ISSN 2278-3091
Volume 8, No.3, May - June 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse22832019.pdf
https://doi.org/10.30534/ijatcse/2019/22832019

Armoring Client and Servers Running on Linux Based Android Platform

[1] G. Rama Koteswara Rao, [2] R. Satya Prasad
[1]Department of Information Technology, V R Siddhartha Engineering College, Vijayawada, AP, India
[2]Department of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, A.P, India

ABSTRACT

Android is an operating system for present-day cell phones and depends on the Linux part. The base of the Android working framework-programming stack contains Linux Kernel, so the Android Operating System is developed over the Linux Kernel. In a versatile market, it is a mainstream working framework. Because of exponential increment in the use of cell phones by the client's step by step, the Android cell phone has been contacted by the most extreme level of the clients on the planet. There is a need to concentrate on research on the Android framework to give security to Android applications, information, the gadget, and the system. This paper concentrates on researching the different genuine security issues associated with the Android gadgets; anticipating these issues, strategies are proposed to alleviate the security hazards on the Android gadgets since the clients are putting away crucial data in their cell phones.

Key words: Android, malicious Apps, Metasploit, SMS permissions, Web View

1. INTRODUCTION

The equipment of the Android cell phones is associated by the Linux Kernel. The Linux Kernel is responsible for a portion of assets, overseeing memory, memory store, distribution and de-allotment of memory for the record framework, booking of the procedure and systems administration. The Android application conveys an incentive to the clients of the cell phone by utilizing a tremendous measure of equipment and programming assets for innovation [7]. The surveyors of the University of Cambridge and Zimperium labs researched that around 90% of keen gadgets of Android were influenced by unsafe vulnerabilities and furthermore hacked through SMS messages. In Android telephones, there is aprospect of mounting malware and Android Apps created by the outsider through social engineering assaults by the assailant. Android displays an open-source-stage in addition to sharpness condition for portable contraptions and all Android applications keep running on the Application Sandbox. Regularly the restricted scope of the framework assets is accessed by the Android applications [3]. The central running device is for the most part dependent on the Linux kerne[4]l. Normally, Android applications get written in Java programming language and continue running in the Dalvik virtual machine. In any case, applications are competent to be written in the usual code. Applications are mounted from a lone record with the .Apk record expansion **.**The application assets are perceived and isolated on the Android stage by using the Linux client based assurance. The upside of the Android working framework is that it permits the Android application to keep running with the individual procedure with extraordinary client id, which is not quite the same as other operating frameworks[13]. The Android advancement group made an examination on different security issues and vulnerabilities on other cell phones, work areas and server stages and saw that there is a need to give solid security show in the Android stage to empower a solid ecosystem of utilization. .

2. PROPOSED WORK

This paper centres around recognizing the vulnerabilities on the Android working framework by infusing the noxious App on the Android cell phone through SMS, email, and so forth and to test the effect of the helplessness on the gadget[5]. All things considered, the Android working framework is in charge of recognizing the vulnerabilities in the gadget and ought not to permit introducing the vindictive App on the gadget. This paper centres around the accompanying errands(i) Attacking and anticipation stage on the Android cell phone qnd(ii) Attacking and anticipation stage on the Android cell phone utilizing Metasploit.

2.1. Assaulting and anticipation stage on the Android cell phone.

2.1.1 Attacking Phase: The malignant Apps can be created by the assailant which can perform noxious exercises on the gadgets, for example, perusing the log record, propelling DoS assaults, perusing SMS messages, perusing exhibition data, dealing with the whole cell phone, taking or change of information, and so on. In assaulting stage the aggressor sends the pernicious App to the injured individual which can peruse the log record of the person in question and can send it to assailant's telephone by means of SMS/mail when it was actuated. The activities that can perform in this stage are as per the following(i). Making a malignant App to peruse the log record and exhibition data from the cell phone [10]. Vindictive App shows the phoney promotion demonstrating the unique offers offered by Amazon with high limits.(ii). The noxious .apk record is infused into the unfortunate casualty gadget through SMS, WhatsApp message or email. (iii). the alluring message like limits offered on the items or message identified with welcome is shown to the client at whatever point the App is opened. (iv). at the point when a typical client taps on the connection, the malevolent application covered up inside the connection will be consequently be introduced on the injured individual gadget. (v). The App will run naturally when it is actuated or it can keep running out of sight. When the App is actuated, it peruses the log record from the compromised individual's gadget and advances it to the aggressor through an SMS message or email.

