Cyber Forensics Case Study Report

Mobile Malware: V-Bucks App Forensic Analysis

By Joshua Hartzfeld, Parker Cummings, Liam Dumbell, and John Vitali

Overview

Our team, specializing in Mobile Malware research and prevention, leads the Cyber Forensic branch at our company. Recently, we received a concerning call from a customer regarding a suspicious app named "V-Bucks" that their 9-year-old son had unknowingly installed on their smartphone. The customer expressed uncertainty about whether this app had introduced a virus onto the phone.

After a thorough discussion and analysis of the customer's report, our team decided to undertake the case. Our goal was to conduct a comprehensive examination of the smartphone to assess the potential threat posed by the app. If deemed necessary, we planned to develop a strategy for safely removing the app from the device, ensuring the customer's peace of mind and the security of their smartphone.

Objectives

Our primary objectives for this case are as follows:

- Safely acquire the device to ensure no outside variables affect the integrity of our investigation
- Once in our hands, obtain as many logs from the device as possible, such as call logs, message logs, location data, device information, etc. These may be useful later when we're attempting to understand what the app is doing to the device.
- Obtain a Disk Image of the device
- Analyze the Disk Image in Autopsy to determine a timeline of relevant events
- Extract the .apk from the Disk Image and do further analysis within an Android Emulator as well as JADX
- Compile our findings and conclude if the app is malicious in any way.
- Give a list of steps to the customer to ensure the app is properly removed and the device is safe to use again

Acquisition

Before we could handle the device physically, we had to make sure the customer handled the device properly so we could get as much of the information from the device as possible. We told the customer to do three things before bringing the device in.

- Make sure the device doesn't lose power before we can extract volatile data (RAM). Meaning don't let the device die and don't shut it off. If the device powers off, all of that data is lost.
- Do not attempt to connect the device to a computer to back up files or charge the phone due to a potential chance of spreading malware to an uninfected device.
- Try to gather any resources relating to the installation of the app, for example: The time of installation, where it was installed, what the app is called, etc.

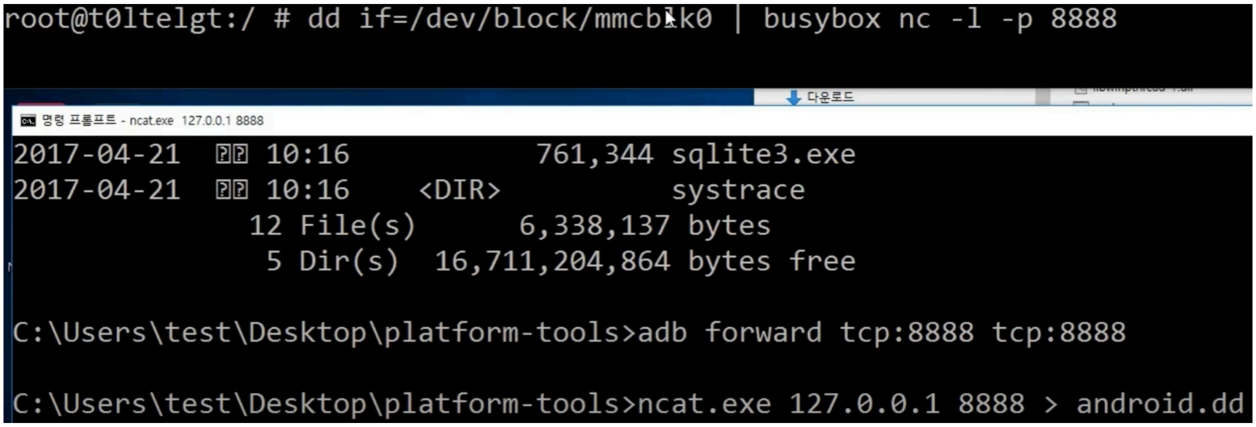
Once the device has been handed over to us, we can begin the isolation process of the device. Since the phone would still be connected to cell towers, we must isolate it. We can do this in a few different ways, but we will use a Paraben Wireless StrongHold Bag so we can continue to keep the device powered. If we chose any other isolation method, the battery would drain much quicker because the device is in roaming mode (which it still is in the Paraben bag, but we can charge it). We can now begin the data acquisition, which includes, but is not limited to:

- SIM card data extraction
    - User Data, Contacts, Carrier and Network Data, International Mobile Subscriber Identity (IMSI), etc
- Call data
    - Numbers dialed, calls received, etc
- Message information
- Location information
- Disk Dump of the entire phone

It is important to note that if the device loses power, PINs or any other access codes may be required to view files.

Disk Dump Autopsy Analysis

We Firstly can acquire the Disk Dump of the device using adb. adb provides access to a Unix shell that you can use to run a variety of commands and scripts on a device to download files to your local machine.



```
root@t0ltelgt:/ # dd if=/dev/block/mmcblk0 | busybox nc -l -p 8888
```

