Android Application testing with Burp Suite

OVERVIEW

The main objective of this project was to analyze malicious apk files through Burp Suite. For this task , we decided to use Kali Linux as our host machine as it comes with Burp Suite pre-installed. We began our testing by configuring and installing Android Studio into our Virtual Machine. Upon installing Android Studio , we chose Google Pixel 3XL as our Android Emulator and we chose Android 5.1 Lollipop as our Android Version to test upon. Once the emulator was set up , we had to configure Burp suite to intercept both HTTPS and HTTP traffic. A certificate had to be installed on to the emulator in order for Burp-Suite to intercept the entire traffic flow. Once everything was configured and set up correctly , we began testing out malicious android applications by installing it into our Android emulator. We used a GitHub repository to get the malware apk samples used for testing. We tried numerous apk files from the Github Repository and VirusShare as well but only a few were compatible and successful in our testing. We began intercepting the entire traffic flow from our Android Emulator via BurpSuite. We discovered interesting requests made by these malicious applications and also discovered unusual HTTP requests which were successfully captured by BurpSuite. In addition to capturing traffic from these malicious apk files , we also explored some other features of BurpSuite , where we could manipulate the intercepted traffic and craft a response to our liking.

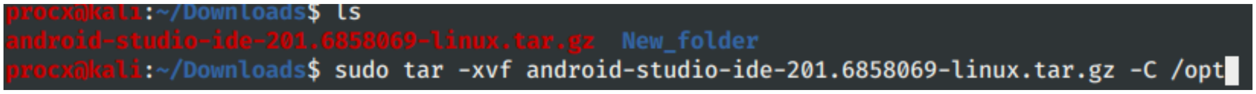
TOOLS AND SERVICES USED

- ★ Kali Linux - A Debian based Linux distribution designed for digital forensics and penetration testing.
- ★ VMWare Workstation - An application used for running multiple OS as Virtual Machines.
- ★ Android Studio - Android Studio is the official integrated development environment for Google's Android operating system designed specifically for Android development.
- ★ Alacarte - Alacarte is an easy-to-use menu editor for GNOME that can add and edit new entries and menus.
- ★ BurpSuite - BurpSuite is an integrated platform for performing security testing of various applications.
- ★ Android Debug Bridge(ADB) - Android Debug Bridge, is a command-line utility included with Google's Android SDK. ADB can control your device over USB from a computer, copy files back and forth, install and uninstall apps, run shell commands, and more.

1) Downloading and configuring Android Studio and AVD Emulator on Kali Linux

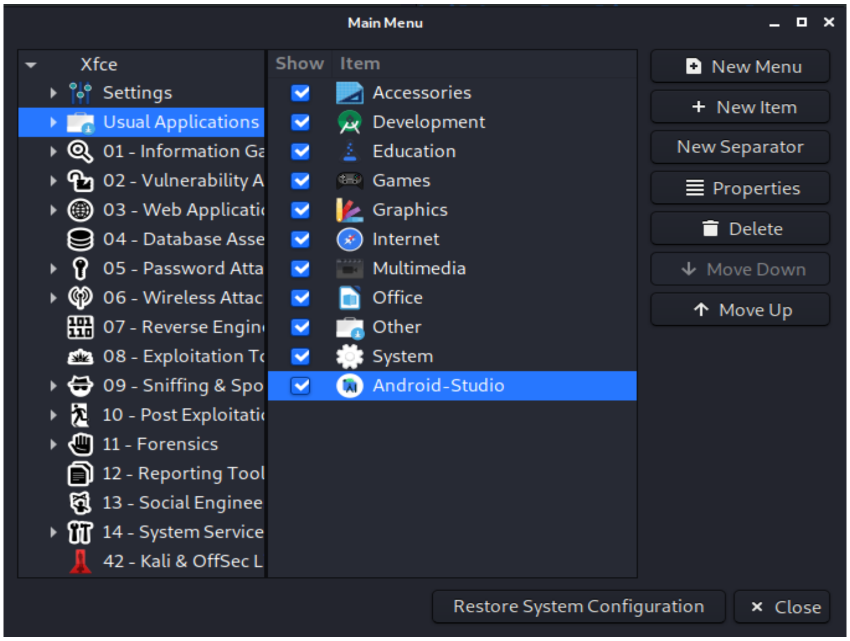
https://developer.android.com/studio

We downloaded Android studio from the official website. Upon download , we extracted the files and stored the android studio in our /opt directory.



```
procx@kali:~/Downloads$ ls
android-studio-ide-201.6858069-linux.tar.gz  New_folder
procx@kali:~/Downloads$ sudo tar -xvf android-studio-ide-201.6858069-linux.tar.gz -C /opt
```

We used an application called Alacarte to map Android Studio into the Main Menu for quick access.



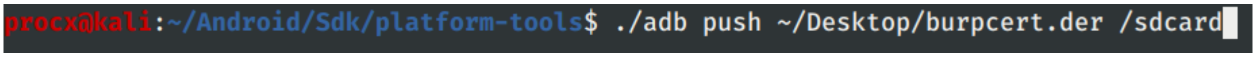
Once the application was mapped successfully , we were able to easily access Android Studio without having to manually run the executable each time.

2) Configuring AVD in Android studio and installing Burpsuite certificates for proxy

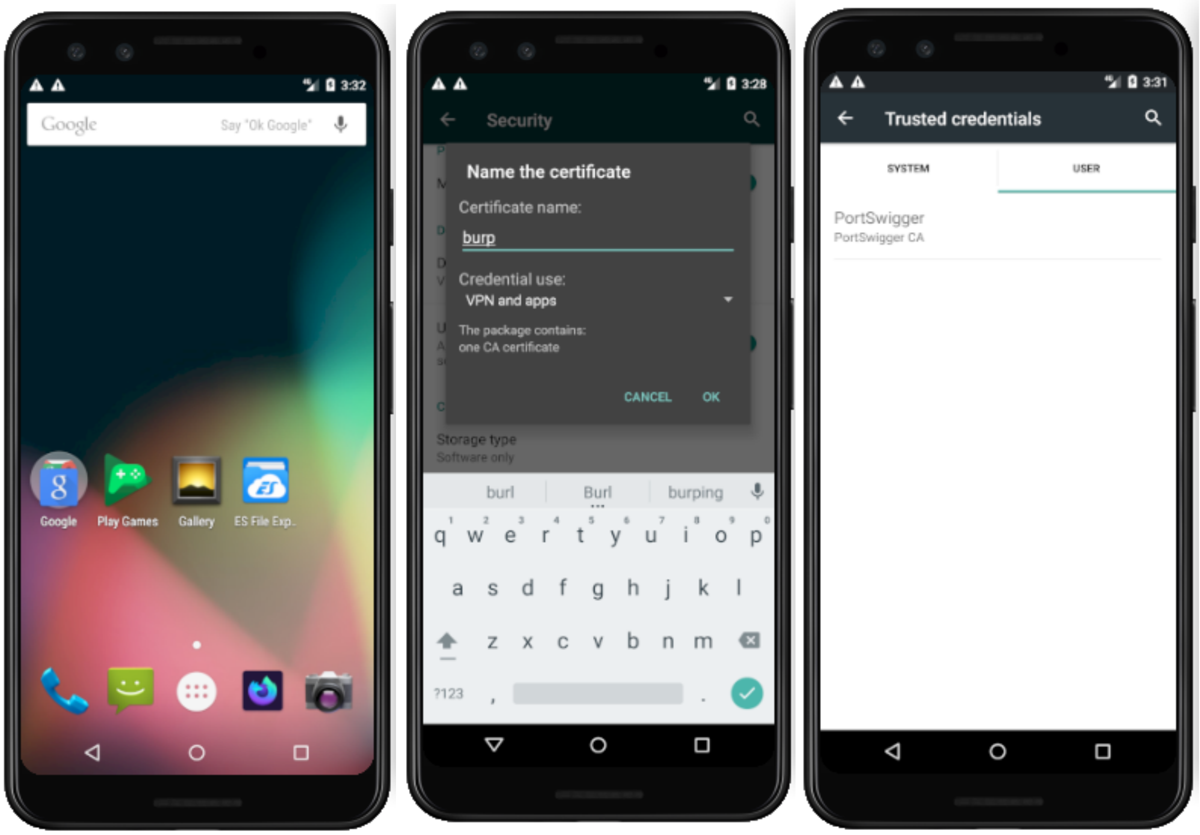
Burpsuite certificate for android device for HTTPS proxy configuration :-

https://portswigger.net/support/installing-burp-suites-ca-certificate-in-an-android-device

After configuring quick access for Android Studio , we configured our Android Virtual Device within Android Studio . We chose Android 5.1 Lollipop for our android version for testing purposes and went for a Pixel 3 device as our emulator. We had initially opted for a higher version of android but configuration of Burp certificates on higher versions were tricky and not fully supported. We downloaded the certificate needed for intercepting and proxying all network traffic from Burpsuite's website and used adb push to copy the certificate from our device to the emulator . Once copied ,we selected the certificate and authorized it as a trusted user. This enables us to intercept all the traffic flowing from our android device via Burp Suite.



```
procx@kali:~/Android/Sdk/platform-tools$ ./adb push ~/Desktop/burpcert.der /sdcard
```



Name the certificate

Certificate name:
burp

Credential use:
VPN and apps

The package contains:
one CA certificate

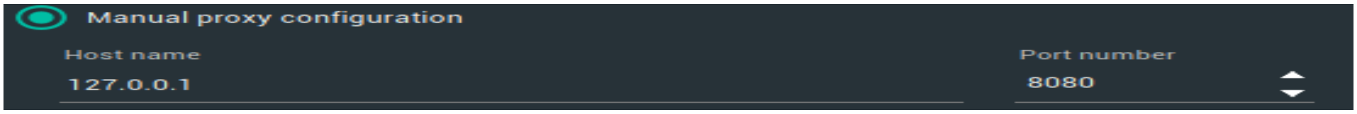
CANCEL OK

Trusted credentials

SYSTEM USER

PortSwigger
PortSwigger CA

We manually set the proxy address for Burp Suite which was 127.0.0.1 and port number 8080.