Steps included during the assaulting stage:

(i). at the back end, the malignant App gets the information of the call log data from the unfortunate casualty gadget utilizing the Android Call Log Content Provider Managed inquiry [9]. This strategy restores the outcome set with lines and sections way which contains various fields like telephone number, call type, call date, and contact span. (ii). Call code contains three kinds of calls, for example, approaching, active and missed calls. The calling code containing messages is assembled into three kinds utilizing the strategies present in the cursor class. (iii). the fields containing the call data are added to the StringBuffer variable. Next, the information in the current StringBuffer variable is changed over into string information type and the string message containing the injured individual call data is transmitted to the assailant through an SMS message or email utilizing the strategies in the SmsManagerclass.

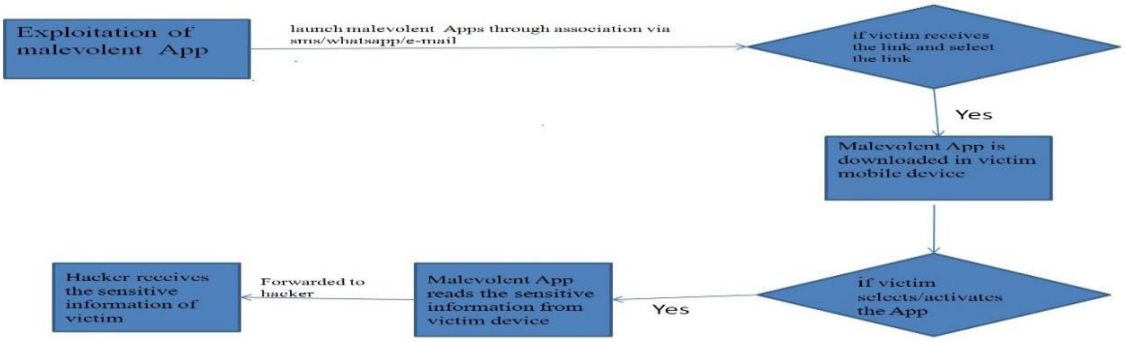

Exploitation of malevolent App
launch malevolent Apps through association via sms/whatsapp/e-mail
if victim receives the link and select the link
Yes
Malevolent App is downloaded in victim mobile device
if victim selects/activates the App
Yes
Malevolent App reads the sensitive information from victim device
Forwarded to hacker
Hacker receives the sensitive information of victim

Figure 1: Flow chart of assaulting stage

The App has been created to contain the malevolent code to peruse the log record and store it in a document[2]. Each column containing the following data, for example, telephone number, sort of call, and length of the call with information are changed over into string information type to advance it to the programmer's telephone through SMS or email as exemplified in the calculation underneath in figure[1].

Table 1: Algorithm to get critical data from a cell phone

```
Begin
Read the call log and store it in a file
                    Read first row in a file
                    while(!eof) do
                    begin
                              Extract phonenumber, typeofcall,date,callduration from the row
                              switch(typeofcall)
                               begin
                                        case : OUTGOING
                                                  calltype = "Outgoing Call"
                                                  break
                                         case : INCOMING
                                                  calltype = "Incoming Call"
                                                  break
                                        case : MISSED
                                                  calltype = "Missed Call"
                                                  break
                              enddo
                    Append phonenumber, calltype,date,callduration tostringbuffer
                    enddo
          Forward the stringbuffer to the attacker's smartphone via sms or email.

end.
```

2.1.2 Prevention of malignant App

These days, every single individual uses an advanced mobile phone and there is a fast increment of utilizing the applications in cell phones. Regularly, these PDAs are in peril in light of some pernicious applications that are in a roundabout way introduced through malignant connections on our cell phones [5]. The issue with the Android gadget is that it enables us to download the Apps from outside other than Google play store and there will be odds of downloading the malevolent Apps created by the evil programmers. Indeed, even these malignant applications take a portion of our private information like call data [8]. There is a requirement for research to keep from taking the critical data on the cell phone[14]. This paper centresaround executing applications to verify the Android gadget from taking the crucial data. This paper centresaround following errandsScanning the gadget to inspect for malignant or tainted App, Examining the running application in the gadget ,Examine the Apps introduced on the gadget ,Identifying and uninstalling the App that peruses the crucial data from the gadget and Identifying and uninstalling the App that utilizes illicit authorizations

