

# **Mobile Malware Analysis**

**De Silva K.R.K.D**

## **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.

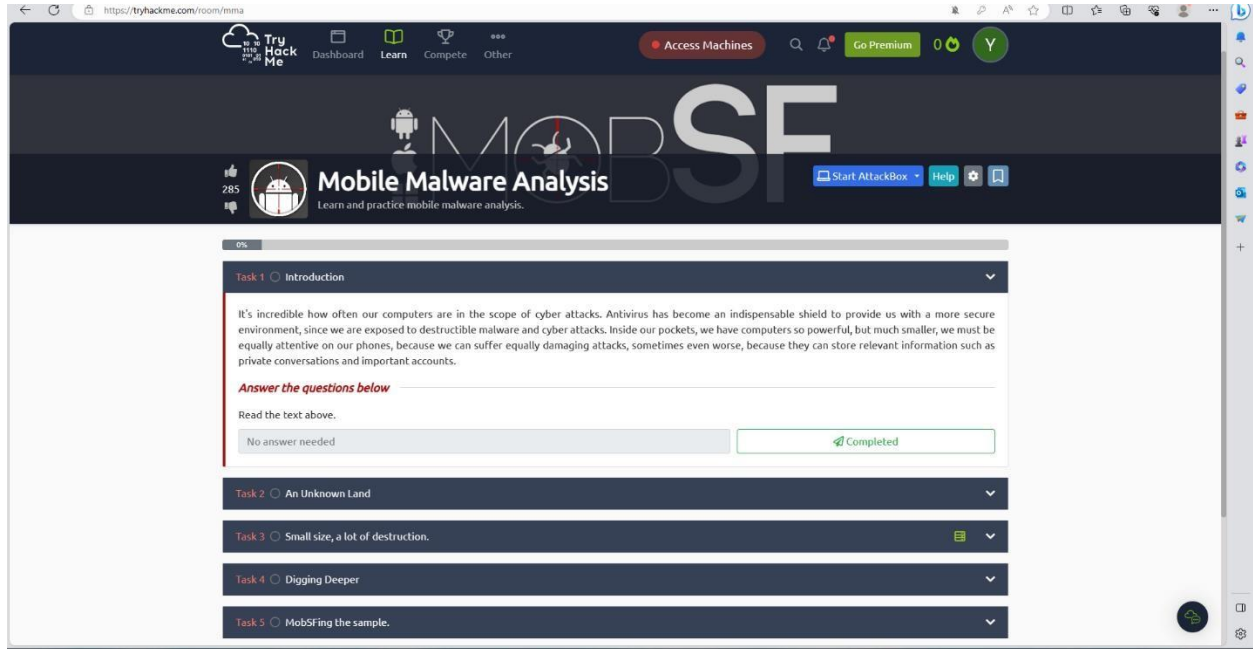
## Methodology

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



TryHackMe Dashboard Learn Compete Other Access Machines Go Premium 0 Y

# MOBSF

## Mobile Malware Analysis

Learn and practice mobile malware analysis.

Start AttackBox Help

0%

**Task 1** Introduction

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 destructible 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.

**Answer the questions below**

Read the text above.

No answer needed Completed

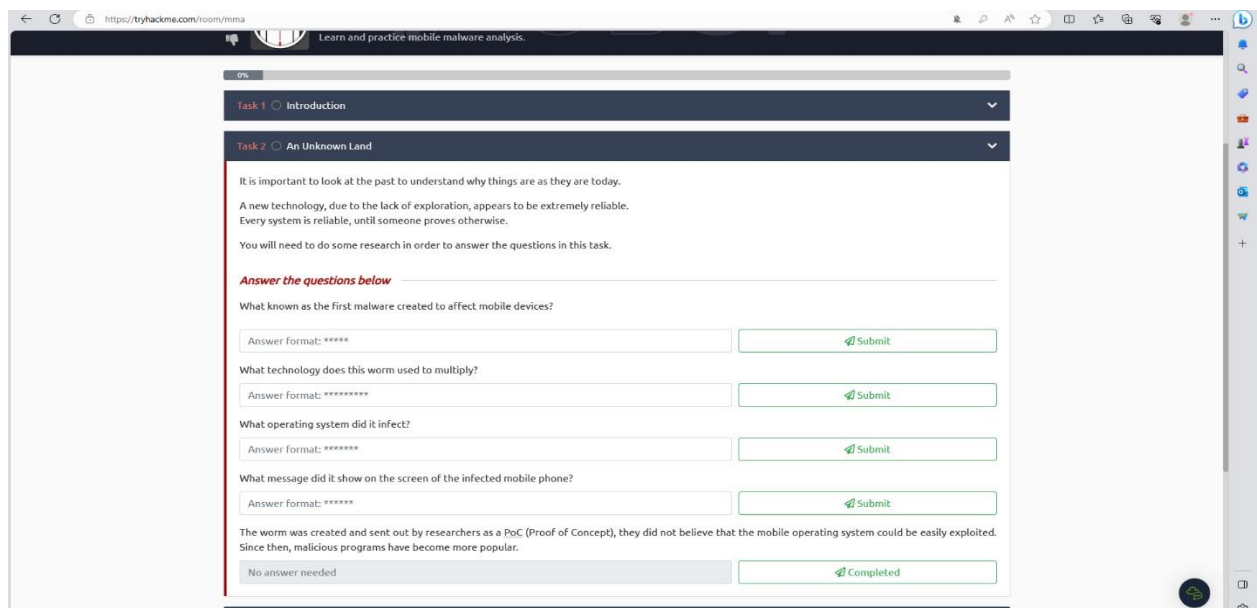
**Task 2** An Unknown Land

**Task 3** Small size, a lot of destruction.

**Task 4** Digging Deeper

**Task 5** MobSFing the sample.

## Task 2:



TryHackMe Dashboard Learn Compete Other Access Machines Go Premium 0 Y

# MOBSF

## Mobile Malware Analysis

Learn and practice mobile malware analysis.

Start AttackBox Help

0%

**Task 1** Introduction

**Task 2** An Unknown Land

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.

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?

Answer format: \*\*\*\*\* Submit

What technology does this worm used to multiply?

Answer format: \*\*\*\*\* Submit

What operating system did it infect?

