Intelligent IDS for Mobile Devices: Modeling and Prototyping

Aline Lopes da Silva, Zair Abdelouahab and Denivaldo Lopes

Federal University of Maranhão - UFMA,
Department of Electrical Engineering, Campus do Bacanga, São Luís-MA, Brazil
{aline.lopes, zair, dlopes}@dee.ufma.br

Abstract: Recently, mobile devices and wireless networks are more common for supporting human activities providing mobility and flexibility. Mobile devices inside a wireless environment contribute to make easily accessible and available the information. However, wireless environment is susceptible to vulnerabilities due to information which is propagated through the air and thus exposed to interception or theft. Mobile devices present some physical limitations such as little capacity of processing, memory and battery duration. These limitations are critical in wireless environment when non identified threats attack mobile devices. Intrusion Detection Systems (IDSs) specialized to mobile devices are necessary in order to identify intrusive behavior taking into account their physical limitations. In this paper, we propose an IDS for wireless networks and mobile devices. This proposal is an extension of Intrusion Detection System-Network Intrusion Detection System based on Intelligent Agents (IDS-NIDIA).

Keywords: Security, Intrusion Detection, Mobile Devices, Wireless Networks.

1. Introduction

The advent of the wireless network IEEE 802.11 [1] also called WLAN has brought to users and organizations with new types of threats.

In wired networks a malicious person needs to obtain physical access to practice any malicious activity. On the other hand in wireless networks the attacks can occur from anywhere and can compromise some network nodes, once the propagation of information comes through the air.

A wide range of devices are using wireless technology, but mobile devices are the major benefited from wireless technology for enabling communication.

Wireless networks have increased in popularity due to the facility of installation and configuration. However, users are looking for efficient solutions to meet the basic requirements of information security: confidentiality, availability and integrity.

Recently, mobile devices using wireless networks become the major victims of attacks in this type of unsafe environment.

Intrusion Detection System (IDS) are installed in strategic places for monitoring the security violations or unauthorized accesses coming from inside or outside of the network.

In this paper, we propose an IDS for mobile devices based on extensions and adaptations of IDS-NIDA [17] that uses intrusion detection mechanisms based on monitoring of devices through their behavior profiles and data packets transmitted in a wireless network. Thus users of mobile devices which access information through a wireless network have a security mechanism adapted to their reality observing the limitations of hardware and software.

This paper is organized as follows: section 2 presents an overview of technologies used in this research work. Section 3 describes the model in UML diagrams of our solution of IDS for mobile devices. Section 4 shows a prototype of our proposed architecture of IDS for mobile devices. Section 5 presents related works. Section 6 concludes a discussion about our contributions and directions of future research.

2. Overview

In this section, we present the concepts and technologies that serve as base for the development of our research work.

2.1 Wireless network and mobile devices

Wireless networks allow devices equipped with wireless interfaces to use network resources without being physically connected to them. These devices need to be inside the environment covered by electromagnetic waves (called extension) of the wireless network infrastructure. A Wireless Local Area Network (WLAN) consists of a group of wireless nodes that are inside a limited geographic region and that is capable of establishing communication via radio [9].

WLANs are typically used by devices inside of an extension clearly delimited, as inside a building, and are generally implemented as extensions of Local Area Networks (LAN) to provide an enhanced mobility to end users. The major organization responsible for wireless network is the Institute of Electrical and Electronics Engineers (IEEE).

Mobile devices are a gamma of network nodes such as tablet PCs (Personal Computer), PDAs, smartphones, mobile phone and iPhones. The advent of mobile devices opens new opportunities for automating industrial process and making use of information technology. Information systems are no more confined to the physical limits of buildings and users can continue making their activities in the environment covered by a wireless communication infrastructure.

Technologies for mobile computation and wireless network have evolved quickly, but their capacity is inferior to desktops processing capabilities, storage and communication.

Physically, mobile devices presents memory size up to 100 Mbytes, memory extension through memory cards like such as Secure Digital card, processors with clock up to 300 MHz, color screen with high resolution and with touchscreen technology, light weight, wireless network interface like as Wi-fi, Bluetooth and Infrared.

These devices allow access to Internet, reading, writing and storing files containing personal data, contact list, documents, pictures and videos.

2.2 Limitations of mobile devices

Due to the small capacity of resources (e.g. processing power and memory size) of mobile devices compared to desktop computers, software that has an important demand of resources can become unavailable for utilization.

A mobile device also depends on energy provided by a battery that has a limited useful life. The life expectative of battery of these devices can be measured in hours or weeks. However, the life time of a battery is related to its activity; bigger the demand of processing power, smaller will be the life expectative of a battery. This leaves mobile devices exposed to resource scarcity and exhaustion attack. This latter consists in reduction of life time of batteries and consumes all the power of the device that becomes inappropriate for utilization.

2.3 Security in wireless networks

WLANs should adopt a security mechanism. The aim is to reach security through the combination of security tools embedded in the wireless network specification. The fundamental mechanisms are:

- Confidentiality: assures that data are not read by unauthorized persons or entities.
- Integrity: assures that data are not modified during the transmission.
- Availability: assures that network resources are available when required to legitimate users.
- Access control: restricts users and devices that require access to a network resource.

Security for wireless networks and wired networks present some common aspects, and the major category of threats and attacks are common to both. In [14], NIST provides a general taxonomy for attacks in wireless network classifying them in two basic types of attacks: passive and active.

In passive attacks, the attacker simply listens and analyzes network traffic with the objective to capture sensitive information of the target. These kinds of attacks compromise confidentiality of the end user traffic. In passive attack, an unauthorized user gain gal access to the network traffic without modifying it [5, 6].

In active attacks, the attacker may create serious routing disruptions by altering or dropping the packets or selectively forwards packets toward the destination [6, 20]. Active attacks can take one of the following forms or a combination: masking, repeat, message change, denial of service.

2.4 Attacks in mobile devices

There are some attacks that compromise the communication in wireless environment. For example, some attacks intend to compromise the availability of services to users or steal information that can be personal or confidential [7]. Wireless networks and mobile devices increase the number of nodes exposed to attacks. Therefore, attacks that are specialized in interrupt and steal the communication of mobile devices are acquiring each time more fans.

The weak performance of mobile devices is actually the indication that attacks are focused to stress the resources in order to obtain denial of service or unavailability. For example, loading the processor and discharging the battery are the more common attacks to mobile devices.

Therefore, mobile devices are vulnerable to a type of denial of service known as battery exhaustion attack in which an attacker tries exhaust quickly the battery of the device [4]. This attack is firstly identified in ad-hoc networks on wireless network sensors [11]. In these networks, the energy source that is consumed during the communication is the major factor of the batteries life time.

Mobile devices are often in power sleep mode for retaining energy and extend the battery life time. The aim of sleep deprivation attack is to consume the battery charge through increasing in the processor load. These types of attacks explore the energy management of mobile devices in order to inhibit the ability of moving to a state of reduction in battery consummation.

There are three categories of attacks in mobile devices [9]: malignant power attacks, benign power attacks and service request power attacks. Malignant power attacks are programs designed or modified with the aim of surcharging the processor and therefore consume quickly the battery charge. Benign power attacks have the aim of increasing the consummation of the resources through the repeated call of normal process. Service request power attacks have the aim to consume the resources of the mobile device through the repeated call of services. These attacks are more common requests in a network.