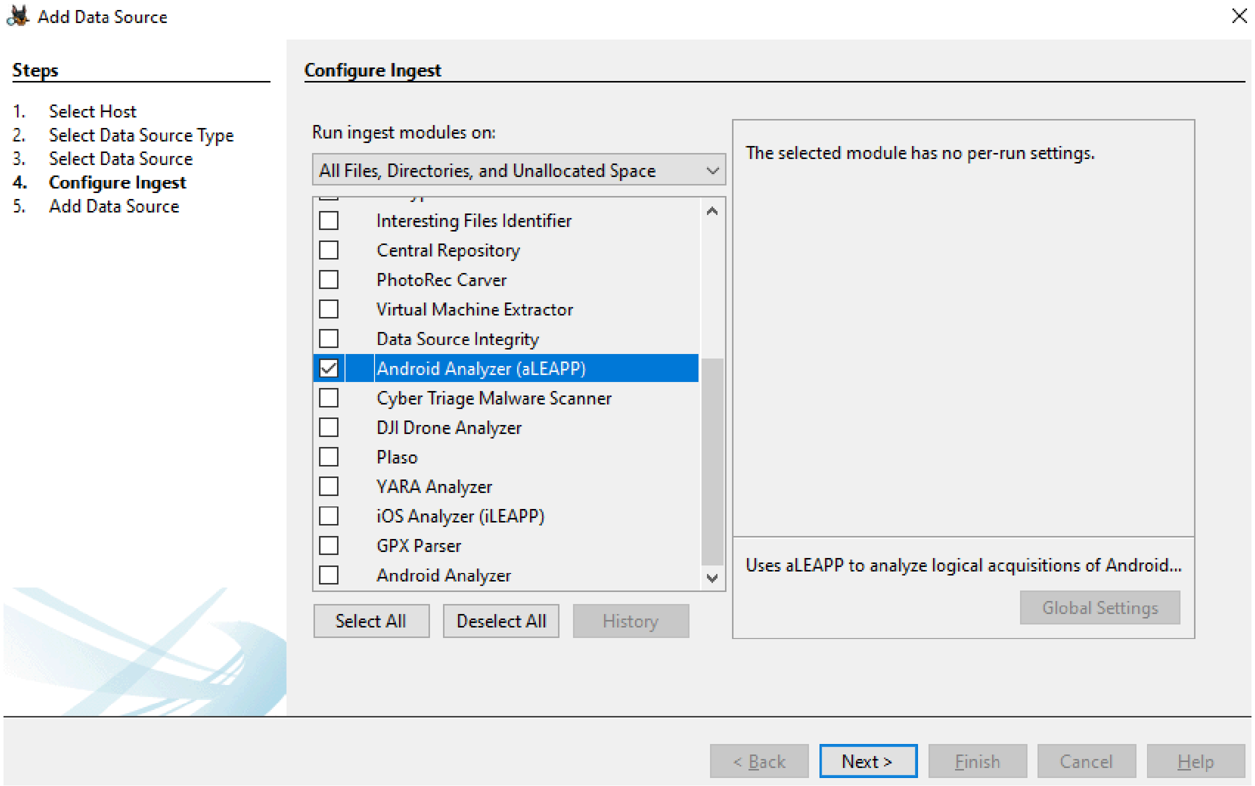
```
2017-04-21  ?? 10:16           761,344 sqlite3.exe
2017-04-21  ?? 10:16    <DIR>          systrace
              12 File(s)      6,338,137 bytes
               5 Dir(s)  16,711,204,864 bytes free

C:\Users\test\Desktop\platform-tools>adb forward tcp:8888 tcp:8888

C:\Users\test\Desktop\platform-tools>ncat.exe 127.0.0.1 8888 > android.dd
```

Once we have the image on one of our machines we can start setting up Autopsy to properly analyze the disk dump of the client's phone.

We've chosen to use Android Analyzer in this situation, which runs aLEAPP, a tool that specializes in parsing Android Logs Events (https://github.com/abrignoni/aLEAPP)



Add Data Source

Steps

1. Select Host
2. Select Data Source Type
3. Select Data Source
4. **Configure Ingest**
5. Add Data Source

Configure Ingest

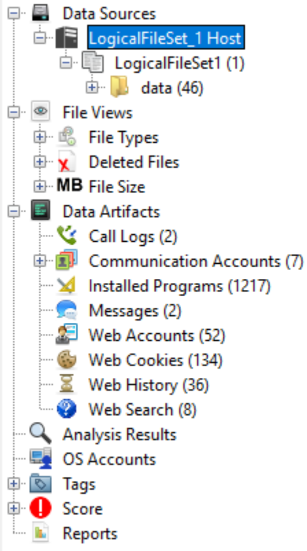
Run ingest modules on: All Files, Directories, and Unallocated Space

□ Interesting Files Identifier
□ Central Repository
□ PhotoRec Carver
□ Virtual Machine Extractor
□ Data Source Integrity
☑ Android Analyzer (aLEAPP)
□ Cyber Triage Malware Scanner
□ DJI Drone Analyzer
□ Plaso
□ YARA Analyzer
□ iOS Analyzer (iLEAPP)
□ GPX Parser
□ Android Analyzer

The selected module has no per-run settings.

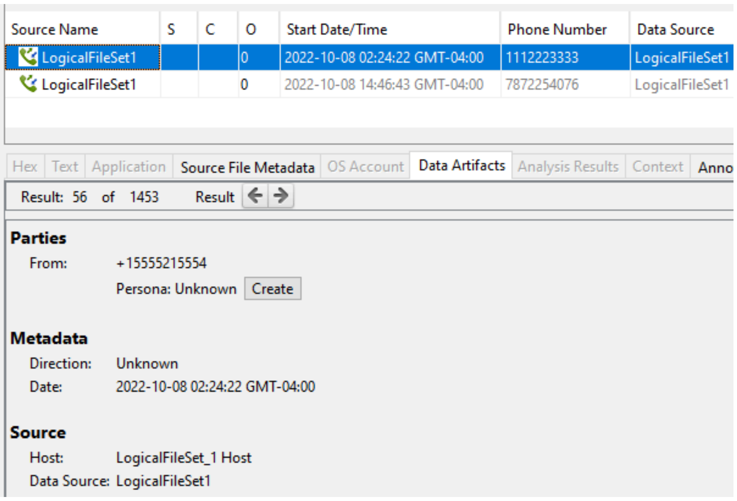
Uses aLEAPP to analyze logical acquisitions of Android...

When we run aLEAPP against this image, we get some interesting data that we can view



Here we can see logs related to various parts of the device

Firstly, the call logs show the number associated with the device which is good for reference



Source Name	S	C	O	Start Date/Time	Phone Number	Data Source
LogicalFileSet1			0	2022-10-08 02:24:22 GMT-04:00	1112223333	LogicalFileSet1
LogicalFileSet1			0	2022-10-08 14:46:43 GMT-04:00	7872254076	LogicalFileSet1

