Event-Based Method for Detecting Trojan Horses in Mobile Devices

Daniel Fuentes[1], Juan A. Álvarez[1], Juan A. Ortega[1],
Luis González-Abril[2], and Francisco Velasco[2]

[1] Computer Languages and Systems Department
University of Seville
Avda. Reina Mercedes s/n, 41012, Seville, Spain
[2] Applied Mathematics Department
University of Seville
Avda. Ramón y Cajal 1, 41018, Seville, Spain
{dfuentes,jaalvarez,jortega,luisgon,velasco}@us.es

Abstract. Mobile phones and wireless technology and its constant evolution have, in the last years, revolutionized the way in which we communicate and work. However, one of the main barriers encounter in the use of these technologies is data security. Trojan horses are dangerous software to attack phones, PDAs and Smartphones. New versions are created everyday to attack the functionality, theft the stored information and propagate themselves. In this paper, we present a new real-time method to detect Trojan horses in mobile devices. We study the events in the device to detect programs which can be suspected to be Trojan horses. By doing so, we can detect not only the known Trojan horses with more accuracy, but also detect new trojans. Practical experiences on different devices have been carried out and results show the effectiveness of the method.

Keywords: Trojan horses, mobile devices, infection, propagation.

1 Introduction

Today, mobile devices represents the most extended and changed technology. Even personal computers do not grow, or improve as quickly and in the same way as this technology on the rise. Moreover, the growing number of services and benefits, are becoming more essential in our daily life because they provide not only the basic voice communication service, but also contain other forms of communication such as SMS, MMS, Bluetooth, or Email. And all of these services can be an infection source. Security experts are finding a growing number of infections that target cellular phones. Nowadays, none of the new attacks has done extensive damage in the wild, but it could only a matter of time before this occurs. Researches' attack simulations have shown that before long, hackers could infect mobile phones with malicious software that deletes personal data or runs up a victim's phone bill by making toll calls. The attacks could also degrade or overload mobile functions, and eventually causing them to crash. We are not more concerned with the security of our mobile

phones and the importance of the information which is stored. Furthermore, these devices do not come with antivirus software. There are various attacks, and one of these is made by Trojan horses. Nowadays, mobile devices are more and more threatened by Trojan horses. It is easy to infect a mobile device by Trojan horses when it has not any protection. When a phone is compromised, a Trojan horse may be planted in the computer as a back door so that the intruder can control the victim's device thereafter, unless the user can delete it. A Trojan horse may be also planted into a mobile phone through SMS, MMS, Bluetooth message or Email attachment. When a user downloads a file like game or a game from some malicious web sites, a Trojan horse may be downloaded too. A mobile phone can also be infected with Trojan horses when the user of the mobile device browses some malicious web sites.

Traditionally, when a Trojan horse is planted in computer, the intruder can only send information to the intruder's device and it started its propagation to other devices. But a complex Trojan horse can destroy all the system by deleting some files in the computer, even formatting the disks. Up to know, various strategies [1-2] for Trojan horse detection in a PC are developed. One way is to scan the hard disks through file name matching. Another way is to scan registry database. The third way is to scan open ports. If a port that matches the one used by a Trojan horse is found to be open, it implies that there is a Trojan horse. All these methods can detect only some known Trojan horses, but not unknown ones. But mobile-device technology is still relatively new, and vendors have not developed mature security approaches. This is an area of constant change, where everyday appears new devices, platforms, software and, obviously, malware. With new Trojan horses appearing daily, new way should be developed to detect unknown Trojan horses. In this paper, we discuss the problem of Trojan horse detection in mobile devices, and present a new and novel method to detect Trojan horses. In this way, we can detect not only the known Trojan horses with more accuracy, but also detect new trojans in real time.

The structure of this paper is as follows. In section 2 we present some related works. In particular, section 2.1 focuses on Trojan horse attacks whose main objective is, in addition to its own quickly propagation, the theft of private user information while the victim is unaware. In Section 2.2 and Section 2.3 the existing techniques and our solution to detect Trojan horses are described respectively. Section 3 describes the behavior of the Trojan horse in mobile devices through practical experience with real infections. Finally, in Section 4, we summarize our conclusions and present some lines of future work.