Manual proxy configuration

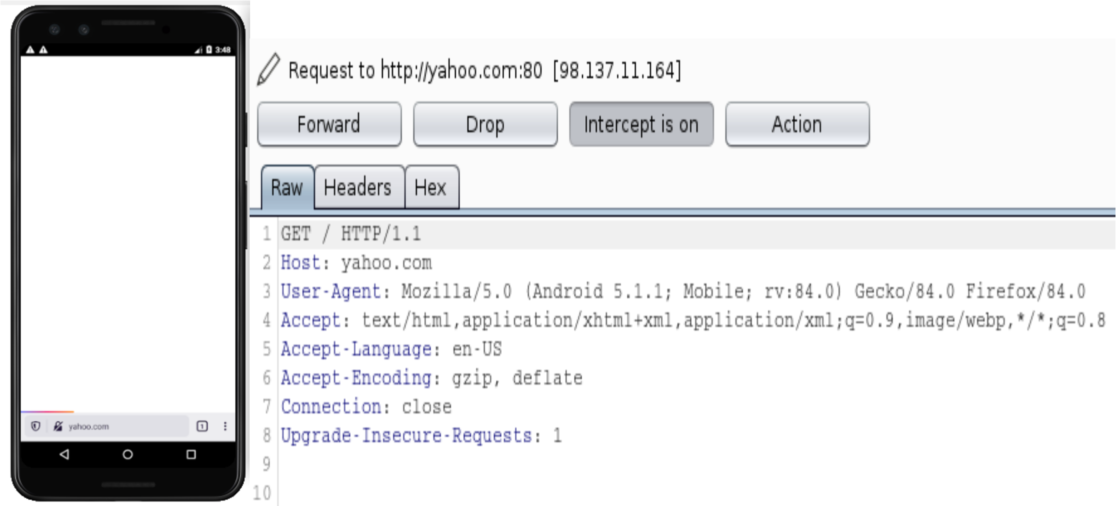
Host name
127.0.0.1

Port number
8080

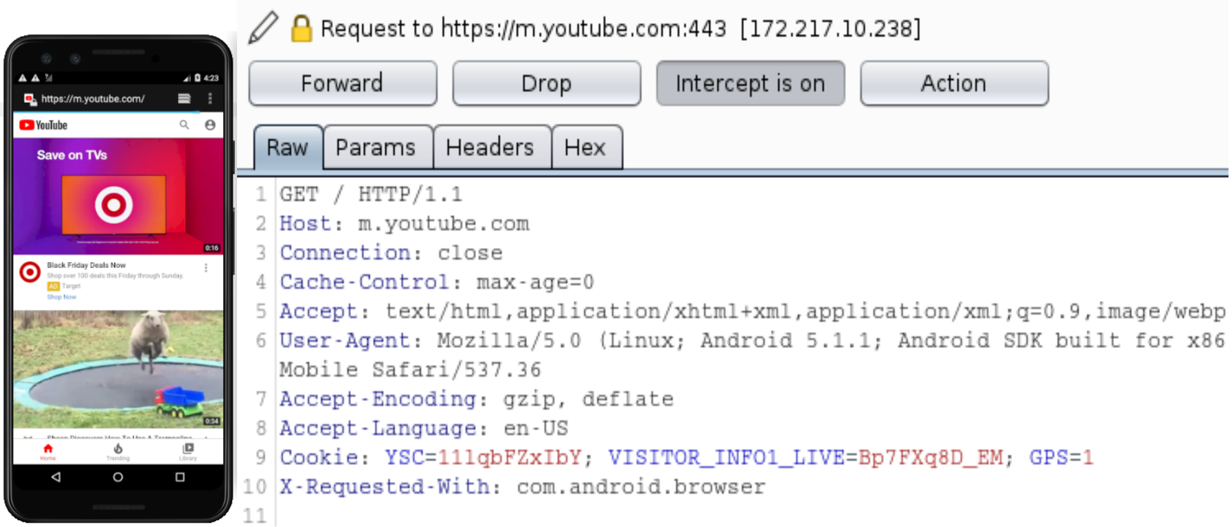
3) Testing traffic interception in BurpSuite (HTTP and HTTPS)

We tried to open yahoo.com through Firefox on our virtual device. We were able to successfully intercept the traffic from yahoo. We could see all the packet details from the header information captured by Burpsuite. We could also see that this was a GET request sent from us and we are also able to see the respective host getting the request and the device configurations as well.

We proceeded to test out if we could capture HTTPS traffic as well. We tried opening youtube.com and we were able to intercept the request and view detailed information of the device and view and manually edit all the header information in the request. We were successful in capturing web traffic in Burp Suite. Now we proceed with capturing traffic from android applications.



```
GET / HTTP/1.1
Host: yahoo.com
User-Agent: Mozilla/5.0 (Android 5.1.1; Mobile; rv:84.0) Gecko/84.0 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

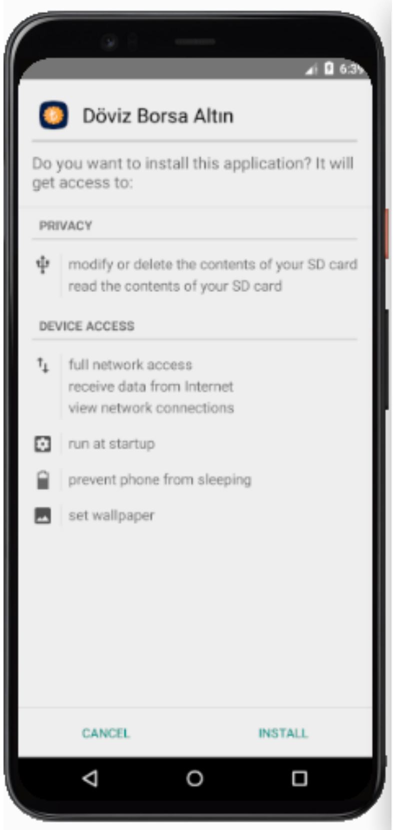


