious
	No	No	No	No	5%↓	Check num
Has_num	No	Yes	Yes	No	0%	Malicious
	No	Yes	No	No	20%↑	Malicious
	No	Yes	No	No	20%↓	Check num
	No	No	No	No	5%↓	Malicious
	No	Yes	No	No	10%↓	Malicious
	No	No	Yes	No	0%	Malicious
	No	No	No	No	2%↑	Malicious
	No	No	No	No	2%↓	Check command
Has_command	No	Yes	Yes	No	0%	Malicious
	No	Yes	No	No	2%↑	Malicious
	No	Yes	No	No	2%↓	Check command
	No	No	No	No	2%↓	Malicious
	No	Yes	No	No	2%↓	Malicious
Content	No	Yes	Yes	No	40%↑	Malicious
	No	No	Yes	No	10%↑	Malicious
	No		Yes	Yes	0%	Malicious
Outgoing SMS	No		Yes	No	0%	Malicious
	No		No	Yes	0%	Malicious
	No		No	No	0%	Apply above rules

3.4.1 Signature generation

The idea behind signature generation is to generate signatures that are representative of attack patterns. To ensure acceptable rates of false positives and false negatives during the signature detection process, we consider many exploits, and frequently update the signatures. The central agent sends the signature updates to all Android mobile devices.

The first line of defence against SMS botnet activities is the signature detection module in Section 3.2, which scans incoming and outgoing SMS messages. Obviously, known SMS botnet attacks would be easily stopped by

blocking SMS messages that match the corresponding signatures. Unknown SMS botnet attacks, however, if they match the defined rules, will be reported to the detection module as suspicious messages in order that we may perform further investigation and label them either malicious or normal. For all messages labelled malicious, signatures will be created based on selected features that are described in Table 1.

3.4.1.1 *Signature generation engine*

The defence module receives the malicious SMS messages reported by the detection module. At first we compare the new SMS messages with the existing malicious SMS messages. If an SMS message already has a signature, the algorithm will attempt to match the message's other features. If there is any match, the SMS message will be ignored. If there is no match, the algorithm will generate a signature of the following features: FromPhone#, ToPhone#, URLs, Command, Phones#, Content, and Phishing Words. It will then repeat the same process until it has generated a signature for each malicious SMS message. The signature updates will be sent to all subscribed Android mobile devices.

3.4.2 Phone Number Blacklist (PNBL)

Blocking malicious SMS is the primary defence against SMS botnet attacks. Clearly, SMS-based attacks would be defendable by filtering if there were regularities in one or more of the attributes of the malicious SMS on Android smartphones. A phone number blacklist (PNBL) contains a list of phone numbers that the SMS botnet detection app should block and should not accept any SMS text messages from. A PNBL can be queried with the signature detection module and allows an efficient way to perform lookups. As an example, when detection results report that a set of malicious SMS messages having the same phone number (a common feature of malicious traffic) is initiating harm (sending SMS spam, commands, etc.), we would generate a signature of the phone number and then send it to the signature detection module in Android smartphones, so that the module could perform a signature scan and block SMS text messages from this phone number.

3.4.3 Malicious applications analysis

Malicious apps are the primary means by which SMS botnets, receiving commands through the SMS service, perform attacks. Analyzing reported apps and extracting their features is therefore a strong method of defence against SMS botnets. The profiling analysis step is done in the detection module, and

the outputs are shown to a security administrator, who can perform static and dynamic analysis using common tools.

The profile outputs represent the degree of risk presented by an installed application (low, medium, or high), as gauged by a specific set of security rules. For example, the use of permissions is not as dangerous in some apps as it is in others. The profile outputs can include a normal feature of an attacked smartphone and can be part of a totally legitimate profile. However, a malicious app can also exploit this feature. Research experiments show that, it may not be always possible to confirm the intent of using permission to recognize an attack. Nevertheless, security administrators are be able to use this technique to understand the functionality of the malicious app, and to confirm features and characteristics of malware.