2 Background and Related Work

The popularity of research on mobile devices is growing. Mobile devices like cell phones, Smartphones, PDAs (Personal Digital Assistants) or laptops have become popular and widespread available due to factors such as their ease of transportation, flexibility, storage capacity and increasing computing capabilities. Because of that, there are a lot of mobile services victims of this malware:

Mobile agents. Recent studies on mobile agents [3-4] describe the security and reliability of these software platforms, which can be migrated from one device to another and continue their execution. Security is the most important problem for mobile agent systems, especially when money transactions are concerned.

Mobile services. They provide many applications like mobile TV services [5-6] (it combines mobile phone services with television content and represents a logical step both for consumers and operators and content providers), mobile payment [7], tourism services [8] or m-learning [9].

Mobile advertising. Mainly, this form of advertising is a subset of mobile marketing. There are published papers discussing different methods (or approaches) for mobile advertising [10-11].

Domotic systems. For several years, the use of mobile devices has become essential to new domotic systems that improve your home life. [12-13].

In the last years, many types of malware for mobile devices have appeared. They can degrade mobile functions, delete or steal personal data, increase the victim's phone bill or disable the device completely. Each service that allows the user to connect to another device can be a source for a virus intrusion or other threats. For that reason, an infection can attack a device through different ways, as we can see in Figure 1. SMS, MMS and Bluetooth are together the most common ways for a possible infection. The small size of SMS (only 160 bytes, 160 characters) is the main disadvantage and the reason why there is not yet been a large-scale infection via SMS. However, MMS is one of the most used routes of infections. The size of the MMS is imposed by the service provider: usually it is more than 300KB, which seems an appropriate size to accommodate the malware. Bluetooth technology develops different levels of security based on the identification of the devices involved but, in spite of that, the number of vulnerabilities via Bluetooth has increased considerably. One of the most dangerous is e-mails, since there are no size restrictions and they can spread more easily to other tools, mainly PCs. Other ways, like USB connections, allow an infection to move from one device to another. Finally, WI-FI networks provide interoperable wireless access but sometimes the network origin and reliability is unknown.

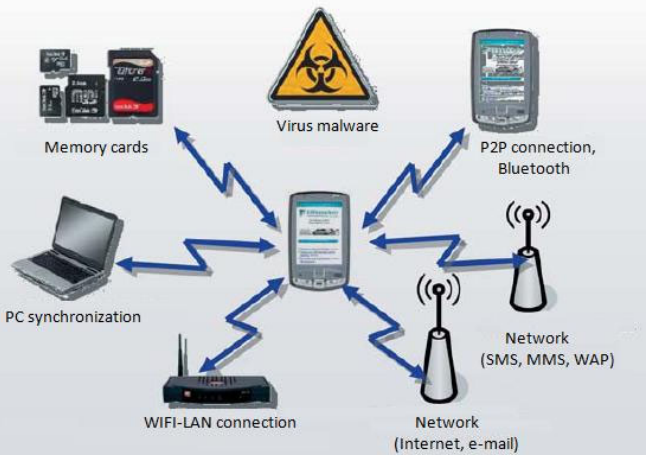

Fig. 1. Spyware infection routes in mobile devices

The security solutions that currently exist for mobile devices were originally created for PCs and, consequently, they approach the key challenges of the mobile environment such as limited processing power and secondary issues. Recently, products like Flexilis[14] or Airscanner [15], which are dedicated exclusively to mobile devices security, protect mobile devices against threats including viruses, malwares or spam.

2.1 Trojan Horses in Mobile Devices

What happens if your mobile phone or PDA is lost or stolen? The device may contain confidential data and legal liabilities could arise if it contains confidential information such as medical records. We have already seen that there are many ways in which a mobile device can be attacked, but this article focuses on attacks with Trojan horses, where their primary goals are to get information in a maliciously way and its own propagation.