This paper centres around recognizing the consents on the Android App. There are two sorts of authorizations typical and risky. Programmers utilize hazardous consents on the App to take private data or to perform pernicious exercises on the injured individual gadget[1]. This paper centres around distinguishing the unsafe authorizations on the App and uninstalling the App. At the point when the App contains the perilous authorizations, the Android working framework will alarm the client with warning messages. The malevolent exercises on App can be seen by looking at the consent on the App[12]. The accompanying activity will be framed by the aggressor utilizing the unsafe authorizations. Stealing the secret phrase data and important log information, Reading the contact data and perusing/composing framework settings, Examining the active calls, Using the unfortunate casualty camera and following the area of the person in question, Sending SMS messages to aggressor telephone from injured individual gadget, Tracking the area of the unfortunate casualty gadget andExamining or taking the display data of the incapacitated individual gadge.t

The App has been created to recognize the noxious and uninstall the malevolent App containing the hazardous authorizations as exemplified in the calculation underneath .

Table 2: Algorithm to uninstall malevolent Apps

```
Begin
Check the Android version running on the device.
if( Android version is less than 6.0) then
send an SMS message to the user
endif
Check if permissions are granted to App
if(permission granted) then
    Check the App is requesting for dangerous permission
    if (the permissions are sendsms||readsms||receivesms||readexternalstorage||
    writeexternalstorage|| writecalllog||getaccounts||processoutgoingcalls||receivemms)then
            if(above any of the permissions are granted) then
                send notification message tovictim
                uninstall the App
            endif
        endif
endif
end
```

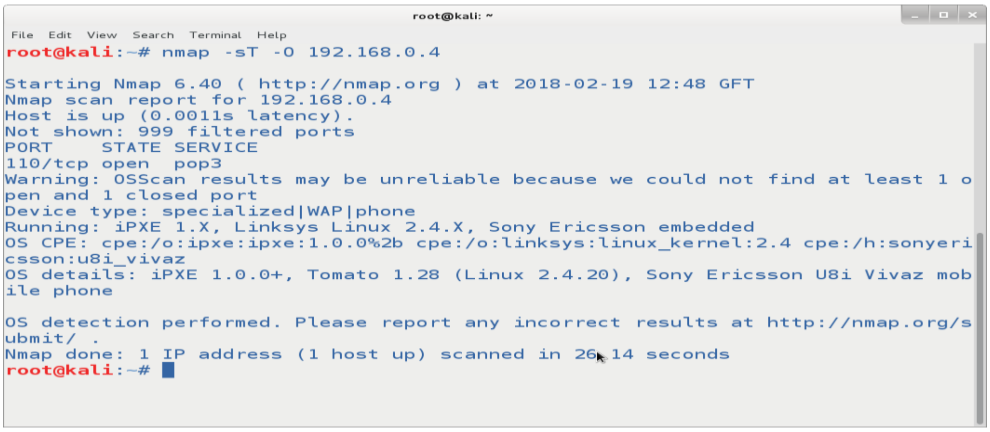
2.2 Attack on Android telephone utilizing Metasploit

The dangers on the Android cell phone are expanding step by step as utilization of the telephones is expanding [11]. Today the few assaults are performing on the Android working framework. The malware can be made in App utilizing Metasploit device and once the App is introduced on the gadget makes an indirect access on the gadget to access to gadget on a similar system which encourages the assailant to perform following activities remotely on injured individual gadget, for example, phishing, banking-trojans, sending counterfeit SMS, picking up control on the unfortunate casualty gadget camera to take pictures, and so forth. The means associated with playing out the assault utilizing Metasploit instrument is

Step1: Retrieving the data of the Android gadget utilizing Operating System Fingerprinting instruments as exemplified in the figure underneath:

