Integrated Software Platform for Mobile Malware Analysis – A Potential Vision

George Suciu[1,3(✉)], Laurentiu Bezdedeanu[1], Cristiana Istrate[1,3], Mari-Anais Sachian[1], Houssam Boukoulo[2], Corentin Boscher[2], Fabien Souleyreau[2], and Eduard-Cristian Popovici[3]

[1] R&D Department, Beia Consult International, Peroni 16, 041386 Bucharest, Romania
{george, laurentiu.bezdedeanu, cristiana.istrate, proiecte}@beia.ro
[2] Institut National Polytechnique de Bordeaux, Avenue du Dr Albert Schweitzer 1, 33400 Talence, France
[3] ETTI Faculty, University POLITEHNICA of Bucharest, Bucharest, Romania
eduard.popovici@upb.ro

Abstract. With the evolution of technology, we are witnessing the development of mobile terminals that are getting closer to a personal computer in terms of features and applications. At the same time, there is an increase in the number of mobile device users, which also leads to an increase in the use of online shopping or finance management applications. Hence, mobile terminals become a target for cyber criminals. Starting from the analysis of the current situation in the world regarding cyber security technology and solutions, we aim to build an integrating software platform for mobile malware analysis. The aim of the research is to develop a software platform that integrates the malware analysis procedures for most of the existing mobile terminals. So, the main objective of this article is to analyze the quality of cyber-protective solutions for mobile devices. We present our experiments which may bring solutions for some of the major vulnerabilities like active development of mobile malware, and hacking. Moreover, in this article we discuss the issue of security on Android, making use of security platform like Kali which permits to use different kinds of security analysis programs.

Keywords: Malware · Cyber security · Integrated platform · Mobile malware · Kali

1 Introduction

Going forth in the last years, the threat to mobile phones has risen as many people are using them on a daily basis for all kind of needs. Malware is the most generic term for different kind of threats. Those threats include trojan horses, keyloggers, rootkits, viruses and worms and cybercrime [1].

To examine a malware, dynamic analysis [2] is usually used, which is a set of techniques for analyzing an application or software running in a controlled and

V. Poulkov (Ed.): FABULOUS 2019, LNICST 283, pp. 431–436, 2019.
https://doi.org/10.1007/978-3-030-23976-3_38

monitored environment. The idea is to observe the malware actions so we can draw conclusions about how it behaves.

An easier way to analyze the malware within the mobile device would be an integrated platform for dynamic and static analysis [3]. Mainly speaking, an integrated platform focuses on the system integration of more gentle methods used for analysis or other acts. Security is important for data protection but critical for mobile banking and intellectual property. As the mobile phones are used everyday by a lot of people and they are also connected to a lot of personal information, there is a concern to make a change in order to avoid leaks of personal information [4]. We propose a hybrid approach by developing a new software platform that enables the analysis of malware on mobile phones. The main objective is to identify the operational requirements and the capacity needed to develop this platform through a secure way. The solution that would prevent malware attacks will be presented through some tries that we made and how our integrated software platform should function and look like.

The paper is structured as follows: Sect. 2 describes related work. Section 3 shows the methodology and types of mobile threats. Section 4 analyzes research experiments and results. Finally, Sect. 5 presents the conclusions and forthcoming plans.

2 Related Work

A complete traditional anti-malware/anti-spyware software for mobile devices proactively checks applications and files for malware and viruses, scans the built-in memory and SD card, finds Potentially Unwanted Programs (PUPs) for removal and scans automatically the apps and files when accessed [5]. It also needs to automate the processes like scheduling automatic scans, updating the protection database automatically, updating over a WiFi network. It should have a privacy manager who identifies every application's access privileges in detail and breaks down access privileges by category: Contacts, Identity Information, Simple Message Service (SMS), and Security Settings. The application manager has to identify which applications are currently running, to notice installed applications and to enable custom whitelisting of approved apps. For the security audit, it identifies security vulnerabilities on the device and suggests remediation.

In [6] was proposed a system architecture for automated Android malware analysis, which is able to make predictions regarding malicious applications. The system architecture, which has a core server that runs AndroSandX server applications. For avoiding backdoor exploitation, the system has a peripheral firewall to restrict inbound connection. This system allows static, dynamic and hybrid analysis approach.

3 Methodology for Assessing Mobile Attacks

In this section we present a short classification of mobile threats and attacks, according to the NIST technology areas [7] where attacks can be exploited. Table 1 presents the assessment of attacks on hypervisors and virtualization environments running on mobile devices.