2.5 Intrusion detection and prevention systems for wireless networks

An IDS makes the monitoring of events on a computational system or network, analyses them in order to find signals of possible incidents such as violations or menaces [9, 14, 16, 17].

Security incidents can have many causes such as malicious code and attackers trying to obtain unauthorized access to systems from Internet, and authorized users that profit of their privileges and try to obtain additional privileges to obtain unauthorized access to system from inside of their intranets. Intrusion prevention is the process to make the detection of intrusions and try stopping possible incidents.

An IDS consists of a software that automates the intrusion detection process identifying the attack after it occurs. An Intrusion Prevention System (IPS) consists of software that has all the capacities of an intrusion detection system and can also stop possible incidents before they occur.

IDS tehnologies are associated to many methodologies to detect incidents [9, 13, 19]: *signature-based*, *anomalies-based* and *protocol state analysis*. *Signature-based* relies on identifying known signatures that corresponds to a menace. However this technique is susceptible to a slight variation of the attack signature and also to an unknown attack. *Anomalies-based* is a process of comparing activities and depends on the pattern of computer usage. It flags any computer activity that runs differently from the acceptable profile. However, the main weakness of this approach is its susceptibility to false positives. It fails to recognize a legitimate activity when the activity is completely new. It also needs periodical update on its profile, which requires substantial time. *Protocol state analysis* relies solely on the frequency of input data based on system calls, or protocols such as IP, TCP, and UDP. This technique is computationally light, since it does not require numerous types of parameters or maintenance of activity profiles. However, it requires detailed design to avoid missed attack types [8].

IDS for wireless network provide many types of capacities of security. The fact is that IDS for wireless networks is relatively a new class of IDS, and its capacities still vary between products. For a near future, the trend is the maturity of its capacities. Among capacities that are more common: information collection, record, detection and prevention. Device identification of WLANs and identification of WLANS are example of information collection.

3. IDS Architecture for Mobile Devices

The IDS architecture specialized to end users of mobile device is based on [14].

The proposed architecture employs detection by anomalies with strategies of analysis with the aim to identify abnormal behaviors of monitored devices. This monitoring is based on information collected from mobile devices jointly with information captured in the network packets addressed to these devices.

The normal behavior of a mobile device is diagnosed through historic data collected during a period of normal of its operation. All processes of analysis and diagnosis of a possible attack are done by an IDS_Proxy which informs the device being attacked by threats and their countermeasure. Figure 1 presents a scenario of usage of our proposed solution.

According to Figure 1, the components present in the scenario are as follows:

- IDS_Proxy: captures the network traffic addressed and sent to the mobile devices being monitored. It also receives data of each mobile device in order to analyze their behaviors.
- IDS_Client: is responsible to collect information in the mobile device being monitored. This information is sent to the IDS_Proxy in regular periods for analysis.
- Access Point: allows mobile devices have access to resources of the network and coordinate the communication between them.
- Mobile device: represented in Figure 1 by Smarth Phones, PDAs and Tablet PCs.
- Threat: represents attacks to the mobile devices.

The proposed architecture is based on client-server model using techniques of detection based on a hybrid IDS that collects information about mobile devices and at same time makes the monitoring of network traffic. Figure 2 presents the architecture for IDS_Client and IDS_Proxy.

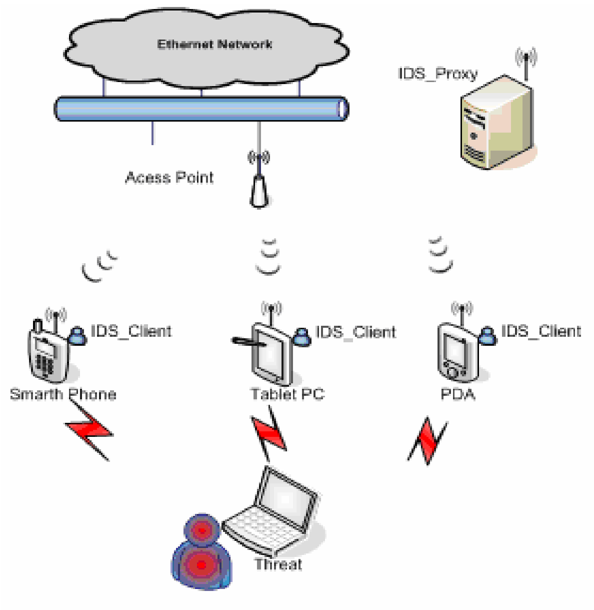
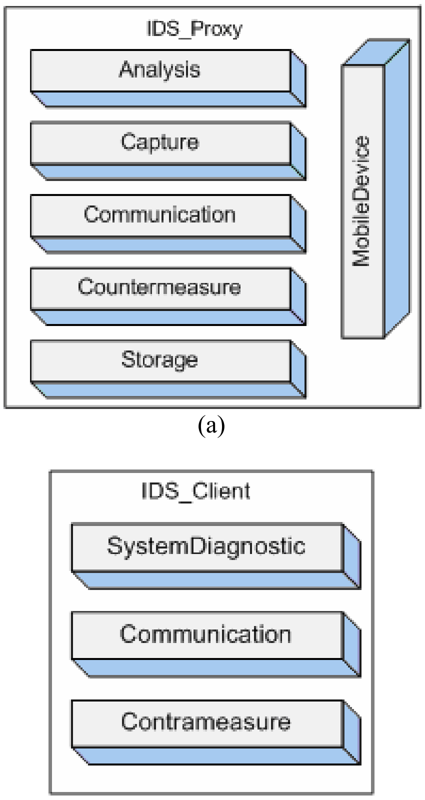


Figure 1. Scenario of usage of the proposed solution



(a)

(b)

Figure 2. Architecture: (a) IDS-Proxy and (b) IDS-Client

IDS_Proxy and IDS_Client are composed of layers that perform specific functionalities in the detection process. The description of these functionalities is described as follows.

IDS_Proxy is composed of the following layers:

- Analysis Layer: is responsible for information analysis in order to identify possible attacks through information obtained in the capture layer and based on data sent by mobile devices being monitored.

- Capture Layer: responsible for the capture of wireless network traffic. The process of capture is made by a sensor that acts in promiscuous mode in the network.
- Communication Layer: is responsible for the communication between server and client.
- Countermeasure Layer: is a layer responsible to achieve the identification of a possible attack.
- Storage Layer: is responsible for storage of information captured in the network traffic and information sent by mobile devices.
- MobileDevice Layer: is responsible to manage the information about monitored mobile devices.

IDS_Client is composed by the following layers:

- System Diagnostic Layer: is responsible for the lecture of information about monitored mobile devices.
- Communication Layer: is responsible for communication between a mobile device and the server for sending information manipulated in the analysis process and receiving of notification in situations of threat detections;
- Countermeasure Layer: is responsible to perform actions against detected intrusion. The actions depend of information sent by IDS_Proxy that determines the action to be performed by the mobile device in response to the threat.

4. IDS-NIDIA for Mobile Devices

The proposed architecture is an adaptation and an extension of the architecture of IDS-NIDIA for taking care mobile devices and wireless networks.

4.1 IDS-NIDIA

The proposal of IDS-NIDIA is an intrusion detection system based on agents capable to detect attacks in real time using a neural network.

IDS-NIDIA is based on CIDF (Common Intrusion Detection Framework) model [18] and contains agents for detecting events (sensor agents), data analysis mechanism (monitoring agents and security evaluation), data storage mechanism (database historic) and a module for executing countermeasure (agent control action). In addition, other agents are responsible for the integrity of the IDS-NIDIA and by the coordination of the IDS activity.

