



Sri Lanka Institute of Information Technology

System And Network Programming – IE2012

Lab 06

Mobile Malware Analysis

IT22151056

De Silva K.R.K.D

Group – WD.CS 01.02

Table of contents

Introduction To The Topic.....	3
Methodology.....	4
Task 1.....	5
Task 2.....	5
Task 3.....	8
Task 4.....	10
Task 5.....	14
Task 6.....	17
Conclusion.....	23
References.....	23

Introduction To The Topic

Mobile malware has become a much greater concern in our increasingly connected society where mobile devices are components of our daily lives. Smartphones and tablets, two examples of mobile devices, have developed into powerful computing platforms that provide a wide range of services and applications. However, because of their adaptability, they have become attractive targets for hackers looking to steal user data and exploit security flaws. People, companies, and even organizations are at serious risk from mobile malware, a specific type of malicious software created for mobile platforms like iOS and Android. These dangerous programs, which can infect devices and cause trouble on both a personal and professional level, exist in a variety of varieties, ranging from advertising and spyware to Trojans and malware.

To comprehend, analyze, and reduce the risks caused by these threats, malware for mobile device analysis is a crucial subject in the area of cybersecurity. To find and eliminate dangerous parts, this analytical technique looks at mobile applications, analyzes code structures, and evaluates behavior. The main objectives for smartphone malware analysis are to strengthen security protocols, preserve user privacy, and secure sensitive data. “Mobile malware is an increasing danger to consumer devices even if it is not as common as malware that targets conventional desktops. As assaults multiply and are more powerful, mobile malware is posing a threat to the mobile safety sector.”[1]

In addition to giving you the necessary skills, the “Mobile Malware Analysis” program also gives you the ability to contribute to the ongoing fight against mobile device malware threats. By the completion of this training, you’ll be better equipped to guard against the always-changing array of mobile threats, protecting both your digital life and the mobile ecosystems that we all depend on a daily basis. Come along with me as I study the exciting, difficult, and important topic of mobile malware investigation.

Methodology

Analysis of mobile malware is a complex, diverse process that is essential to cybersecurity. These mobile powerhouses are essential to our everyday lives in an age where mobile devices rule, but they are also excellent targets for hackers looking to exploit flaws, compromise user data, or engage in various other crimes. As the guide for understanding, analyzing, and minimizing the risks offered by these digital adversaries, an effective approach for mobile malware investigation is crucial.

It starts with careful planning, guaranteeing the availability of necessary tools and forms for documenting findings. Following sample gathering, there is a focus on confirming the legitimacy and applicability of the malware. Putting up a separate analysis environment is a key next step since it stops malware from spreading accidentally. While dynamic analysis entails running the virus in a secure setting and watching its activity, including network traffic, static analysis looks at the app's code, credentials, and API calls to find possible dangers.

TryHackMe lab link: [TryHackMe | Mobile Malware Analysis](https://tryhackme.com/room/mma)

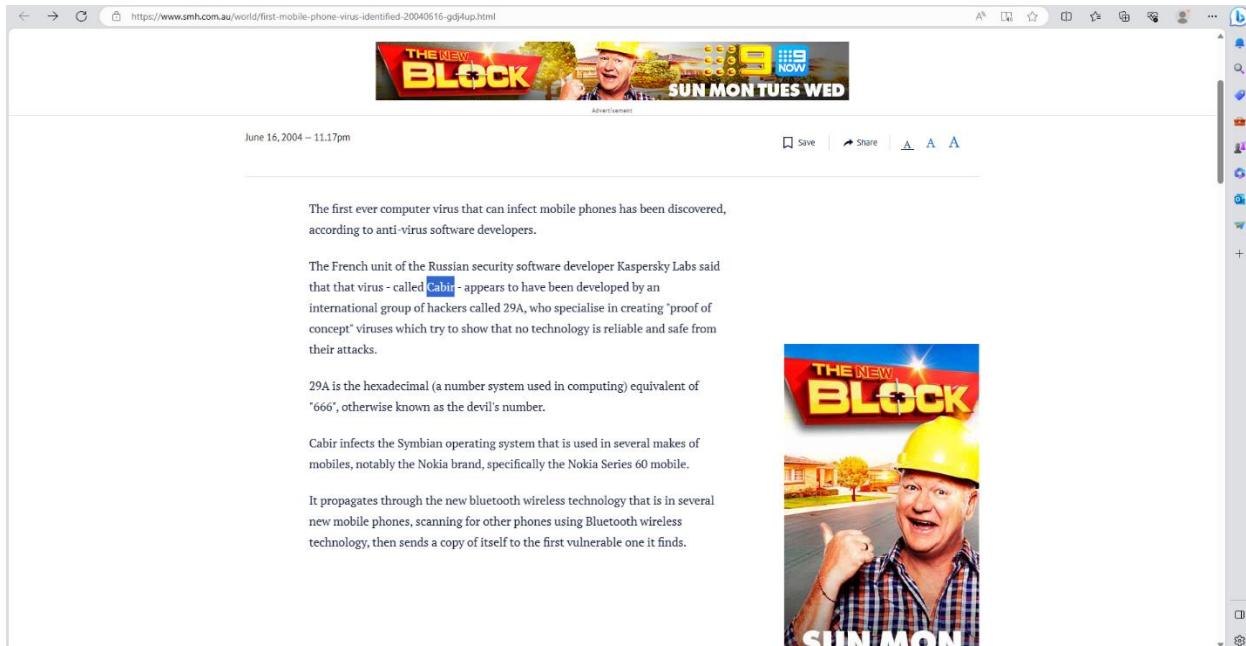
Task 1: Introduction – Read the text – No answer is needed

The screenshot shows the TryHackMe interface for the 'Mobile Malware Analysis' room. At the top, there's a navigation bar with links for Dashboard, Learn, Compete, and Other, along with an 'Access Machines' button and a search bar. Below the navigation is a large 'MOBSF' logo with 'Mobile Malware Analysis' text and a subtitle 'Learn and practice mobile malware analysis.' On the left, there's a sidebar with icons for various tools like Start AttackBox, Help, and Settings. The main content area displays 'Task 1 ○ Introduction'. The task text reads: 'It's incredible how often our computers are in the scope of cyber attacks. Antivirus has become an indispensable shield to provide us with a more secure environment, since we are exposed to destructive malware and cyber attacks. Inside our pockets, we have computers so powerful, but much smaller, we must be equally attentive on our phones, because we can suffer equally damaging attacks, sometimes even worse, because they can store relevant information such as private conversations and important accounts.' Below the text is a section titled 'Answer the questions below' with the instruction 'Read the text above.' A text input field contains 'No answer needed' and a green 'Completed' button.

