Two tales and one Antidot(e) — a new mobile malware campaign in Poland

medium.com/@mvaks/two-tales-and-one-antidot-e-a-new-mobile-malware-campaign-in-poland-de704997096f

February 12, 2025

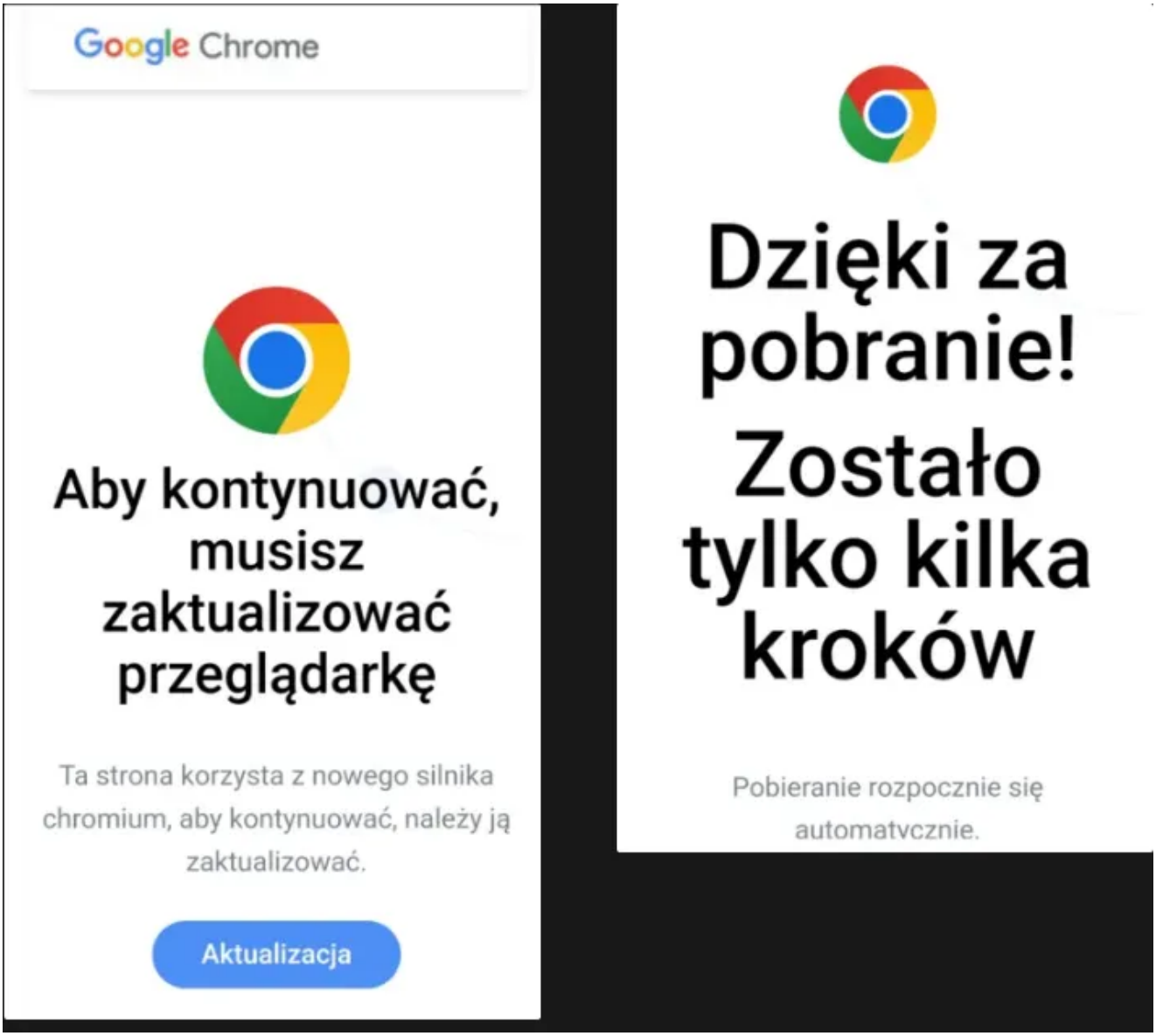
--

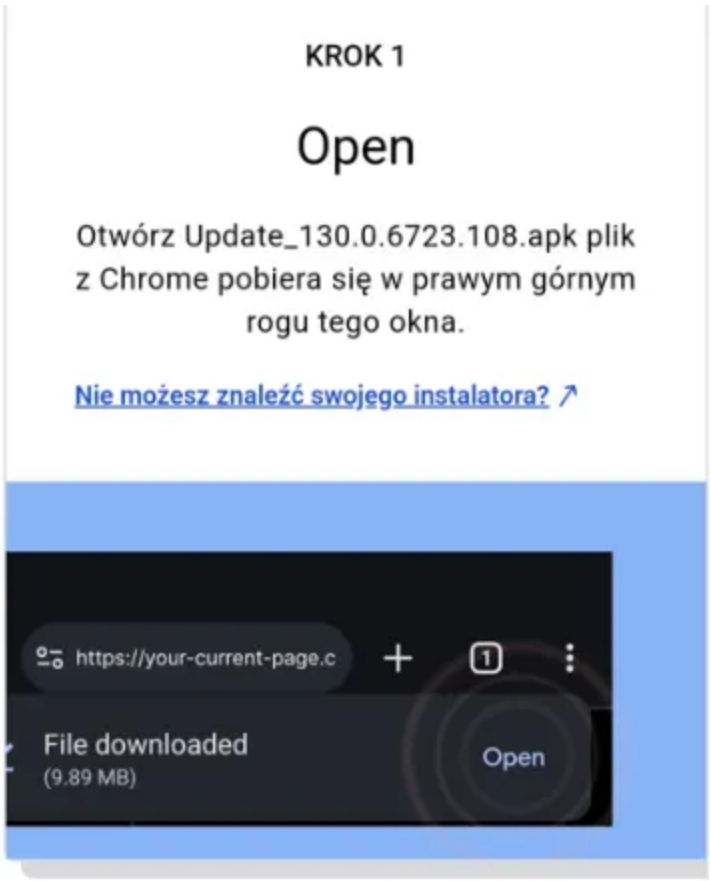
Recently, the Polish cyber threat landscape has seen a growing number of malicious mobile applications. In addition to identifying the apps impersonating shopping platforms such as OLX and Allegro, as well