Result: 56 of 1453 Result

Parties

From: +15555215554

Persona: Unknown Create

Metadata

Direction: Unknown

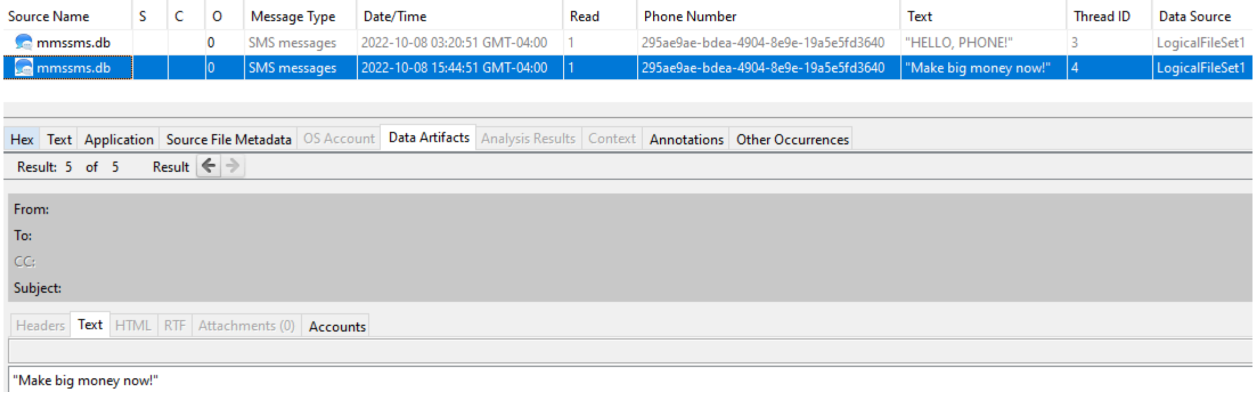
Date: 2022-10-08 02:24:22 GMT-04:00

Source

Host: LogicalFileSet_1 Host

Data Source: LogicalFileSet1

We can also see the text messages saved on the device



Source Name	S	C	O	Message Type	Date/Time	Read	Phone Number	Text	Thread ID	Data Source
mmssms.db			0	SMS messages	2022-10-08 03:20:51 GMT-04:00	1	295ae9ae-bdea-4904-8e9e-19a5e5fd3640	"HELLO, PHONE!"	3	LogicalFileSet1
mmssms.db			0	SMS messages	2022-10-08 15:44:51 GMT-04:00	1	295ae9ae-bdea-4904-8e9e-19a5e5fd3640	"Make big money now!"	4	LogicalFileSet1

Result: 5 of 5 Result

From:

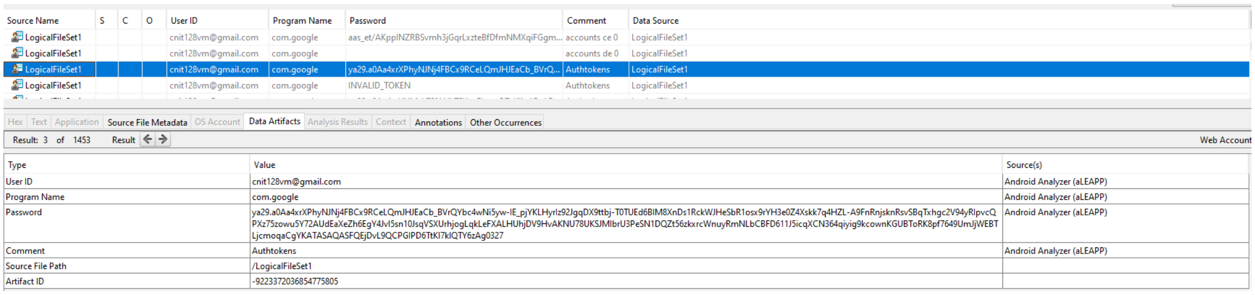
To:

CC:

Subject:

"Make big money now!"

We can also view the email address associated with the device as well as their web history
cnit129vm@gmail.com

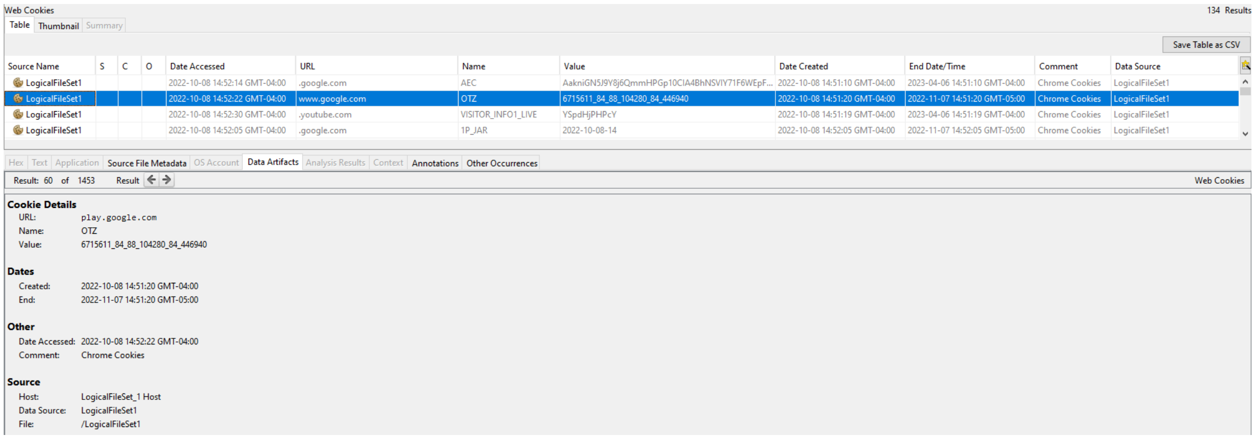


Source Name	S	C	O	User ID	Program Name	Password	Comment	Data Source
LogicalFileSet1				cnit128vm@gmail.com	com.google	aas_et/AKppINZRBSvmh3jGqrLxzteBfDfmNMXqiFGgm...	accounts ce 0	LogicalFileSet1
LogicalFileSet1				cnit128vm@gmail.com	com.google		accounts de 0	LogicalFileSet1
LogicalFileSet1				cnit128vm@gmail.com	com.google	ya29.a0Aa4xrXPhyNJNj4FBCx9RCeLQmJHJEaCb_BVrQ...	Authtokens	LogicalFileSet1
LogicalFileSet1				cnit128vm@gmail.com	com.google	INVALID_TOKEN	Authtokens	LogicalFileSet1