3.4.4 Response

The results of the detection module determine the degree of threat or severity of an attack against the Android smartphone. Although identifying malicious SMS messages will help to block SMS botnets by taking down SMS bots and cutting the C&C channel, it also requires the Android user to cooperate by removing the malicious application.

The security administrator is able to send a request to users, asking them to perform an action, for the protection of their smartphones. We have developed an SMS botnet detection app that runs agents, performs signature detection, and provides an interface to allow the administrator to communicate and interact with the Android user. In the Android platform, users themselves have to uninstall the apps based on the information provided. The administrator provides an extensive explanation about the malicious app, including information about its publisher, and other apps from the same publisher. Also, the administrator indicates what dangerous permissions the app used, and notifies the user that the app is sending out SMS messages without the user's knowledge.

4 Experiments and Results

To determine the capability of the multi-agent system, the SMS signature-based detection module, the anomaly-based detection module, and the defence module in accurately detecting SMS botnets, we conducted different experiments. We analyzed the overall performance of our proposed framework and provided a thorough analysis of JADE agents monitoring mechanism after

demonstrating the capability of each module individually. Table 5 summarizes the details of the five datasets.

To generate signatures, we used five types of datasets as shown in Table 6.

4.1 Signature Detection Module Results

In this experiment, we used two datasets. The first dataset is a labeled dataset called the British English SMS that has 425 malicious text messages. The second dataset is the NUS dataset that has over 55,000 unlabeled text messages to evaluate the proposed framework. First, the experiment used the British English SMS; we loaded all SMS messages and ran our application prototype, then reported the results. In the second experiment, we randomized the NUS dataset that has total of 55,835 text messages that have the threats that are defined in Section 5. We divided the 55,196 SMS messages into 11 sets, each set having approximately 5,000 SMS messages. We used 11 Android emulators to load each set to an emulator and ran our SMS botnet detection application prototype. As is shown in Figure 1, we first ran the feature extractor and then applied our SMS signature detection algorithm to classify the SMS message to normal, suspicious, or malicious.

In this experiment, Figure 3 shows the experiment distribution results of signature detection module for each set. Also, the summary of NUS dataset signature detection results are shown in Table 7. The signature detection agents

Table 5 Details of the datasets used for experiments

		Labelled		
Dataset Name	Dataset Creator	Hams	Spams	Unlabelled
DIT SMS Spam	Delany et al. [10]	0	1,353	
smsSpamCollection	Almeida et al. [4]	4827	747	
IIIT-D SMS Spam	Kuldeep et al. [20]	1000	1000	
British English SMS	Nuruzzaman et al. [17]	450	425	
NUS SMS Corpus	Chen et al. [7]			55,835

Table 6 The number of signatures records

Dataset Name	Content	URLs	Phone Num	Phishing Words	Commands
DIT SMS Spam	1353	95	1165	0	0
IIIT-D SMS Spam	1000	129	304	0	0
SMS Spam Collection	747	104	490	0	0
British English SMS	425	56	248	0	0
Our dataset	0	824	0	170	397

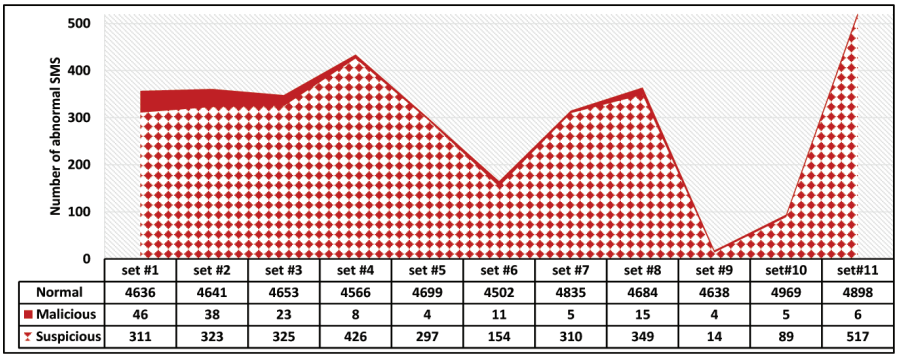

Figure 3 The experiments distribution result of signature detection.