as the Polish bank PKO, which were described in the previous article, a new mobile malware campaign involving the Antidot malware has been detected.

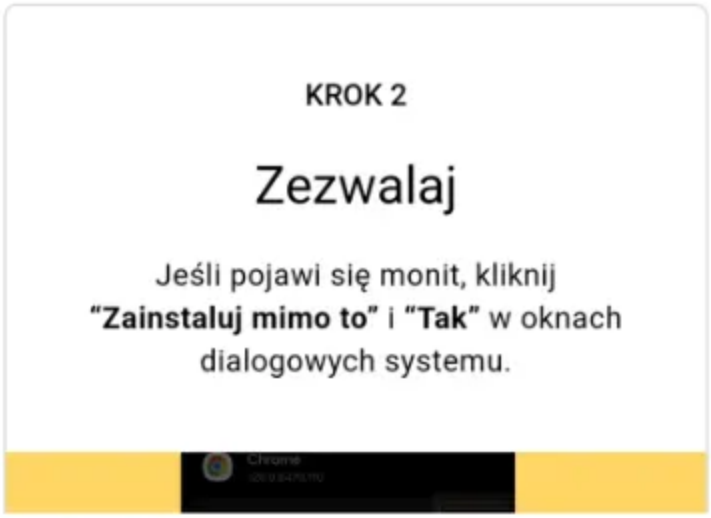
Antidot was first described by researchers from Cyble in May 2024. At that time, it was found masquerading as Google Play updates in detected campaigns. The malware features overlay and keylogging capabilities and includes a VNC module, allowing attackers to remotely control infected devices.