Task 2:

The screenshot shows the TryHackMe interface for the 'Mobile Malware Analysis' room, specifically Task 2. The task title is 'Task 2 ○ An Unknown Land'. The task text reads: 'It is important to look at the past to understand why things are as they are today. A new technology, due to the lack of exploration, appears to be extremely reliable. Every system is reliable, until someone proves otherwise.' It also states: 'You will need to do some research in order to answer the questions in this task.' Below the text is a section titled 'Answer the questions below' with the instruction 'What known as the first malware created to affect mobile devices?'. There are four answer fields, each with an 'Answer format: *****' placeholder and a green 'Submit' button. The fourth answer field contains the text: 'The worm was created and sent out by researchers as a PoC (Proof of Concept), they did not believe that the mobile operating system could be easily exploited. Since then, malicious programs have become more popular.' A green 'Completed' button is shown next to the fourth answer field.

What is known as the first malware created to affect mobile devices? Read :
<https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html>



The first ever computer virus that can infect mobile phones has been discovered, according to anti-virus software developers.

The French unit of the Russian security software developer Kaspersky Labs said that that virus - called Cabir - appears to have been developed by an international group of hackers called 29A, who specialise in creating 'proof of concept' viruses which try to show that no technology is reliable and safe from their attacks.

29A is the hexadecimal (a number system used in computing) equivalent of '666', otherwise known as the devil's number.

Cabir infects the Symbian operating system that is used in several makes of mobiles, notably the Nokia brand, specifically the Nokia Series 60 mobile.

It propagates through the new bluetooth wireless technology that is in several new mobile phones, scanning for other phones using Bluetooth wireless technology, then sends a copy of itself to the first vulnerable one it finds.

What technology does this worm used to multiply? Read: [First mobile phone virus identified \(smh.com.au\)](https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html)



Once the worm is running, it will constantly search for Bluetooth-enabled devices, vastly shortened battery life because of the constant scanning.

If the virus succeeds in penetrating the phone, it writes the inscription 'Caribe' on the screen and is then activated every time that the phone is turned on.

29A sent the code to anti-virus software developers on Tuesday, who have since verified in lab test that it can be spread from phone to phone.

As the virus has only been circulated in a controlled laboratory setting, it poses no risk to the wider public. There are no known cases of it "in the wild".

29A is credited with the release of a recent virus called "Rugrat" that targets Windows 64 bit operating systems.

In May, researchers from the Symantec anti-virus software group identified W634.Rugrat.3544 and linked it to a family of six viruses that are all believed to be the work of the same author or group of authors. Each of the viruses demonstrates a different "first ever" infection technique.

According to the anti-virus software developer F-Secure, the discovery of Cabir is proof that the technologies are now available to create viruses for mobile phones.

What operating system did it infect? : <https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html>

The first ever computer virus that can infect mobile phones has been discovered, according to anti-virus software developers.

The French unit of the Russian security software developer Kaspersky Labs said that that virus - called Cabir - appears to have been developed by an international group of hackers called 29A, who specialise in creating "proof of concept" viruses which try to show that no technology is reliable and safe from their attacks.

29A is the hexadecimal (a number system used in computing) equivalent of '666', otherwise known as the devil's number.

Cabir infects the [Symbian operating system](#) that is used in several makes of mobiles, notably the Nokia brand, specifically the Nokia Series 60 mobile.

It propagates through the new bluetooth wireless technology that is in several new mobile phones, scanning for other phones using Bluetooth wireless technology, then sends a copy of itself to the first vulnerable one it finds.

Thank you for reading the Herald. This article is complimentary.

Register or log in now to read more articles and unlock extra benefits.

No payment required

READ MORE

Advertisement

THE NEW BLOCK

SUN MON TUES WED

9 NOW

What message did it show on the screen of the infected mobile phone?:

<https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html>

Once the worm is running, it will constantly search for Bluetooth-enabled devices, vastly shortened battery life because of the constant scanning.

If the virus succeeds in penetrating the phone, it writes the inscription [Caribe](#) on the screen and is then activated every time that the phone is turned on.

29A sent the code to anti-virus software developers on Tuesday, who have since verified in lab test that it can be spread from phone to phone.

As the virus has only been circulated in a controlled laboratory setting, it poses no risk to the wider public. There are no known cases of it "in the wild".

29A is credited with the release of a recent virus called "Rugrat" that targets Windows 64 bit operating systems.

In May, researchers from the Symantec anti-virus software group identified W634.Rugrat.3344 and linked it to a family of six viruses that are all believed to be the work of the same author or group of authors. Each of the viruses demonstrates a different "first ever" infection technique.

According to the anti-virus software developer F-Secure, the discovery of Cabir is [a breakthrough in mobile computing security](#).

Read more for free. Register or log in now to unlock more articles.

FROM OUR PARTNERS

AFR

How AI is reshaping the corporate landscape

Brought to you by Salesforce

Advertisement

THE NEW BLOCK

SUN MON TUES WED

9 NOW

A screenshot of a web browser window displaying the TryHackMe platform. The URL is <https://tryhackme.com/room/mma>. The main content area shows a challenge titled "Task 2" with the following details:

A new technology, due to the lack of exploration, appears to be extremely reliable. Every system is reliable, until someone proves otherwise.

You will need to do some research in order to answer the questions in this task.

Answer the questions below

What known as the first malware created to affect mobile devices?

Cabir Correct Answer

What technology does this worm used to multiply?

bluetooth Correct Answer

What operating system did it infect?

Symbian Correct Answer

What message did it show on the screen of the infected mobile phone?

Caribe Correct Answer

The worm was created and sent out by researchers as a PoC (Proof of Concept), they did not believe that the mobile operating system could be easily exploited. Since then, malicious programs have become more popular.

No answer needed Correct Answer

Below this, there is a list of tasks:

- Task 3: Small size, a lot of destruction.
- Task 4: Digging Deeper
- Task 5: MobSFing the sample.
- Task 6: It doesn't smell good!

The status bar at the bottom right indicates "Woop woop! Your answer is correct."

Task 3

What is the format of the file?