Agents of IDS-NIDIA have the following main objectives: generate indices of suspected attack through data analysis collected from logs of host and packets of the network; perform countermeasures according to the rates obtained; learn with experiences and update their knowledge base.

The proposed model provides a methodology for detection based on abuse and anomaly in order to ensure robustness to the system.

The architecture of IDS-NIDIA is composed of layers. Each layer has specific activities to perform the behavior that is implemented in the agents. The communication between layers is done through communication between agents. Figure 3 shows the architecture of IDS-NIDIA.

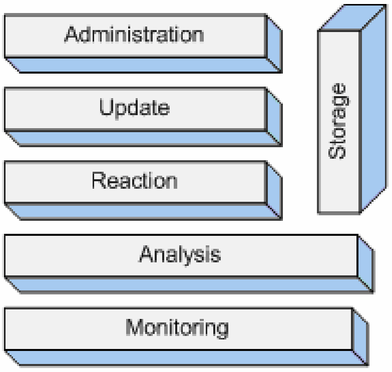


Figure 3. NIDIA Model

The Monitoring layer is responsible for capturing the occurrence of events in the outside world and providing their data for the other components of the system. In this layer, the System Monitoring Agent (SMA) is located, and it is classified into two categories: Network Sensor Agent (NSA) that captures the packets that are traveling in the network, and Host Sensor Agent (HSA) that works collecting information from a host and providing them for analysis.

The analysis layer has the task of analyzing the events received from the monitoring layer. In this layer, the events collected are formatted so that patterns of attack can be identified and confirmed as a real attack. At his point, the following knowledge bases are used: Incidents of Intrusion and Forensic Information DataBase (IIDB), Pattern of Intruders and Intrusions DataBase (PIDB) and Strategy DataBase (STDB). In this layer, the System Evaluation Agent (SEA) analyzes and provides a degree of suspicion about the events that have been previously formatted.

The reaction layer is responsible for taking countermeasures if a security incident is confirmed. In this layer, the System Control Agent (SCA) performs the countermeasures according to the Strategy Database (STDB) and action (RADB).

Update layer is responsible for maintaining all databases always updated. In this layer, the System Update Agent (SUA) have a responsibility to maintain the integrity and consistency of information stored, and it is the only one able to change the databases.

Management Control layer is composed by Main Controller Agent (MCA) that is responsible for the administration and integrity of all agents of the system.

Storage Layer is responsible for managing the persistent information in IDS-NIDIA. In this layer, the databases of IDS-NIDIA are present.

4.2 Adaptation and extensions of the IDS-NIDIA

Originally, the IDS-NIDIA is designed to run on an environment of local network, in other words, the features are developed to provide services to protect and detect intrusion in a local network. During the course of the evolution of IDS-NIDIA, several proposals are incorporated in its model due to the continued growth and variety of attacks and changes in the physical structure of local networks [2]. The diffusion of wireless networks and how they have been integrated to local networks contributed to appear new requirements and further adjustments in IDS-NIDIA.

The aim of our new architecture is to provide a safe environment for users of mobile devices using wireless networks. For this purpose, IDS-NIDIA is extended and adapted to acquire the capabilities of our proposed architecture for IDS in mobile devices presented in section 3. Thus, IDS-NIDIA is redesigned to identify intrusive behavior based on the analysis of information travelling on the wired and wireless networks and from the host or from mobile devices, and execution of countermeasures performed in real time against threat in hosts or mobile devices.

The integration of IDS-NIDIA and IDS for mobile devices is possible because both architectures are structured in layers that have very similar features as shown in Figures 2 and 3. Each layer of the proposed architecture of IDS for mobile devices can be mapped into an intelligent agent.

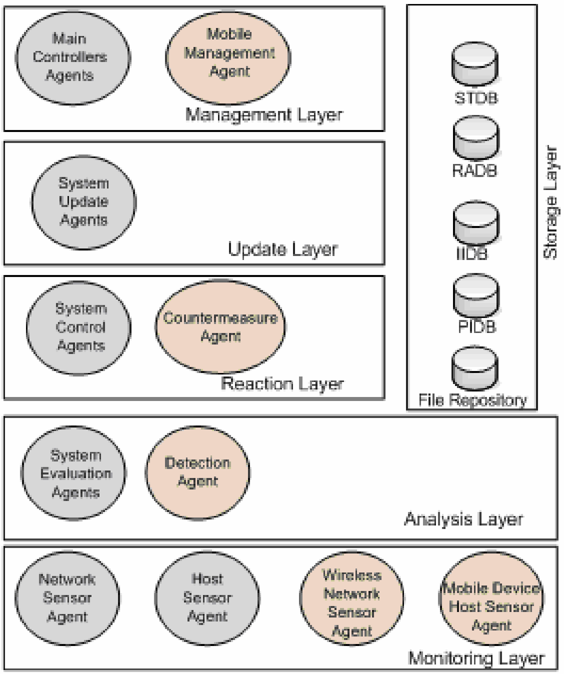


Figure 4. Integration between proposed architecture and NIDIA

Figure 4 presents the integration between our proposed IDS architecture for mobile devices and the architecture of NIDIA. Elements colored in blue represent agents and database belonging to NIDIA, while the elements in brown represent agents and repository of files of the proposed architecture for IDS in mobile devices.

The Wireless Network Sensor Agent and Mobile Device Host Sensor Agent are inserted in the monitoring layer together with the Network Sensor Agent and Host Sensor Agent. The Detection Agent is inserted into the analysis layer with inclusion of the wireless detection environment and abnormal behavior identification for mobile devices being monitored by System Evaluation Agent. The Countermeasure Agent is inserted into the reaction layer jointly with System Control Agent and it can include the ability to perform the appropriate countermeasures, i.e., perform effective actions against intrusions in the wireless environment. The Mobile Management Agent is inserted into the management layer jointly with the Main Controller Agent and it allows the management of devices being monitored. In the storage layer, File Repository is added for storing captured information.

4.3 Modeling

The extension of IDS-NIDIA is designed and modeled using Object Oriented Analysis and Design, and UML. We present a class diagram for IDS_Proxy and IDS_Client.

Figure 5 shows the class diagram for IDS_Proxy. The class FrameWifi represents the captured frames on a wireless network; the attributes of the class are established according to the format of the frame of MAC layer of IEEE 802.11b/g standard. The class ParseFrameWifi processes the captured frame and produces the attributes represented in the class FrameWifi, i.e. It makes the processing of packets in order to extract the information concerning the network layers: link, network and transport. Class AnalysisFrameWifi makes the analysis of all frames and compare them to the behavior patterns.

The class CapturePacket is responsible for the capture of frames using the class PacketCapture of net.sourceforge.jpcap.capture package, which is the core class for capturing packets of library Jpcap [11]. It provides a high-level interface for the capture of network packets through the tunnel of the library Libpcap [11] and inherits the behavior of the class Thread of package java.lang in order to be concurrently executed for capturing network packets.

The class SensorRadio has a relationship with the class SensorRadioGUI that represents the system graphical interface for viewing the captured frames and the information sent by devices. The class MobileDevice defines the attributes to identify the mobile device to be used for identification and creation of the profile of the device. The profile of the device is represented by the class ProfileMobileDevice that contains the data received from the class IDSProxyServer that is responsible for direct communication between the device and the server. The class AnalysisProfileDevice makes the analysis of information of the device and compares it with the device profile taken as pattern of normality. Manager manages the IDS_Proxys and provides an interface between the network administrator and the IDS_Proxy.