Nowadays, when a user downloads a song, video, image or video game in his terminal, he can download an attached infection too. These downloads are usually done by sending SMS or MMS messages from unknown origins. As we will see in the demonstration section, a Trojan horse will be camouflaged inside an image, but it can be introduced in other file (into a video, music or game file) to the terminal.

One clear example of a Trojan horse is the Mosquito Trojan 2.0, which accompanied the pirated version of the game for mobile devices with the same name. The Trojan did not affect the functionality of the device, but sent SMS messages to premium services (1 €/SMS approximately) while the user was playing with an illegal copy of the game. In fact, it is very probable that there are still websites where a user can download the game and, although there are two warnings before installing it, some users may be tempted to install it. Despite everything, this Trojan horse disappeared when the mobile game was deleted.

2.2 Existing Detection Techniques

A program file of a Trojan horse can be planted to the victim's system in different ways as pointed out in the section below. Some Trojan horse detections use fixed file names when they are installed into a device. For this reason, one way for Trojan horse detection us scan the hard disks through file name matching [1-2]. Another way is to scan registry database and check the sentences detection to find the trace of these malicious files. In mobile platforms, there are file managers or software development kits (SDKs) to permit access to these files. Both techniques are simple and efficient, and they can work in real time. Furthermore, when an infection is detected, it is deleted automatically for the computer.

The last technique is based on the feature that a Trojan horse must use a port to make net connection. Especially, some Trojan horses use fix port number [1-2]. If a port that matches the one used by an unauthorized application, it can be suspected to be a Trojan horse with high possibility. With all these techniques, known Trojan horses infections can be detected but not new ones. With our detection technique, known or new Trojan horses can be detected in a mobile device.

2.3 Trojan Horse Detection through Events Tracing

Some trojans detection techniques have been described in a previous point. But all these techniques cannot detect new infections. In mobile world, everyday appears new malicious software, hence it is necessary a technique to detect known and unknown Trojan horses. This paper proposes an events-based technique to detect Trojan horses in mobile devices. For example, we have realized if a user send a SMS, MMS, Email, etc. he needs to press a bottom of his device. However, if a Trojan horse sends the same file, no button is pressed. Furthermore, the messaging rate of a normal user is 0-10.07 messages/hour [16] but, usually when a Trojan horse does his own propagation, it sends a lot of messages at the same time.

We monitor the messages transmitting of the device in real time, to identify the applications which the messages are transmitted. When a program sends various messages in a continuously way and without previous events (like a bottom click) our analyzing procedure starts. We study every process to determinate the path of the program that creates it. If any program is found in an unauthorized path, it can be suspected to be a Trojan horse with very high possibility, otherwise, it is not. In Windows Mobile, the operating system limits the number of processes running to 32, hence the scan is fast. In this way, we can detect known and new Trojan horses with more accuracy.

3 Experimentation

There are some programs like Windows Mobile Pro-X FlexiSPY which performs mobile espionage. This application allows you to control all sent and received SMS messages and all call records and their duration, listen to telephone conversations, remote control software functions using SMS, or download directly into the device without a PC or cables. Moreover, if the device has a GPS function, it can be used as a crawler to get the coordinates to locate the device. It works with all versions of Windows Mobile 2003, except with Pocket PC, and it costs approximately 350 US$.

We have carried out simulations of a Trojan horse infections. This software allows a malicious user to steal all the contacts information. Firstly, we describe the Trojan horse implementation and the infection consequences. Secondly, we apply our solution to detect the Trojan horse and delete it.

3.1 Experiment on Detecting the Trojan-SMS.Win-CE.Sejweek Trojan

Windows Mobile 6 Professional Software Development Kit was installed on the Microsoft Visual Studio 2008 programming environment to implement the Trojan-SMS.Win-CE.Sejweek prototype. The attacked user will be simulated by the Visual Studio Simulator. When the simulation begins, the Trojan horse starts to run in the background while the user only sees a picture in his PDA. Moreover, the Microsoft Tool Cellular Emulator v1.43 was used to enable the malicious user to send and receive MMS and SMS messages and make calls (including other services) to the Visual Studio's emulator. Thus, the simulation of information exchange through MMS or SMS between mobile devices has been done in an efficient manner without using the services of any company. The main objective of this Trojan is to obtain

private data from the attacked phone. In this application that information will consist of the contacts' names and telephone numbers.