'TWFsd2FyZQ.apk'

A screenshot of a web browser window displaying the TryHackMe platform. The URL is <https://tryhackme.com/room/mma>. The main content area shows a challenge titled "Task 3" with the following details:

This view means that MobSF is ready to be used.

Answer the questions below

Deploy the machine & use MobSF to scan the file named "TWFsd2FyZQ.apk" that is located on the Desktop.

No answer needed Correct Answer

What is the format of the file?

.apk Correct Answer

The sample's size is 10,1 bytes, so it seems that it is not a complex application.

No answer needed Completed

Decode the name of the sample.

Answer format: ***** Submit Hint

Which is the target platform?

Answer format: ***** Submit

Below this, there is a screenshot of the MobSF analysis interface. The desktop background shows a "Try Hack Me" logo with binary code. The taskbar includes icons for Microsoft Edge, TWFsd2FyZQ.apk, and sample.apk. The desktop also has a Recycle Bin icon. The status bar at the bottom right indicates "Woop woop! Your answer is correct."

Decode the name of the sample. Decode Link : <https://www.base64decode.org/>

The screenshot shows the homepage of base64decode.org. At the top, there are tabs for 'Decode' and 'Encode'. Below them, a banner for '123RF AI Generated Images' is visible. The main content area has a heading 'Decode from Base64 format' and a text input field containing the string 'TWFd2FyZQ'. Below the input field are several configuration options: 'Source character set' set to 'UTF-8', 'Decode each line separately', and 'Live mode OFF'. A large green button labeled 'DECODE' is centered. To the right of the input field, a sidebar lists 'Other tools' and 'URL Decode'. The background features a repeating pattern of various icons.

Which is the target platform? Read: <https://fileinfo.com/extension/apk>

The screenshot shows two windows side-by-side. On the left is a web browser displaying the tryhackme.com room for challenge 'MIMI'. It shows a question about the file format (.apk), size (10.1 bytes), and a task to decode the sample name ('Malware'). On the right is a desktop application window titled 'Static Analysis' from 'MobSF'. It shows the file information for 'TWFd2FyZQ.apk', including the file name, size (0.01MB), MD5 hash, SHA1 hash, SHA256 hash, and APK scores (Icon: Hidden, Average CVSS: 6.7, Security Score: 75/100). Below this, there are four cards: 'ACTIVITIES' (1), 'SERVICES' (1), 'RECEIVERS' (1), and 'PROVIDERS' (0). The status bar at the bottom indicates the system is running on 'ENG' with a battery level of '16/99%', a signal strength of '18/50', and a connection speed of '32m 44s'.

Task 4

What does “Avast-Mobile” can tell us about this software?

<https://www.virustotal.com/gui/file/e201a1d2cecf1d04d97d59abec0863c716dcf9cad89b85d036f9163a48057e7/detection>

The screenshot shows the VirusTotal analysis interface for the file TWFd2FyZQ.apk. The main summary indicates a high score of 43/64, with 43 security vendors flagging it as malicious. Below this, a detailed table lists various security vendors and their findings:

Virus Vendor	Findings
AhnLab-V3	PUP/Android.Metasploit.54109
AntiY-AVL	Trojan/Generic.ASMalwAD.DCI
Avast	Android.Metasploit-G [PUP]
AVG	Android.Metasploit-G [PUP]
BitDefender	Application.HackTool.Meterpreter.AQR
Cynet	Malicious (score: 99)
DrWeb	Android.RemoteCode.6833
eScan	Application.HackTool.Meterpreter.AQR
F-Secure	Malware:ANDROID/Droid.FJNR.Gen
GData	Application.HackTool.Meterpreter.AQR
Ikarus	Trojan-Downloader.AndroidOS.Agent
Alibaba	HackTool.Android/Metasploit.07e03416
Arcabit	Application.HackTool.Meterpreter.AQR
Avast-Mobile	Android.Evo-gen [Trj]
Avira (no cloud)	ANDROID/TrojanDownloader.FNAAGen
BitDefenderFalk	Android.Riskware.SMSSend.RR
Cyren	AndroidOS/Downloader.M.gen/Eldorado
Emsisoft	Application.HackTool.Meterpreter.AQR (B)
ESET-NOD32	A Variant Of Android/TrojanDownloader.A..
Fortinet	Android/Agent_JNtr
Google	Detected
K7GW	Trojan-Downloader (004ff8551)

What program was used to create the malware?

The screenshot shows the TryHackMe challenge interface for Task 4, titled "Digging Deeper". It includes a question about what program was used to create the malware, with the answer "Metasploit" being correct. To the right, the MobSF static analysis tool provides detailed information about the malware, including its file name (TWFd2FyZQ.apk), package name (com.metasploit.stage), and various API permissions and activity counts.

What is the package name?

The screenshot shows the VirusTotal analysis page for the APK file. Key details include:

- File type:** Android executable mobile android apk
- Magic:** Zip archive data, at least v2.0 to extract, compression method=deflate
- TrID:** Java Archive (72.7%) ZIP compressed archive (21.6%) PrintFox/PageFox bitmap (640x800) (5.4%)
- File size:** 9.95 KB (10187 bytes)
- History:** First Submission: 2020-10-18 22:00:39 UTC; Last Submission: 2023-07-28 14:09:35 UTC; Last Analysis: 2023-09-16 12:52:41 UTC; Earliest Contents Modification: 2020-10-18 18:49:26; Latest Contents Modification: 2020-10-18 18:49:28.
- Names:** TWFsd2FyZQ.apk, t1.apk, apkanal.apk
- Android Info:** Summary: Android Type APK, Package Name com.metasploit.stage, Main Activity com.metasploit.stage.MainActivity, Internal Version 1, Displayed Version 1.0, Minimum SDK Version 10, Target SDK Version 17.
- Certificate Attributes:** Valid From: 2017-11-04 08:26:42, Valid To: 2034-03-02 01:25:14, Serial Number: 1, Thumbprint: b8ea694b40dactle715d2ecb270c10795974fa5e.
- Certificate Subject:** Distinguished Name: C=US/O=Android/CN=Android Debug, Country Code: US/O=Android/CN=Android Debug.
- Certificate Issuer:** (Information partially cut off)

What is the SHA-1 signature?