In recent campaigns in Poland, cybercriminals have employed an intriguing scenario involving a supposed update for the Google Chrome application. On compromised Polish websites, they placed scripts that, when visited by an unsuspecting victim, displayed a message urging them to update their software. If accessed from a computer using Safari, Google Chrome, or Edge, a `.dmg` file belonging to the SocGholish malware family was downloaded, ultimately leading to an infection with the Lumma Stealer.

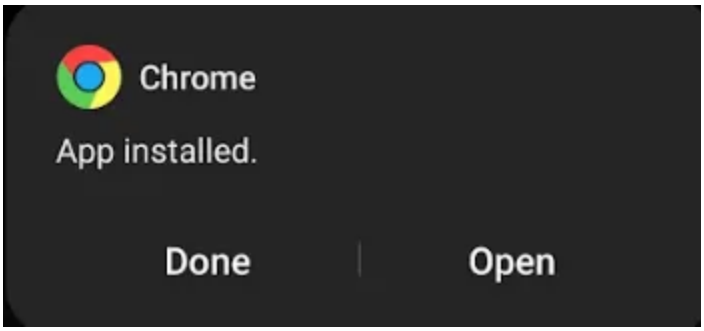
Meanwhile, if accessed from a mobile device, a message appeared stating that the site was using a new Chromium engine, prompting the download of an `.apk` file named `Update_130.1.6723.108.apk`. The file's name, resembling a legitimate update, was intended to make the victim's less suspicious. The victim was then instructed to grant permission for installing apps from third-party sources.


Google Chrome
Aby kontynuować, musisz zaktualizować przeglądarkę
Ta strona korzysta z nowego silnika chromium, aby kontynuować, należy ją zaktualizować.
Aktualizacja
Dzięki za pobranie!
Zostało tylko kilka kroków
Pobieranie rozpocznie się automatvcznie.

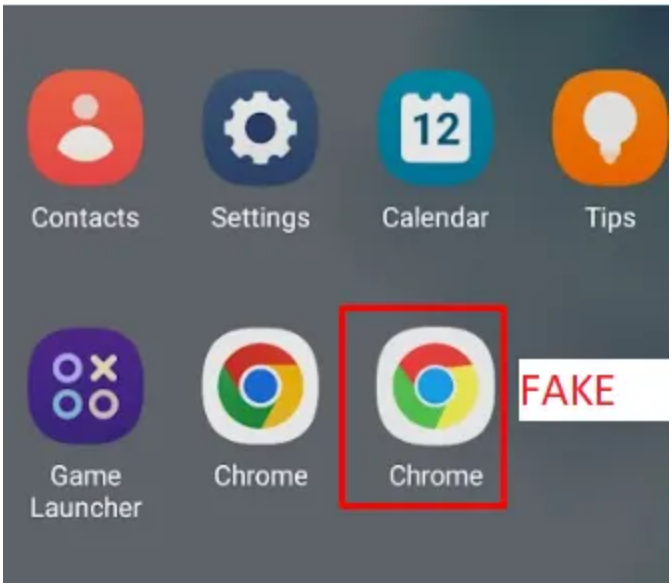




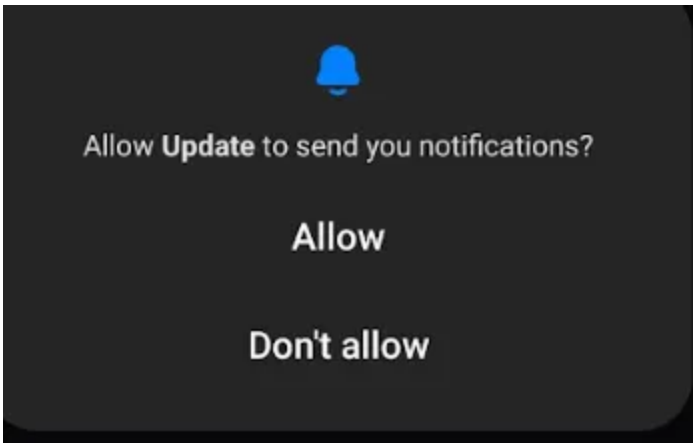
After installation on the device, an icon impersonating Google Chrome appeared on the home screen.