Table 7 The proposed framework experimental results

Types	Features	# of SMS	Total	Percentage
Malicious	SMS body	0	165	0.5%
	Phones	3		
	URLs	23		
	Commands	139		
	FromPhone#	0		
	ToPhone#	0		
Suspicious	Phones#	869	3081	5.5%
	URLs	144		
	Commands	2182		
Normal			51721	94%

send 3,115 suspicious SMS text messages and 165 malicious SMS messages to the SMS profiling agent and sends commands to app-profile and user-profile agents requesting the current profiles be sent to the Android profiling agent. 139 of SMS messages contained C&C botnet commands that have corresponding command signatures and 26 malicious SMS messages have malicious URLs. The signature detection agents reported 869 suspicious phone numbers, 144 suspicious URLs, and 2,182 suspicious commands.

4.2 Anomaly-Based Detection Module Results

In this section, we evaluate the proposed detection approach using the standard metric. The SMS botnet detection module receives the reported SMS messages and Android profiles, and then performs anomaly detection. We performed the experiments using various datasets. The input to the detection module consisted of three types of data: well-known datasets, reported SMS messages,

and reported Android profiles. It was vital to select appropriate datasets to evaluate the precision of the proposed approach. IIIT-D SMS Spam Dataset [20], a labeled dataset which has 1,000 spam SMS messages and 1,000 normal SMS.

In order to get the anomaly-based detection module to perform well and to detect SMS botnets intelligently, we used four steps of evaluation methodology to detect SMS botnets. First, the anomaly-based detection module takes the labeled datasets that contain malicious and normal SMS [4] and cluster them based on content similarities using the X-means algorithm. The result of the clustering produces a number of clusters that are analyzed and categorized into four class labels. Second, the anomaly-based detection module uses the 353 reported SMS messages that need to be classified into one of the four class labels using the SMS classification approach. Third, the anomaly-based detection module applies profile analysis to the Android profiles using aggregation and prioritization techniques to produce an abnormal profile table (APT). Finally, the anomaly-based detection module applies rule-based correlations to SMS messages in the four label classes and the profiles outputs in order to label each message in each class label as a malicious or normal message.

In the service provider server, where all agents reported the SMS messages and its profile. We started by combining all the datasets that are described in Table 5 and we removed duplicated from the datasets. Although SMS spam messages are characterized by obfuscation, we kept many of the non-identical messages that might still be close matches. We randomized spam SMS messages and normal SMS messages and chose 500 normal SMS messages and 500 spam SMS messages.

In the first step, we clustered 1,000 SMS messages using X-mean clustering technique and then applied our clustering analysis method. The results of SMS clustering algorithm are described in Table 8. In the second step, we applied an SMS classification algorithm by taking 165 reported malicious SMS messages that were received from signature detection agents and classified all malicious SMS to malicious class labels which help to classify suspicious SMS messages. The SMS classification algorithm also classified the 3,081

Table 8 The SMS clustering and classification results

Type of Data	M	MM	N	MN	Total
Dataset SMSes	338	107	153	402	1000
Malicious SMSes	165	0	0	0	165
Suspicious SMSes	56	39	2891	95	3081

Table 9 The detection module results for NUS dataset

Label	Total	Phones#	URLs	Commands	Phishing Words
malicious	941	818	136	281	144
normal	2152				

reported suspicious SMS messages to one of four class labels. The result of the classification are given in Table 8. For the NUS dataset, 2,891 of SMS messages are classified as normal and 95 SMS messages are classified as majority normal. 56 SMS messages are labeled as malicious and 39 SMS messages are labeled as majority malicious. In the third steps, Analyze the reported profiles by applying the Android profiles analysis algorithm that produces the abnormal profile table. In the fourth step, employed SMS correlation algorithm that applied correlation rules 4 to label the instances in each one of four class labels. Table 9 shows the results of the detection on NUS dataset. 941 of SMS messages are labeled as malicious and 2,152 of SMS messages are labeled as normal.