The screenshot shows the VirusTotal analysis page for the APK file. Key details include:

- MD5:** 54d5d55a08d1c32a8d049794a33a5dc
- SHA-1:** **76dd42594ac1f1a4c3492ffaa5ebfb54af0000**
- SHA-256:** e10ba1d2dec1b94d97d9f9ebecc96a1c716dcf9cad9b85d036f9163a48057e7
- Vhash:** bf0b094a11794443f9547200a48e44
- SSDEEP:** 192:31VdHnTJaxP1Yn1582+vwC7WvqgbSL4GBLxJxKuAn9yB00NnWuJn7Wf28JXy
- TLSH:** TlRQ29f7AA7A461BF107ABBC50432B877DFA03486219335d4/COEB481527ACD33E64A
- Permalink:** 79fe1c93a40f489be7f9a0578105707f74+973e0e9782bc50504a9f991ec30
- File type:** Android executable mobile android apk
- Magic:** Zip archive data, at least v2.0 to extract, compression method=deflate
- TrID:** Java Archive (72.7%) ZIP compressed archive (21.6%) PrintFox/PageFox bitmap (640x800) (5.4%)
- File size:** 9.95 KB (10187 bytes)
- History:** First Submission: 2020-10-18 22:00:39 UTC; Last Submission: 2023-07-28 14:09:35 UTC; Last Analysis: 2023-09-16 12:52:41 UTC; Earliest Contents Modification: 2020-10-18 18:49:26; Latest Contents Modification: 2020-10-18 18:49:28.
- Names:** TWFsd2FyZQ.apk, t1.apk, apkanal.apk
- Android Info:** Summary: Android Type APK, Package Name com.metasploit.stage, Main Activity com.metasploit.stage.MainActivity, Internal Version 1, Displayed Version 1.0, Minimum SDK Version 10, Target SDK Version 17.

What is the unique XML file?

The screenshot shows the VirusTotal analysis interface for a specific file. At the top, it lists 'Contacted IP Addresses (45)' with columns for IP, Detections, Autonomous System, and Country. Most entries show 0/89 detections and are from the US. Below this is a section for 'Bundled Files (6)' with columns for Scanned, Detections, File type, and Name. It shows several AndroidManifest.xml files from different dates. A 'Graph Summary' section is also present. The URL in the address bar is <https://www.virustotal.com/gui/file/14e52da70277a5560f72e8094cbcd44eb22ff9520d40ed7e033200c19>.

How many permissions are there inside?

The screenshot shows a TryHackMe challenge titled 'MMA C/HN V2'. The challenge asks: 'How many permissions are there inside?'. The answer '22' is entered in the text field and marked as a 'Correct Answer'. Other questions and answers shown include: 'What is the package name?' (com.metasploit.stage), 'What is the SHA-1 signature?' (74d442594acf1dc6e3492ffa5eb8956af000d), and 'What is the unique XML file?' (AndroidManifest.xml). The challenge interface includes a sidebar with various icons.

Which permission allows the application to take pictures with the camera?

The screenshot shows the VirusTotal analysis interface for a file. At the top, the URL is https://www.virustotal.com/gui/file/e201a1d2cecf1d04d97d59abec0863c716dcf9cad89b85d036f9163a48057e7/details. The main content area displays the following information:

- Thumbprint:** D8EAE94D40A8C7E7B52ECD2/UC10/9599/41a8e
- Certificate Subject:**
 - Distinguished Name: C=US/O=Android/CN=Android Debug
 - Country Code: US/O=Android/CN=Android Debug
- Certificate Issuer:**
 - Distinguished Name: C=US/O=Android/CN=Android Debug
 - Country Code: US/O=Android/CN=Android Debug
- Permissions:**
 - android.permission.ACCESS_COARSE_LOCATION
 - android.permission.INTERNET
 - android.permission.ACCESS_FINE_LOCATION
 - android.permission.SEND_SMS
 - android.permission.WRITE_CALL_LOG
 - android.permission.READ_CALL_LOG
 - android.permission.WRITE_EXTERNAL_STORAGE
 - android.permission.RECORD_AUDIO
 - android.permission.WRITE_CONTACTS
 - android.permission.CALL_PHONE
 - android.permission.READ_PHONE_STATE
 - android.permission.READ_SMS
 - android.permission.CAMERA
 - android.permission.CHANGE_WIFI_STATE
 - android.permission.RECEIVE_SMS
 - android.permission.READ_CONTACTS
 - android.permission.ACCESS_WIFI_STATE
 - android.permission.ACCESS_NETWORK_STATE
 - android.permission.SET_WALLPAPER
 - android.permission.RECEIVE_BOOT_COMPLETED
 - android.permission.WRITE_SETTINGS
 - android.permission.WAKE_LOCK
- Activities:** com.metasploit.stage.MainActivity

What is the message left by the community?

The screenshot shows the comment section for the same file analysis page. The URL is https://www.virustotal.com/gui/file/e201a1d2cecf1d04d97d59abec0863c716dcf9cad89b85d036f9163a48057e7/community.

The comments are as follows:

- lucaslapinho** 7 months ago: -1
- solomon** 8 months ago: o/
- Fulmine** 1 year ago: nice room :)
- i0wk3y** 2 years ago: Great room, Happy Hacking! 😊
- mKmelo** 2 years ago: very good Room TKS
- farinsp5** 2 years ago: THM{Virus5-T014aL-TWard2Fyz23@bmfaXhXpcw}

A message at the bottom states: You must be signed in to post a comment.

The results provided by VirusTotal shows that we have a generic malware. It does not serve for attack purposes because we can see that a good part of the Antiviruses are detecting it, this malware is a good one for searching purposes, but it is also used for post exploitation.

No answer needed Correct Answer

What is the package name?
com.metasploit.stage Correct Answer

What is the SHA-1 signature?
74d442594acf11dc6e3492fffea5eb8956af000d Correct Answer

By extracting the content, it will create a folder with some files inside, one of which is a XML. It describes some important information about the application for Android build tools, for Android operating system and for Google Play. This file declares items, shows some stuff as the package name and the permissions required to the device. The information that will be needed for the next questions can be found on VirusTotal also.

No answer needed Correct Answer

What is the unique XML file?
AndroidManifest.xml Correct Answer

How many permissions are there inside?
22 Correct Answer

Which permission allows the application to take pictures with the camera?
android.permission.CAMERA Correct Answer

What is the message left by the community?
THM{V1rus5-T0t4al-TWFsd2FyZ51BbmFseXNpcw} Correct Answer

Task 5 ○ MobSFing the sample.