Answer format: \*\*\*\*\* Submit

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

Answer format: \*\*\*\*\* Submit

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 Completed

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>

← → ↻ 🔍 https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html

Advertisement

June 16, 2004 – 11:17pm

Save Share A A A

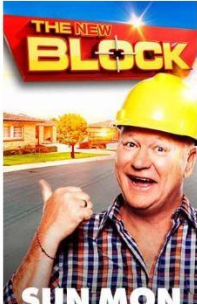
The first ever computer virus that can infect mobile phones has been discovered, according to anti-virus software developers.

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)

← → ↻ 🔍 https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html

The Sydney Morning Herald

SUBSCRIBE Log in

Advertisement

SUN MON TUES WED 9 NOW

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 proof that the technologies are now available to create viruses for mobile.

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

FROM OUR PARTNERS

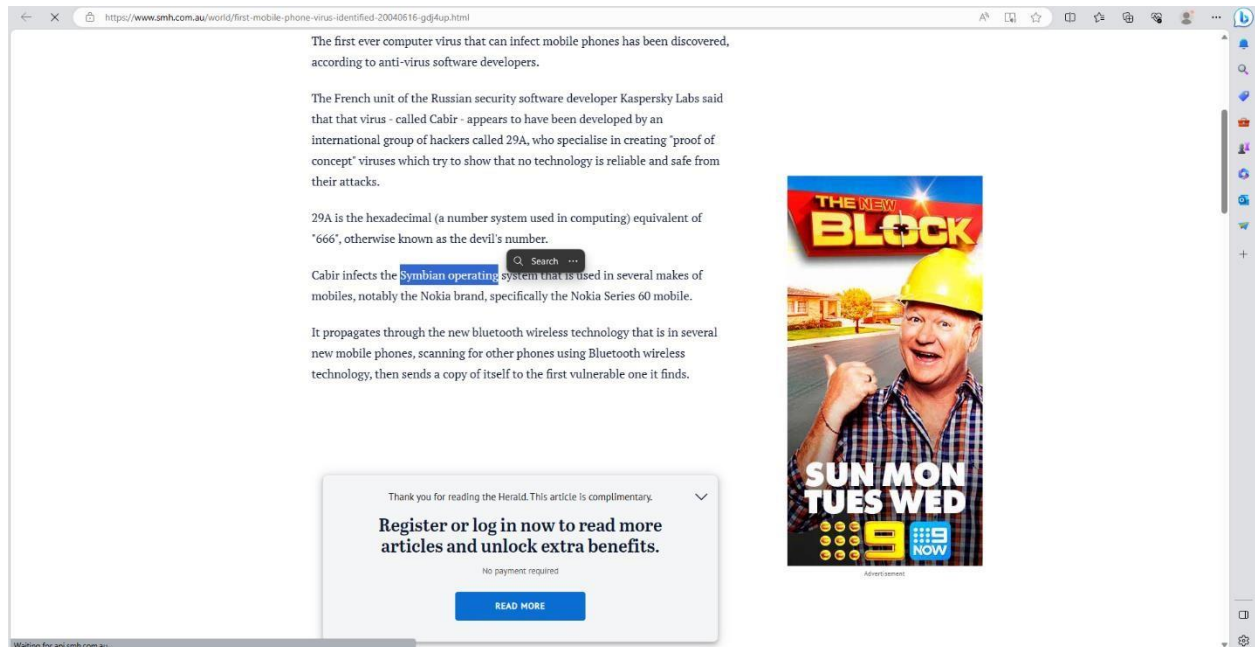
AFR

How AI is reshaping the corporate landscape

Brought to you by Salesforce



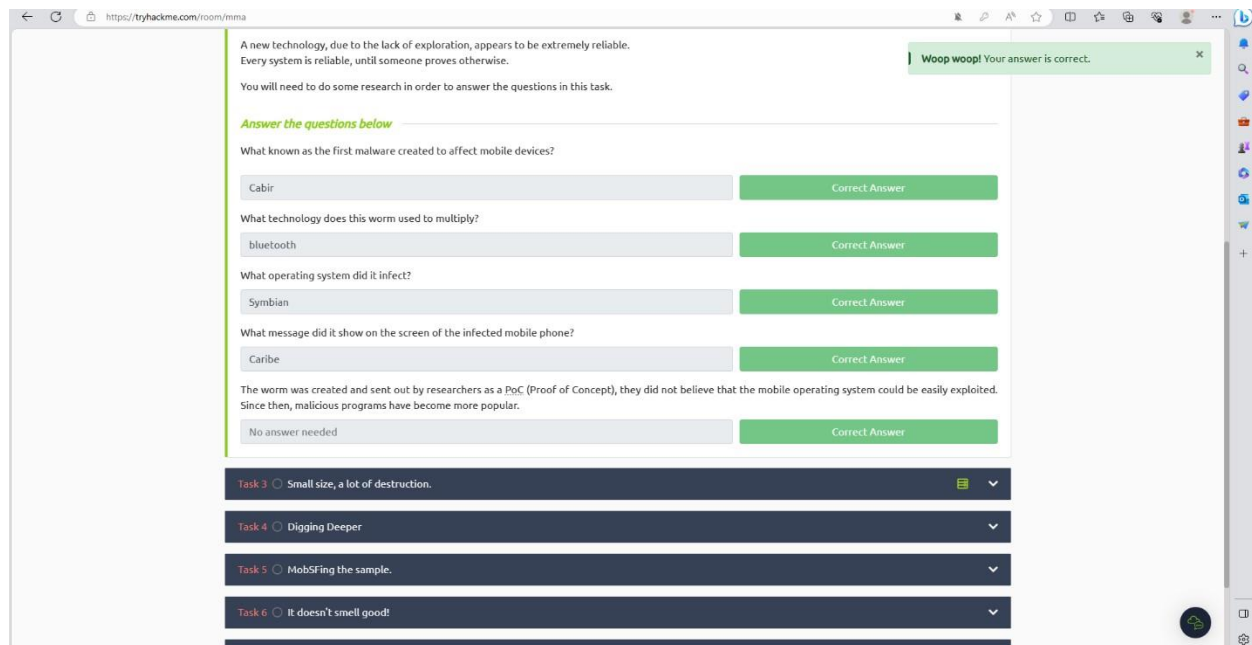
What operating system did it infect? : <https://www.smh.com.au/world/first-mobile-phone-virus-identified-20040616-gdj4up.html>