Result: 3 of 1453 Result

Type	Value	Source(s)
User ID	cnit128vm@gmail.com	Android Analyzer (aLEAPP)
Program Name	com.google	Android Analyzer (aLEAPP)
Password	ya29.a0Aa4xrXPhyNJNj4FBCx9RCeLQmJHJEaCb_BVrQYbc4wNi5yw-IE_pjYKLHyrlz92JgqDX9ttbj-T0TUEd6BIM8XnDs1RckWJHeSbR1osx9rYH3e0Z4Xskk7q4HZL-A9FnRnjsknRsvSBqTxhgc2V94yRlpvcQ PXz75zowu5Y72AUdEaXeZh6EgY4Jvl5sn10JsqVSXUrhjogLqkLeFXALHUhjDV9HvAKNU78UKSJMlbrU3PeSN1DQZt56zkxrcWnuyRmNLbCBFD611J5icqXCN364qiyig9kcownKGUBToRK8pf7649UmJjWEBT LjcmoqaCgYKATASAQASFQEjDvL9QCPGIPD6TtKI7kIQTY6zAg0327	Android Analyzer (aLEAPP)
Comment	Authtokens	Android Analyzer (aLEAPP)
Source File Path	/LogicalFileSet1	
Artifact ID	-9223372036854775805	

We can also see what web cookies the device has set, which has references to the Google Play Store



Source Name	S	C	O	Date Accessed	URL	Name	Value	Date Created	End Date/Time	Comment	Data Source
LogicalFileSet1				2022-10-08 14:52:14 GMT-04:00	.google.com	AEC	AakniGN5J9Y8j6QmmHPGp10CIA4BhNSVIY71F6WEpF...	2022-10-08 14:51:10 GMT-04:00	2023-04-06 14:51:10 GMT-04:00	Chrome Cookies	LogicalFileSet1
LogicalFileSet1				2022-10-08 14:52:22 GMT-04:00	www.google.com	OTZ	6715611_84_88_104280_84_446940	2022-10-08 14:51:20 GMT-04:00	2022-11-07 14:51:20 GMT-05:00	Chrome Cookies	LogicalFileSet1
LogicalFileSet1				2022-10-08 14:52:30 GMT-04:00	.youtube.com	VISITOR_INFO1_LIVE	YSpdHjPHPcY	2022-10-08 14:51:19 GMT-04:00	2023-04-06 14:51:19 GMT-04:00	Chrome Cookies	LogicalFileSet1
LogicalFileSet1				2022-10-08 14:52:05 GMT-04:00	.google.com	1P_JAR	2022-10-08-14	2022-10-08 14:52:05 GMT-04:00	2022-11-07 14:52:05 GMT-05:00	Chrome Cookies	LogicalFileSet1

Result: 60 of 1453 Result

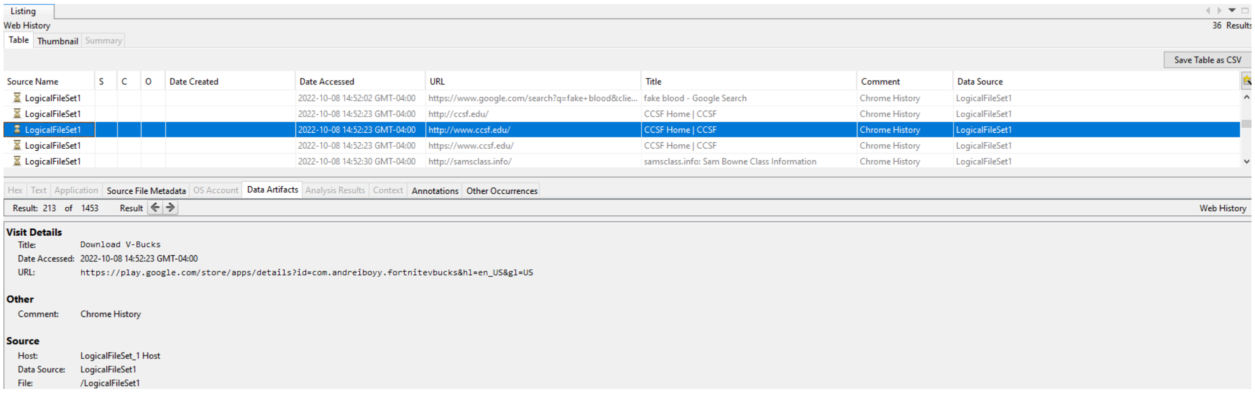
Cookie Details
URL: play.google.com
Name: OTZ
Value: 6715611_84_88_104280_84_446940

Dates
Created: 2022-10-08 14:51:20 GMT-04:00
End: 2022-11-07 14:51:20 GMT-05:00

Other
Date Accessed: 2022-10-08 14:52:22 GMT-04:00
Comment: Chrome Cookies

Source
Host: LogicalFileSet_1 Host
Data Source: LogicalFileSet1
File: /LogicalFileSet1

This is also reflected in the web history/searches



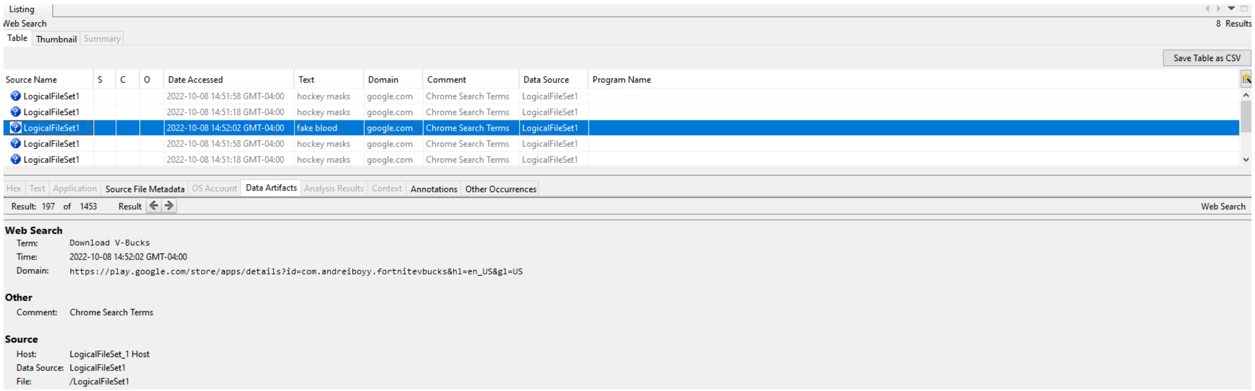
Source Name	S	C	O	Date Created	Date Accessed	URL	Title	Comment	Data Source
LogicalFileSet1					2022-10-08 14:52:02 GMT-04:00	https://www.google.com/search?q=fake+blood&clie...	fake blood - Google Search	Chrome History	LogicalFileSet1
LogicalFileSet1					2022-10-08 14:52:23 GMT-04:00	http://ccsf.edu/	CCSF Home | CCSF	Chrome History	LogicalFileSet1
LogicalFileSet1					2022-10-08 14:52:23 GMT-04:00	http://www.ccsf.edu/	CCSF Home | CCSF	Chrome History	LogicalFileSet1
LogicalFileSet1					2022-10-08 14:52:23 GMT-04:00	https://www.ccsf.edu/	CCSF Home | CCSF	Chrome History	LogicalFileSet1
LogicalFileSet1					2022-10-08 14:52:30 GMT-04:00	http://samsclass.info/	samsclass.info: Sam Bowne Class Information	Chrome History	LogicalFileSet1