4.3 Discussion

We are focusing on detecting malicious applications that misuse the Short Message Service (SMS). Considering the privacy of users' information, the proposed approaches always show the user what the agent will send and get the user's permission to send it. We have collected all the features that we think will play an important role in identifying these malicious applications. These features are critical to user privacy and the user may be concerned about the data that the agents capture. One solution is to give the user full decision-making power on whether or not to allow the agents to send the information to the service provider. Our approach provides full details about the detected SMS to the user, who may take action by deleting the SMS and removing all the reported applications. The goal of the proposed framework is to protect SMS from malicious interference. Google has improved the use of SMS provision starting with Android version 4.4 but attackers always have ways to obfuscate Android security, and thousands of mobile devices are still using older Android versions.

To avoid loss of connection, Android agents are in charge of confirming that the data are received and stored safely in the service provider. JADE has the ability to maintain the connection between agents and has a store-and-forward mechanism; if the connection is lost, the agent will re-establish the connection as soon as it is available and re-transmit the data as soon as the connection is up [6].

5 Conclusions and Future Work

In this work, we have proposed an SMS-based botnet detection framework that uses multi-agent technology based on observations of SMS and Android smartphone features. The proposed detection framework is based on a multi-layer model which consists of three processing modules with the use of JADE agents: 1) SMS signature-based detection; 2) SMS anomaly-based detection; and 3) a defence module. In addition, multi-agent technology is a powerful tool that can monitor certain environments and report abnormal behaviour in order to protect user data. investigation to confirm whether the SMS message has C&C instruction.

The work performed in this paper provides a basis for future research of intrusion detection systems based on a multi-agent system in mobile devices. One area of future work is applying a broader range of features for intrusion detection. These features need to be calculated in real-time to enable the detector to keep up with large number of the reported SMS messages and their profile. Another interesting area that can be investigated in the future is to extend the framework. If an agent can make decisions on the fly about suspicious SMS messages. These decisions are based on other agents findings if they report the same SMS message with its characteristics.

References

[1] Aggarwal, C. C., and Zhai, C. (2012). "A survey of text clustering algorithms," in *Mining Text Data*, eds C. C. Aggarwal and C. X. Zhaipages (New York, NY: Springer), 77–128.

[2] Aguero, J., Rebollo, M., Carrascosa, C., and Julián, V. (2010). "Developing intelligent agents on the android platform," in *Proceedings of Seventh Asia Joint Conference on Information Security*, (Washington, DC: IEEE).

[3] Alam, M., Cheng, Z., and Vuong, S. (2014). "Context-aware multi-agent based framework for securing android," in *Proceedings of International Conference on Multimedia Computing and Systems*, (Washington, DC: IEEE), 961–966.

[4] Almeida, T. A., Hidalgo, J. M. G., and Silva, T. P. (2013). Towards sms spam filtering: results under a new dataset. *Int. J, Inform. Sec. Sci.* 2:1–18.

[5] Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. (2008). Jade: a software framework for developing multi-agent applications. lessons learned. *Inform. Soft. Technol.* 50:10–21.

[6] Bergenti, F., Caire, G., and Gotta, D. (2014). "Agents on the move: Jade for android devices,". In *Proceedings of 2014 Workshop From Objects to Agents*. (New York, NY: ACM).

[7] Chen, T., and Kan, M.-Y. (2013). Creating a live, public short message service corpus: the nus sms corpus. *Lang. Resour. Eval*. 47, 299–335.

[8] Cheng, Z. (2012). *A Multi-Agent Security System for Android Platform*. Master's thesis, University of British Columbia, Vancouver, BC.

[9] Cuppens, F., and Miege, A. (2002). "Alert correlation in a cooperative intrusion detection framework," in *Proceedings of 2002 IEEE Symposium on Security and Privacy*, (Washington, DC: IEEE) 202–215.

[10] Delany, S. J., Buckley, M., and Greene, D. (2012). Sms spam filtering: methods and data. *Exp. Syst. With Appl.* 39, 9899–9908.