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>

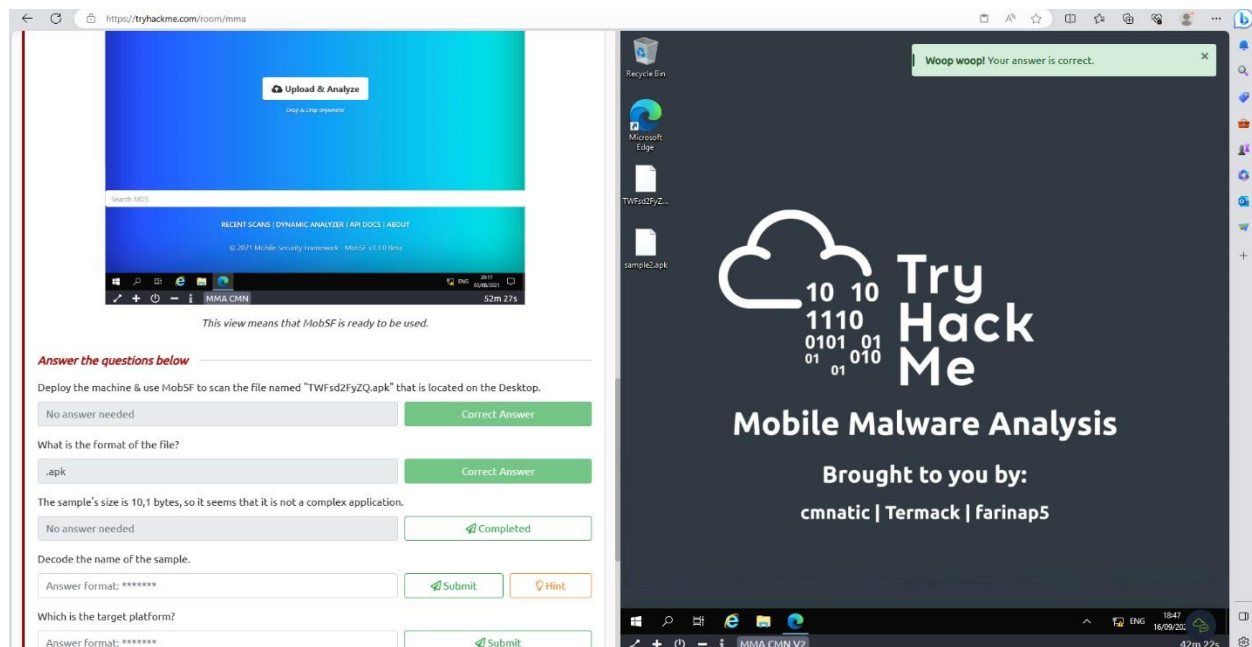




## Task 3

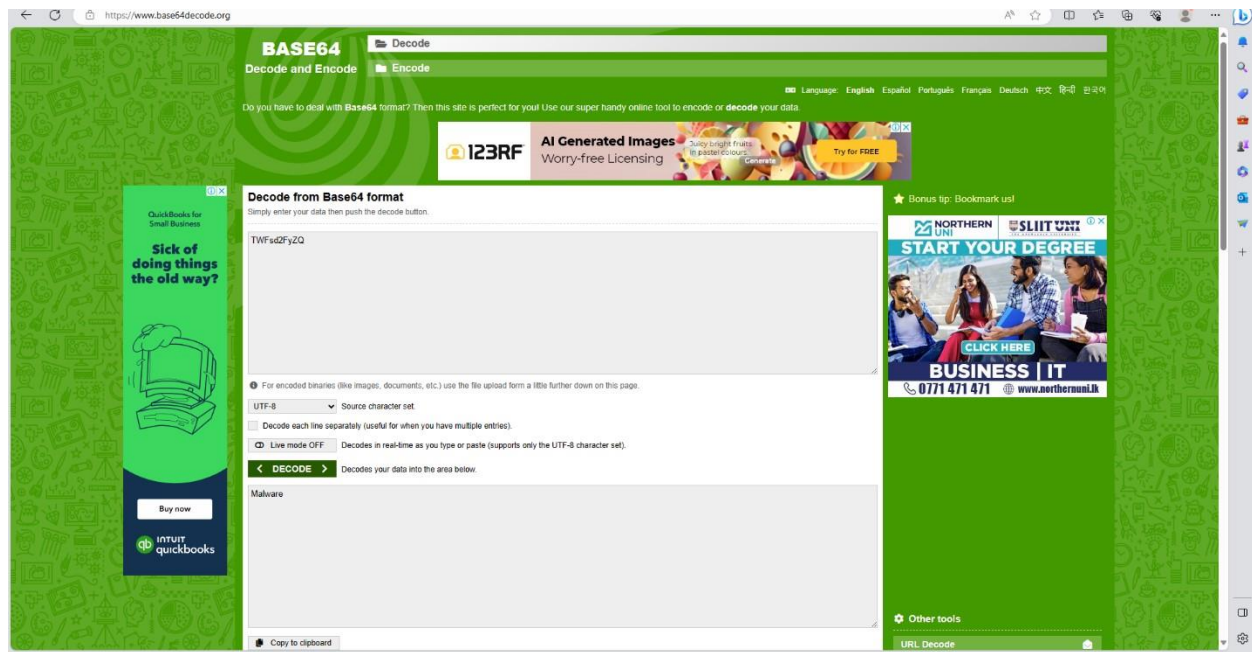
What is the format of the file?