Task 5

What is the programming language used to create the program?

Free to install it in a virtual machine you own to understand more how the application works, you can install it in GitHub - <https://github.com/MobSF/mobile-Security-Framework-MobSF>.

The machine is configured to start MobSF when deployed, if you accidentally closed the web page you can visit the MobSF page by visiting the link <http://127.0.0.1:8000> inside the deployed machine. Press the "Upload & Analyze" button and select the file we have been working on.

Answer the questions below

What is the programming language used to create the program?
java Correct Answer

How many signatures does the package has?
Answer format: * Submit

Application is signed with v1 signature scheme, what is it vulnerable to on Android <7.0?
Answer format: ***** Submit

MobSF gives all the code decompiled. Just a base of programming make us able to understand a little bit of what is happening.
No answer needed Completed

This malware is used to create a connection with the victim that is called a reverse shell.
No answer needed Completed

What is the App name?
Answer format: ***** Submit

It looks like there is a function calling for the package manager, so it can see all the installed applications. What function is that?
Answer format: ***** Submit Hint

Returning to the manifest.
The flag "android:allowBackup" allows the user to backup application data via USB debugging. It is recommended that

How many signatures does the package has?

The image shows two windows side-by-side. On the left is a browser window for the TryHackMe challenge 'MobsFing the sample'. It contains several questions about the Android app 'TWFsd2f2yZQ.apk'. The right window is the 'Static Analysis' interface of the MobSF tool, showing detailed analysis results for the same APK file.

TryHackMe Challenge (Left Window):

- Task 5:** MobsFing the sample.
- Question: What is the programming language used to create the program? Answer: java. **Correct Answer**
- Question: How many signatures does the package has? Answer: 1. **Correct Answer**
- Question: Application is signed with v1 signature scheme, what is it vulnerable to on Android <7.0? Answer: Janus. **Submit**
- Question: MobSF gives all the code decompiled. Just a base of programming make us able to understand a little bit of what is happening. Answer: No answer needed. **Completed**
- Question: This malware is used to create a connection with the victim that is called a reverse shell. Answer: No answer needed. **Completed**
- Question: What is the App name? Answer format: *****. Answer: *****. **Submit**

MobSF Static Analysis (Right Window):

SIGNER CERTIFICATE

```

APK is signed
v1 signature: True
v2 signature: False
v3 signature: False
Found 1 unique certificates
Subject: C=US/O=Android/CN=Android Debug
Signature Algorithm: rsassa_pkcs1v15
Valid From: 2017-11-08:08:26:42+00:00
Valid To: 2034-03-02 01:25:14+00:00
Issuer: C=US/O=Android/CN=Android Debug
Serial Number: 0x1
Hash Algorithm: sha1
md5: 9d7f41a21ae3bd0e8c009456ea4bc8b
sha1: 18ee9a04b8ac11e715d2e270c107955974fa5e
sha256: f542876dd1a0dc95dc4910570e4889c4053136f31ce6df80b5edd966af21ef
sha512: 769427cb4fc12bc3c1b8c6668755fa50a22ed19a70b053c3ebff0a9a27c56d0e385b349be3d9375fdef52d27e2819f
  
```

Search: [Search Bar]

Application is signed with v1 signature scheme, what is it vulnerable to on Android<7.0?

The image shows two windows side-by-side. On the left is a browser window for the TryHackMe challenge 'MobsFing the sample'. It contains several questions about the Android app 'TWFsd2f2yZQ.apk'. The right window is the 'Static Analysis' interface of the MobSF tool, showing detailed analysis results for the same APK file, including a summary of findings.

TryHackMe Challenge (Left Window):

- Task 5:** MobsFing the sample.
- Question: What is the programming language used to create the program? Answer: java. **Correct Answer**
- Question: How many signatures does the package has? Answer: 1. **Correct Answer**
- Question: Application is signed with v1 signature scheme, what is it vulnerable to on Android <7.0? Answer: Janus. **Correct Answer**
- Question: MobSF gives all the code decompiled. Just a base of programming make us able to understand a little bit of what is happening. Answer: No answer needed. **Completed**
- Question: This malware is used to create a connection with the victim that is called a reverse shell. Answer: No answer needed. **Completed**
- Question: What is the App name? Answer format: *****. Answer: *****. **Submit**

MobSF Static Analysis (Right Window):

STATUS + DESCRIPTION

STATUS	DESCRIPTION
bad	Application is signed with v1 signature scheme, making it vulnerable to Janus vulnerability on Android <7.0
bad	Application signed with a debug certificate. Production application must not be shipped with a debug certificate.
bad	Application is signed with SHA1withRSA. SHA1 hash algorithm is known to have collision issues.
secure	Application is signed with a code signing certificate

Showing 1 to 4 of 4 entries

What is the App name?

The machine is configured to start MobSF when deployed, if you accidentally closed the web page you can visit the MobSF page by visiting the link <http://127.0.0.1:8000> inside the deployed machine. Press the "Upload & Analyze" button and select the file we have been working on.

Answer the questions below

What is the programming language used to create the program?

java Correct Answer

How many signatures does the package has?

1 Correct Answer

Application is signed with v1 signature scheme, what is it vulnerable to on Android <7.0?

Janus Correct Answer

MobSF gives all the code decompiled. Just a base of programming make us able to understand a little bit of what is happening.

No answer needed Correct Answer

This malware is used to create a connection with the victim that is called a reverse shell.

No answer needed Correct Answer

What is the App name?

MainActivity Correct Answer

It looks like there is a function calling for the package manager, so it can see all the installed applications. What function is that?

Answer format: *.***** Submit Hint

Returning to the manifest.

The flag "android:allowBackup" allows the user to backup application data via USB debugging. It is recommended that this be set as "False", even if by default it is "True".

What is the result of the configuration?

Static Analysis | 127.0.0.1:8000/static_analyzer?name=TWFsd2FyZQ.apk&checksum=566d0c5... | Woop woop! Your answer is correct.