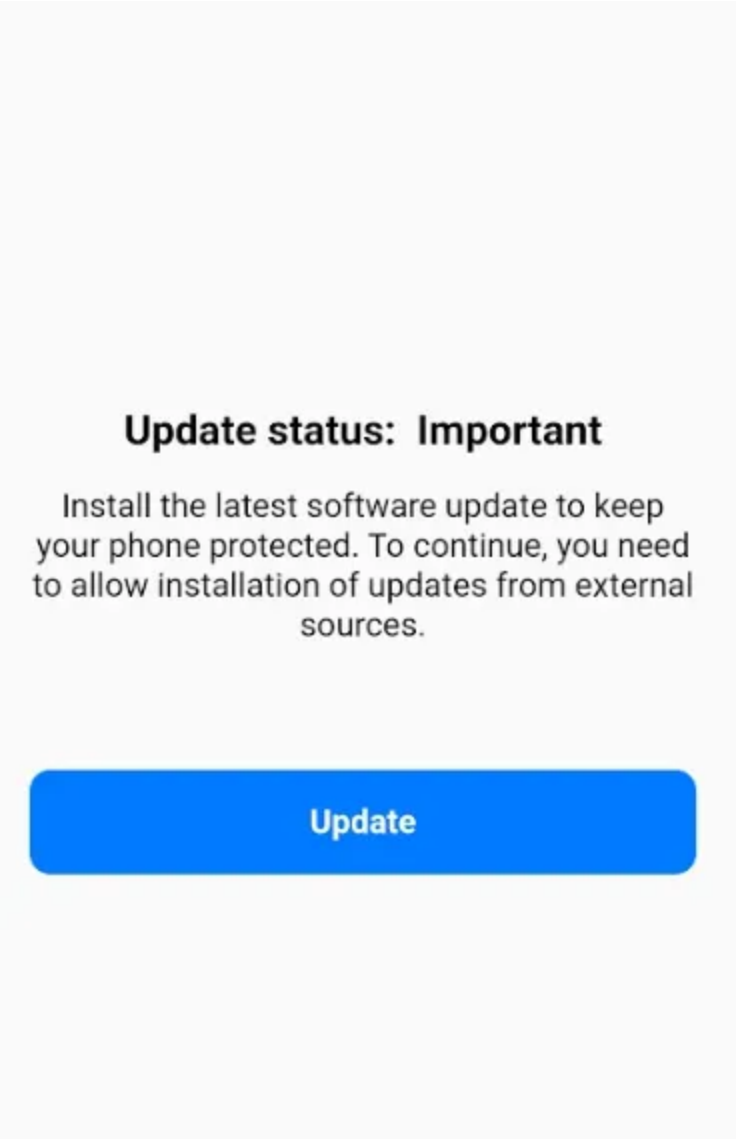
The icon of the fake application (on the right) slightly differs from the original application (on the left).