`TWfsd2FyZQ.apk`

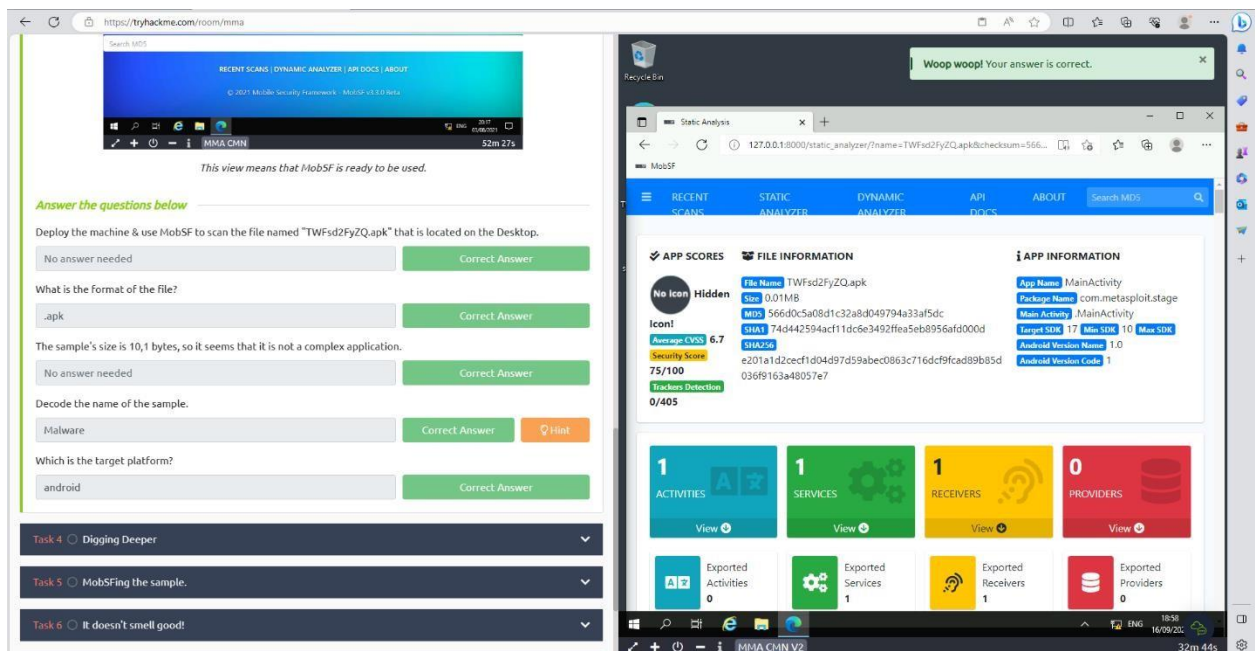




Decode the name of the sample. Decode Link : <https://www.base64decode.org/>



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



## Task 4

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

<https://www.virustotal.com/gui/file/e201a1d2cecf1d04d97d59abec0863c716dcf9cad89b85d036>

43 / 64

43 security vendors and no sandboxes flagged this file as malicious

e201a1d2cecf1d04d97d59abec0863c716dcf9cad89b85d036f9163a48057e7

TWfsd2FyZQ.apk

Size: 9.95 KB | Last Analysis Date: 5 hours ago | APK

android obfuscated apk runtime-modules reflection

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

Popular threat label: **hacktool:metasploit:metasploit** | Threat categories: hacktool, trojan, downloader | Family labels: meterpreter, metasploit, remotecode

Security vendors' analysis

Vendor	Detection
AhnLab-V3	PURAndroid.Metasploit.54109
Antiy-AVL	Trojan.Generic.ASMalw.ADDC
Avast	Android.Metasploit-G [PUP]
AVG	Android.Metasploit-G [PUP]
BitDefender	Application.HackTool.Meterpreter.AQR
Cyren	Malicious (score: 99)
DrWeb	Android.RemoteCode.6833
eScan	Application.HackTool.Meterpreter.AQR
F-Secure	Malware.ANDROID.Agent.FJNR.Gen
GData	Application.HackTool.Meterpreter.AQR
Ikarus	Trojan-Downloader.AndroidOS.Agent
Alibaba	HackTool.Android.Mesplit.07a03416
Arcabit	Application.HackTool.Meterpreter.AQR
Avast-Mobile	Android.Evo-gen [Trj]
Avira (no cloud)	ANDROID/Trojan.Old.FNAA.Gen
BitDefenderFalx	Android.Riskware.SMS.Send.RR
Cyren	Android.OS/Downloader.M.gen/Eldorado
Emsisoft	Application.HackTool.Meterpreter.AQR (B)
ESET-NOD32	A Variant Of Android/TrojanDownloader.A...
Fortinet	Android.Agent.LJNtr
Google	Detected
K7GW	Trojan-Downloader (004f8551)

What program was used to create the malware?

Task 3: Small size, a lot of destruction.

Task 4: Digging Deeper

Let's make a deeper analysis.

VirusTotal is an incredible service, this web site can give us the power of analyze a package with the database of more than seventy Anti-Virus, and the result is fast and accurate.

<https://www.virustotal.com/>

To analyze the file in VirusTotal, you will need the file hash, you can get it by using the powershell cmdlet "Get-FileHash" or you can analyze the file with MobSF and it will show the file hash (we will get back to this tool in the next task).

**Answer the questions below**

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

Android:Metasploit-G [PUP] | Correct Answer

What program was used to create the malware?

Metasploit | Correct Answer

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 | Completed

What is the package name?

Answer format: \*\*\* | Submit

What is the SHA-1 signature?

Answer format: \*\*\*\*\* | Submit

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

Static Analysis

127.0.0.1:8000/static\_analyzer/?name=TWfsd2FyZQ.apk&checksum=566...

APP SCORES

FILE INFORMATION

APP INFORMATION

1 ACTIVITIES | 1 SERVICES | 1 RECEIVERS | 0 PROVIDERS

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

What is the package name?

File type: Android executable (mobile android apk)

Magic: Zip archive data, at least v2.0 to extract, compression method=deflate

TrID: Java Archive (72.9%) 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

TWFd2FyZQ.apk  
t1.apk  
apkanall.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	b8ea694b40ac716715d2ecb270c10795974fa5e

#### Certificate Subject

Distinguished Name	C=US/O=Android/CN=Android Debug
Country Code	US/O=Android/CN=Android Debug

#### Certificate Issuer

What is the SHA-1 signature?

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

### Basic properties

MD5	56A0C5a0B6fC3a8d049794a33bf5dc
SHA-1	7044289ccf18c3a3421f7a8b8896a6d0000
SHA-256	e201a1d2cec1d04d97d59bec0863c716dcf9cad89b65d036f9163a48057e7
Vhash	bfc0bc094c1f41784483f854700c48a44
SSDEEP	192R31V2fTNTJeePWYHESB2+wwC7WYhqb5L4cIBLxJXulKAndy8100rHeWuJh7Wf2BdJXty
TLSH	TfY0229f7AA7A4a18F107ABBC50432B877DFAD3486219335DxCOEBC48152AACD33E764A
Permhsh	77e1e293a40f48f8e71a9a067810b7d787a4973ae9782bc504b4c9a99fec30
File type	Android executable (mobile android apk)
Magic	Zip archive data, at least v2.0 to extract, compression method=deflate
TrID	Java Archive (72.9%) 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

TWFd2FyZQ.apk  
t1.apk  
apkanall.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?