```
GET / HTTP/1.1
Host: m.youtube.com
Connection: close
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp
User-Agent: Mozilla/5.0 (Linux; Android 5.1.1; Android SDK built for x86
Mobile Safari/537.36
Accept-Encoding: gzip, deflate
Accept-Language: en-US
Cookie: YSC=11lqbFZxIbY; VISITOR_INFO1_LIVE=Bp7FXq8D_EM; GPS=1
X-Requested-With: com.android.browser
```

4) Downloading and installing malware apk samples from GitHub and VirusShare

https://github.com/ashishb/androidmalware/tree/master/fraud_financial_apps

We downloaded a fraud financial application from GitHub and used ./adb push to copy the apk file on to our virtual device . Installation of the application was a bit tricky as it was a Turkish application. We used google translation on the application and found out that this was an international currency exchange application.



Döviz Borsa Altın

Do you want to install this application? It will get access to:

PRIVACY

modify or delete the contents of your SD card
read the contents of your SD card

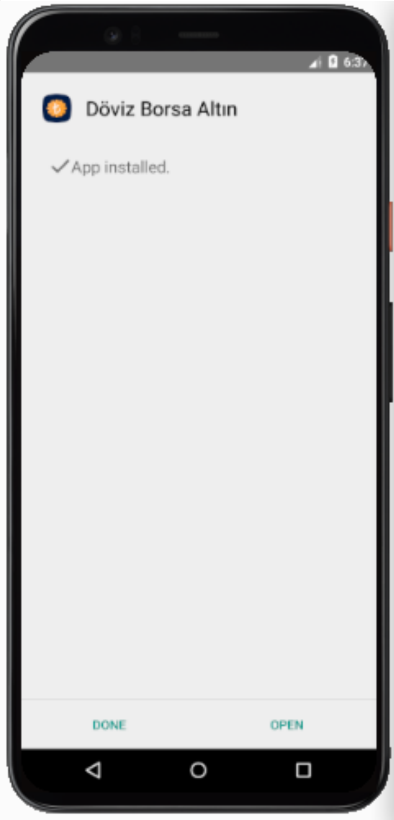
DEVICE ACCESS

full network access
receive data from Internet
view network connections

run at startup

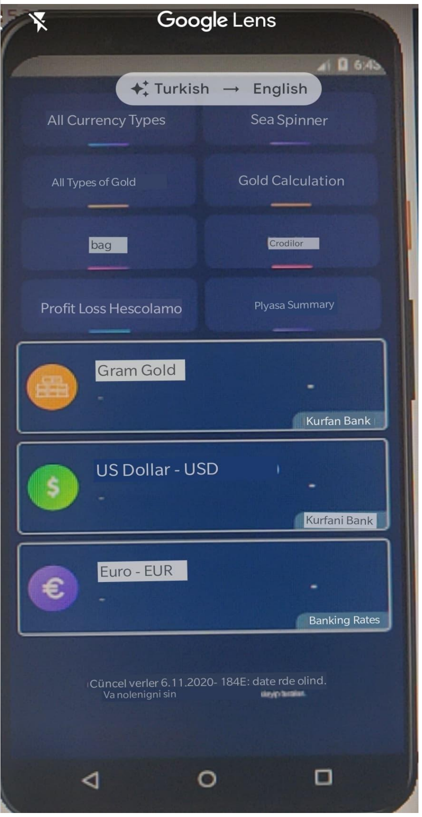
prevent phone from sleeping

set wallpaper



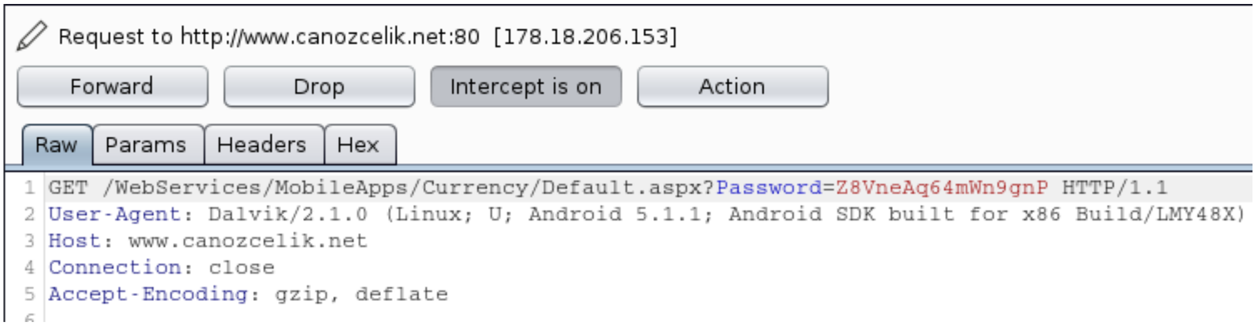
Döviz Borsa Altın

✓ App installed.



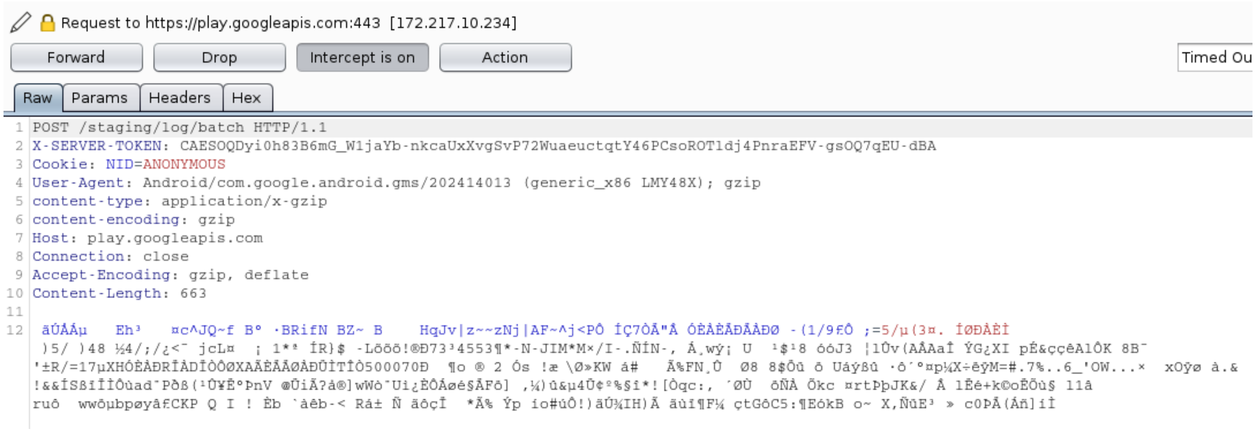
Cüncel verler 6.11.2020- 184E: date rde olind.
Va nolenigni sin

As soon as we open the money exchanging application , we intercept its GET request to the host canozcelik.net . We saw that the GET request is submitting a password which can be seen in plaintext .This configuration in the application causes the contents of the password field to appear in the URL and be captured by an attacker. All passwords need to be submitted via the POST method to ensure safety.



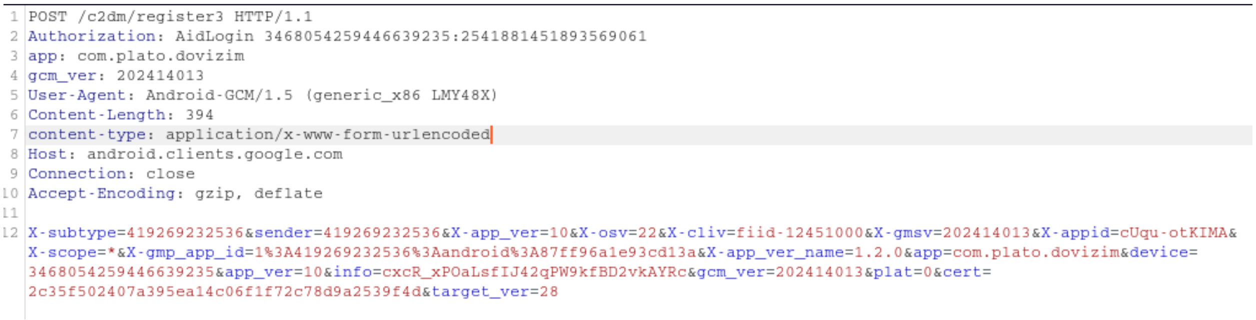
```
GET /WebServices/MobileApps/Currency/Default.aspx?Password=Z8VneAq64mWn9gnP HTTP/1.1
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; Android SDK built for x86 Build/LMY48X)
Host: www.canozcelik.net
Connection: close
Accept-Encoding: gzip, deflate
```

In addition to the GET request that was captured by Burpsuite , we also saw BurpSuite capturing the POST request from Google's Api . Upon the interception of the POST request , we could see that the host is requesting for a log batch and in addition to that we also see that the content length is 663 bytes which is encrypted.



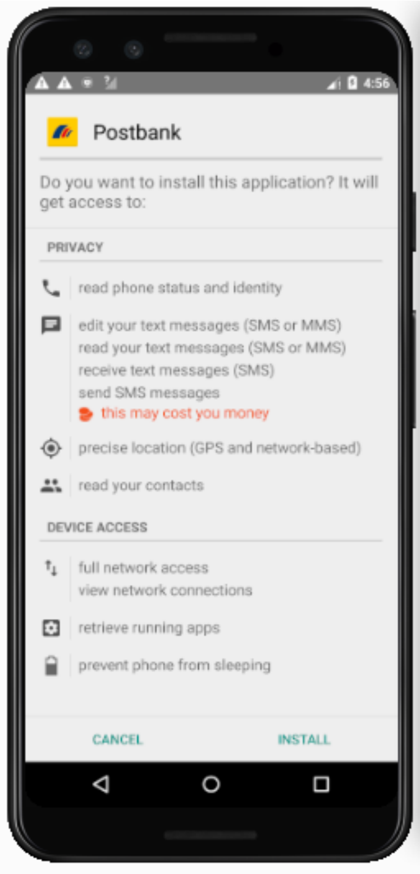
```
POST /staging/log/batch HTTP/1.1
X-SERVER-TOKEN: CAESOQDyi0h83B6mG_W1jaYb-nkcaUxXvgSvP72WuaeuctqtY46PCsoROT1dj4PnraEFV-gsOQ7qEU-dBA
Cookie: NID=ANONYMOUS
User-Agent: Android/com.google.android.gms/202414013 (generic_x86 LMY48X); gzip
content-type: application/x-gzip
content-encoding: gzip
Host: play.googleapis.com
Connection: close
Accept-Encoding: gzip, deflate
Content-Length: 663