The main Windows Mobile 6 SDK classes used for the demonstration were:

OutlookSession. It allows, among other functions, to access and modify data in the Contacts. In this case, it uses the nickname and the phone number.
MessageInterceptor. Personalized message receiver. It implements the channel that allows the infection to remain pending for an incoming SMS. The purpose of the MessageInterceptor class will be a key factor in the implementation of the Trojan horse and the proposed solution (as we shall see below), because it contains the event which receives SMS messages (*MessageReceived()*).
SmsMessage. Implements the creation and sending of SMS.
MmsMessage. Implements the creation and sending of MMS.

Information flow. The operation on the user's attacked device is shown in the next code:

```
image_download();
trojan_horse_installation();
execution_in_background();
if (sms_received ())
      if(is_malicious_user(SMS_received))
            while(is_all_information_contacts_sent())
                  SMS_send(information_contacts,
                        SMS_received.telephone_number);
            end_while
            contact = contacts.first();
            while(contacts.hasNext())
                  contact = contacts.next();
                  MMS_send(trojan_CAB, contact_number);
        end_while
      else
            send_SMS_to_inbox();
      end_if
end_if
```

1. The attacked user receives the picture where the Trojan horse infection is packed. The file can be transferred from the Internet through MMS or via Bluetooth to the terminal.
2. Once the virus reaches the mobile phone, it automatically installs itself.
3. The malicious program awaits orders. The attacker's instructions are introduced by means of a SMS with a default structure.
4. If the received SMS is in the correct format, in this case with the head @spy@, the content processing begins. Otherwise, the message goes to the user's inbox.
5. Then, the Trojan horse checks the label-value pairs. The parser recognizes the pairs <sms>telephone_number.
6. The program automatically sends the Contacts data in the Phonebook to each phone <sms>telephone_number pair which appears in the SMS received (stage 3). In

the demo, it was sent via SMS to each contact name and phone number with the format contact_name:telephone_number, but different data could also be sent.

7. The Trojan program, in CAB format (Windows native compressed archive format) is sent by MMS (another ways could be Bluetooth, WIFI or Email) to all contacts for the propagation of this malicious software. It is necessary a new MMS for each contact telephone number because the size of this file is 52KB.

8. Once every all SMSs and MMSs have been sent, the Trojan horse infection awaits for new orders.

Now, the Trojan horse is installed in the victim's mobile phone. In turn, the Trojan horse behavior on the attacker device is described in the next code:

```
send_SMS();
while(no_response_received())
      if(SMS_received())
            processing_contacts_information();
      end_if
end_while
```

1. The malicious user sends an order to the Trojan horse by a SMS to the user under attack in an appropriate format, hich in the test application is

@spy@<sms>telephone_number<sms>telephone_number<sms>, etc.

2. The malicious user awaits the response of the Trojan horse.

3. The malicious user begins to receive SMS messages to the structure contact_name:telephone_number-contact_name:telephone_number-, etc the messages are processed through a second parser in which the malicious user decides how to process information. The process is completed when the user decides to send an SMS with new orders that lead to begin the attack process.

Obviously, the whole process takes place without the user's awareness of the attack because the Trojan horse remains running in the background. In addition, messages sent from the device to the attacked phone do not arrive to the mailboxes, so they do not arouse suspicion. Furthermore, sending SMS and MMS messages entails an economic cost. For a possible estimation, we will assume that the average length of the contact's name or nickname is 10 characters approximately and we know that mobile phone numbers consist of nine characters. If we add the spaces, a contact's information consumes twelve characters. An average user may have 100 contacts stored in his phone book and, in Spain, a SMS costs 0.15€ on average. Therefore, when a malicious user requests data from the contacts in the agenda of the attacked device it will cost 8.62€ approximately and the malicious user can begin this process when he wants. Finally, we have to add the cost of the Trojan propagation. In Spain, one MMS costs 1€ approximately, hence the infection consequences could cost about 108.62€.