**Contacted IP addresses (45)**

IP	Detections	Autonomous System	Country
108.177.119.136	0 / 89	15169	US
108.177.119.139	0 / 89	15169	US
108.177.126.100	0 / 89	15169	US
108.177.126.102	0 / 89	15169	US
108.177.126.113	0 / 89	15169	US
108.177.126.132	0 / 89	15169	US
108.177.126.139	0 / 89	15169	US
108.177.126.140	0 / 89	15169	US
108.177.126.94	0 / 89	15169	US
108.177.127.101	0 / 89	15169	US

**Bundled Files (6)**

Scanned	Detections	File type	Name
2023-09-09	31 / 61	Android	classes.dex
2023-08-27	12 / 60	Android	AndroidManifest.xml
2022-10-19	0 / 61	Android	resources.arsc
?	?	?	META-INF/SIGNATURE.SF
?	?	?	META-INF/MANIFEST.MF
?	?	?	META-INF/SIGNATURE.SF

**Graph Summary**

How many permissions are there inside?

**Title:** MMA CHN V2  
**IP Address:** 10.10.46.153  
**Expires:** 17m 14s

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?  
74d442594dcf11dc6e3492ffa5eb8956afd000d 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

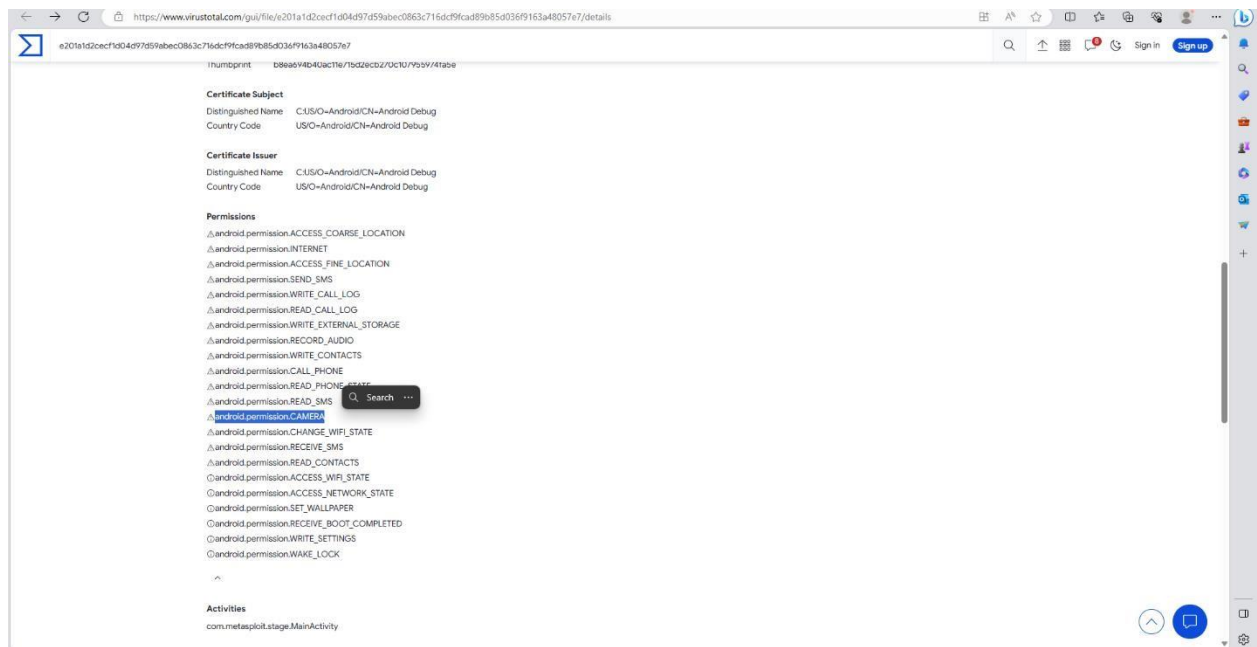
How many permissions are there inside?  
22 Correct Answer

Which permission allows the application to take pictures with the camera?  
Answer format: \*\*\*\*\* Submit

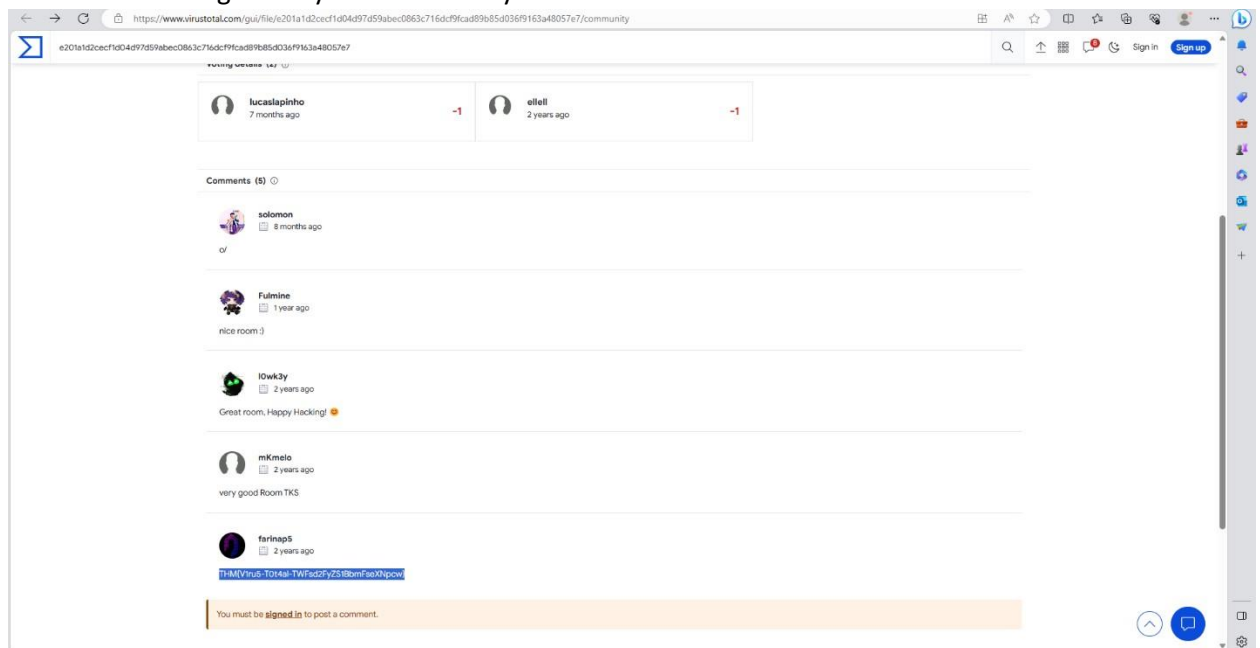
What is the message left by the community?  
Answer format: \*\*\*[\*\*\*\*\*] Submit

**Task 5** MobSfing the sample.

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



What is the message left by the community?