Result: 213 of 1453 Result

Visit Details
Title: Download V-Bucks
Date Accessed: 2022-10-08 14:52:23 GMT-04:00
URL: https://play.google.com/store/apps/details?id=com.andreiboyy.fortnitevbucks&hl=en_US&gl=US

Other
Comment: Chrome History

Source
Host: LogicalFileSet_1 Host
Data Source: LogicalFileSet1
File: /LogicalFileSet1



Listing

Web Search

Table | Thumbnail | Summary

8 Results

Save Table as CSV

Source Name	S	C	O	Date Accessed	Text	Domain	Comment	Data Source	Program Name
LogicalFileSet1				2022-10-08 14:51:58 GMT-04:00	hockey masks	google.com	Chrome Search Terms	LogicalFileSet1	
LogicalFileSet1				2022-10-08 14:51:18 GMT-04:00	hockey masks	google.com	Chrome Search Terms	LogicalFileSet1	
LogicalFileSet1				2022-10-08 14:52:02 GMT-04:00	fake blood	google.com	Chrome Search Terms	LogicalFileSet1	
LogicalFileSet1				2022-10-08 14:51:58 GMT-04:00	hockey masks	google.com	Chrome Search Terms	LogicalFileSet1	
LogicalFileSet1				2022-10-08 14:51:18 GMT-04:00	hockey masks	google.com	Chrome Search Terms	LogicalFileSet1	

Hex | Text | Application | Source File Metadata | OS Account | Data Artifacts | Analysis Results | Context | Annotations | Other Occurrences

Result: 197 of 1453 Result ← →

Web Search

Web Search

Term: Download V-Bucks

Time: 2022-10-08 14:52:02 GMT-04:00

Domain: https://play.google.com/store/apps/details?id=com.andreiboyy.fortnitevbucks&hl=en_US&gl=US

Other

Comment: Chrome Search Terms

Source

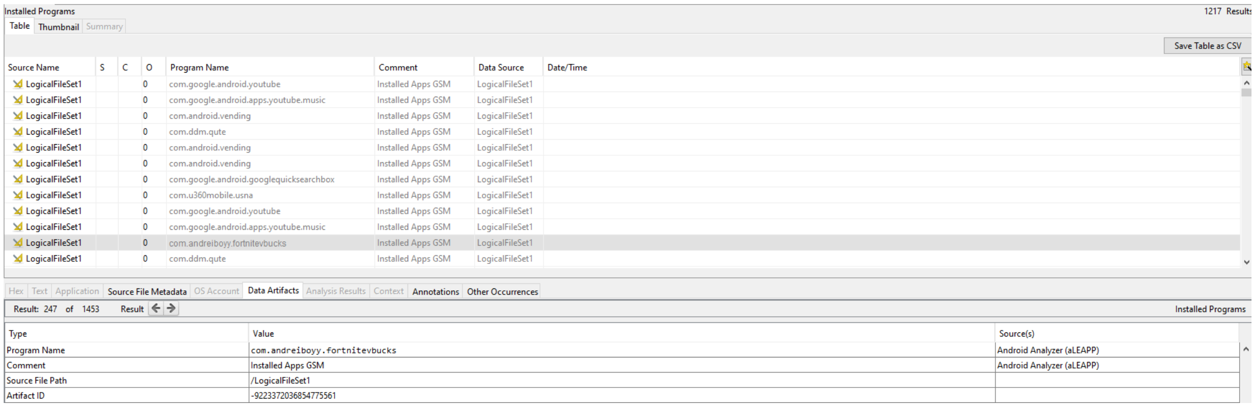
Host: LogicalFileSet_1 Host

Data Source: LogicalFileSet1

File: /LogicalFileSet1

Something interesting about this is that it also gives us the username of the developer of the app we're investigating, andreiboyy.

We can also use this link to source the app downloaded and download a copy for ourselves, but just to be sure we'll investigate the Installed Programs section and look for V-Bucks



Installed Programs

Table | Thumbnail | Summary

1217 Result

Save Table as CSV

Source Name	S	C	O	Program Name	Comment	Data Source	Date/Time
LogicalFileSet1			0	com.google.android.youtube	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.google.android.apps.youtube.music	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.android.vending	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.ddm.qute	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.android.vending	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.android.vending	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.google.android.googlequicksearchbox	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.u360mobile.usna	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.google.android.youtube	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.google.android.apps.youtube.music	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.andreiboyy.fortnitevbucks	Installed Apps GSM	LogicalFileSet1	
LogicalFileSet1			0	com.ddm.qute	Installed Apps GSM	LogicalFileSet1	

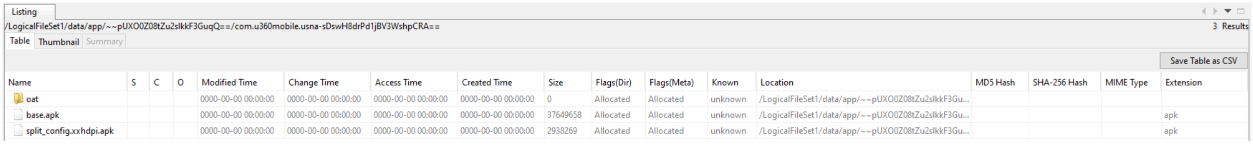
Hex | Text | Application | Source File Metadata | OS Account | Data Artifacts | Analysis Results | Context | Annotations | Other Occurrences

Result: 247 of 1453 Result ← →

Installed Programs

Type	Value	Source(s)
Program Name	com.andreiboyy.fortnitevbucks	Android Analyzer (aLEAPP)
Comment	Installed Apps GSM	Android Analyzer (aLEAPP)
Source File Path	/LogicalFileSet1	
Artifact ID	-9223372036854775561	

Double-clicking on the logical file set brings us to the data section of the disk dump where we can extract the app as an .apk file and analyze it further.