RECENT SCANS STATIC ANALYZER DYNAMIC ANALYZER API DOCS ABOUT Search M3n

APP SCORES FILE INFORMATION APP INFORMATION

No icon Hidden

File Name: TWFsd2FyZQ.apk
Size: 0.01MB
MD5: 566d0c5a08dc132a8d049794a33f5dc
SHA1: 744d245594acf11dc6e3492fe5eb8956af0d000d
SHA256: e201a1d2cef1d04d97d59abec0863c716dcf9fcad89b85d036f
SHA512: 9163a48057e

Icon: MainActivity
Average CVSS: 6.7
Security Score: 75/100
Trackers Detection: 0/405

App Name: MainActivity
Package Name: com.metasploit.stage
Main Activity: MainActivity
Target SDK: 17
Min SDK: 10
Max SDK: 1
Android Version Name: 1.0
Android Version Code: 1

ACTIVITIES: 1 View
SERVICES: 1 View
RECEIVERS: 1 View
PROVIDERS: 0 View

Exported Activities: 0
Exported Services: 1
Exported Receivers: 1
Exported Providers: 0

Windows Search Internet Mail File Home MMA CMN V2 ENG 19:56 16/09/2023 47m 38s

It looks like there is a function calling for the package manager, so it can see all the installed applications. What function is that?

The flag “android:allowBackup” allows the user to backup application data via USB debugging. It is recommended that this be set as “False”, even if by default it is “True”. • What is the severity of this configuration?

The image shows two side-by-side screenshots. On the left is the TryHackMe MMA challenge interface for challenge 1. It displays several questions with dropdown answers and green "Correct Answer" buttons. One question asks about the app name, with "MainActivity" being the correct answer. Another asks about the severity of the "android:allowBackup" flag, with "medium" being the correct answer. Below these are sections for "Task 6" and "Task 7". On the right is the MobSF static analysis tool interface. It shows a manifest analysis table with three findings: 1. Application Data can be Backed up [android:allowBackup] flag is missing (medium severity). 2. Broadcast Receiver (.MainBroadcastReceiver) is not Protected. An intent-filter exists (high severity). 3. Service (.MainService) is not Protected (high severity). The MobSF interface also includes tabs for RECENT SCANS, STATIC ANALYZER, DYNAMIC ANALYZER, API DOCS, and ABOUT.

Task 6

Our next sample located on the Desktop, the name of the file is sample2.apk, let's start a MobSF analysis on it. What is the SHA-256 hash of the file?

This screenshot shows the continuation of the challenge. The MMA interface (left) has a new task: "Answer the questions below". It asks for the SHA-256 hash of the file, which is provided as "bd8cda80aae3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94d1". A "Correct Answer" button is available. The MMA interface also shows other questions related to the malware's history and detection. The MobSF interface (right) shows the static analysis results for sample2.apk. It provides detailed information about the file, including its name, size, MD5, SHA1, and SHA256 hashes. It also displays a summary of the app's permissions and behaviors, such as 3 activities, 5 services, 8 receivers, and 0 providers. A message at the bottom of the MMA interface says "Your streak has increased. You're 4 away from a badge!"

After finding the sample on VirusTotal, what does the “Avast” anti-virus engine recognizes it as?

<https://www.virustotal.com/gui/file/bd8cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f540>

44 security vendors and 1 sandbox flagged this file as malicious

bd8cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94db52a
pegasus.apk

Community Score: 44 / 65

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 24+

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan, pegasus, chrysaor

Threat categories: trojan

Family labels: pegasus, chrysaor, fkmn

Security vendors' analysis:

VirusTotal	Description	Family	Description
AhnLab-V3	Trojan.Android.Agent.860476	Allsafe	Trojan/Spy.Android/Pegasus.d33dbca
Anti-AVL	Trojan(Spy)Android.Chrysaor	Arcabit	Android.Pegasus
Avast	Android.Obfusc-BM [Trj]	Avast-Mobile	Android/Evo-gen[Trj]
AVG	Android.Obfusc-BM [Trj]	Avira (no cloud)	ANDROID/DSpyAgent.FKMN.Gen
BitDefender	Trojan.GenericID.46667348	BitDefenderFax	Android.Trojan.Pegasus.F
Cynet	Malicious (score: 99)	DrWeb	Android.SignGen.Sip.3172
Emsisoft	Trojan.GenericID.46667348 (B)	eScan	Trojan.GenericID.46667348
ESET-NOD32	Multiple Detections	F-Secure	Malware.Android/Spy.Agent.FKMN.Gen
Fortinet	Android/Obfus.NS[tr]	GData	Trojan.GenericID.46667348
Google	Detected	Ikarus	Trojan.AndroidOS.Obfus

With what we have, try to find out the name of the sample.

Task 2: An Unknown Land

Task 3: Small size, a lot of destruction.

Task 4: Digging Deeper

Task 5: MobSFing the sample.

Task 6: It doesn't smell good!

I think that now we have the necessary knowledge to analyze bigger stuff.

Our next sample located on the Desktop, the name of the file is sample2.apk, let's start a MobSF analysis on it.

Answer the questions below

What is the SHA-256 hash of the file?

bd8cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94db52a

Correct Answer

After finding the sample on VirusTotal, what does the “Avast” anti-virus engine recognizes it as?

Android.Obfusc-BM [Trj]

Correct Answer

With what we have, try to find out the name of the sample.

pegasus

Correct Answer

It seems like it is a very dangerous malware and has a big history of destruction.

This became news for spying journalists, what year was that?

Answer format: ****

Submit Hint

It was reported that the malware was developed by a legitimate intention: The idea behind it was to use the software as a government tool designed to track and combat terrorism and crime.

This malware has been found infecting people's smartphones and political activists in more than 44 countries.

Static Analysis

Woop woop! Your answer is correct.