Title	IP Address	Expires
MMA CMN V2	10.10.46.153	15m 33s

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?

74d442594dcf11dc6e3492ffea5eb8956afd000d 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[V1ru5-T0t4al-TWfsd2FyZ51BbmfsXNpcw] 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/0x09cr4ck/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?

Task 5 MobSFing the sample.

Let's use MobSF (Mobile Security Framework) to make a deeper analysis of this file. MobSF is a software created to make a security focused analysis of Android and iOS files. It can check for misconfigurations, leaked data and much more in a mobile program.

This tool can be used for static and dynamic analysis, in this room we will focus only in the static analysis but you are 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?

1 Correct Answer

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

Static Analysis

Woop woopl Your answer is correct.

127.0.0.1:8000/static\_analyzer/?name=TWfSd2fY2Q.apk&checksum=566d0c5...

RECENT SCANS STATIC ANALYZER DYNAMIC ANALYZER API DOCS ABOUT Search MobSF

View AndroidManifest.xml View Source View Smali

Download Java Code Download Smali Code Download APK

**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: 2017-11-04 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: 9d7f41a21ae3b0e8c809456ae4bc0b  
sha1: b8ea694b4ac11e715d2ec270c107955974fa5e  
sha256: f542876dd1e4bc95dc49105707e4889c4053136f31ce6dcf89b5edd966af21ef  
sha512: 769427cb4fc12bcc3c1b8c668755dfa50a22ed19a700b53cc3ebf0a9a9c27c56d06385b349be309375fde52d27e2819f

Search:

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

Task 5 MobSFing the sample.

Let's use MobSF (Mobile Security Framework) to make a deeper analysis of this file. MobSF is a software created to make a security focused analysis of Android and iOS files. It can check for misconfigurations, leaked data and much more in a mobile program.

This tool can be used for static and dynamic analysis, in this room we will focus only in the static analysis but you are 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?

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 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

Static Analysis

Woop woopl Your answer is correct.

127.0.0.1:8000/static\_analyzer/?name=TWfSd2fY2Q.apk&checksum=566d0c5...

RECENT SCANS STATIC ANALYZER DYNAMIC ANALYZER API DOCS ABOUT Search MobSF

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: 9d7f41a21ae3b0e8c809456ae4bc0b  
sha1: b8ea694b4ac11e715d2ec270c107955974fa5e  
sha256: f542876dd1e4bc95dc49105707e4889c4053136f31ce6dcf89b5edd966af21ef  
sha512: 769427cb4fc12bcc3c1b8c668755dfa50a22ed19a700b53cc3ebf0a9a9c27c56d06385b349be309375fde52d27e2819f

Search:

STATUS	DESCRIPTION
Bad	Application is signed with v1 signature scheme, making it vulnerable to <b>Janus</b> 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 Previous 1 Next

What is the App name?





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 left screenshot shows a series of questions about Android security. The final question asks: "What is the severity of this configuration?" with the answer "medium".

The right screenshot shows the MobSF static analysis tool interface. It displays a table of issues found in the APK:

NO	ISSUE	SEVERITY	DESCRIPTION
1	Application Data can be Backed up [android:allowBackup] flag is missing.	medium	The flag [android:allowBackup] should be set to false. By default it is set to true and allows anyone to backup your application data via adb. It allows users who have enabled USB debugging to copy application data off of the device.
2	Broadcast Receiver [MainBroadcastReceiver] is not Protected. An intent-filter exists.	high	A Broadcast Receiver is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Broadcast Receiver is explicitly exported.
3	Service [MainService] is not Protected.	high	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Service is explicitly exported.

## 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?

The left screenshot shows a series of questions about the sample2.apk file. The final question asks: "What is the SHA-256 hash of the file?" with the answer "bd8cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94dl".

The right screenshot shows the MobSF static analysis tool interface. It displays the file information and app information for sample2.apk:

APP SCORES	FILE INFORMATION	APP INFORMATION
No icon Hidden	File Name: sample2.apk	App Name: Media Sync
Size: 1.06MB	MD5: 8d4b77fa3546149f25bd17357d41fb80	Package Name: seC.dujmehn.qdthet
SHA1: 7289737c1d462726abbe8933547702c130bbdccc	SHA256: bd8cda80aaee3e4a17e9967a1c062ac5c8e4aef7eaa3362f54044c2c94dl	Main Activity: seC.dujmehn.qdthet.Dujmehnpayd
AppScan CWS: 7.5	Target SDK: 9	Min SDK: 9
Security Score: 85/100	Android Version Name: 2.9.3	Android Version Code: 292
Trackers Detection: 0/405		

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

<https://www.virustotal.com/gui/file/bd8cda80aee3e4a17e9967a1c062ac5c8e4aefd7eaa3362f54042c94db52a/detection>

Security vendors' analysis	Threat categories	Family labels
AhnLab-V3	Trojan.Android.Agent.860476	Trojan.Spy.Android.Pegasus.ds33dbca
Antiy-AVL	Trojan(Spy)Android.Chrysaora	Android.Pegasus
Avast	Android.Obfusc-BM [Trj]	Android.Evo-gen [Trj]
AVG	Android.Obfusc-BM [Trj]	ANEKOD.SpyAgent.FKMN.Gen
BitDefender	Trojan.GenericKD.46667348	Android.Trojan.Pegasus.F
Cynet	Malicious (score: 99)	Android.Siggen.Susp.3172
Ensisoft	Trojan.GenericKD.46667348 (B)	Trojan.GenericKD.46667348
ESET-NOD32	Multiple Detections	Malware.ANDROID.SpyAgent.FKMN.Gen
Fortinet	Android.Obfusc.NB/tr	Trojan.GenericKD.46667348
Google	Detected	Trojan.AndroidOS.Obfusc

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

bd8cda80aee3e4a17e9967a1c062ac5c8e4aefd7eaa3362f540442c94db52a

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.

127.0.0.1:8000/static\_analyzer/?name=sample2.apk&checksum=bd4b77...