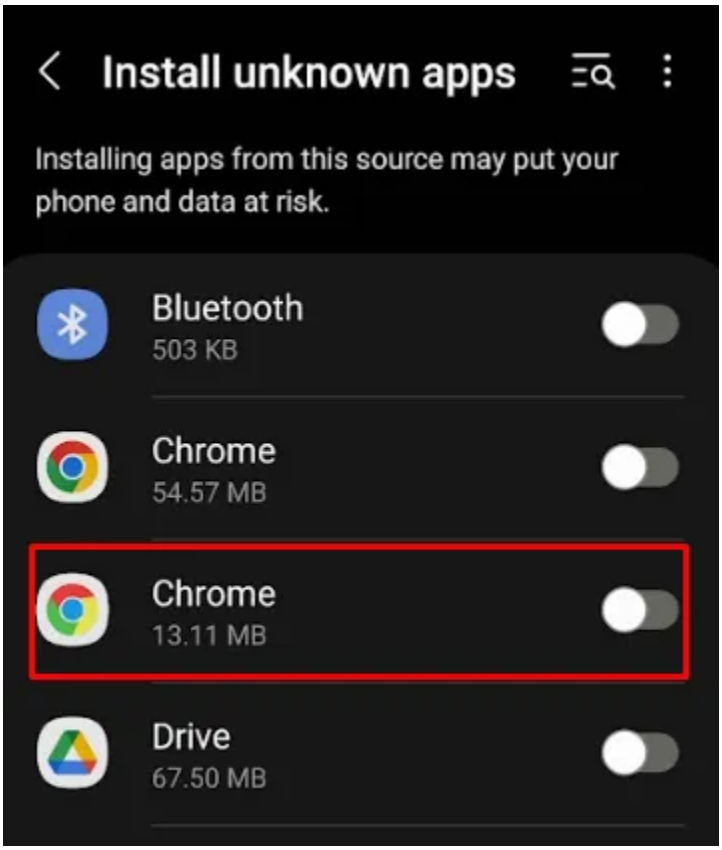
The application also requested permission to send notifications.



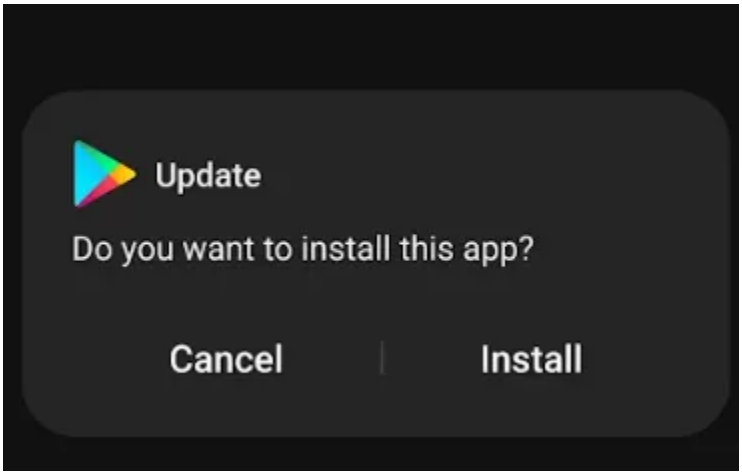
Additionally, it asked for permissions to install extra applications — the downloaded app was a dropper, meaning it contained another malicious application.



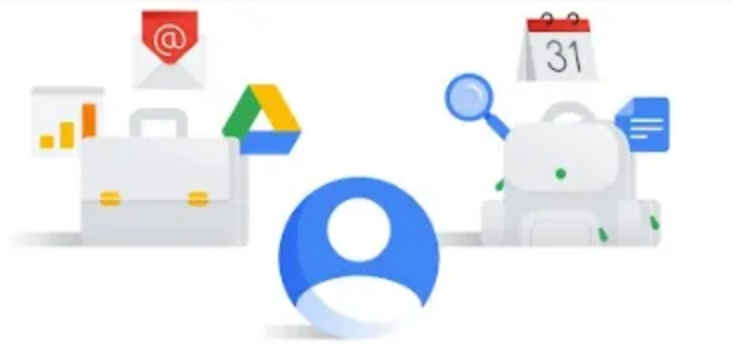
After approving the installation, differences in the app icons could be noticed — the original one on top and the fake one below.



After granting permission, a request to install the *Update* application appears.



The hidden secondary application also requested Accessibility access to take control of the device.