Listing

/LogicalFileSet1/data/app/~~pUXO0Z08tZu2slkkF3GuqQ==/com.u360mobile.usna-sDswH8drPd1jBV3WshpCRA==

3 Results

Table | Thumbnail | Summary

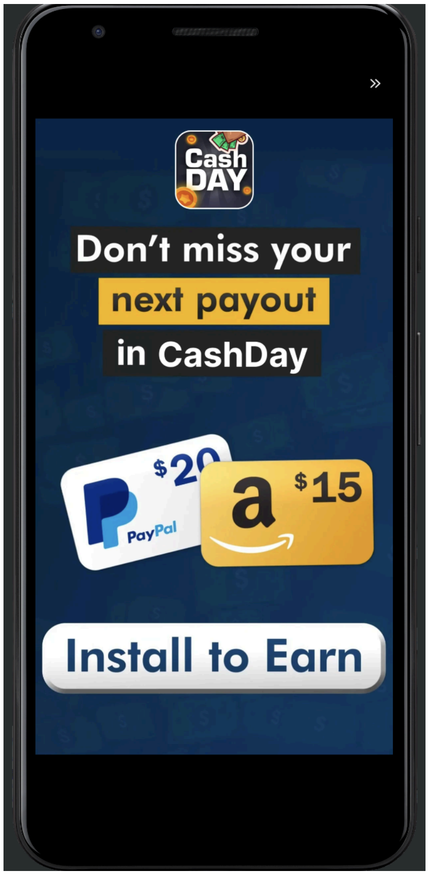
Save Table as CSV

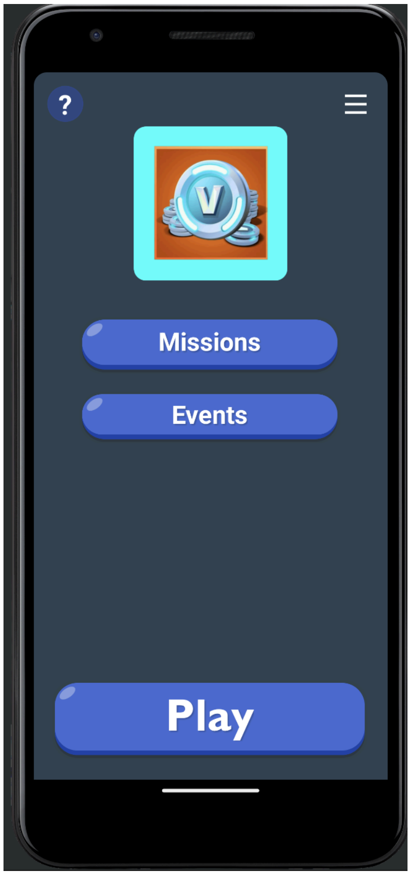
Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location	MD5 Hash	SHA-256 Hash	MIME Type	Extension
oat				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0	Allocated	Allocated	unknown	/LogicalFileSet1/data/app/~~pUXO0Z08tZu2slkkF3Gu...				
base.apk				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	37649658	Allocated	Allocated	unknown	/LogicalFileSet1/data/app/~~pUXO0Z08tZu2slkkF3Gu...				apk
split_config.xxhdpi.apk				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	2938269	Allocated	Allocated	unknown	/LogicalFileSet1/data/app/~~pUXO0Z08tZu2slkkF3Gu...				apk

Emulation Overview and Analysis:

Once we were able to obtain the device and confirm the app was correct, we decided it would be best to view the app under an emulated version of an Android Phone using Android Studio's built-in emulation function

Viewing the app in emulation mode, it already seems very suspicious and has some key clues that show this app could be malicious. First off, it has a very lazy UI design with simple buttons and only a picture of the V-bucks symbol present. The purpose of this app is to watch videos (ads), share accounts on both Twitter and Facebook, take quizzes, and play mini-games to earn these in-app coins and supposedly redeem them for V-bucks.





As you can see, almost every single attempt to access a new page results in a new ad, usually clocking in at no less than 30 seconds. We believe that this is a tactic used to keep the phone awake and keep users on the app so that it can perform actions under the hood that the user cannot see. This could be a form of adware, made to keep users going down the rabbit hole of constantly clicking on ads to get what they believe is free V-bucks. To view what the app may be doing under the hood, we as a team decided to view the .apk (source code for Android apps) in a program for analysis. The program we decided to use is called JADX, it is a popular choice for examining applications during mobile forensics. In the following sections the permissions, source code, and other aspects of the application will be examined in this program and discussed.

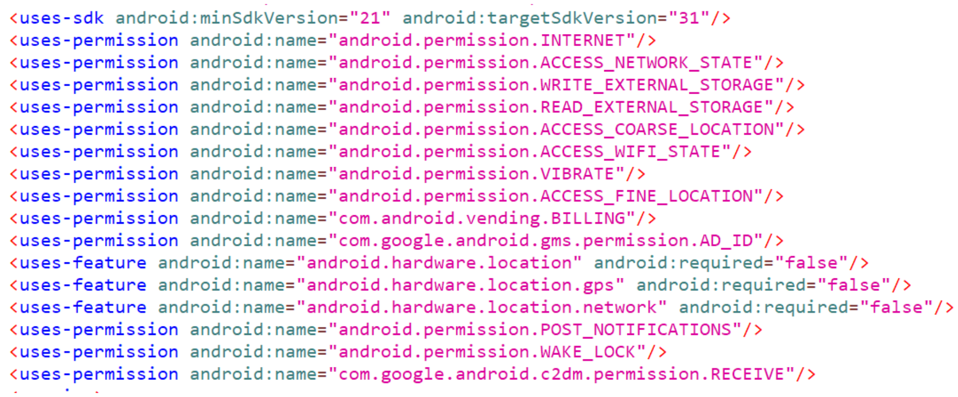
Jadx Analysis:

JADX is a command line and GUI tool for producing Java source code from Android Dex and Apk files. This allows for Static Analysis of Android .apk files.
This allows us to view the Android app's:

- Source Code and API Usage
- Permissions and Third Party Libraries

Analyzing each one of these sections will give us a better understanding of the purpose of an Android app and if it has the potential to be malicious.