Table 1. Examples of mobile attacks

Hypervisors systems	– Hyperjacking attack – Installing system firmware rootkits – Attacking hypervisors through system firmware – Exploitation of privileged interfaces provided by the hypervisor	– Attacking hypervisor emulation of hardware devices – Management of VMs – VM sprawl – Denial of service
Virtualization environments	– VM Escape – Cache covert channel attack – VM breaking the isolation – IP or MAC address spoofing – VLAN hopping	– Traffic snooping – Resource starvation – Secure privileged/administrative guests – Unsecured VM migration

4 Experiments and Results

In this section we analyze the mobile malware experiments using Kali [8], a Linux distribution which is used for penetration, testing, and security auditing. It has been adhering completely to Debian development standards and includes many tools which are used in data security. Kali Linux provides already installed and updated penetration testing, security tools, frameworks and their updated repositories, all pre-compiled set of word-lists that is needed during penetration testing (to brute-force logins) and command line interface.

Within the experimentation work, we installed Kali Linux on the Nexus 5 smart phone and on a computer as a Virtual Machine. We used VirtualBox with the ISO that has been downloaded from the official website of Kali. Then we tried with the package air cracking, to hack the password of a WiFi network. At the end of it, we tried to find, thanks to various dictionaries, the password that was set. If the WiFi is using a WEP encryption with common words, the algorithm searches on a database of known words. Else, if it is a complicated password it would take many hours or even days, but it depends on the algorithm that it is used. For example, it can try all the French words ordered alphabetically or all the numbers starting from 0, etc. Depending on the performance of the computer it would take more or less time.

4.1 Hardware and Software Used for Installation of Kali Linux

We will describe further the installation process and the experiments made with Kali Linux to hack the WIFI password. We used the phone LG NEXUS 5X version N4F261 for installing Kali Linux Nethunter 2017. Specification of the phone is OS Android 7.1.1, Qualcomm MSM8992 Snapdragon 808 processor, Hexa-core CPU (4 × 1.4 GHz Cortex-A53 & 2 × 1.8 GHz Cortex-A57), 32 GB internal memory, 2 GB RAM, M8994F-2.6.36.2.20 network, Wi-Fi 802.11 a/b/g/n/ac, dual-band.

We used VirtualBox version 5.1.24 for the installation of Kali and it has the following performance parameters: 2048 MB of base memory, 2 processors at 2 GHz.

To experiment what a malware is, we used Kali Linux which integrates tools to create a malware and to get back, for example, some log in and password thanks to a keylogger that we can send to a pool of a computer. Firstly we created the malware with the command line presented in Fig. 1.

```
root@kali:/home/kali# msfvenom -p windows/meterpreter/reverse_tcp LHOST=172.23.5
6.199 LPORT=8080 -f exe -e x86/bikata_ga_nai -i 12 -o Bureau/malware.exe
```

Fig. 1. Creation of the malware.

Then we configured the host and the port that we are going to capture using the following command lines:

```
set payload windows/meterpreter/revers_tcp
set LHOST 172.23.56.199
set LPORT 8080
exploit -j
```

Furthermore, we launched the malware on the target computer and then we connected it to the session on Kali Linux with the following command line:

```
session -i
```

Finally, we connected to the session on Kali Linux with the previous command line. The final result is a malware which has the following repository:/home/kali and takes screenshots from the user desktop (as presented in Fig. 2).

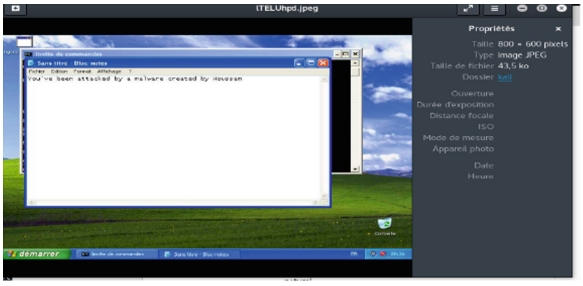

Fig. 2. Screenshot of the target computer.

4.2 Hacking a WiFi

The first attempt at securing these access points was termed Wired Equivalent Privacy (WEP). This encryption method has been around for quite a while and a number of weaknesses have been discovered. It has been largely replaced by WPA (Wi-Fi Protected Access) and WPA2.