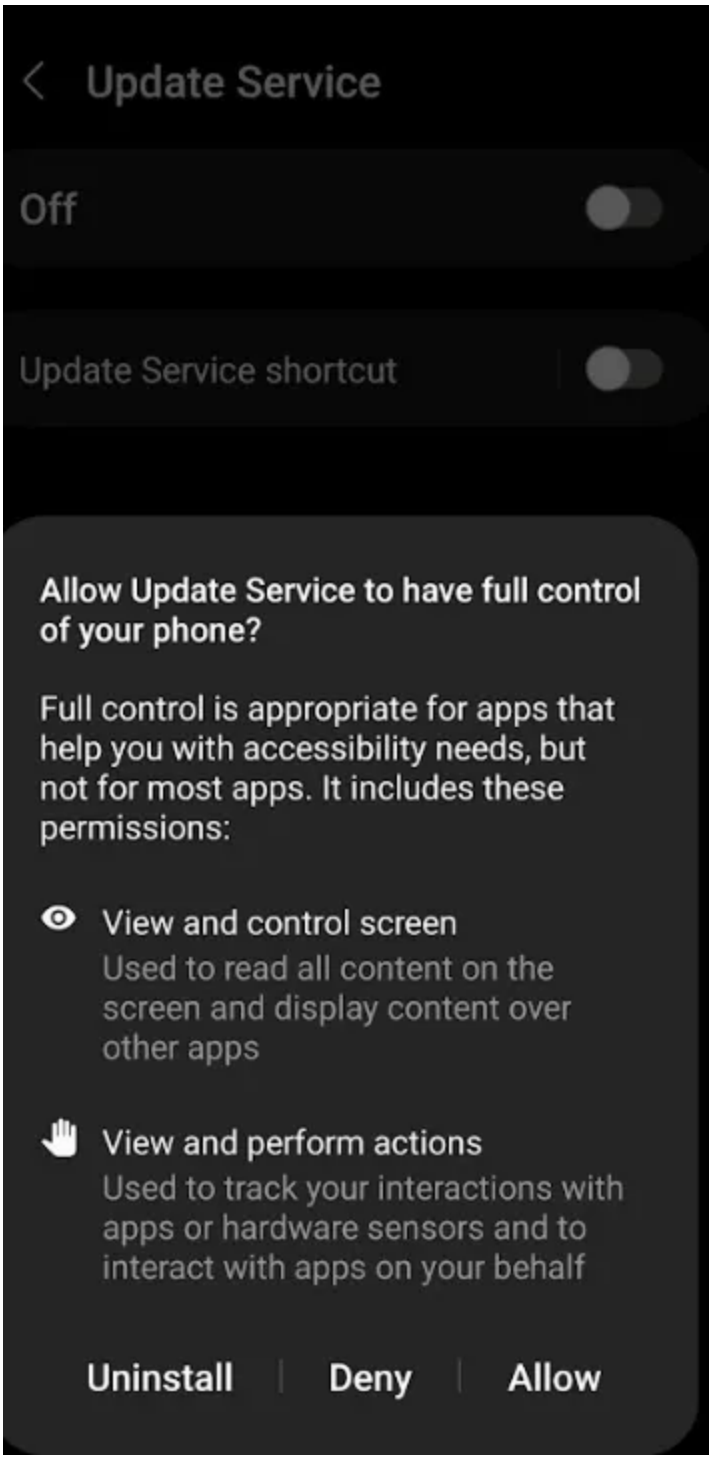
For the app to work correctly,
Accessibility needs to be enabled.

1 - Tap the Settings button

2 - Go to "Installed apps"