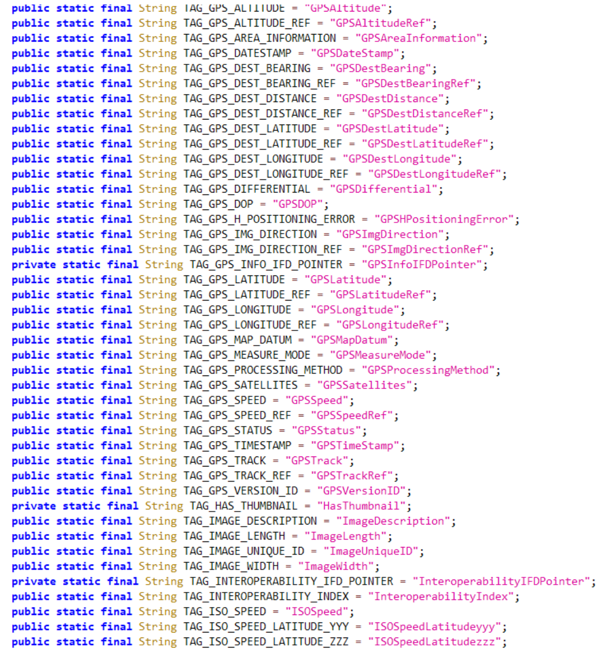
Here's a list of the app's permissions:



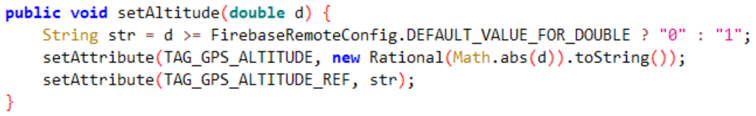
```
<uses-sdk android:minSdkVersion="21" android:targetSdkVersion="31"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="com.android.vending.BILLING"/>
<uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
<uses-feature android:name="android.hardware.location" android:required="false"/>
<uses-feature android:name="android.hardware.location.gps" android:required="false"/>
<uses-feature android:name="android.hardware.location.network" android:required="false"/>
<uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
```

.WRITE_EXTERNAL_STORAGE and **.READ_EXTERNAL_STORAGE**: These permissions are commonly used for apps that need to save data or access files on the device's external storage (such as photos, videos, voice recordings, messages, etc).
.INTERNET: This permission allows the app to access the internet and send sensitive user data (like personal information or device details) to remote servers controlled by the app developer without the user's knowledge.
.ACCESS_COARSE_LOCATION and **.ACCESS_FINE_LOCATION**: These permissions allow the app to access the device's location. This can be used to track users without their consent or for targeted advertising.
.BILLING: This permission is related to in-app purchases and billing through the Google Play Store. As this app is marketed towards children, the app can likely use this permission to trick users into making unauthorized purchases or subscriptions.
.WAKE_LOCK: Probably the most suspicious of all, allows the app to prevent the device from going into sleep mode. The app can abuse this permission to keep the device active which allows for prolonged tracking.

Knowing this, we can now look into the source code looking for terms related to spying. ExifInterface defines many variables related to the device's location and contains functions that set these values to the current state of the device



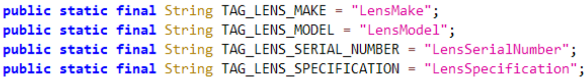
```
public static final String TAG_GPS_ALTITUDE = "GPSAltitude";
public static final String TAG_GPS_ALTITUDE_REF = "GPSAltitudeRef";
public static final String TAG_GPS_AREA_INFORMATION = "GPSAreaInformation";
public static final String TAG_GPS_DATESTAMP = "GPSDateStamp";
public static final String TAG_GPS_DEST_BEARING = "GPSDestBearing";
public static final String TAG_GPS_DEST_BEARING_REF = "GPSDestBearingRef";
public static final String TAG_GPS_DEST_DISTANCE = "GPSDestDistance";
public static final String TAG_GPS_DEST_DISTANCE_REF = "GPSDestDistanceRef";
public static final String TAG_GPS_DEST_LATITUDE = "GPSDestLatitude";
public static final String TAG_GPS_DEST_LATITUDE_REF = "GPSDestLatitudeRef";
public static final String TAG_GPS_DEST_LONGITUDE = "GPSDestLongitude";
public static final String TAG_GPS_DEST_LONGITUDE_REF = "GPSDestLongitudeRef";
public static final String TAG_GPS_DIFFERENTIAL = "GPSDifferential";
public static final String TAG_GPS_DOP = "GPSDOP";
public static final String TAG_GPS_H_POSITIONING_ERROR = "GPSHPositioningError";
public static final String TAG_GPS_IMG_DIRECTION = "GPSImgDirection";
public static final String TAG_GPS_IMG_DIRECTION_REF = "GPSImgDirectionRef";
private static final String TAG_GPS_INFO_IFD_POINTER = "GPSInfoIFDPointer";
public static final String TAG_GPS_LATITUDE = "GPSLatitude";
public static final String TAG_GPS_LATITUDE_REF = "GPSLatitudeRef";
public static final String TAG_GPS_LONGITUDE = "GPSLongitude";
public static final String TAG_GPS_LONGITUDE_REF = "GPSLongitudeRef";
public static final String TAG_GPS_MAP_DATUM = "GPSMapDatum";
public static final String TAG_GPS_MEASURE_MODE = "GPSMeasureMode";
public static final String TAG_GPS_PROCESSING_METHOD = "GPSProcessingMethod";
public static final String TAG_GPS_SATELLITES = "GPSSatellites";
public static final String TAG_GPS_SPEED = "GPSSpeed";
public static final String TAG_GPS_SPEED_REF = "GPSSpeedRef";
public static final String TAG_GPS_STATUS = "GPSStatus";
public static final String TAG_GPS_TIMESTAMP = "GPSTimeStamp";
public static final String TAG_GPS_TRACK = "GPSTrack";
public static final String TAG_GPS_TRACK_REF = "GPSTrackRef";
public static final String TAG_GPS_VERSION_ID = "GPSVersionID";
private static final String TAG_HAS_THUMBNAIL = "HasThumbnail";
public static final String TAG_IMAGE_DESCRIPTION = "ImageDescription";
public static final String TAG_IMAGE_LENGTH = "ImageLength";
public static final String TAG_IMAGE_UNIQUE_ID = "ImageUniqueID";
public static final String TAG_IMAGE_WIDTH = "ImageWidth";
private static final String TAG_INTEROPERABILITY_IFD_POINTER = "InteroperabilityIFDPointer";
public static final String TAG_INTEROPERABILITY_INDEX = "InteroperabilityIndex";
public static final String TAG_ISO_SPEED = "ISOSpeed";
public static final String TAG_ISO_SPEED_LATITUDE_YYY = "ISOSpeedLatitudeyyy";
public static final String TAG_ISO_SPEED_LATITUDE_ZZZ = "ISOSpeedLatitudezzz";
```



```
public void setAltitude(double d) {
    String str = d >= FirebaseRemoteConfig.DEFAULT_VALUE_FOR_DOUBLE ? "0" : "1";
    setAttribute(TAG_GPS_ALTITUDE, new Rational(Math.abs(d)).toString());
    setAttribute(TAG_GPS_ALTITUDE_REF, str);
}
```