Solution and Detection. We have used our proposed solution to detect this Trojan horse application. We have applied the event listeners (for SMS, MMS, GPS, Bluetooth, etc.) provided by Windows Mobile SDK to detect Trojan horses. In the case of our demonstration, the MessageReceived() event is used to receive SMS or a MMS. This event is implemented by the MessageInterceptor() method to detect any

access via SMS or MMS that occurs in the system. Furthermore, the OnClick() event helps us to know the user interaction with the device. When the program sends various messages without previous click events and when the difference between two sent mes-sages are less than 3 seconds the analyzing procedure starts. We access to the process manager to scan all the process (in Windows Mobile, the number of processes running is limited to 32). Previously, we have saved in a text file the information data of authorized programs to check if the program file is a possible Trojan horse or not. When a program does not appear in the known programs list, a message is showed to ask for user confirmation to delete the program. If the user wants to delete the application, the solution finds the program path (through its process) and deletes the Trojan horse. In the next code the whole process is described:

```
if(message_sent())
      if( (lastEvent != onClick()) and
        (time_between_last_message() < 3))
            for p: processes
                  if (not (process_list.contains(p)))
                                          trojan_alert();
                        if(delete_confirmation_recieved())
                              start_scan();
                        end_if
                  end_if
            end_for
      end_if
end_if
```

Although in this demonstration we use SMS and MMS malware to demonstrate the defense, our approach is equally effective in combating malware propagating through Email, Bluetooth and Wi-Fi.

Simulation results. Experiments are carried out to test our method in a Windows Mobile 6 system installed in a HTC 3300 PDA. It follows from the tests that when the device is attacked (after the Trojan horse was installed) it is not aware of the entry or exit of information through MMS or SMS. Moreover, the device does not save copies of those MMS and SMS messages in inbox or outbox. However, when the user receives a message with the malicious Trojan horse the screen light turns on but still nothing happens. We used several Antivirus Mobile Programs such as AirScanner 3.0 and BullGuard 2.0. It shows that all of them do not detect our Trojan horse implementation. However, our solution can detect the Trojan horse, and after the user confirmation, the infection is deleted.

Symbian variation. This Trojan has been successfully implemented for Symbian SO using a Nokia 6120 Classic. We have used using Net60 library [17]. This library permits the automatic translation from C# to C++. Hence, we have reused the Windows Mobile experiment code. However, Symbian SO does not contain any event listener for SMS or MMS. The event handling functions have been simulated using sockets, SMS and MMS protocols and default ports.

3.2 Experiment on Detecting Neo-Call Spysoftware

Neo-Call [18] is a spyware program that runs on most of Symbian handhelds. Neo-Call software has been installed in a Nokia 6120 Classic. Neo-Call conducts eave-dropping, call interception, GPS tracking, etc. It monitors phone calls and SMS text messages. The malicious receives the spy information through SMS and he can control spy actions sending SMS to the attacked phone using the Neo-Control tool. This tool creates and sends formatted SMS automatically with specific codes and tags. We have tested the SMS and Bluetooth misbehaviors. In both situations, the connections to send and receive information through SMS are established through sockets. In our method, the system can control the port range for SMS protocol (KSMSDatagramProtocol). Therefore, using the Graphical User module, in all tests the user was informed when Neo-call tried to send/receive SMS to/from the malicious user.

3.3 Experiment on Detecting Cabir

Cabir is the first network worm capable of spreading via Bluetooth; it infects mobile phones which run Symbian OS. It searches nearby Bluetooth equipments and then transfers a sis file to them once found. In 2008, the Cabir code was published in the web [19]. Like other Symbian malware, Cabir use a socket with Bluetooth RFCOMM protocol and 0x00000009 port. Usually, port 0 is the normal port used to transfer files with the UI in Symbian. However, Cabir uses this port to avoid notification. We conduct our experiments to discover when this malware tried to connect with other devices, not when the file was transferred. In our approach, the Connections submodule includes the RFCOMM protocol and a wide port range. Hence, the attacked was informed of the connection attempts through a message in the screen.