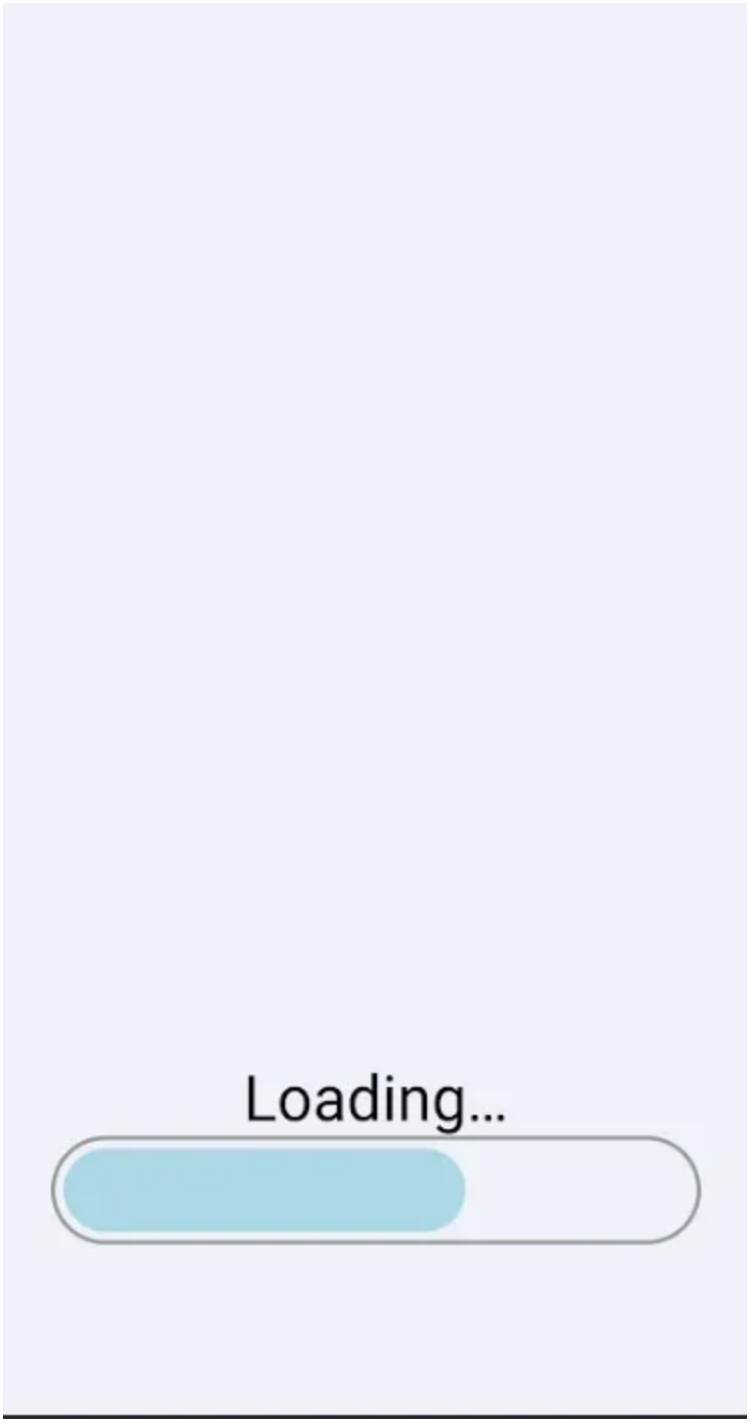
3 - Enable **"Update Service"**

Settings



Once permissions were granted, a loading screen appeared for the victim, serving as a cover for malicious activities running in the background.

The victim's screen becomes locked and it is very difficult to get out of it.



So far, two compromised websites following this attack scenario have been identified in Poland. Both distributed applications were connecting to the same C2, but their checksums differed.

The malware is obfuscated using a custom encryption code and packed with JSONPacker.