RECENT SCANS STATIC ANALYZER DYNAMIC ANALYZER API DOCS Search MDS

APP SCORES FILE INFORMATION APP INFORMATION

Icon! Hidden File Name: sample2.apk
Icon! Average CVSS: 7.5 Package Name: se.cdujmehn.qdthetyl
Security Score: 85/100 SHA1: 7289737c1d0462726bbe8935a7702c130bbdc
Trackers Detection: 0/405 Target ADK: 9 Max Sdk: 9 Max SWD: 292
0/405 Android Version Name: 2.9.3
Android Version Code: 292

3 ACTIVITIES View 5 SERVICES View 8 RECEIVERS View 0 PROVIDERS View

Exported Activities: 0 Exported Services: 1 Exported Receivers: 5 Exported Providers: 0

17:59 17/09/2023 37m 35s

This became news for spying journalists, what year was that? Read:
[https://en.wikipedia.org/wiki/Pegasus_\(spyware\)](https://en.wikipedia.org/wiki/Pegasus_(spyware))

Left Screenshot (tryhackme.com):

- Task 5:** MobsFing the sample. (Completed)
- Task 6:** It doesn't smell good! (Completed)
- Message: I think that now we have the necessary knowledge to analyze bigger stuff.
- Message: Our next sample located on the Desktop, the name of the file is sample2.apk, let's start a MobSF analysis on it.
- Answer the questions below:**
 - What is the SHA-256 hash of the file?
bd8cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94d1 (Correct Answer)
 - After finding the sample on VirusTotal, what does the "Avast" anti-virus engine recognizes it as?
Android:Obfusc-BM [Trj] (Correct Answer)
 - With what we have, try to find out the name of the sample.
pegasus (Correct Answer)
 - It seems like it is a very dangerous malware and has a big history of destruction.
 - This became news for spying journalists, what year was that?
2017 (Correct Answer)
 - It was reported that the malware was developed by a legitimate intention: The idea behind it was to use the software as a government tool designed to track and combat terrorism and crime.
 - This malware has been found infecting people's smartphones and political activists in more than 44 countries.
No answer needed (Completed)
 - If we search the name we found of the malware in MITRE ATT&CK (<https://attack.mitre.org/>), we can find some interesting information.
 - What is the ID of the MITRE ATT&CK that is associated with our sample?
S0316 (Completed)

Right Screenshot (MobSF):

- App Scores:** Average CVSS: 7.5, Security Score: 85/100, Trackers Detection: 0/405
- File Information:** File Name: sample2.apk, Size: 1.06MB, MD5: 8d4b77fa3546149f25bd17357d41fb0, SHA1: 7289737c1cd062726abe8935a7702c130bbdc, SHA256: b38cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94d52, Target SDK: 9, Min SDK: 0, Max SDK: 29, Android Version Name: 2.9.3, Android Version Code: 292
- App Information:** App Name: Media Sync, Package Name: seC.dujmehr.qdthet, Main Activity: seC.dujmehr.qdthet.Dujmehnpqyd
- Metrics Summary:**
 - ACTIVITIES: 3 (View)
 - SERVICES: 5 (View)
 - RECEIVERS: 8 (View)
 - PROVIDERS: 0 (View)
- Exported Metrics:**
 - Exported Activities: 0
 - Exported Services: 1
 - Exported Receivers: 5
 - Exported Providers: 0

What is the ID of the MITRE ATT&CK that is associated with our sample? Review:
<https://attack.mitre.org/software/S0316/>

MITRE | ATT&CK Software Page for Pegasus for Android:

- Software Overview:** Pegasus for Android is the Android version of malware that has reportedly been linked to the NSO Group. The iOS version is tracked separately under Pegasus for iOS.
- Associated Software Descriptions:** Chrysaor
- Techniques Used:**

Domain	ID	Name	Use
Mobile	T1649	Audio Capture	Pegasus for Android has the ability to record device audio.
Mobile	T1645	Compromise Client Software Binary	Pegasus for Android attempts to modify the device's system partition.
Mobile	T1624	Event Triggered Execution	Pegasus for Android listens for the <code>com.sec.svc.event</code> broadcast intent in order to maintain persistence and activate its
- Details Panel:**
 - ID: S0316
 - Associated Software: Chrysaor
 - Type: MALWARE
 - Platform: Android
 - Version: 1.2
 - Created: 25 October 2017
 - Last Modified: 24 October 2022

What technique has the ability to exploit OS vulnerabilities to escalate privileges? Review: <https://attack.mitre.org/techniques/T1404/>

Software			
Overview			
3PARA RAT			
4H RAT			
AADInternals			
ABK			
AbstractEmu			
ACAD/Medre.A			
Action RAT			
adbupd			
AdFind			
Adups			
ADVSTORESHELL			
Agent Smith			
Agent Tesla			
Agent.btz			
Allwinner			
Amadey			
Anchor			
Android/AdDisplay.Ashas			
Android/Chuli.A			
AndroidOS/MalLocker.B			
ANDROIDOS_ANSERVER.A			
AndroBAT			
References			
1. Mike Murray. (2017, April 3). Pegasus for Android: the other side of the story emerges. Retrieved April 16, 2017.		2. Rich Cannings et al. (2017, April 3). An investigation of Chrysaor Malware on Android. Retrieved April 16, 2017.	

There is a permission that when accepted, allows the application to access the list of accounts in the Accounts Service. What is the status shown by MobSF regarding this permission.
(`android.permission.GET_ACCOUNTS`)

https://tryhackme.com/room/mma

Correct Answer Hint

It was reported that the malware was developed by a legitimate intention: The idea behind it was to use the software as a government tool designed to track and combat terrorism and crime.

This malware has been found infecting people's smartphones and political activists in more than 44 countries.

No answer needed

Correct Answer

If we search the name we found of the malware in MITRE ATT&CK (<https://attack.mitre.org/>), we can find some interesting information.

What is the ID of the MITRE ATT&CK that is associated with our sample?

S0316

Correct Answer

What technique has the ability to exploit OS vulnerabilities to escalate privileges?

T1404

Correct Answer Hint

Now, let's go back to the MobSF analysis.

No answer needed

Correct Answer

There is a permission that when accepted, allows the application to access the list of accounts in the Accounts Service. What is the status shown by MobSF regarding this permission. (android.permission.GET_ACCOUNTS)

dangerous

Correct Answer

What org.eclipse.paho.client file refers to properties of Portuguese from Brazil (pt-br)?

Answer Format: ***://*****/*****/*****/*****/* static analysis */

RECENT SCANS STATIC ANALYZER DYNAMIC ANALYZER API DOCS ABOUT Search MDS

Woop woop! Your answer is correct.

widgets can be used by which application. With this permission, applications can give access to personal data to other applications. Not for use by common applications.

Permission	Status	Description
android.permission.GET_ACCOUNTS	dangerous	list accounts
android.permission.GET_PACKAGE_SIZE	normal	measure application storage space
android.permission.GET_TASKS	dangerous	retrieve running applications

18:18 17/09/2024

THM AttackBox MMA CMN V2

18m 29s

What org.eclipse.paho.client file refers to properties of Portuguese from Brazil (pt_br)?