 ãÚÅÂµ   Eh³   ¤c^JQ~f B° ·BRifN BZ~ B    HqJv|z~~zNj|AF~^j<PÔ ÎÇ7ÒÃ"Â ÓÈÀÈÃÐÃÀÐØ -(1/9£Ô ;=5/µ(3¤. ÎØÐÀÈÎ
 )5/ )48 ½4/;/¿<¯ jcL¤  ¡ 1*ª ÎR}$ -Lõõõ!®Ð73³4553¶*-N-JIM*M×/I-.ÑÎN-, Á¸wý¡ U  ¹$¹8 óóJ3 ¦1Ûv(AÅAaÎ ÝG¿XI pÉ&ççêA1ÔK 8B¯
'±R/=17µXHÓÈÀÐRÎÀDÎÒÔØXAÃÈÃÃØÀÐÛÌTÎÒ500070Ð  ¶o ® 2 Ós !æ \Ø»KW á#   Ã%FN¸Û  Ø8 8$Õû ō Uáýßû ·ô´°¤p¼X÷êÿM=#.7%..6_'OW...×  xOÿø à.&
!&&ÍSßîÎÌÔùad¨Pðß(¹Û¥Ê°ÞnV ®ÛiÃ?å®]wWò¨Ui¿ÈÒÁøé§ÃFõ] ,¼)û&µ4Ũ¢º%§î*![Òqc:, ´ØÙ  õÑÀ Õkc ¤rtÞþJK&/ Â 1Êé+k©oÉÕù§ 11â
ruô  wwõµbpøyâ£CKP Q I ! Èb `àêb-< Rá± Ñ ãôçÎ  *Ã% Ýp ío#úÔ!)ãÛ¾IH)Ã ãùî¶F¼ çtGôC5:¶EókB o~ X,ÑûE³ » c0ÞÂ(Áñ]íÌ
```

We also see another POST request being intercepted from the host com.plato.dovizim , which is the Turkish fraudulent bank application itself.



```
POST /c2dm/register3 HTTP/1.1
Authorization: AidLogin 3468054259446639235:2541881451893569061
app: com.plato.dovizim
gcm_ver: 202414013
User-Agent: Android-GCM/1.5 (generic_x86 LMY48X)
Content-Length: 394
content-type: application/x-www-form-urlencoded
Host: android.clients.google.com
Connection: close
Accept-Encoding: gzip, deflate

X-subtype=419269232536&sender=419269232536&X-app_ver=10&X-osv=22&X-cliv=fiid-12451000&X-gmsv=202414013&X-appid=cUqu-otKIMA&
X-scope=*&X-gmp_app_id=1%3A419269232536%3Aandroid%3A87ff96a1e93cd13a&X-app_ver_name=1.2.0&app=com.plato.dovizim&device=
3468054259446639235&app_ver=10&info=cxcR_xPOaLsfIJ42qPW9kfBD2vkAYRc&gcm_ver=202414013&plat=0&cert=
2c35f502407a395ea14c06f1f72c78d9a2539f4d&target_ver=28
```

We proceeded with testing another application named postbank from GitHub. The installation process was smooth and did not give us any issues. Upon installation , the application did not appear in the application drawer . It obfuscated itself from view and was only available on the application list under settings. A few moments after installation , an activate device administrator prompt appeared . Upon clicking cancel , the prompt would constantly reappear and would not go unless
Activate was selected.



Postbank

Do you want to install this application? It will get access to:

PRIVACY

read phone status and identity

edit your text messages (SMS or MMS)
read your text messages (SMS or MMS)
receive text messages (SMS)
send SMS messages
this may cost you money

precise location (GPS and network-based)

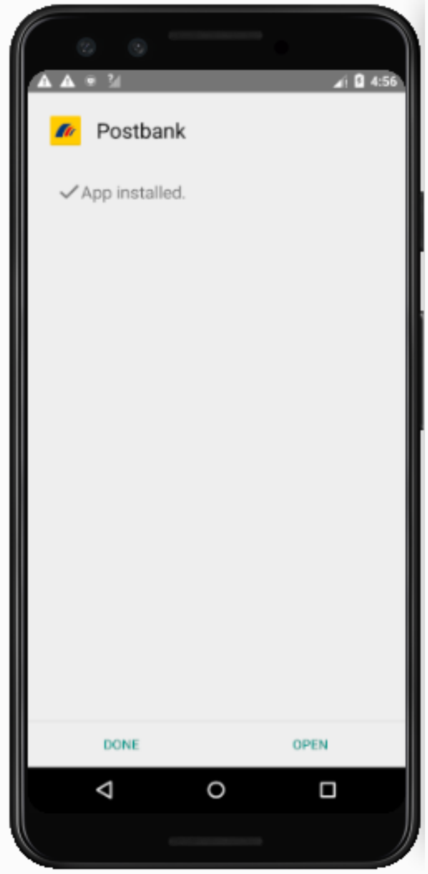
read your contacts

DEVICE ACCESS

full network access
view network connections

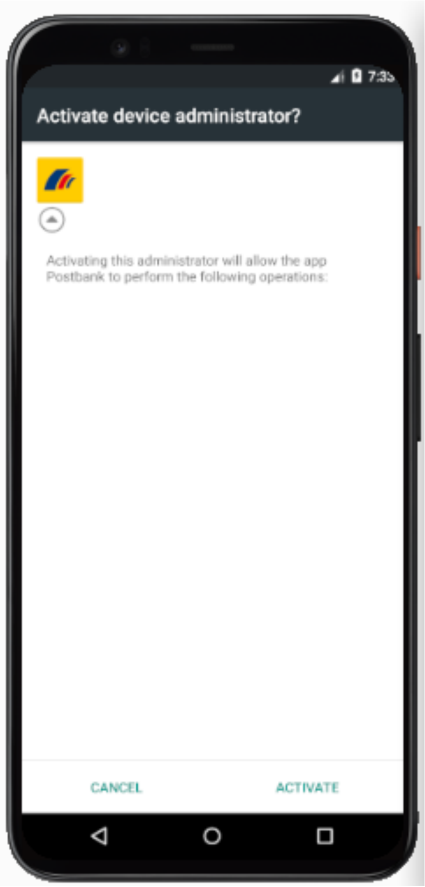
retrieve running apps

prevent phone from sleeping



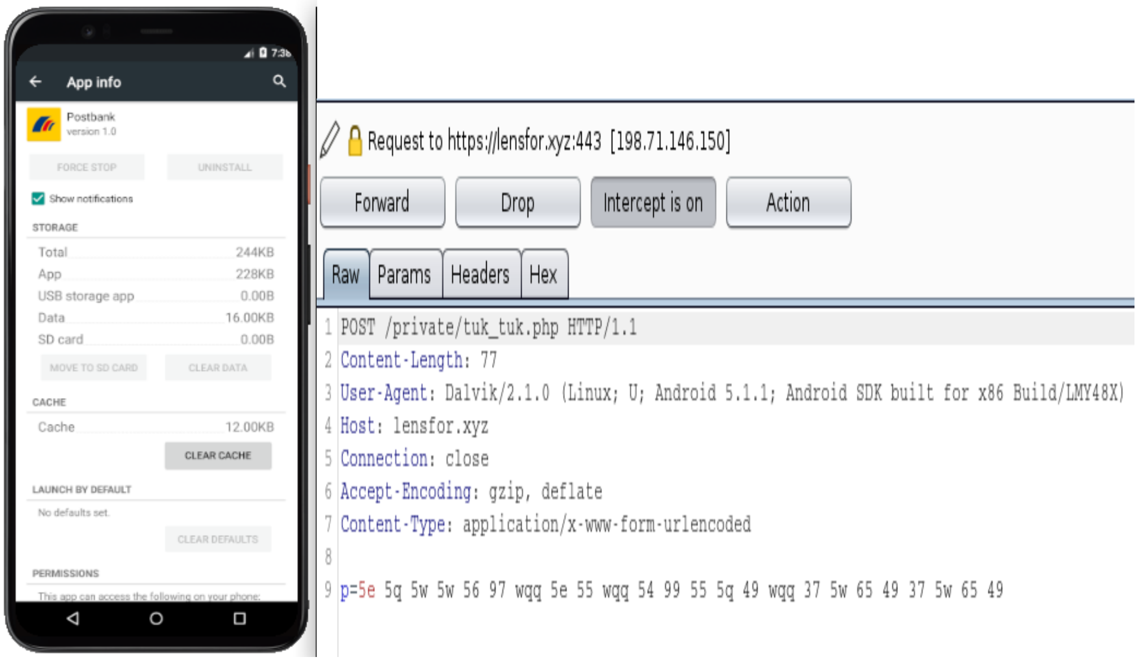
Postbank

✓App installed.



Activate device administrator?

Activating this administrator will allow the app Postbank to perform the following operations:



Postbank
version 1.0

☑ Show notifications

STORAGE