Step2: Build up endeavour apparatus in Kali Linux

Step3: Check the injured individual Android gadget is helpless against the endeavor apparatus.
Step4: Build up payload utilizing misuse apparatus.
Step5: Implementing application to the departure the malevolent payload from Intrusion Prevention framework



```
root@kali:~# nmap -sT -O 192.168.0.4

Starting Nmap 6.40 ( http://nmap.org ) at 2018-02-19 12:48 GFT
Nmap scan report for 192.168.0.4
Host is up (0.0011s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE
110/tcp open  pop3
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Device type: specialized|WAP|phone
Running: iPXE 1.X, Linksys Linux 2.4.X, Sony Ericsson embedded
OS CPE: cpe:/o:ipxe:ipxe:1.0.0%2b cpe:/o:linksys:linux_kernel:2.4 cpe:/h:sonyeri
csson:u8i_vivaz
OS details: iPXE 1.0.0+, Tomato 1.28 (Linux 2.4.20), Sony Ericsson U8i Vivaz mob
ile phone

OS detection performed. Please report any incorrect results at http://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.14 seconds
root@kali:~# 
```

Figure 2: Gathering data aboutsmartphone-utilizing port checking

Step6: Implementing the adventure on the unfortunate casualty gadget

2.2.1 Prevention

One method for avoiding the noxious App on the gadget is by inspecting the applications containing the unsafe consents on the gadget physically and erase it physically if hazardous authorizations are seen on the App[13-15]. The application is created to shield from malevolent Apps executed from Metasploit instrument and work process of taking care of vindictive Apps of the Metasploit apparatus is exemplified in figure 3 underneath[16-18].

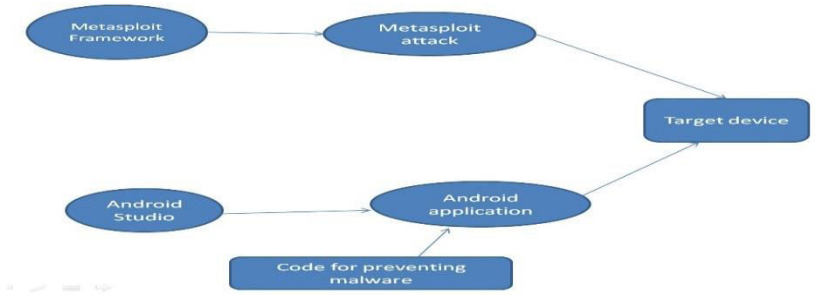
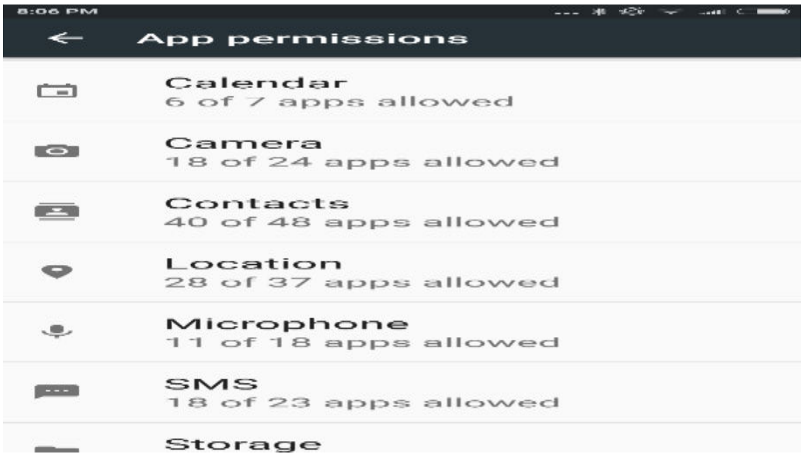


Figure 3: Workflow handing Metasploit attacks

Figure 4 beneath demonstrates a rundown of consents and when the client taps onthe SMS authorizations will show InTheFigure Underneath: the rundown of Apps that contains SMS consents as exemplifie[19-20]