First, the motivation to hack someone's Wi-Fi router or access point (AP) was to navigate around the web anonymously, or more precisely, with someone else's IP address. Second, once the Wi-Fi router is hacked, the victim's traffic can be decrypted using a sniffing tool to capture and spy on all their traffic. Third, if their AP is compromised, it can be used as a node in the dark-net to share large files over torrent protocol, using someone else's bandwidth, rather than the attackers own, or as zombie node in malware attacks.

After installing Kali Linux into a Virtual Machine, the tool air-crack-ng was used. In order to hack the WEP Wi-Fi we used a network wireless USB card Gigaset 108 and the following command:

```
Airodump-ng -c channel -w (file name) –bssid BSSID
wlan0
```

The next step is to wait for data to be present from other legitimate stations and open another terminal, as presented in Fig. 3.

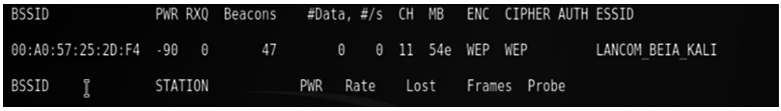

Fig. 3. Target SSID for attack.

The above command will start capturing packets from the SSID "LANCOM_-BEIA_KALI" on channel 11 and write them to file WEP crack. This command will now to capture packets in order to crack the WEP key. To do that, we will need to inject packets into the AP.

Further step is to wait for someone to connect to the AP so that we can get the MAC (Media Access Control) address from their network card. After their MAC address is captured, it can be spoofed and injected packets into their AP. To do this, the aireplay-ng command can be used. Needed are the BSSID of the AP and the MAC address of the client who connected to the AP.

The attack works by capturing an ARP packet and then replaying that ARP thousands of times in order to generate the IVs that we need for cracking the WEP, which on our setup takes approximately 1 min.

After that, the ARPs are injected into the AP, and the packets are captured in order to generate the needed airodump file WEPcrack.

If enough data is present, aircrack-ng will display the key on the platform screen, as presented in Fig. 4.

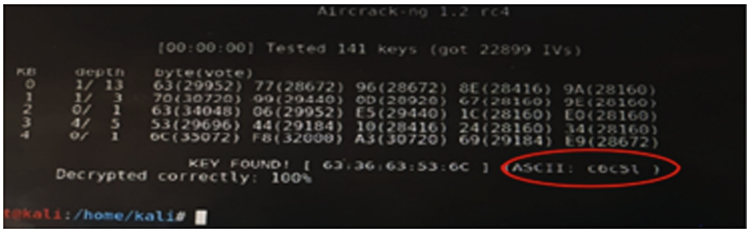

Fig. 4. The found key

5 Conclusions

In this paper, we presented related work regarding integrated software platforms for mobile malware analysis and detailed experiments we made on Kali Linux, in order to gain a better view over the platform functionalities. We demonstrated how to analyze the malware developing tools and succeeded to create a malware using Kali. Also, we demonstrated how Kali Linux can be used to hack a Wi-Fi password. As future work, we will investigate how to integrate other open source forensics tools.

Acknowledgments. This work was supported by a grant of the Ministry of Innovation and Research, UEFISCDI, project number 5 Sol/2017 and ODSI within PNCDI III.

References

1. Cristodorescu, M., Jha, S., Sanjit, A., Song, S.D., Bryant, R.E.: Semantics-aware malware detection, Carnegie Mellon University Research Showcase @CMU, p. 2 (2005)
2. Blasing, T., Batyuk, L., Schmidt, A.D., Camtepe, S.A., Albayrak, S.: An Andriod application sandbox system for suspicious software detection. In: 5th International Conference on Malicious and Unwanted Software, pp. 55–56 (2010)
3. Islam, R., Tian, R., Batten, L.M., Versteeg, S.: Classification of malware based on integrated static and dynamic features. J. Netw. Comput. Appl. **36**(2), 646–656 (2013)
4. Malhotra, A., Bajaj, K.: A survey on various malware detection techniques on mobile platform. Int. J. Comput. Appl. **139**(5), 15–20 (2016)
5. Simmonds, M.: How businesses can navigate the growing tide of ransomware attacks. Comput. Fraud. Secur. **3**, 9–12 (2017)
6. Jadhav, S.: An assistive system for android malware analysis to increase malware analysis efficiency. In: 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 370–374 (2017)
7. Chapman, C.: Network Performance and Security: Testing and Analyzing Using Open Source and Low-cost Tools. Syngress (2016)
8. Johansen, G., Allen, L., Heriyanto, T., Ali, S.: Kali Linux 2–Assuring Security by Penetration Testing. Packt Publishing Ltd. (2016)