The modeling ofe IDS_Client is based on the functions provided by the application development environment for Palm OS [15] and Access [15]. Figure 6 shows the class SystemDiagnostic of the IDS_Client.

The methods of the class SystemDiagnostic are described as follows:

- getValuesMemorySizeHeap returns the amount of memory used by applications running on the device.
- getValuesMemorySizeFreeHeap returns the amount of free memory for applications that are running.
- getValuesMemorySizeFlash returns the total amount of RAM in the device.
- getValuesMemoryFreeFlash returns the amount of memory available.

- sysBatteryInfo is responsible for reading information on the device's battery with amount of left charge, the current battery voltage, the minimum permitted voltage.
- getAddress is responsible for the passage of the parameters for communication with server in order to send the captured information.
- dmNumDatabases returns the number of applications installed on the device.
- dmDatabaseInfo is responsible for consultation for more specific information for each application installed.
- netLibSocketOpen opens the socket for a subsequent communication.
- netLibSocketConnect is responsible for carrying out the connection to the server.
- netLibSend is responsible for sending the information to the server.
- netLibReceive is responsible for the receipt of any information sent to the device through sockets.

We can specify the sequence diagrams of IDS_Proxy as presented in Figure 7 and Figure 8.

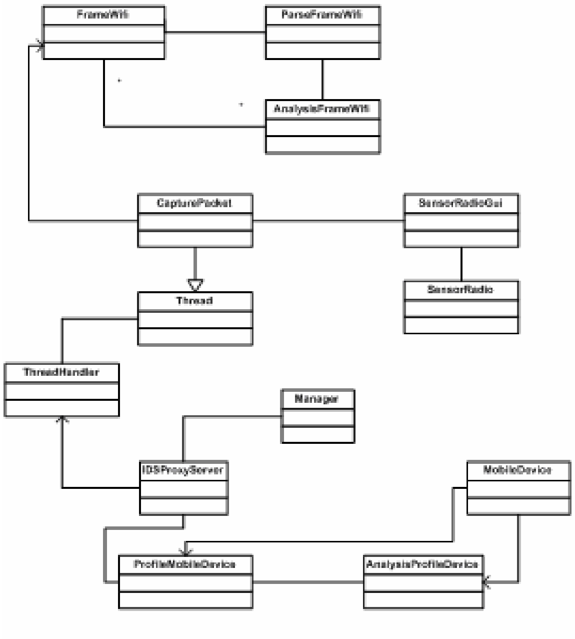


Figure 5. Class Diagram of IDS_Proxy

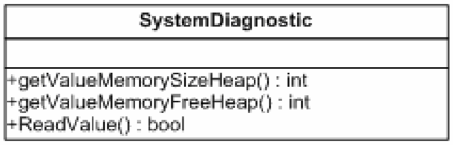

SystemDiagnostic
+getValueMemorySizeHeap() : int
+getValueMemoryFreeHeap() : int
+ReadValue() : bool

Figure 6. Class Diagram IDS_Client

The sequence diagram begins with a registration of the information about the devices to be monitored (1). In IDS_Proxy, two services are initialized and run concurrently: service for receiving data sent by clients (3) and service for capturing packets that occurs in the wireless network (4). The captured packets are submitted to an analysis for identifying the types of packets and for identifying whether they represent any threat (6). When the data from the client are received, the data is processed and compared to the information of the profile taken as normal behavior pattern for the device (7). The analysis of the information about the mobile device and the analysis of the information about network traffic are used by the system to infer if the device has or not some kind of threat (8). Thus, if an attack is detected, then the device is notified about the threat in order to (9) take the adequate countermeasure (10 and 11).

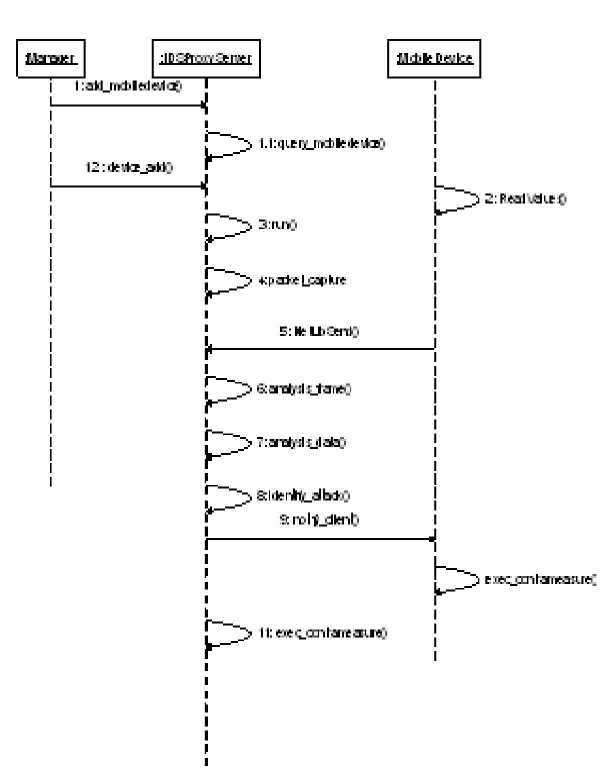


Figure 7. Sequence Diagram: IDS_Proxy capturing and analyzing the packages of wireless network.

As illustrated in Figure 8, the application needs to be initialized in the device (1) for reading the data needed (3) to be sent to the server (6) which will process the data (7) and will add these values to the device profile (9) for verification of current data with the default profile of the device (10). The inspection includes the amount of battery for the device (11), the voltage of the same (12) and programs installed on the device (13).

4.4 Prototyping

IDS_Client and IDS_Proxy were implemented as follows.

The IDS_Proxy is an adaptation and an extension of a tool proposed by R. Ataide and Z. Abdelouahab [2]. This tool captures the traffic of the wireless network as an intrusion-detection service offered by NIDIA in wireless networks. The adaptation of the tool consisted of the following features:

- Analysis of the information captured on the wireless network extended to layers of network and transport to identify the IP, ICMP and TCP.
- Filtering the traffic of mobile devices that are being monitored, and storage of information and records of attacks in the archives of the XML format.
- Support to communicate directly with device for

collecting data and sending notifications.

- Analysis of the data sent by the device to detect irregular behavior of the device and identification of records of attacks addressed to mobile devices;
- A module for capturing information about the mobile device for further examination.