**APP SCORES**

File Name: sample2.apk

Size: 1.06MB

MD5: 8d4b77fa3546149f25bd17357d41fbf0

SHA1: f289737c1d462726abbe8933547702c130bbdccc

SHA256: bd8cda80aee3e4a17e9967a1c062ac5c8e4aefd7eaa3362f540442c94db52a

Security Score: 7.5

85/100

Tracked Detection: 0/405

**FILE INFORMATION**

App Name: Media Sync

Package Name: sec.dujmehn.qdthet

Main Activity: sec.dujmehn.qdthet.Dujmehnpqd

Target SDK: 9

Min SDK: 9

Max SDK: 9

Android Version Name: 2.9.3

Android Version Code: 292

**APP INFORMATION**

3 ACTIVITIES

5 SERVICES

8 RECEIVERS

0 PROVIDERS

Exported Activities: 0

Exported Services: 1

Exported Receivers: 5

Exported Providers: 0

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))

The screenshot displays a mobile security analysis interface. On the left, a task list shows 'Task 5' completed and 'Task 6' in progress. The main area contains a series of questions and answers related to the analysis of a sample named 'sample2.apk'. The questions and answers are:

- What is the SHA-256 hash of the file?  
bd8cda80aace3e4a17e9967a1c062ac5c8e4aefd7eaa3362f54044c2c94dl
- After finding the sample on VirusTotal, what does the "Avast" anti-virus engine recognizes it as?  
Android:Obfus-BM [Trj]
- With what we have, try to find out the name of the sample.  
pegasus
- 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
- 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
- If we search the name we found of the malware in MITRE ATT&CK (<https://attack.mitre.org/>), we can find some interesting information.  
Completed
- What is the ID of the MITRE ATT&CK that is associated with our sample?

On the right, a 'Static Analysis' window shows the results of a MobSF analysis. It includes sections for 'APP SCORES', 'FILE INFORMATION', and 'APP INFORMATION'. The 'APP INFORMATION' section shows the app name 'Media Sync', package name 'seC.dujmeh.qdthet', and version '2.9.3'. Below this, there are four cards representing different types of activities: '3 ACTIVITIES', '5 SERVICES', '8 RECEIVERS', and '0 PROVIDERS'. Each card has a 'View' button and an 'Exported' button.

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

The screenshot shows the MITRE ATT&CK Software page for 'Pegasus for Android'. The page is titled 'Pegasus for Android' and includes a description: 'Pegasus for Android is the Android version of malware that has reportedly been linked to the NSO Group. [1] [2] The iOS version is tracked separately under Pegasus for iOS.' The page also features a sidebar with a list of software categories, a table of associated software descriptions, and a table of techniques used.

**Associated Software Descriptions**

Name	Description
Chrysaor	[1] [2]

**Techniques Used**

Domain	ID	Name	Use
Mobile	T1429	Audio Capture	Pegasus for Android has the ability to record device audio. [1]
Mobile	T1645	Compromise Client Software Binary	Pegasus for Android attempts to modify the device's system partition. [1]
Mobile	T1471	Event Trimming/Execution	Pegasus for Android listens for the device's broadcast intent in order to maintain persistence and activate its

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

Platform	Technique ID	Technique Name	Description
Mobile	T1429	Audio Capture	Pegasus for Android has the ability to record device audio. <sup>[1]</sup>
Mobile	T1645	Compromise Client Software Binary	Pegasus for Android attempts to modify the device's system partition. <sup>[1]</sup>
Mobile	T1624	Event Triggered Execution: Broadcast Receivers	Pegasus for Android listens for the <code>android.intent.action.BOOT_COMPLETED</code> broadcast intent in order to maintain persistence and activate its functionality at device boot time. <sup>[1]</sup>
Mobile	<b>T1404</b>	Exploitation for Privilege Escalation	Pegasus for Android attempts to exploit well-known Android OS vulnerabilities to escalate privileges. <sup>[1]</sup>
Mobile	T1644	Out of Band Data	Pegasus for Android uses SMS for command and control. <sup>[1]</sup>
Mobile	T1636	Protected User Data: Calendar Entries	Pegasus for Android accesses calendar entries. <sup>[1]</sup>
		Protected User Data: Call Log	Pegasus for Android accesses call logs. <sup>[1]</sup>
		Protected User Data: Contact List	Pegasus for Android accesses contact list information. <sup>[1]</sup>
Mobile	T1418	Software Discovery	Pegasus for Android accesses the list of installed applications. <sup>[1]</sup>
Mobile	T1409	Stored Application Data	Pegasus for Android accesses sensitive data in files, such as messages stored by the WhatsApp, Facebook, and Twitter applications. It also has the ability to access arbitrary filenames and retrieve directory listings. <sup>[1]</sup>
Mobile	T1422	System Network: Configuration Discovery	Pegasus for Android checks if the device is on Wi-Fi, a cellular network, and is roaming. <sup>[1]</sup>
Mobile	T1512	Video Capture	Pegasus for Android has the ability to take pictures using the device camera. <sup>[1]</sup>

**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)

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

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?

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

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

No answer needed

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)

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

Answer format: \*\*\*\*/\*\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*

This software has several features that make the identification and the processes it performs to explore the target, harder to handle, even when it is being analyzed.

No answer needed

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?

Answer format: \*\*\*\*/\*\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*/\*\*\*\*

Static Analysis

Woop woop! Your answer is correct.

127.0.0.1:8000/static\_analyzer/?name=sample2.apk&checksum=8d4b77...