[11] Geng, G., Xu, G., Zhang, M., Guo, Y., Yang, G., and Wei, C. (2012). The design of sms based heterogeneous mobile botnet. *J. Comput*. 7, 235–243.

[12] Hua, J., and Sakurai, K. (2011). "A sms-based mobile botnet using flooding algorithm," in *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*, (New York, NY: Springer), 264–279.

[13] Kok, J., and Kurz, B. (2011). "Analysis of the botnet ecosystem," in *Proceedings of the 10th Conference of Telecommunication, Media and Internet Techno-Economics (CTTE)*, (Berlin: VDE), 1–10.

[14] Larsen, B., and Aone, C. (1999). "Fast and effective text mining using linear-time document clustering," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY: ACM.

[15] Li, Y., Zhai, L., Wang, Z., and Ren, Y. (2013). *Control Method of Twitter and SMS-Based Mobile Botnet*. Berlin: Springer.

[16] Mulliner, C., and Seifert, J.-P. (2010). "Rise of the ibots: owning a telco network," in *Proceedings of the 5th International Conference on Malicious and Unwanted Software*, (Washington, DC: IEEE), 71–80.

[17] Nuruzzaman, M. T., Lee, C. and Choi, D. (2011). "Independent and personal sms spam filtering," in *Proceedings of the 11th International Conference on Computer and Information Technology*, (Washington, DC: IEEE), 429–435.

[18] Rosenberg, D. (2013). *CarrierIQ: The Real Story*. Available at: http://vulnfactory.org/blog/2011/12/05/carrieriq-the-real-story/

[19] Vural, I., and Venter, H. (2010). "Mobile botnet detection using network forensics," in *Future Internet-FIS 2010*, eds A. J. Berre, A. Gómez-Pérez, K. Tutschku, and D. Fensel (New York: Springer), 57–67.

[20] Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A., and Naik, V. (2011). "Smsassassin: crowdsourcing driven mobile-based system for sms spam filtering," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, (New York, NY: ACM), 1–6.

Biographies

A. J. Alzahrani is assistant professor at the college of computer science and engineering (CCSE), University of Hail (UOH), Saudi Arabia. He earned his Ph.D. from the faculty of Computer Science, University of New Brunswick, Canada in October 2016. His research interests include botnet detection, Android security, network security, malware analysis and reverse engineering. Dr. Alzahrani was the project manager for the Smart Campus at King Saud University, Riyadh, Saudi Arabia. Dr. Alzahrani received the Saudi Arabian Cultural Bureau's Academic Excellence Award in Ph.D. Program 2016 and the Saudi Arabian Cultural Mission's Academic Excellence Award in Master Program 2008. He received the Lawrence Technological University's Academic Honor Award in Master Program 2008. Dr. Alzahrani is a member of the IEEE, a member of the ACM, and Canadian Information Processing Society. He is a member of the Information Security Centre of Excellence, University of New Brunswick. He is member of Saudi Security group (Hemaya), Riyadh, Saudi Arabia.

A. A. Ghorbani has held a variety of positions in academia for the past 35 years and is currently the Canada Research Chair (Tier 1) in Cybersecurity, the Dean of the Faculty of Computer Science, and the Director of the Canadian Institute for Cybersecurity. He is the co-inventor on 3 awarded patents in the area of Network Security and Web Intelligence and has published over 200 peer-reviewed articles during his career. He has supervised over 160 research associates, postdoctoral fellows, graduate and undergraduate students during his career. His book, Intrusion Detection and Prevention Systems: Concepts and Techniques, was published by Springer in October 2010. In 2007, Dr. Ghorbani received the University of New Brunswick's Research Scholar Award. Dr. Ghorbani has developed a number of technologies that have been adopted by high-tech companies. He co-founded two startups, Sentrant and EyesOver in 2013 and 2015. Dr. Ghorbani is the Co-Editor-In-Chief of Computational Intelligence Journal. He was twice one of the three finalists for the Special Recognition Award at the 2013 and 2016 New Brunswick KIRA award for the knowledge industry.