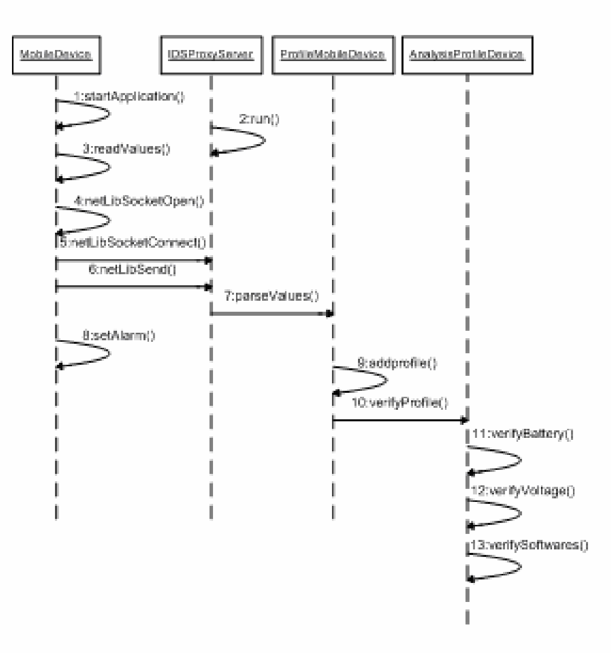


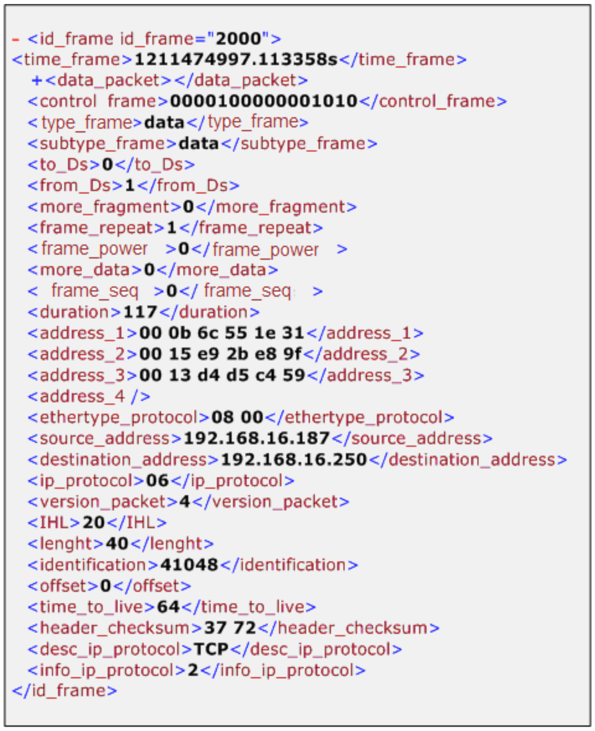
Figure 8. Sequence Diagram: sending and processing data monitored in the mobile device

The IDS_Proxy is developed using Java technology with the use of two external API: one to capture traffic on the network, Jpcap, and the other responsible for the generation of files in XML format for storage captured information.

The identification of frames from the wireless network is stored in the class ParseFrameWifi and this class is stored in an XML document that is presented in Listing 1.

The main elements of this XML document are:

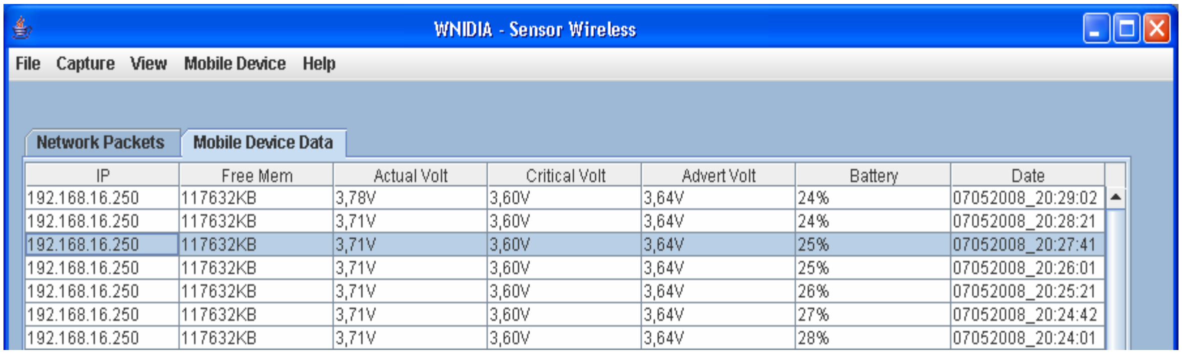
- id_frame: identifies the frame from a wireless network.
- data_packet: captured flows in hexadecimal format on the wireless network.
- control_frame: field of control of the frame, information used to identify the function of the frame in the MAC layer.
- type_frame: type of frame (data, control or management).
- subtype_frame: subtype of the frame according to their type.
- to_Ds: indicates whether this frame is going to the distribution system.
- from_Ds: indicates whether this frame is out of the distribution system.
- more_fragment: indicates whether there are more pieces of the frame to be received for season.
- frame_power: indicates whether the station comes into power save mode.



```
- <id_frame id_frame="2000">
<time_frame>1211474997.113358s</time_frame>
  +<data_packet></data_packet>
  <control  frame>0000100000001010</control_frame>
  < type_frame>data</ type_frame>
  <subtype_frame>data</subtype_frame>
  <to_Ds>0</to_Ds>
  <from_Ds>1</from_Ds>
  <more_fragment>0</more_fragment>
  <frame_repeat>1</frame_repeat>
  < frame_power   >0</ frame_power   >
  <more_data>0</more_data>
  <  frame_seq   >0</ frame_seq   >
  <duration>117</duration>
  <address_1>00 0b 6c 55 1e 31</address_1>
  <address_2>00 15 e9 2b e8 9f</address_2>
  <address_3>00 13 d4 d5 c4 59</address_3>
  <address_4 />
  <ethertype_protocol>08 00</ethertype_protocol>
  <source_address>192.168.16.187</source_address>
  <destination_address>192.168.16.250</destination_address>
  <ip_protocol>06</ip_protocol>
  <version_packet>4</version_packet>
  <IHL>20</IHL>
  <lenght>40</lenght>
  <identification>41048</identification>
  <offset>0</offset>
  <time_to_live>64</time_to_live>
  <header_checksum>37 72</header_checksum>
  <desc_ip_protocol>TCP</desc_ip_protocol>
  <info_ip_protocol>2</info_ip_protocol>
</id_frame>
```

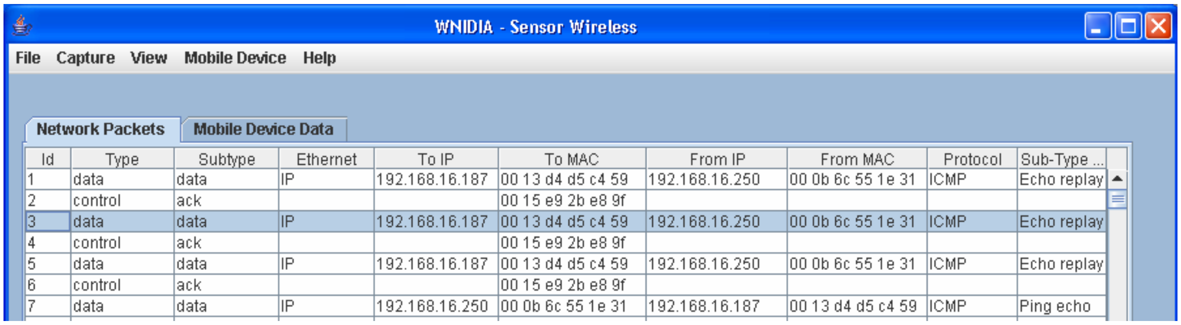
Listing 1. The identification of frames

- more_repeat If the frame is being retransmitted.
- more_data: for the station indicates that more frames are to be sent.
- frame_order: indicates whether frames received must be strictly ordered.
- duration: indicates the time needed to receive the next transmission.
- address_1: MAC address of the origin host.
- address_2: MAC address of the destination host.
- address_3: MAC address of BSSID.
- address_4: indicates the MAC address of the host.
- source_address: IP address of the host.
- destination_address: IP address of the destination host.
- ethertype_protocol: protocol encapsulated within the specified package sub layer SNAP.
- version: : version of the packet datagram.
- IHL: size of the header on the Internet.
- lenght: full-size datagram including header and data.