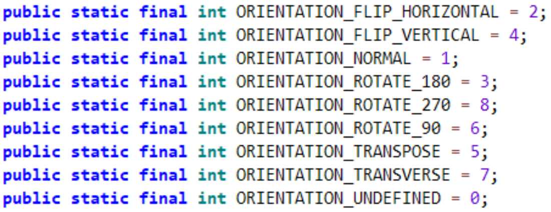
This is also true for other properties of the device:

Camera Specifications



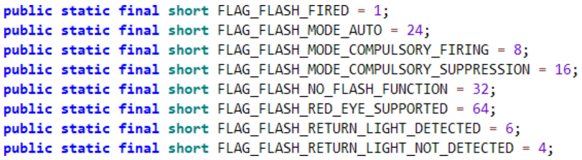
```
public static final String TAG_LENS_MAKE = "LensMake";
public static final String TAG_LENS_MODEL = "LensModel";
public static final String TAG_LENS_SERIAL_NUMBER = "LensSerialNumber";
public static final String TAG_LENS_SPECIFICATION = "LensSpecification";
```

Device Orientation



```
public static final int ORIENTATION_FLIP_HORIZONTAL = 2;
public static final int ORIENTATION_FLIP_VERTICAL = 4;
public static final int ORIENTATION_NORMAL = 1;
public static final int ORIENTATION_ROTATE_180 = 3;
public static final int ORIENTATION_ROTATE_270 = 8;
public static final int ORIENTATION_ROTATE_90 = 6;
public static final int ORIENTATION_TRANSPOSE = 5;
public static final int ORIENTATION_TRANSVERSE = 7;
public static final int ORIENTATION_UNDEFINED = 0;
```

Flash Usage



```
public static final short FLAG_FLASH_FIRED = 1;
public static final short FLAG_FLASH_MODE_AUTO = 24;
public static final short FLAG_FLASH_MODE_COMPULSORY_FIRING = 8;
public static final short FLAG_FLASH_MODE_COMPULSORY_SUPPRESSION = 16;
public static final short FLAG_FLASH_NO_FLASH_FUNCTION = 32;
public static final short FLAG_FLASH_RED_EYE_SUPPORTED = 64;
public static final short FLAG_FLASH_RETURN_LIGHT_DETECTED = 6;
public static final short FLAG_FLASH_RETURN_LIGHT_NOT_DETECTED = 4;
```

The fact that the app is recording all of this information without any reason for using it within the app is highly suspicious. This indicates that this app may be collecting user data to sell to third parties for a profit.

Findings:

We’ve determined the Android app installed on the customer’s device is very likely to be a combination of a Scam, SpyWare, and AdWare. The App portrays itself as a free way to make in-game money, but there is no possible way to do so. This is indicative of the app being a scam. The app serves an excessive amount of targeted Ads to generate profit. This is indicative of the app being a form of AdWare. The App has an extensive list of suspicious permissions that it uses to record information on the device and its users. This information can be sold to third parties for a profit or used to track users of the app for an extended period of time. (SpyWare)

Recommendations:

As a team, we’d recommend uninstalling the V-Bucks App from the device and doing these extra steps to ensure the customer’s device is safe:

- Restart the device in Safe Mode
- Clear App Cache and Data
- Install and use reputable Antivirus or Anti-Malware software from trusted developers
- Change Account Credentials
- Back up important files/data and perform a factory reset if issues persist
- In the case of the customer, regularly monitor the device from now on to ensure no suspicious apps are installed in the future.

Conclusion:

Once the case was closed, the team met for an after-action report meeting to discuss the case and what results were found. Overall, this app is most definitely employing tactics like spyware and adware to perform actions on a device unbeknownst to the user. With the insane amount of permissions the app has access to, specifically access to write/read storage, access the internet, track GPS location, and keep the device alive, it is apparent that the app plans to get users on it and keep them on it. While the users are going down the adware rabbit hole of different ads and videos in order to get these free V-bucks, the app can continuously track the device, perform actions via the internet including offloading data in the storage unit of the app, and more. It is our conclusion that this app is malicious, and apps created by the same author as this one as well as similar apps should be avoided at all costs. All in all, most apps that are promising a reward for no payment at all are not products, but they are making the user one. They will reveal and steal data from the device and export it using permissions given to the app via the device. It is smart to avoid scam apps like this and to always use your best judgment. It is also a good idea to place a password on the payment verification or installation of an app on a mobile device, so children who may not know better cannot perform these installations without a user’s knowledge, like the case of our client. Lastly, be wary of any app published under the username andreiboyy on the Google Play Store, as we have verified as a team that he is a scam artist.