App permissions

Calendar
6 of 7 apps allowed

Camera
18 of 24 apps allowed

Contacts
40 of 48 apps allowed

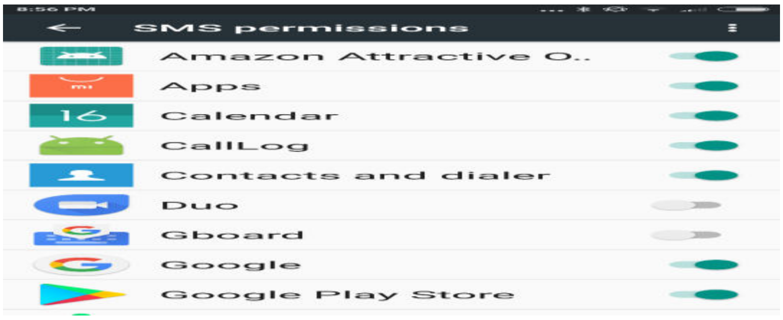
Location
28 of 37 apps allowed

Microphone
11 of 18 apps allowed

SMS
18 of 23 apps allowed

Storage

Figure 4: List of App Permission


8:56 PM

SMS permissions

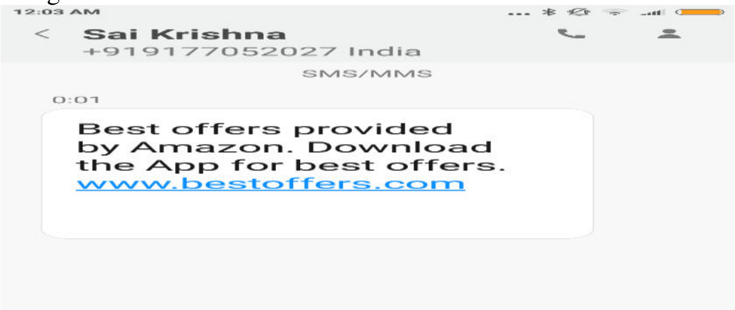
Amazon Attractive O..	●
Apps	●
Calendar	●
CallLog	●
Contacts and dialer	●
Duo	○
Gboard	○
Google	●
Google Play Store	●

Figure 5: List Apps containing SMS permissions

Figure 5 given above demonstrates that the two applications in particular 'call log' and 'Amazon aattractiveoffer' contains the SMS consents. On the off chance that any of these Apps or both are infused by the programmer to steal data from the gadget, the App can abuse the SMS authorizations by sending the unfortunate casualty gadget crucial data through SMS.

3. RESULTS

Theassailant will inserthateful App to theVictimDevicethroughSMS/WhatsApp/Gmail.In the figure 6below indicates that the attacker injects the hateful App through SMS messages


12:03 AM

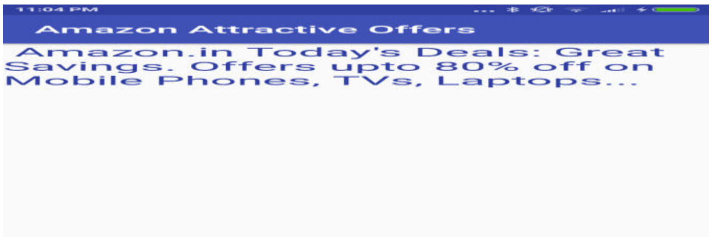
Sai Krishna
+919177052027 India

SMS/MMS

0:01

Best offers provided by Amazon. Download the App for best offers. www.bestoffers.com

Figure 6: SMS message received by the victim


11:04 PM

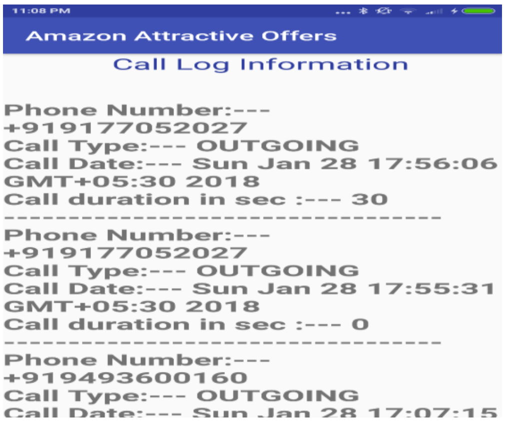
Amazon Attractive Offers

Amazon.in Today's Deals: Great Savings. Offers upto 80% off on Mobile Phones, TVs, Laptops...

Figure 7: Malevolent App is open

When the user clicks on theabove link as shown in Figure 7 the App will be activated and read the sensitiveinformation from the victim such as calllog file as shown in the figure 8below

When the user clicks on theabove link as shown infigure 6 ,the hateful App is inserted on the prey gadget. The hatefulApp shows a link as shown in the figure 7 above.