IP	Free Mem	Actual Volt	Critical Volt	Advert Volt	Battery	Date
192.168.16.250	117632KB	3,78V	3,60V	3,64V	24%	07052008_20:29:02
192.168.16.250	117632KB	3,71V	3,60V	3,64V	24%	07052008_20:28:21
192.168.16.250	117632KB	3,71V	3,60V	3,64V	25%	07052008_20:27:41
192.168.16.250	117632KB	3,71V	3,60V	3,64V	25%	07052008_20:26:01
192.168.16.250	117632KB	3,71V	3,60V	3,64V	26%	07052008_20:25:21
192.168.16.250	117632KB	3,71V	3,60V	3,64V	27%	07052008_20:24:42
192.168.16.250	117632KB	3,71V	3,60V	3,64V	28%	07052008_20:24:01

Figure 9. Information about mobile devices



Id	Type	Subtype	Ethernet	To IP	To MAC	From IP	From MAC	Protocol	Sub-Type ...
1	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.250	00 0b 6c 55 1e 31	ICMP	Echo replay
2	control	ack			00 15 e9 2b e8 9f				
3	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.250	00 0b 6c 55 1e 31	ICMP	Echo replay
4	control	ack			00 15 e9 2b e8 9f				
5	data	data	IP	192.168.16.187	00 13 d4 d5 c4 59	192.168.16.250	00 0b 6c 55 1e 31	ICMP	Echo replay
6	control	ack			00 15 e9 2b e8 9f				
7	data	data	IP	192.168.16.250	00 0b 6c 55 1e 31	192.168.16.187	00 13 d4 d5 c4 59	ICMP	Ping echo

Figure 10. Screenshot of traffic on the wireless network

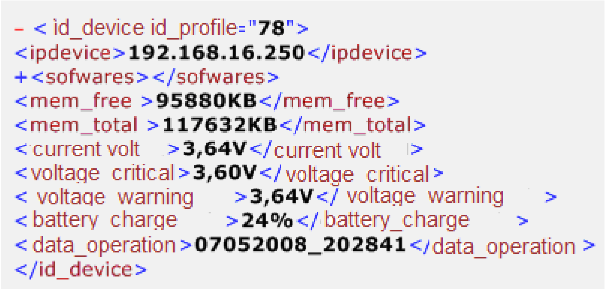
- identification: identifier of fragments of the original datagram.
- offset: datagram indicates where the fragment belongs to.
- time_to_live: service life of the packeta in the network.
- header_checksum: indicates the value of verification.
- ip_protocol: defines the protocol encapsulated in IP datagram.
- info_protocol: information on the functionality of the protocol encapsulated in the datagram.

The elements of Listing 2 are:

- id_profile: counter of information sent by the device.
- softwares: list of software installed on the device in the future, the list was hidden because (indicated by the "+" sign next to the element) extension of the list.
- mem_total: total of memory on the device.
- mem_free: a free memory on the device.
- current _volt: current voltage of the battery.
- voltage_critical: value of the voltage considered critical for the device.
- voltage_warning: voltage warning for the device.
- battery_charge: amount of battery in the device, shown in percentage.
- date_operation: records the moment when the data was received.

Figure 9 shows a screenshot about the wireless network traffic. As shown in the figure the following fields after decoding are shown to the manager:

- Id: identifier in the order in which it is captured.
- Type: type of captured frame (date, control,
- management), information obtained from the MAC layer frame.
- Subtype: defined according to type of frame.
- Ethernet: sublayer protocol specified by SNAP (Subnetwork Access Protocol).
- To IP: IP address of the destination frame.
- To MAC: MAC address of the destination frame, represented in hexadecimal format.
- From IP: IP address origin of the frame.
- From MAC: MAC address of origin, represented in hexadecimal format.
- Protocol: header encapsulated within the IP packet.
- SubType Protocol: identifies the functionality of the protocol encapsulated within the IP packet.



```
- < id_device id_profile="78">
<ipdevice>192.168.16.250</ipdevice>
+<sofwares></sofwares>
<mem_free >95880KB</mem_free>
<mem_total >117632KB</mem_total>
< current volt    >3,64V</current volt    >
<voltage  critical>3,60V</voltage  critical>
<  voltage  warning     >3,64V</ voltage  warning     >
< battery  charge      >24%</ battery_charge      >
< data_operation>07052008_202841</data_operation >
</id_device>
```

Listing 2. Information sent by the device

Figure 10 shows the screenshot about information sent by mobile device. The following information is presented to the network manager:

- IP: IP address of the device that is sending the data.
- Free Mem: indicates the amount of available memory on the device.
- Vol Current: current voltage of the battery from the

device.

- Vol Advert: index voltage warning of the device; the user will be informed that its battery needs to be loaded again.
- Battery: the percentage of available battery in the device at the time of transmission of information.
- Date: indicates the time of transmission of information sent from the device, a range of 40s (forty seconds) is used for transmission of data from the device.

4.5 Testing

Attacks were simulated as a way of demonstrating the effectiveness of our IDS architecture for mobile devices.

In wireless networks the battery of mobile devices is an important issue, especially when it comes to small devices such as PDAs and Smartphones. These devices have physical limitations on the ability of processing and storage. Based on these constraints, generally, attacks require greater impact on these devices. Two attacks are simulated: Sleep deprivation and Syn Flood Attack. These attacks generate requests through the network with the aim to increase the processing load of the device and the consequent depletion of its battery charge.

4.5.1 Simulating the attack sleep deprivation

The attack sleep deprivation is a type of attack that leads to depletion of the battery from the device preventing it to enter into the sleep mode power, status, in which the device reduces the demand for activities with the aim of preserving resources, especially the charge of battery. The objective of this type of attack is against the maximum of activities of the device, consequently unload it and prevent that user has access to information stored in it.

The simulation for this type of attack consists in the demand of an excessive amount of requests through the network interface of mobile device in order to keep it busy.

To prevent the device into sleep mode power, it is generated attacks based on ping flood attack, where hundreds of requests were generated for the purpose device to keep certain level of the operation's network interface device preventing it comes to a situation of inactivity and between state power in sleep.

Two types of attack based on the following commands were done: the first command *ping 0.000001-i-s 136 192.168.16.250* and the second command *ping 0.000001 -i-s 512 192.168.16.250*. Where:

- -i: is the interval for recovery of command.
- -s: size of the package sent.
- "129.168.16.250" the target where the command is run.

For the first simulation scenario, we have opened twenty terminals from one machine runs Fedora Core Linux operating system; each terminal executes the command: *ping –I -0.000001 -s 136 192.168.16.250* during 40 min (forty minutes). The command is implementing the packet with a size of 136 bytes at intervals of 0.000001 seconds against the device with the specified IP. The time in which this command is running is sufficient to represent the effects that have taken significant proportions in the level of charge and battery voltage of the device. These data are shown in the charts below. Figure 11 shows a comparison between the device's battery charge versus the time during attack execution.