```
Total                 244KB
App                   228KB
USB storage app       0.00B
Data                16.00KB
SD card               0.00B
```

CACHE

```
Cache               12.00KB
```

LAUNCH BY DEFAULT

No defaults set.

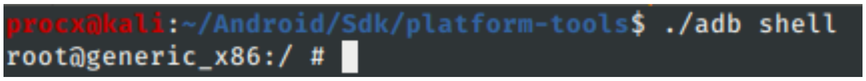
PERMISSIONS

This app can access the following on your phone:

Request to https://lensfor.xyz:443 [198.71.146.150]

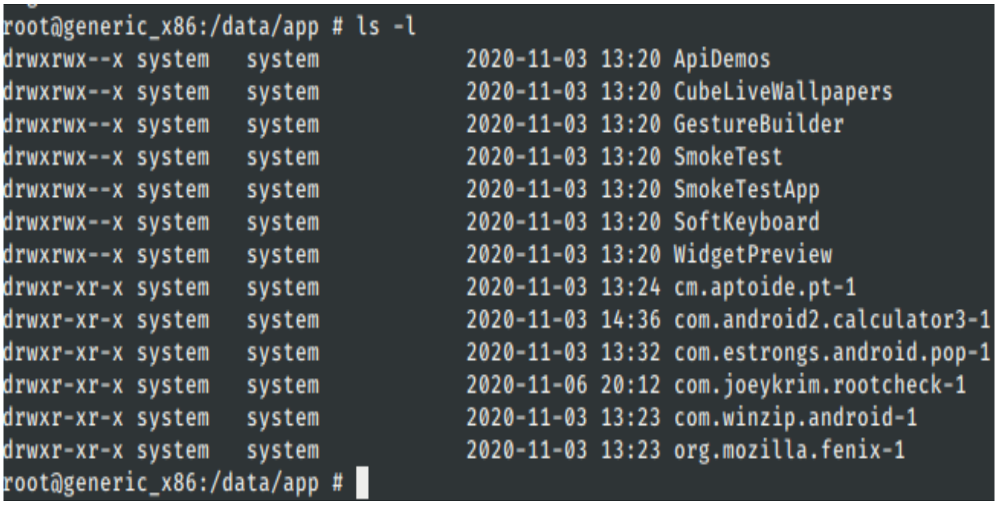
```
POST /private/tuk_tuk.php HTTP/1.1
Content-Length: 77
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; Android SDK built for x86 Build/LMY48X)
Host: lensfor.xyz
Connection: close
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded

p=5e 5q 5w 5w 56 97 wqq 5e 55 wqq 54 99 55 5q 49 wqq 37 5w 65 49 37 5w 65 49
```

At the very moment , Burpsuite intercepted a POST request from the application. The host requesting the data was lensfor.xyz and it was requesting a php file named tuk_tuk.php.The application had made itself as an administrator and gained a foothold in the device could not be uninstalled. We tried to uninstall the application through settings but the uninstall button was greyed out.



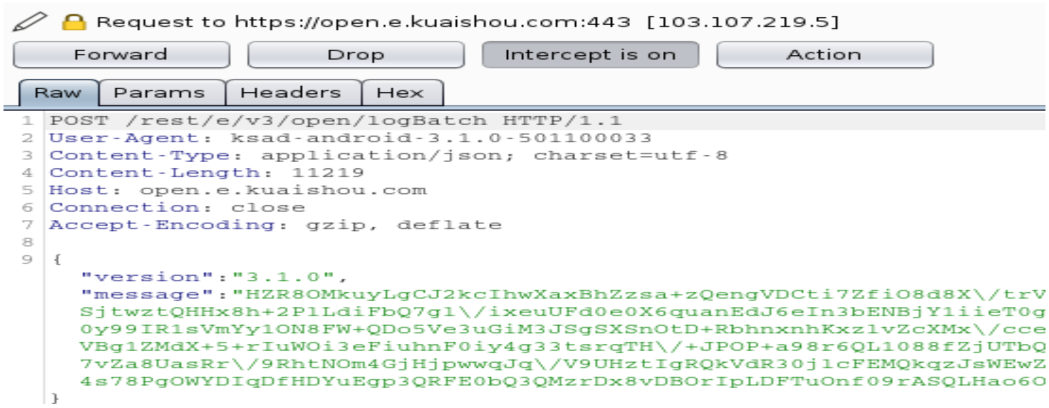
```
procx@kali:~/Android/Sdk/platform-tools$ ./adb shell
root@generic_x86:/ #
```

We had to start adb as root and then go into adb shell and manually delete the Postbank application in order to get rid of it.



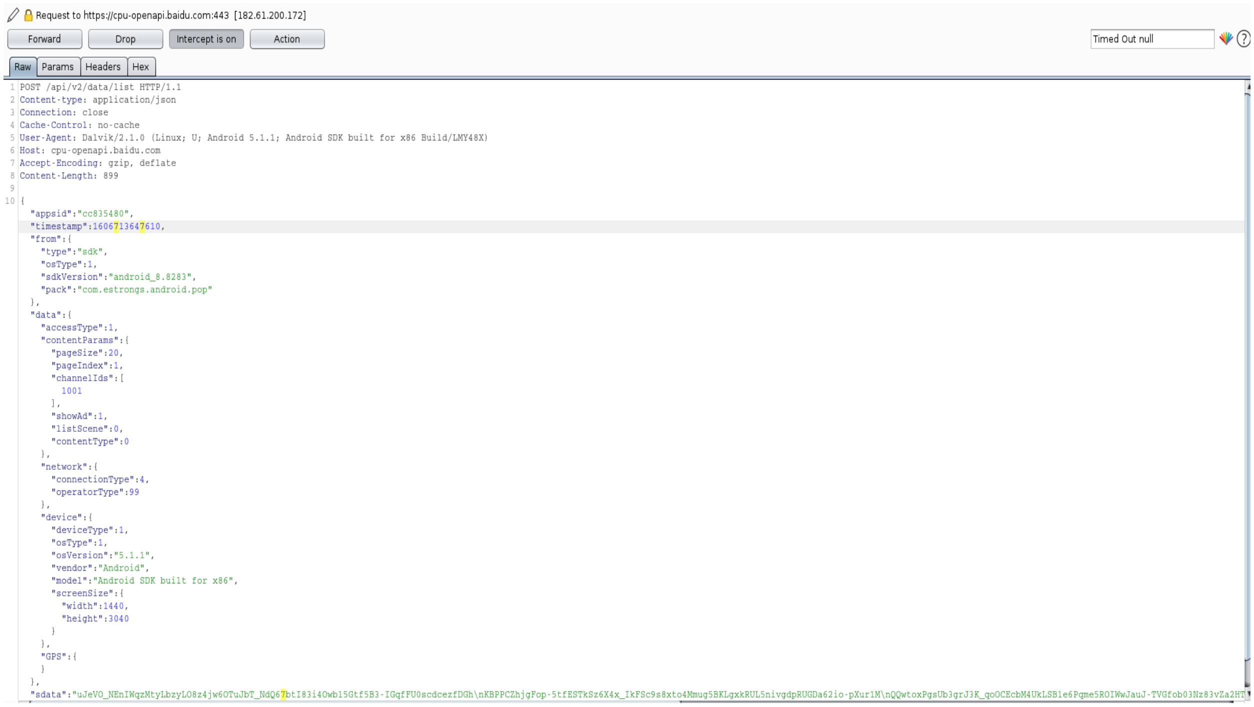
```
root@generic_x86:/data/app # ls -l
drwxrwx--x system   system            2020-11-03 13:20 ApiDemos
drwxrwx--x system   system            2020-11-03 13:20 CubeLiveWallpapers
drwxrwx--x system   system            2020-11-03 13:20 GestureBuilder
drwxrwx--x system   system            2020-11-03 13:20 SmokeTest
drwxrwx--x system   system            2020-11-03 13:20 SmokeTestApp
drwxrwx--x system   system            2020-11-03 13:20 SoftKeyboard
drwxrwx--x system   system            2020-11-03 13:20 WidgetPreview
drwxr-xr-x system   system            2020-11-03 13:24 cm.aptoide.pt-1
drwxr-xr-x system   system            2020-11-03 14:36 com.android2.calculator3-1
drwxr-xr-x system   system            2020-11-03 13:32 com.estrongs.android.pop-1
drwxr-xr-x system   system            2020-11-06 20:12 com.joeykrim.rootcheck-1
drwxr-xr-x system   system            2020-11-03 13:23 com.winzip.android-1
drwxr-xr-x system   system            2020-11-03 13:23 org.mozilla.fenix-1
root@generic_x86:/data/app #
```

5) Unusual traffic interception from Burpsuite



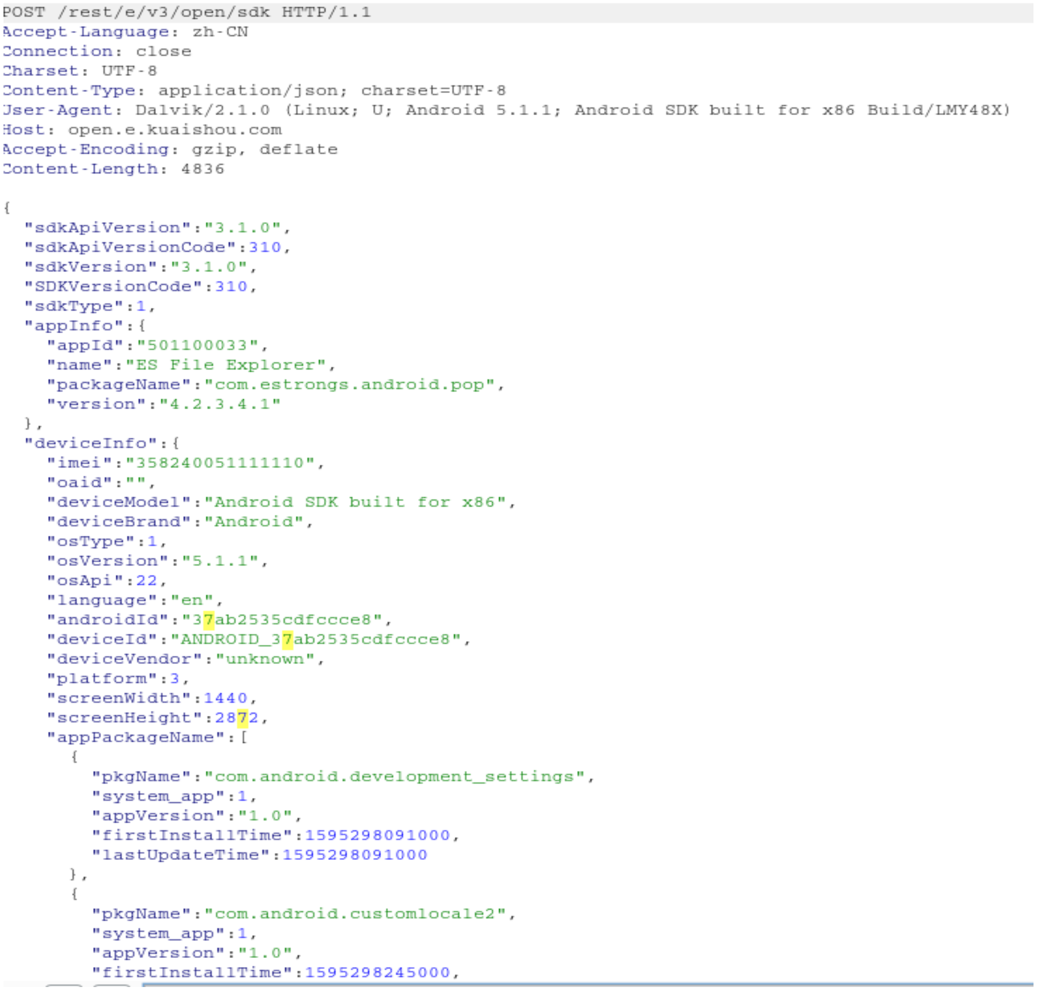
```
POST /rest/e/v3/open/logBatch HTTP/1.1
User-Agent: ksad-android-3.1.0-501100033
Content-Type: application/json; charset=utf-8
Content-Length: 11219
Host: open.e.kuaishou.com
Connection: close
Accept-Encoding: gzip, deflate