RECENT SCANS STATIC ANALYZER DYNAMIC ANALYZER API ENCS

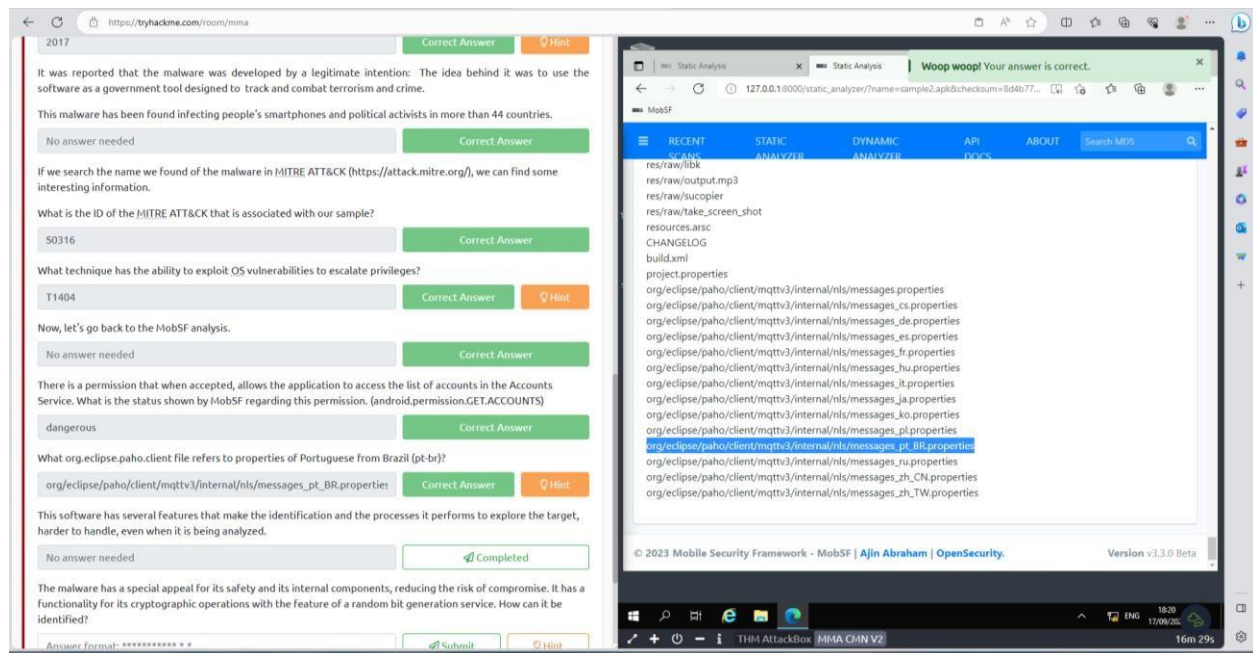
android.permission.GET\_ACCOUNTS  list accounts Allows access to the list of accounts in the Accounts Service.

android.permission.GET\_PACKAGE\_SIZE  measure application storage space Allows an application to find out the space used by any package.

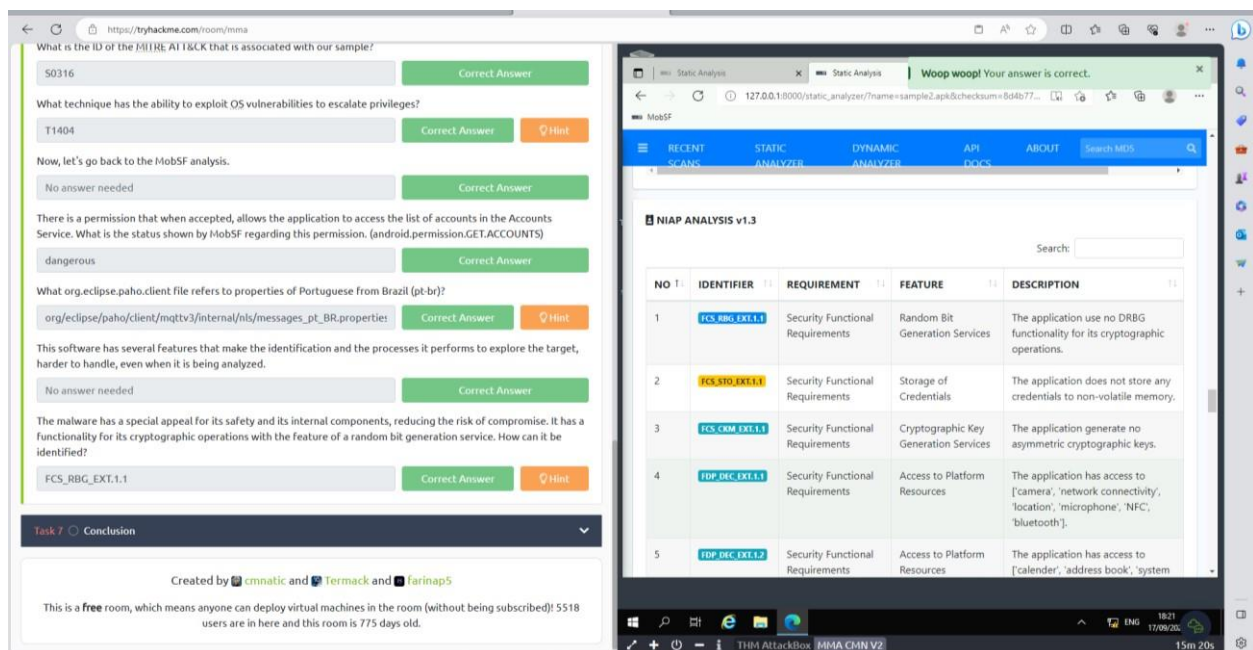
android.permission.GET\_TASKS  retrieve running applications Allows application to retrieve information about currently and

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





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?



Task 4 Digging Deeper

Task 5 MobSfing the sample.

Task 6 It doesn't smell good!

Task 7 Conclusion

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-instg>  
<https://attack.mitre.org/matrices/mobile/android/>  
<https://attack.mitre.org/matrices/mobile/ios/>

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

If you have any feedback, feel free to contact me on discord: farinap5#4535

Answer the questions below

Thank you for your participation!

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.

Static Analysis

127.0.0.1:8000/static\_analyzer/?name=sample?ark/r/thermroom=fr4dn77...

Woop woop! Your answer is correct.

RECENT | STATIC ANALYZER | DYNAMIC ANALYZER | API | ABOUT

NIAP ANALYSIS v1.3

Search:

NO	IDENTIFIER	REQUIREMENT	FEATURE	DESCRIPTION
1	FCS_RBG_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_EXT.1.1	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 ['calendar', 'address book', 'system

THM AttackBox MMA C4M V2

18:21 12/06/2023 15m 14s

Task 4 Digging Deeper

Task 5 MobSfing the sample.

Task 6 It doesn't smell good!

Task 7 Conclusion

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-instg>  
<https://attack.mitre.org/matrices/mobile/android/>  
<https://attack.mitre.org/matrices/mobile/ios/>

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

If you have any feedback, feel free to contact me on discord: farinap5#4535

Answer the questions below

Thank you for your participation!

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.

Woop woop! Your answer is correct.

Congratulations

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

THM AttackBox MMA C4M V2

15m 11s

22 | Page

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