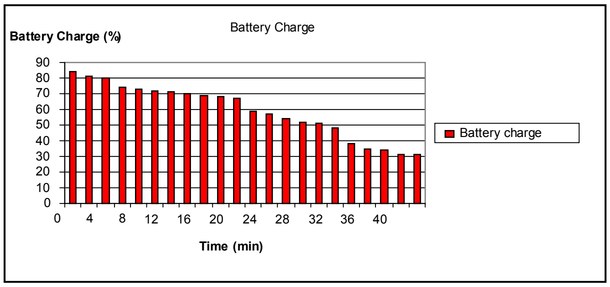


Figure 11. Ping flood attack (battery versus time)

The graph in Figure 12 provides a comparison between the battery's voltage versus time during the same period of execution of the attack.

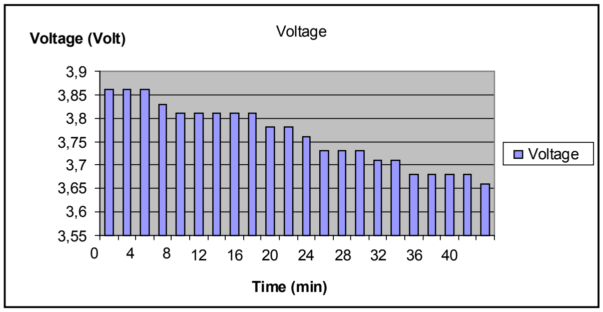


Figure 12. Ping flood attack (voltage *versus* time)

Analyzing information acquired through charts in Figure 11 and Figure 12, it appears that ping flood attack during a period of time can affect considerably the battery charge. Measures at regular intervals of 40s (forty seconds), we observe that in a time of approximately 40min (forty minutes), the remaining battery charge is reduced by less than 50% (fifty percent) of the value obtained in the beginning of the attack realization for the first simulation. A similar situation can be observed in measurements of voltage battery which also has decreased.

To simulate the second scenario, the same variables of the first test are retained except that the executed command is *ping -I -s 512 192.168.16.250*. The command has made usage of packets with 512 bytes size at intervals of 0.000001 seconds against the device. The results are shown in the graphs of Figure 13 and Figure 14.

On the second scenario, we observed that increasing the useful load of the package sent to the device there was an acceleration in the discharge of the battery that has reached 45% (forty-five per cent) of remaining charge with approximately 25min (twenty-five minutes) simulation of the attack. The speed of the discharge is due to the processing required to process packages with greater payload. This is reflected in energy consumption which is a demand, requiring more battery consumption of the device.

4.5.2 Simulating the attack SYN flood

SYN flood is a form of DoS attack in which the attacker sends a stream of SYN requests for the host (victim of the attack), but does not answer with the acknowledgment.

The notification process for this attack is based on the three-way handshake. Initially, the client makes a request for the connection by sending a SYN message to a specified target. The target acknowledges the request by sending out SYN-ACK to return to the customer; this in turn responds with the ACK for the connection to be established.

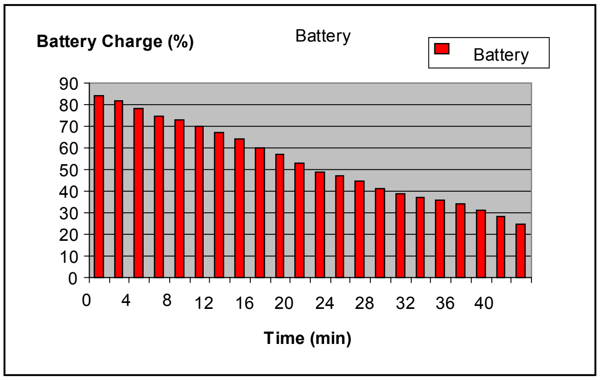


Figure 13. Scenario 2: Simulation ping flood attack (battery *versus* time)

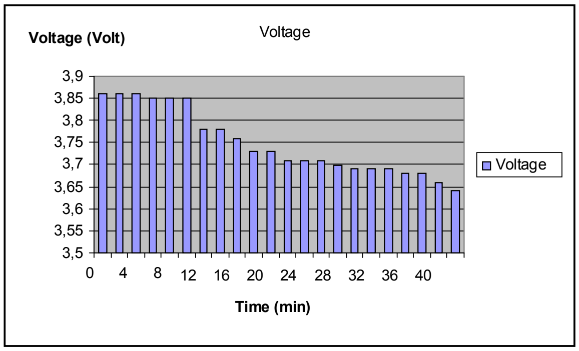


Figure 14. Scenario 2: Simulation ping flood attack (voltage versus time)

The objective of this attack on the device is to maintain certain level of activity in its network interface via the response that is returned by the device even in situations where it is not possible to connect. The intention is really to make the device spend energy and processing in the process of responding to these requests.

To conduct this type of attack, the use of a tool that generated packets with the features of the proposed attack is necessary, i.e. generation of packets with the control SYN flag turned on. The tool hping2 [10] is used for this purpose. This is a command-line tool to create packets containing payloads TCP, UDP or ICMP that can be modified and controlled using command-line syntax.

The attack generated for the prototype is based on the command *hping-I eth0 -S 192.168.1.250 –p++80-i u1000.* This command creates packets over the network where I indicates the interface of origin for generating traffic (eth0, used in the example), S-flag indicates that the control SYN entitled to a destination indicated by 192.168.16.250 and p is a port to which the request must be made. The supplement to "++" before the port number indicates that the command will enhance the port number in an attempt to drive to every request of requisition and -i interval indicates where the requests will be made in the simulation for each thousand microseconds.

This command is run in two different ways: in the first situation the countdown begins in the port number 10 (ten) and in another situation the countdown starts from port 80 (eighty). The attack has started with requests on ports 10 and 80 because of known services running in this range or above it.

These two commands are executed concurrently against the device during an interval of 40 min (forty minutes) time. The measures are based on analysis of the level of load and voltage of the battery.

The impact on these primitives is shown in Figure 15 and Figure 16.

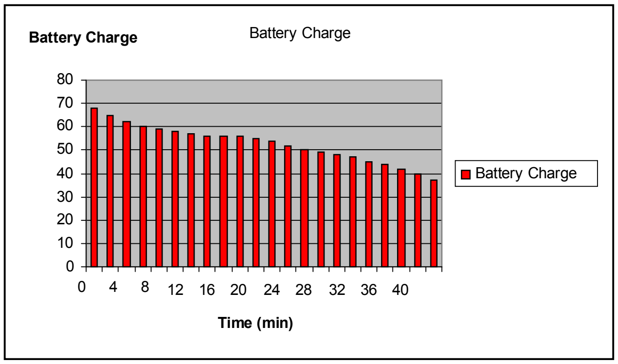


Figure 15. SYN flood attack (battery versus time)

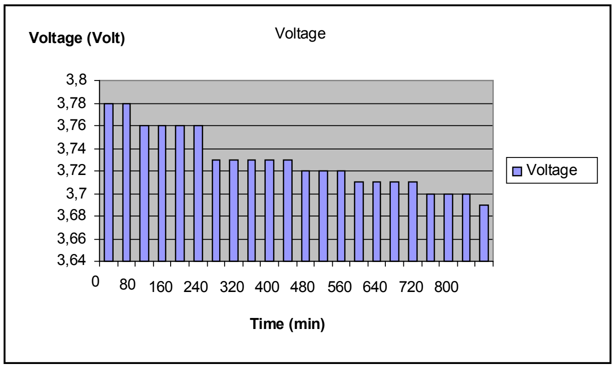


Figure 16. SYN flood attack (voltage versus time)

Comparing the effects of the sleep deprivation and SYN flood attacks, we remark that the former attack decreases more quickly the charge and voltage of the battery than the latter attack.