Code connected with accessibility services (obfuscated names with long numbers)

```
public final void onAccessibilityEvent(android.view.accessibility.AccessibilityEvent r12) {
        r11 = this;
        r0 = 8
        r1 = -142197164544482(0xfffff31136da17ae, double:NaN)
        java.lang.String r1 = defpackage.Mm.q(r1)
        defpackage.Qe.o(r1, r12)
        java.lang.String r1 = defpackage.Ye.b
        long r1 = java.lang.System.currentTimeMillis()
        defpackage.Ye.a = r1
        int r1 = r12.getEventType()
        r2 = 1
        if (r1 == r2) goto L755
        if (r1 == r0) goto L751
        r3 = 16
        r4 = 0
        if (r1 == r3) goto L5ef
        r3 = 32
        if (r1 == r3) goto L5ea
        r3 = 64
        if (r1 == r3) goto L5cf
        r3 = 2048(0x800, float:2.87E-42)
        if (r1 == r3) goto L31
        return
    L31:
        r5 = -233374717700178(0xffff2bbf36da17ae, double:NaN)
        defpackage.Mm.q(r5)
        r5 = -233409077438546(0xffff2bb736da17ae, double:NaN)
        defpackage.Mm.q(r5)
        android.view.accessibility.AccessibilityNodeInfo r1 = r11.getRootInActiveWindow()     // Catch: java.lang.Throwable -> L10e
        if (r1 != 0) goto L49
        goto L758
    L49:
        java.lang.CharSequence r1 = r12.getPackageName()     // Catch: java.lang.Throwable -> L10e
        java.lang.String r1 = r1.toString()     // Catch: java.lang.Throwable -> L10e
        java.lang.CharSequence r3 = r12.getClassName()     // Catch: java.lang.Throwable -> L10e
        java.lang.String.valueOf(r3)     // Catch: java.lang.Throwable -> L10e
        r5 = -233434847242322(0xffff2bb136da17ae, double:NaN)
        java.lang.String r3 = defpackage.Mm.q(r5)     // Catch: java.lang.Throwable -> L10e
        java.lang.String r3 = defpackage.U9.c(r3)     // Catch: java.lang.Throwable -> L10e
        r5 = -233473501947986(0xffff2ba836da17ae, double:NaN)
        java.lang.String r5 = defpackage.Mm.q(r5)     // Catch: java.lang.Throwable -> L10e
        boolean r3 = r3.equals(r5)     // Catch: java.lang.Throwable -> L10e
```

And encryption code:

```
public static String q(long j2) {
    String[] strArr = d;
    long j3 = 4294967295L & j2;
    long j4 = (j3 ^ (j3 >>> 33)) * 7109453100751455733L;
    long M = AbstractC0563s8.M(((j4 ^ (j4 >>> 28)) * (-3808689974395783757L)) >>> 32);
    long j5 = (M >>> 32) & 65535;
    long M2 = AbstractC0563s8.M(M);
    int i2 = (int) (((j2 >>> 32) ^ j5) ^ ((M2 >>> 16) & (-65536)));
    long M3 = AbstractC0563s8.M(M2) ^ (strArr[i2 / 8191].charAt(i2 % 8191) << 32);
    int i3 = (int) ((M3 >>> 32) & 65535);
    char[] cArr = new char[i3];
    for (int i4 = 0; i4 < i3; i4++) {
        int i5 = i2 + i4 + 1;
        M3 = AbstractC0563s8.M(M3) ^ (strArr[i5 / 8191].charAt(i5 % 8191) << 32);
        cArr[i4] = (char) ((M3 >>> 32) & 65535);
    }
    return new String(cArr);
}
```

```
public abstract class Mm {
    public static Km a;
    public static long b;
    public static final String[] d;
    public static final byte[] c = {65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 97, 98, 99, 100, 101, 102, 10
    public static final C0795zn e = new C0795zn("COMPLETING_ALREADY", 1);
    public static final C0795zn f = new C0795zn("COMPLETING_WAITING_CHILDREN", 1);
    public static final C0795zn g = new C0795zn("COMPLETING_RETRY", 1);
    public static final C0795zn h = new C0795zn("TOO_LATE_TO_CANCEL", 1);
    public static final C0795zn i = new C0795zn("SEALED", 1);
    public static final C0658va j = new C0658va(false);
    public static final C0658va k = new C0658va(true);
    public static final Ua l = new Ua(14, false);
    public static final int[] m = {2130903315};
    public static final int[] n = {2130903322};

    static {
        d = r2;
        String[] strArr = [redacted]
    }
```

In the dex file, we find the application's C2:

IOCs

```
Chrome (dropper) com.hilabilu.device 36b70e1789115dc4edfef8b7379f018f
Antidot com.rocanoji.platform f6961a4bbd916f1e85f6a954f1155fb4

Chrome (dropper) com.zabogutajo.associative 83cc7472eb4efc947f3d7c1ebd410e85
Update (Antidot) com.fagulave.data 0772b1116df1586b419acfbff9f8d96c

C2 https://gofromstr.store:8501
```