11:08 PM

Amazon Attractive Offers

Call Log Information

```
Phone Number:---
+919177052027
Call Type:--- OUTGOING
Call Date:--- Sun Jan 28 17:56:06
GMT+05:30 2018
Call duration in sec :--- 30
-----------------------------------
Phone Number:---
+919177052027
Call Type:--- OUTGOING
Call Date:--- Sun Jan 28 17:55:31
GMT+05:30 2018
Call duration in sec :--- 0
-----------------------------------
Phone Number:---
+919493600160
Call Type:--- OUTGOING
Call Date:--- Sun Jan 28 17:07:15
```

Figure 8: Call Log Information

The App recovers the total subtleties of active, approaching and missed call data from the log record. The data will be sent to the aggressor gadgets. This movement will happen when the victimchooses the App or if the App keeps running out of sight and initiated through a specific timeframe. The symptoms for detecting this type of attack islooking at the SMS messages sent on every day surpassing the limit esteem and Looking at Apps with unsafe authorizations running in the background. Three strategies have been produced for taking care of these sorts of assaults by distinguishing introduced Apps, Effected Apps and Running Apps as appeared in the figure 9 beneath The user can examine the list of Apps installed, running and effected Apps at any time. Figure10 below indicates the list of Apps installed on the device.

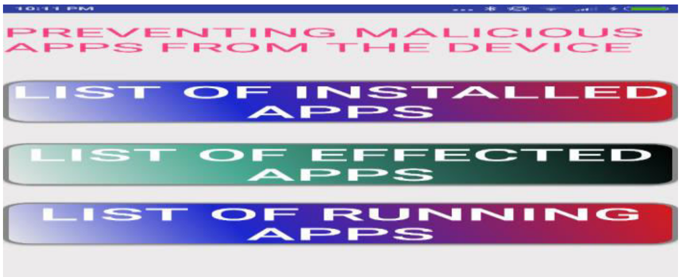
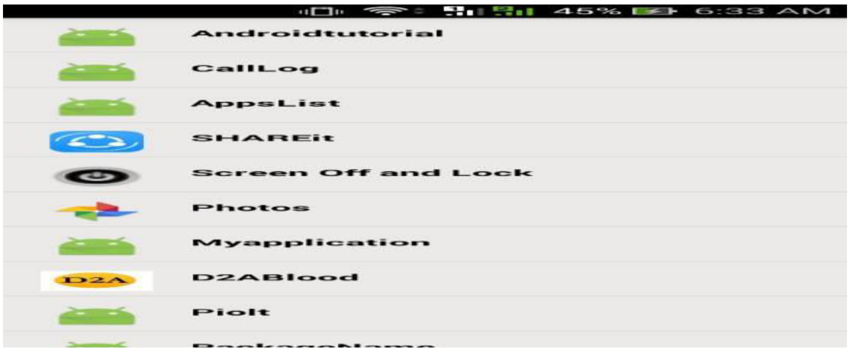
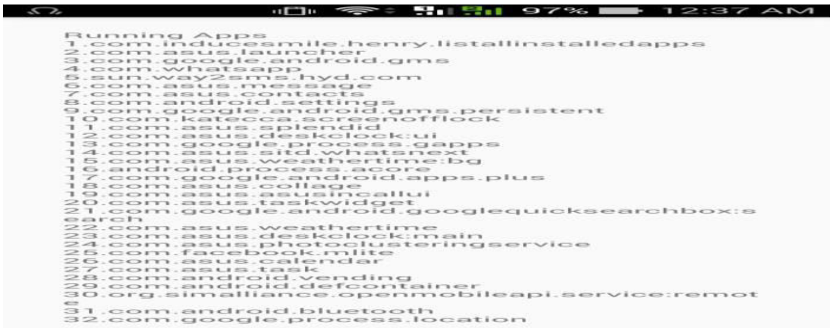


Figure 9: Preventing Phase



Androidtutorial
CallLog
AppsList
SHAREit
Screen Off and Lock
Photos
Myapplication
D2ABlood
Piolt