The left pane shows a challenge from TryHackMe room MMA. It asks for the ID of the MITRE ATT&CK sample, which is 50316. It also asks for a technique to exploit OS vulnerabilities, which is T1404. It then asks about a permission related to accounts, which is dangerous. Finally, it asks for the org.eclipse.paho.client file referring to properties of Portuguese from Brazil, which is org.eclipse/paho/client/mqttv3/internal/nls/messages_pt_BR.properties.

The right pane shows the MobSF static analysis tool. It displays a list of files and resources, including various properties files for MQTT messages in different languages (en, es, de, fr, hu, ja, ko, pl, pt_BR, ru, zh_CN, zh_TW). The analysis results show that the malware has several features making identification difficult, such as safety and internal components, and cryptographic operations with a random bit generation service.

The malware has a special appeal for its safety and its internal components, reducing the risk of compromise. It has a functionality for its cryptographic operations with the feature of a random bit generation service. How can it be identified?

The left pane shows the same challenge as the previous screenshot. It asks for the ID of the MITRE ATT&CK sample (50316), a technique to exploit OS vulnerabilities (T1404), a dangerous permission, and the org.eclipse.paho.client file for Portuguese from Brazil (org.eclipse/paho/client/mqttv3/internal/nls/messages_pt_BR.properties).

The right pane shows NIAP ANALYSIS v1.3 results. Five requirements are listed:

NO	IDENTIFIER	REQUIREMENT	FEATURE	DESCRIPTION
1	FCS_RBG_EXT.1.1	Security Functional Requirements	Random Bit Generation Services	The application uses no DRBG functionality for its cryptographic operations.
2	FCS_SHG_EXT.1.1	Security Functional Requirements	Storage of Credentials	The application does not store any credentials to non-volatile memory.
3	FCS_CRM_EXT.1.1	Security Functional Requirements	Cryptographic Key Generation Services	The application generates no asymmetric cryptographic keys.
4	FOP_DIC_EXT.1.1	Security Functional Requirements	Access to Platform Resources	The application has access to ['camera', 'network connectivity', 'location', 'microphone', 'NFC', 'bluetooth'].
5	FOP_DIC_EXT.1.2	Security Functional Requirements	Access to Platform Resources	The application has access to ['calendar', 'address book', 'system'].

The left side shows the tryhackme room interface with tasks completed:

- Task 4: Digging Deeper
- Task 5: MobSFing the sample.
- Task 6: It doesn't smell good!
- Task 7: Conclusion

Text from Task 7:

If it is normal to think that our mobile phones are harder to be infected, they have characteristics that makes the malware actions limited, as the Sandbox concept, and the fact that we never download things directly from the open internet.

Here I leave some awesome articles and other rooms that may be interesting to get deeper into this subject.

<https://github.com/OWASP/owasp-mstg>
<https://attack.mitre.org/matrices/mobile/android/>
<https://attack.mitre.org/matrices/mobile/os/>

<https://tryhackme.com/room/malmainintroductory>
<https://tryhackme.com/room/androidhacking101>
<https://tryhackme.com/room/iosforensics>

If you have any feedback, feel free to contact me on discord: farinap#4535

Answer the questions below

Thank you for your participation!

No answer needed Correct Answer

Created by cmnatic and Termack and farinap5

This is a free room, which means anyone can deploy virtual machines in the room (without being subscribed)! 5518 users are in here and this room is 775 days old.

The right side shows a static analysis tool window titled "NIAP ANALYSIS v1.3" with the following table:

NO	IDENTIFIER	REQUIREMENT	FEATURE	DESCRIPTION
1	FCS_KBG_EXT.1.1	Security Functional Requirements	Random Bit Generation Services	The application use no DRBG functionality for its cryptographic operations.
2	FCS_STO_EXT.1.1	Security Functional Requirements	Storage of Credentials	The application does not store any credentials to non-volatile memory.
3	FCS_CKM_EXEL.1	Security Functional Requirements	Cryptographic Key Generation Services	The application generate no asymmetric cryptographic keys.
4	FDP_DEC_EXT.1.1	Security Functional Requirements	Access to Platform Resources	The application has access to ['camera', 'network connectivity', 'location', 'microphone', 'NFC', 'bluetooth'].
5	FDP_DEC_EXT.1.2	Security Functional Requirements	Access to Platform Resources	The application has access to ['calender', 'address book', 'system']

The left side shows the tryhackme room interface with tasks completed:

- Task 4: Digging Deeper
- Task 5: MobSFing the sample.
- Task 6: It doesn't smell good!
- Task 7: Conclusion

Text from Task 7:

If it is normal to think that our mobile phones are harder to be infected, they have characteristics that makes the malware actions limited, as the Sandbox concept, and the fact that we never download things directly from the open internet.

Here I leave some awesome articles and other rooms that may be interesting to get deeper into this subject.

<https://github.com/OWASP/owasp-mstg>
<https://attack.mitre.org/matrices/mobile/android/>
<https://attack.mitre.org/matrices/mobile/os/>

<https://tryhackme.com/room/malmainintroductory>
<https://tryhackme.com/room/androidhacking101>
<https://tryhackme.com/room/iosforensics>

If you have any feedback, feel free to contact me on discord: farinap#4535

Answer the questions below

Thank you for your participation!

No answer needed Correct Answer

Created by cmnatic and Termack and farinap5

This is a free room, which means anyone can deploy virtual machines in the room (without being subscribed)! 5518 users are in here and this room is 775 days old.

A congratulatory message box is displayed:

Congratulations
 You've completed the room! Share this with your friends:

[Twitter](#) [Facebook](#) [LinkedIn](#)

Leave feedback

Conclusion

In the field of cybersecurity, mobile malware analysis is a crucial discipline that is necessary for securing user data and preserving the security of mobile ecosystems in a time when tablets and smartphones have become commonplace. Analysts may analyze harmful software and comprehend its inner workings using a methodical and precise approach, enabling the creation of efficient defenses against changing digital threats. Analysts are prepared to meet the challenges offered by mobile malware by following a well-defined approach that includes planning, sample collecting, system setup, dynamic and static analysis, and traffic analysis. This proactive approach not only strengthens security measures but also gives people, organizations, and governments the power to protect against the never-ending swell of cyber-threats.

References

- [1] - <https://www.techtarget.com/searchmobilecomputing/definition/mobile-malware> - Mobile malware
- [2] TryHackMe - [TryHackMe | Mobile Malware Analysis](#)
- [3] <https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html> - First mobile phone virus identified