{
  "version":"3.1.0",
  "message":"HZR8OMkuyLgCJ2kcIhwXaxBhZzsa+zQengVDCti7ZfiO8d8X\/trV
  SjtwztQHHx8h+2PlLdiFbQ7gl\/ixeuUFd0e0X6quanEdJ6eIn3bENBjY1iieT0g
  0y99IR1sVmYy1ON8FW+QDo5Ve3uGiM3JSgSXSnOtD+RbhnxnhKxzlvZcXMx\/cce
  VBg1ZMdX+5+rIuWOi3eFiuhnF0iy4g33tsrqTH\/+JPOP+a98r6QL1088fZjUTbQ
  7vZa8UasRr\/9RhtNOm4GjHjpwwqJq\/V9UHztIgRQkVdR30jlcFEMQkqzJsWEwZ
  4s78PgOWYDIqDfHDYuEgp3QRFE0bQ3QMzrDx8vDBOrIpLDFTuOnf09rASQLHao6O
}
```

While analyzing traffic from an apk application , we encountered a weird POST request from an unusual host. The hostname was open.e.kuaishou.com . While searching for the hostname in Google , we came to know that Kuaishau.com was a Chinese video sharing mobile application platform . Another interesting and unusual request captured by Burpsuite was a POST request from cpu-openapi.baidu.com over HTTPS (port 443). Upon analyzing the request, we can see that the malware is requesting a lot of information from the device. Some examples include device type , device ID , SDK version , and other important information. This was an unusual finding because Burpsuite kept on intercepting traffic from this host even after all malicious applications were removed from the system.



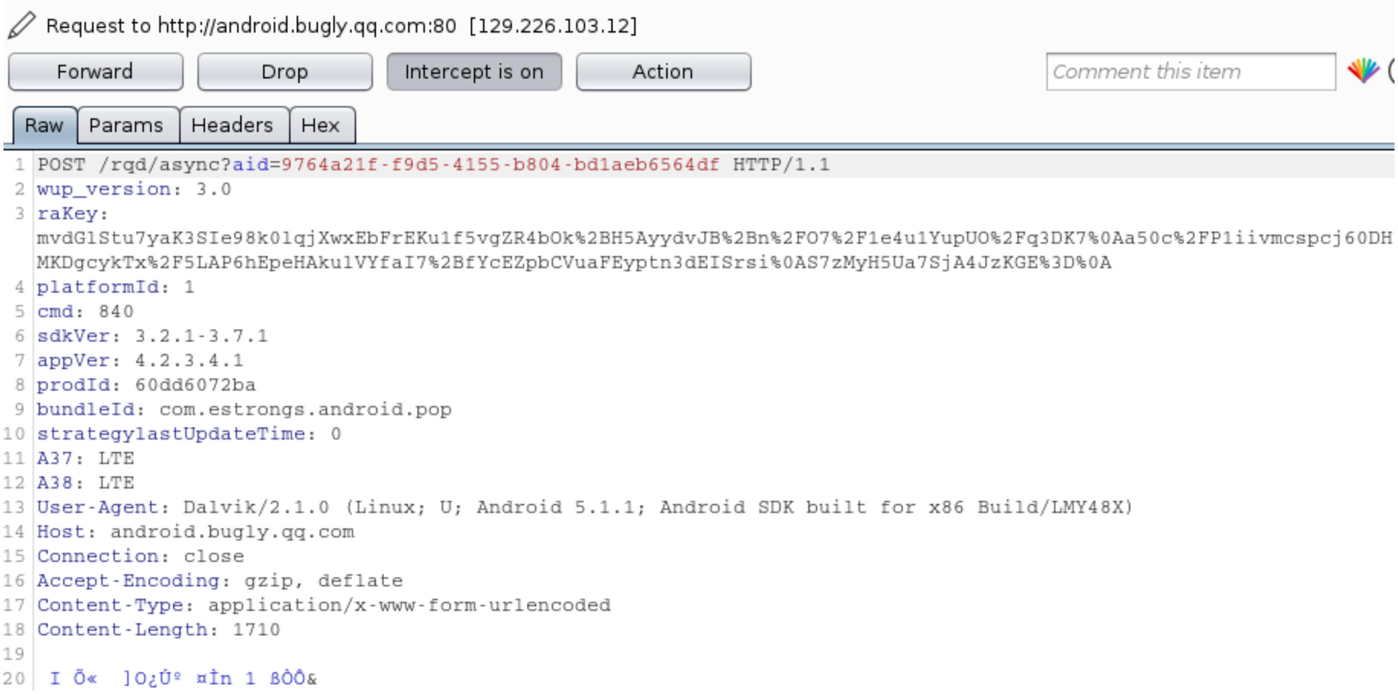
```
POST /api/v2/data/list HTTP/1.1
Content-type: application/json
Connection: close
Cache-Control: no-cache
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; Android SDK built for x86 Build/LMY48X)
Host: cpu-openapi.baidu.com
Accept-Encoding: gzip, deflate
Content-Length: 899

{
  "appsid":"cc835480",
  "timestamp":1606713647610,
  "from":{
    "type":"sdk",
    "osType":1,
    "sdkVersion":"android_8.8283",
    "pack":"com.estrongs.android.pop"
  },
  "data":{
    "accessType":1,
    "contentParams":{
      "pageSize":20,
      "pageIndex":1,
      "channelIds":[
        1001
      ],
      "showAd":1,
      "listScene":0,
      "contentType":0
    },
    "network":{
      "connectionType":4,
      "operatorType":99
    },
    "device":{
      "deviceType":1,
      "osType":1,
      "osVersion":"5.1.1",
      "vendor":"Android",
      "model":"Android SDK built for x86",
      "screenSize":{
        "width":1440,
        "height":3040
      }
    },
    "GPS":{
    }
  },
  "sdata":"uJeVO_NEnIWqzMtyLbzyLO8z4jw6OTuJbT_NdQ67btI83i4Owbl5Gtf5B3-IGqfFU0scdcezfDGh\nKBPPCZhjgFop-5tfESTkSz6X4x_IkFSc9s8xto4Mmug5BKLgxkRUL5nivgdpRUGDa62io-pXur1M\nQQwtoxPgsUb3grJ3K_qoOCEcbM4UkLSBle6Pqme5ROIWwJauJ-TVGfob03Nz83vZa2HT
```

We intercept yet another POST request from another subdomain of kuaishou.com . This time we see that the host is requesting various details regarding the device and applications installed into the device as well. We can see the package name of all the applications that are being requested by the host.