Figure 10: Lists of Apps installed on the device.



```
Running Apps
1.com.inducesmile.henry.listallinstalledapps
2.com.asus.launcher
3.com.google.android.gms
4.com.whatsapp
5.sun.way2sms.hyd.com
6.com.asus.message
7.com.asus.contacts
8.com.android.settings
9.com.google.android.gms.persistent
10.com.katecca.screenofflock
11.com.asus.splendid
12.com.asus.deskclock:ui
13.com.google.process.gapps
14.com.asus.sitd.whatsnext
15.com.asus.weathertime:bg
16.android.process.acore
17.com.google.android.apps.plus
18.com.asus.collage
19.com.asus.asusincallui
20.com.asus.taskwidget
21.com.google.android.googlequicksearchbox:s
earch
22.com.asus.weathertime
23.com.asus.deskclock:main
24.com.asus.photoclusteringservice
25.com.facebook.mlite
26.com.asus.calendar
27.com.asus.task
28.com.android.vending
29.com.android.defcontainer
30.org.simalliance.openmobileapi.service:remot
e
31.com.android.bluetooth
32.com.google.process.location
```

Figure 11: List of Running App

At the point when the clientselects"List Affected apps" as appeared in the figure 9, the application scrutinizes the gadget and distinguishes suspicious apk record in running applications. On selecting the influenced apps , a button which is utilized to uninstall the influenced appis shown below.

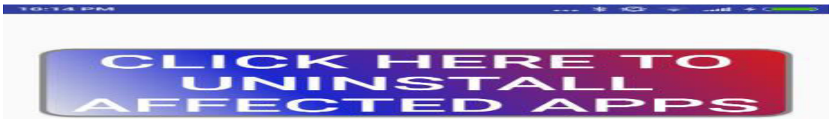
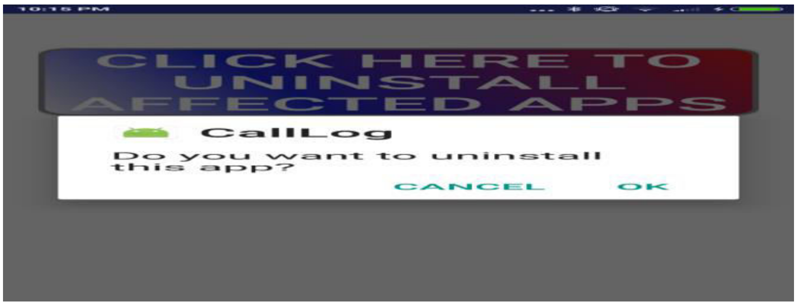


Figure 12: App to Uninstall affected App

At the point when the clienttickon the button"Uninstall Affected Apps" as appeared in figure 12 above, which sweeps to recognize the suspicious apk records running on the gadget. It shows "click here to UNINSTALL" message as appeared in the figure 13 beneath to distinguish suspicious apk record in running applications and to uninstall the suspicious apk documents when the client hits the contaminated app.

When there is an influenced application, it prompts an alarm box i.e., Do you need to uninstall this application? The client can uninstall the influenced app by selecting on "OK" as appeared in the figure 13 underneath. At that point the influenced app will be expelled from the injured individual's gadget.



Do you want to uninstall this app?

Figure 13: Uninstalling effected App

6. CONCLUSION

Contemporary utilization of innovation includes portable Apps with access to individual information being taken off by the specialist organizations and their utilization by means of advanced mobile phones in the hands of shoppers. This clears route for noxious Apps picking up the trust of guiltless portable clients and subsequently imperilling basic individual resources or assets. Averting accidental information stream in cell phones draws in critical intrigue. An App that checks information robbery by malevolent Apps is displayed in the paper.

The assault can be relieved at the source or at the objective since the assault can be performed through advanced mobile phone on the devoted servers. In specific cases, it may not be possible to

alleviate the ambush at the source. This paper also focused on mitigating the routing attacks at the target platform side performed through a smart phone

REFERENCES

[1] Ahmad S. Mashhourand ZakaryaSaleh, "**Community Perception of the Security and Acceptance of Mobile BankingServices in Bahrain: An Empirical Study**", International Journal of Advanced Computer Science and Applications, Vol. 6, No. 9, 2015.

[2] Andre Pereira, Manuel Correia and Pedro Brandao, " **USB Connection Vulnerabilities on Android Smartphones: Default and Vendors' Customizations**", FIP International Conference on Communications and Multimedia Security, Communications and Multimedia Security, pp 19-3,Lecture Notes in Computer Science , LNCS, volume 8735.