4 Conclusions and Future Work

Currently most mobile users do not feel the need to install on their terminals an antivirus program or other software to protect them from potential infections. However, due to the exponential growth of services and capabilities of these devices and the vast amount of information they contain, it is almost indispensable to take any measure against a possible attack. In this article we have discussed the attacks on mobile terminals by Trojan horses. Based on the analysis of the operation mechanism of Trojan horses, an effective method to detect Trojan horse in mobile devices is presented. By this method, we start a scanning process to find infections in the device when a process sends several messages (by SMS, Bluetooth, MMS, etc.) in a short time and without human interaction. By this method, some unknown Trojan horses can be detected while the existing methods cannot do that. Meanwhile, when the Trojan horse is detected the program can be deleted because using the path. This approach has been implemented in Windows Mobile 6 and Symbian systems and it is shown effective using real infections.

As future work, we plan to study the behavior of other vulnerabilities such as worms or viruses that they could affect the operational ability of the device (software or hardware).

Acknowledgments. This research is partially supported by the project of the Ministry of Science and Innovation ARTEMISA (TIN2009-14378-C02-01).

References

1. A-Squared Anti-Trojan Software, http://www.anti.trojan.net/en/
2. Crapanzano, J.: Deconstructing SubSeven, the Trojan horse of choice, http://www.sans.org/rr/toppapers/subseven.php
3. Urra, O., Ilarri, S., Mena, E.: Testing Mobile Agent Platforms Over the Air. In: Proceedings of ICDE Workshop (2008)
4. Naghsh, A.R., Nilchi, A.R.: Evaluation of Security and Fault Tolerance in Mobile Agents. In: Wireless and Optical Communications Networks (2008)
5. Loebbecke, C., Huyskens, C.: Adoption of Mobile TV Services Among Early Users: Convergence of Familiar Technologies and Emergence of Technology Induced Paradoxes. In: 7th International Conference on Mobile Business (2008)
6. Schatz, R., Egger, S.: Social Interaction Features for Mobile TV Services. In: Proceedings of ITI Conference (2007)
7. Zhang, Q.: Mobile Payment in Mobile E-commerce. In: Proceedings of the 7th World Congress on Intelligent Control and Automation (2008)
8. Carlsson, C., Walden, P., Yang, F.: Travel MoCo - A Mobile Community Service for Tourists. In: 7th International Conference on Mobile Business (2008)
9. Moura, M.: Mobile learning: teaching and learning with mobile phones and Podcasts. In: 8th IEEE International Conference on Advanced Learning Technologies (2008)
10. Lauri, A.: Bluetooth and WAP push based location aware mobile advertising system. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (2004)
11. Salo, J., Tähtinen, J.: Retailer Use of Permission-Based Mobile Advertising. In: Proceedings of Advances in Elec-tronic Marketing. Idea Publishing Group, USA (2005)
12. Sandu, F., Romanca, M.: Remote and Mobile Control in Domotics. In: Proceedings of Optimization of Electrical and Electronic Equipment (2008)
13. Fernández-Montes González, A., Álvarez García, J.A., Juan Antonio Ortega Ramirez J.A., Martínez N., Seepold R.: An Orientation Service for Dependent People Based on an Open Service Architecture. In: Lecture Notes in Computer Science. Springer, Heidelberg (2007)
14. http://www.flexibilis.com
15. http://www.airscanner.com
16. Xie, L., Zhang, X., Chaugule, A., Jaeger, T., Zhu, S.: Designing System-level Defenses against Cellphone Malware. In: Proceedings of 28th IEEE International Symposium on Reliable Distributed Systems (2009)
17. RedFiveLabs, http://www.redfivelabs.com
18. Neo-Call, http://www.neo-call.com
19. Cabir code, http://www.offensivecomputing.net/?q=node/773