```
POST /rest/e/v3/open/sdk HTTP/1.1
Accept-Language: zh-CN
Connection: close
Charset: UTF-8
Content-Type: application/json; charset=UTF-8
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; Android SDK built for x86 Build/LMY48X)
Host: open.e.kuaishou.com
Accept-Encoding: gzip, deflate
Content-Length: 4836

{
  "sdkApiVersion":"3.1.0",
  "sdkApiVersionCode":310,
  "sdkVersion":"3.1.0",
  "SDKVersionCode":310,
  "sdkType":1,
  "appInfo":{
    "appId":"501100033",
    "name":"ES File Explorer",
    "packageName":"com.estrongs.android.pop",
    "version":"4.2.3.4.1"
  },
  "deviceInfo":{
    "imei":"358240051111110",
    "oaid":"",
    "deviceModel":"Android SDK built for x86",
    "deviceBrand":"Android",
    "osType":1,
    "osVersion":"5.1.1",
    "osApi":22,
    "language":"en",
    "androidId":"37ab2535cdfccce8",
    "deviceId":"ANDROID_37ab2535cdfccce8",
    "deviceVendor":"unknown",
    "platform":3,
    "screenWidth":1440,
    "screenHeight":2872,
    "appPackageName":[
      {
        "pkgName":"com.android.development_settings",
        "system_app":1,
        "appVersion":"1.0",
        "firstInstallTime":1595298091000,
        "lastUpdateTime":1595298091000
      },
      {
        "pkgName":"com.android.customlocale2",
        "system_app":1,
        "appVersion":"1.0",
        "firstInstallTime":1595298245000,
```

We can see that the apk files with malware have already infected our device as Burpsuite is intercepting a lot of unusual traffic and upon closer inspection , a lot of important details about our devices and applications are being sent out by these applications.



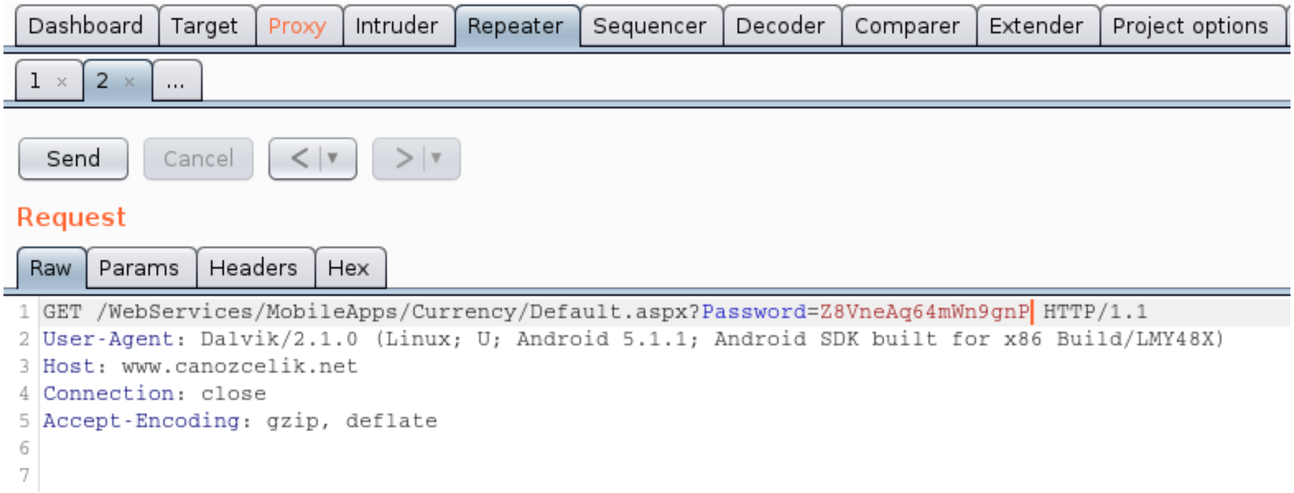
Request to http://android.bugly.qq.com:80 [129.226.103.12]

```
POST /rqd/async?aid=9764a21f-f9d5-4155-b804-bd1aeb6564df HTTP/1.1
wup_version: 3.0
raKey:
mvdGlStu7yaK3SIe98k0lqjXwxEbFrEKu1f5vgZR4bOk%2BH5AyydvJB%2Bn%2FO7%2F1e4u1YupUO%2Fq3DK7%0Aa50c%2FP1iivmcspcj60DH
MKDgcykTx%2F5LAP6hEpeHAkulVYfaI7%2BfYcEZpbCVuaFEyptn3dEISrsi%0AS7zMyH5Ua7SjA4JzKGE%3D%0A
platformId: 1
cmd: 840
sdkVer: 3.2.1-3.7.1
appVer: 4.2.3.4.1
prodId: 60dd6072ba
bundleId: com.estrongs.android.pop
strategylastUpdateTime: 0
A37: LTE
A38: LTE
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; Android SDK built for x86 Build/LMY48X)
Host: android.bugly.qq.com
Connection: close
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 1710

 I Õ«  ]O¿Ú º ¤İn 1 ßÒÔ&
```

6) Exploring other features of Burp Suite

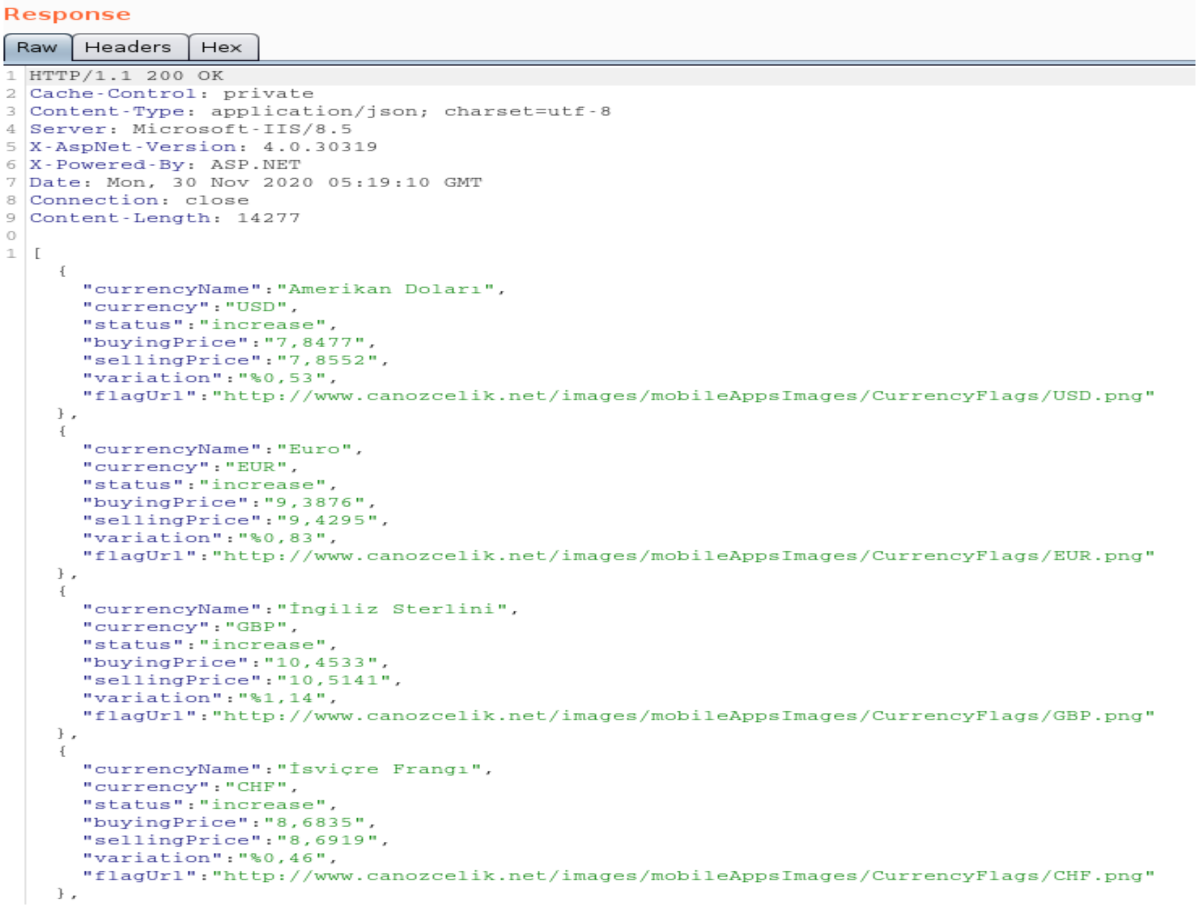
Another useful feature of Burp Suite other than simply intercepting traffic is to modify appropriate header information from HTTP requests and view the desired output. For this purpose, we chose to use REPEATER inside of Burp Suite and manually test out how the responses are crafted for some of the intercepted traffic from our fraudulent banking application.

For this experiment , a default GET request is sent to REPEATER where the application is requesting various information regarding the currency types and other various details relating to its currency. In addition to that we see a default password that was auto set by the application to authorize access and validate its authenticity to its host. REPEATER allows us to change the default values that are present in the header information to anything we desire. This is very useful for testing out the behavior of HTTP requests upon modifying certain header information.