In fact, for each figure relating battery charge versus time, we can trace a curve that can be defined as a pattern of attack. Similarly, for each figure relating voltage versus time, we can trace another curve for another pattern of attack. However, we present the experiments during 44 (forty-four minutes), our proposed IDS architecture for mobile devices can take a countermeasure in the beginning of the attack avoiding low level of charge and voltage. In both simulations of attack, the implementation of our IDS architecture for mobile devices has detected an abnormal behavior on the wireless network in a couple of minutes.

5. Related works

Several studies have been proposed in the area of IDS for wireless devices [3, 4, 12].

In [3], T. Buennemeyer et al use the concept of an IDS that takes as a parameter activities that have the greatest impact on the battery of mobile device. It is composed of two modules: BSIPS (Battery Sensing Intrusion Protection System) and CIDE (Correlation Intrusion Detection Engine). BSIPS provides the monitoring of limits and warning notices when changes are detected outside the range of the devices. The hosts are employed with sensors on the wireless network and form the basis of the Canary-Net IDS.

In [4], G. A. Jacoby et al introduce an IDS based on the behavior of battery of mobile devices. The BBID (Battery Based Intrusion Detection) is composed of two modules, the HIDE (Host Intrusion Detection Engine) and HASTE (Host Analyze Signature Trace Engine). They are based on a set of rules that aims to identify any abnormal behavior of these battery devices.

In [12], D. C. Nash et al propose a model of IDS toward mobile devices with analysis taking activities running on the mobile device such as active processes in the device, access to the disk, to estimate the power consumed by active process in device. Thus, this estimation allows the identification of cases with increase of demand for the device and thus detects processes that can represent potential attacks to exhaust the battery.

6. Conclusions

Security measures provided for wireless networks are a work in constant evolution. Meanwhile, the techniques of security are designed to protect the environment without major wireless solutions for the end user.

A wireless environment is a heterogeneous environment, composed of small-sized equipments. Thus, the measures of protection must take into account the composition of the components of this type of environment.

The mobility within the wireless environment allows the development of equipment which practicality and portability as essential characteristics. However, these facilities impacts on the project of these devices. Limitations of storage, processing and dependence on batteries such as energy source were the factors given in exchange for the freedom and mobility. These factors are crucial when you think about security measures for mobile devices. Thus, security tools for mobile devices must include their limitations, and we need security solutions that have the least possible impact on the computing resources of these devices.

The main contribution of this paper is a proposal of an IDS architecture for mobile devices that is able to detect intruders in hybrid wireless networks based on the model of detection of anomalies and on the model of detection by signatures. The proposed architecture is based on information about the behavior of mobile device. This behavior is inferred through observations about resources available in the device and monitoring of network traffic.

In future research work, we intend to improve our architecture incorporating:

- Techniques for safe and reliable communication between the IDS_Client and IDS_Proxy;
- Adaptation of IDS_Proxy to other forms of attacks focused on mobile devices, such as attacks that occur on the UDP protocol;
- Countermeasures by the IDS_Proxy and expansion of countermeasures in IDS_Client.

Acknowledgments

This work is supported by the National Council for Scientific and Technological Development (CNPq, Brazil), and the Foundation for Support of Research, Science and Technological Development of Maranhão (FAPEMA).

References

[1] ANSI/IEEE Std 802.11, 1999 Edition, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 1999.

[2] R. Ataide, Z. Abdelouahab, “An architecture for wireless intrusion detection systems using artificial neural networks,” International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE), 2008.

[3] T. K. Buennemeyer, M. Gora, R. C. Marchany, and J. G. Tront, “Battery exhaustion attack detection with small handheld mobile computers,” In Proceedings of IEEE International Conference on Portable Information Devices, pp. 1-5, 2007.

[4] G. A. Jacoby, N. J. Davis, and R. Marchany, “Using battery constrains with mobile hosts to improve network security”, IEEE Security e Privacy, Vol. 4, No. 5, pp. 40-49, 2006.

[5] S. Khan, N. Mast, K.-K. Loo, A. Salahuddin, “Cloned access point detection and prevention mechanism in IEEE 802.11 wireless mesh networks,” Journal of Information Assurance and Security, Vol. 3, No. 4, pp.257-262, 2008.

[6] S. Khan, N. Mast, K.-K. Loo, A. Salahuddin, “Passive security threats and consequences in IEEE 802.11 wireless mesh networks,” Journal of Digital Content Technology and its Application, Vol. 2, No. 3, pp. 4-8, 2008.

[7] Kingpin and Mudge, “Security analysis of the Palm operating system and its weaknesses against malicious code threats,” In Proceedings of the 10th USENIX Security Symposium, pp. 135-151, 2001.

[8] M. F. Marhusin, L. Cornforth, D. Henry, “An overview of recent advances in intrusion detection,” In Proceedings of 8th IEEE International Conference on Computer and Information Technology, pp. 432-437, 2008.

[9] T. Martin, M. Hsaio, H. Dong, J. Krinshnaswami, “Denial of service attacks on battery-powered mobile computers,” In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, pp. 309-318, 2004.

[10] O. Bonaventure, “Software Tools for networking”, IEEE Network, Vol. 18, No. 6, pp 4-5, 2004.

[11] C. Morariu, B. Stiller, “DiCAP: Distributed Packet Capturing Architecture for high-speed network links,” In Proceedings of 33rd Annual IEEE Conference on Local Computer Networks (LCN), pp. 168-175, 2008.

[12] D. C. Nash, T. L. Martin, D. S. Ha, M. S. Hsiao, “Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices,” Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 141-145, 2005.

[13] P. Ning, S. Jajodia, “Intrusion detection techniques,” The Internet Encyclopedia, H. Bidgoli, ed., Wiley & Sons, pp. 2–6, 2003.

[14] T. Karygiannis and L. Owens, “Wireless network security: 802.11, bluetooth and handheld devices,” Special Publication 800-48, 2002.

[15] Palm, “Palm API Guide, Palm OS Platform – version 5.6,” Available at: https://pdnet.palm.com/regac/pdn/PalmOSAPIGuide/index.html, Accessed in 02/10//2009.

[16] M. Servilla, R. Heady, G. Luger, and A. Maccabe, “The architecture of a network level intrusion detection system”, Technical Report CS90-20, Department of Computer Science, University of New Mexico, August, EUA, 1990.

[17] M. Silva, D. Lopes and Z. Abdelouahab, “A Remote IDS based on Multi-agent Systems, Web Services and MDA,” Proceedings of the IEEE International Conference on Software Engineering Advances, pp. 64-69, 2006.

[18] S. Staniford-Chen, B. Tung, and D. Schnackenberg, “The Common Intrusion Detection Framework (CIDF),” The Information Survivability Workshop, 1998.

[19] A. Patcha, J. M. Park, “An overview of anomaly detection techniques: existing solutions and latest technological trends,” Computer Networks, Vol. 51, No. 12, pp. 3448-3470, 2007.

[20] S. Khan, K.-K. Loo, T. Naeem, M. A. Khan, “Denial of service attacks and challenges in broadband wireless networks”, Journal of Computer Science and Network Security, Vol. 8, No. 7, pp. 1-6, 2008.