[3] BhavyaSareen, Sugandha Sharma andMayankArora, "Mobile **Cloud Computing Security as a Service using Android**", International Journal of Computer Applications, Vol.99, Number 17 ,2014.

[4] Daniel Walnycky, Ibrahim Baggili, Andrew Marrington, Frank Breitinger and Jason Moore, " **Network and device forensic analysis of Android social-messagingapplications**",Digital Investigation14(2015), http://dx.doi.org/10.1016/j.diin.2015.05.009

[5] Dominic Bucerza, CrinaRatiu and Misu-Jan Manolescu, "**SmartSteg: A New Android Based Steganography Application", International Journal of Computers Communications & Controls**", Vol. 8, No. 5, 2013.

[6]Geumhwan Cho, Jungsung Cho, Youngbae Song , Donghyun Choi and Hyoungshick Kim, " **Combating online fraud attacks in mobile-based advertising**", EURASIP Journal on InformationSecurity,https://jiseurasipjournals.springeropen.com/articles/10.1186/s13635-015-0027-7.

[7]IvoFriedberg, KieranMcLaughlin ,Paul Smith, David Laverty andSakirSezer , " **STPASafeSec: Safety and security analysis for cyber-physical systems**", Journal of Information Security and Applications,Vol.34,pp.183-196.

[8]Jainye Liu and Jainkun Yu, "Research **on Development of Android Applications**" , 4th International Conference onIntelligent Networks and Intelligent Systems (ICINIS), 1-3 Nov. 2011, Kunming, China, DOI: 10.1109/ICINIS.2011.40.

[9] Jang Il Kim, Kyung Shin Kim, Ki Won Nam and Yong-Gyu Jung, "**Automated malware analysis service using mobile virtual box**", Journal of Service Science Research, June 2016, Volume 8, Issue 1, pp 73–83.

[10]Jongsu Lim and Jeon Hyun Yi," **Structural analysis of packing schemes for extracting hidden codes in mobile malware**",EURASIP Journal on Wireless Communications and Networking, pp. 1-12, DOI 10.1186/s13638-016-0720-3.

[11]Mahmood Deypir and AbbasHorri, " **Instance based security risk value estimation for Android applications**", Journal of Information Security and Applications, Vol. 40, June2018,pp.20-30, https://doi.org/10.1016/j.jisa.2018.02.002.

[12]NilayYildirim and AsafVarol, |"**Android Based Mobile Application Development for Web Login Authentication Using Fingerprint Recognition Feature**", International Journal of Computer Science and Mobile Computing, Vol. 5, Issue. 10, October 2016, pp.61-68, ISSN 2320–088X.

[13]Sadeque Reza Khan and Farzana Sultana Dristy, "**Android Based Security and Home Automation System**",International Journal of Ambient Systems and Applications (IJASA) Vol.3, No.1, March 2015.

[14]Syed Farhan AlamZaidi ,Munam Ali Shah andMuhammad Kamran, " **A Survey on Security for Smartphone Device**", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 4, 2016.

[15] Vijaykumar, G., A. Gantala, M. S. L. Gade, P.Anjaneyulu, and S.H.Ahammad.2017."**Micro Controller Base Heartbeat Monitoring and Display on PC**." Journal

of Advanced Research in Dynamical and ControlSystems 9(4):250-260

[16] As 'habi, Keivan, Arman Vafabakhsh, and Saeed Borji. 2016. "**Data Transmission Security in CloudsComputing**." Indian Journal of Fundamental and Applied Life Sciences 6: 2231–6345

[17] Tirthani Ganesan, Neha R. "**Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography**."

[18] S.V.R.K.Rao, M.Saritha Devi, A.R.Kishore, Praveen Kumar "**Wireless sensor Network based Industrial Automation using Internet of Things (IoT**)" Volume 7, No.6, November - December 2018

[19] B.Manoj, K.V.K.Sasikanth, M.V.Subbarao, V Jyothi Prakash"**Analysis of Data Science with the use of Big Data**" Volume 7, No.6, November - December 2018.

[20]N.Saritha Devi, K.S.R.Raju, A.Madhu, R.Raja Sekhar "**Safety and Security for School children's Vehicles using GPS and IoT Technology**" Volume 7, No.6, November - December 2018.