```
GET /WebServices/MobileApps/Currency/Default.aspx?Password=Z8VneAq64mWn9gnP HTTP/1.1
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; Android SDK built for x86 Build/LMY48X)
Host: www.canozcelik.net
Connection: close
Accept-Encoding: gzip, deflate


```

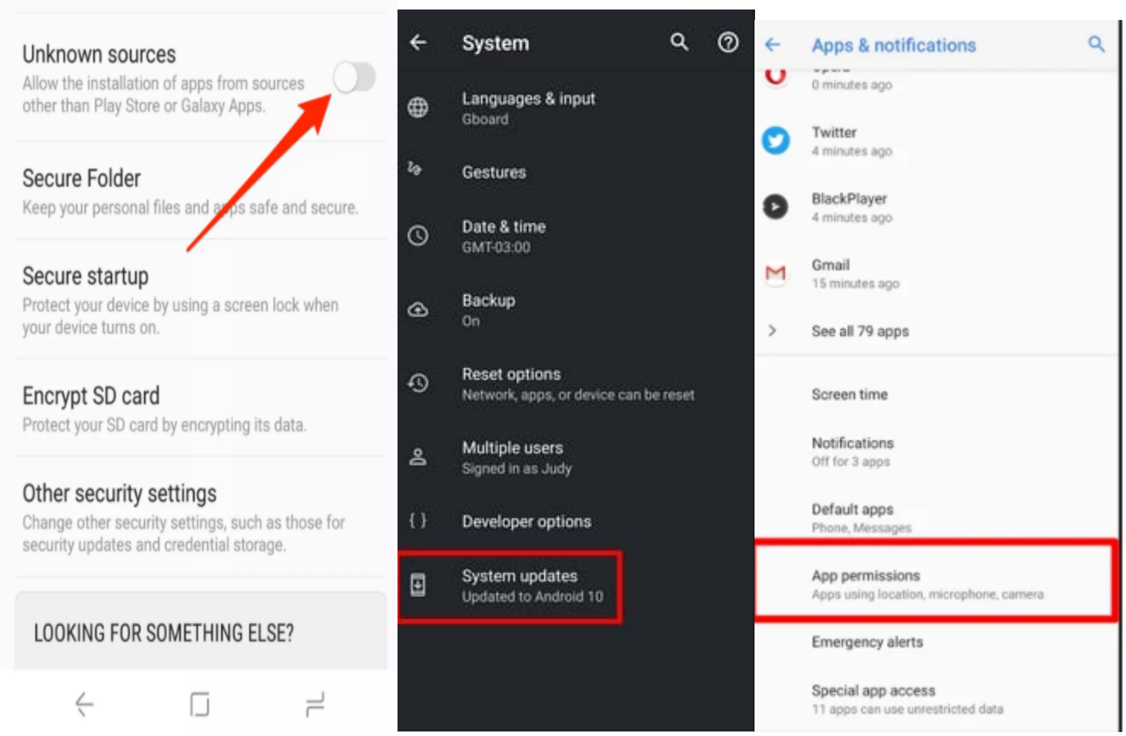
Upon forwarding the request through REPEATER , we can see its response. We see that there is a 200 OK response from HTTP which indicates that the request has been successful. Other header information is also present within the Response. We can see all the currency details requested by the application to its host. In addition to that , we can view that the application is powered by ASP.NET and we can also see the AspNet Version which is 4.0.30319 .



```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/json; charset=utf-8
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Mon, 30 Nov 2020 05:19:10 GMT
Connection: close
Content-Length: 14277

[
  {
    "currencyName":"Amerikan Doları",
    "currency":"USD",
    "status":"increase",
    "buyingPrice":"7,8477",
    "sellingPrice":"7,8552",
    "variation":"%0,53",
    "flagUrl":"http://www.canozcelik.net/images/mobileAppsImages/CurrencyFlags/USD.png"
  },
  {
    "currencyName":"Euro",
    "currency":"EUR",
    "status":"increase",
    "buyingPrice":"9,3876",
    "sellingPrice":"9,4295",
    "variation":"%0,83",
    "flagUrl":"http://www.canozcelik.net/images/mobileAppsImages/CurrencyFlags/EUR.png"
  },
  {
    "currencyName":"İngiliz Sterlini",
    "currency":"GBP",
    "status":"increase",
    "buyingPrice":"10,4533",
    "sellingPrice":"10,5141",
    "variation":"%1,14",
    "flagUrl":"http://www.canozcelik.net/images/mobileAppsImages/CurrencyFlags/GBP.png"
  },
  {
    "currencyName":"İsviçre Frangı",
    "currency":"CHF",
    "status":"increase",
    "buyingPrice":"8,6835",
    "sellingPrice":"8,6919",
    "variation":"%0,46",
    "flagUrl":"http://www.canozcelik.net/images/mobileAppsImages/CurrencyFlags/CHF.png"
  },
```

SOME IMPORTANT POINTS TO KEEP AN EYE ON WHILE INSTALLING ANDROID APPLICATIONS

- Download android applications from the official Play Store.
- Be cautious even when installing applications from the play store as there can be numerous applications out there that could be deceptive and fraudulent and have very similar names and descriptions to legit applications.
- Make sure that installation from unknown sources is turned off as this prevents the device from installing applications that are outside the Play Store.
- Make sure that permissions are read thoroughly while installing the application and make sure to disable any permission that you think the application does not require
- Be sharp and alert and stay away from pirated applications and modded games as these are one of the most common ways on how malwares are injected into a device.
- Make sure you are updating your Android Device frequently as most updates come with the latest security patches.



Unknown sources
Allow the installation of apps from sources other than Play Store or Galaxy Apps.

Secure Folder
Keep your personal files and apps safe and secure.

Secure startup
Protect your device by using a screen lock when your device turns on.

Encrypt SD card
Protect your SD card by encrypting its data.

Other security settings
Change other security settings, such as those for security updates and credential storage.

LOOKING FOR SOMETHING ELSE?

System

Languages & input
Gboard

Gestures

Date & time
GMT-03:00

Backup
On

Reset options
Network, apps, or device can be reset

Multiple users
Signed in as Judy

Developer options

System updates
Updated to Android 10

Apps & notifications

0 minutes ago

Twitter
4 minutes ago

BlackPlayer
4 minutes ago

Gmail
15 minutes ago

See all 79 apps

Screen time

Notifications
Off for 3 apps

Default apps
Phone, Messages

App permissions
Apps using location, microphone, camera

Emergency alerts

Special app access
11 apps can use unrestricted data

CONCLUSION

Overall , we gained a lot of new knowledge doing this project. We got to understand a lot about BurpSuite which we were not very familiar with . In addition to that , we got a deeper understanding of Android Studio and AVD Emulator and learned a lot about Android Debug Bridge. At the same time , we also gained a deeper understanding on some Networking protocols like HTTP requests and understood how the devices communicate with the internet. We got to experiment with other features of Burp Suite as well which was interesting. We encountered a bunch of obstacles during the project and experienced a lot of roadblocks at times and had difficulties moving further , but with proper guidance from our professor , Dr. Namin , we managed to complete the task. Overall , this was a huge learning experience for all of us and we understood how infected malware apk files works and at the same time understood how severe the threats can be and the security measures needed to prevent them.