How Flubot targets Android phone users and their money

nortonlifelock.com/blogs/research-group/flubot-targets-android-phone-users

Malware steals login credentials for banking and cryptocurrency apps — and it could spread around the world

Flubot, also called Cabassous, is an Android banking malware (also a banking trojan) that is pushed by cybercriminals in large-scale campaigns, targeting consumers across Europe this spring.

Access to the botnet is being sold in underground forums by the operators to criminal groups as a so-called malware-as-a-service (MAAS). The actors behind the Flubot botnet sending Smishing (SMS phishing) messages with fake notices of upcoming package deliveries and urge the victim to follow a link to track the shipment.

The landing page then presents a download button supposedly required to track the package. After a victim falls for this social engineering trick, Flubot is downloaded to the mobile device and requests various permissions, including access to the contact list, sending SMS messages, and overlaying other applications.

The contact list is subsequently sent to a command-and-control (C&C or C2) server and used to seed new waves of smishing messages that are sent through infected phones (devices).

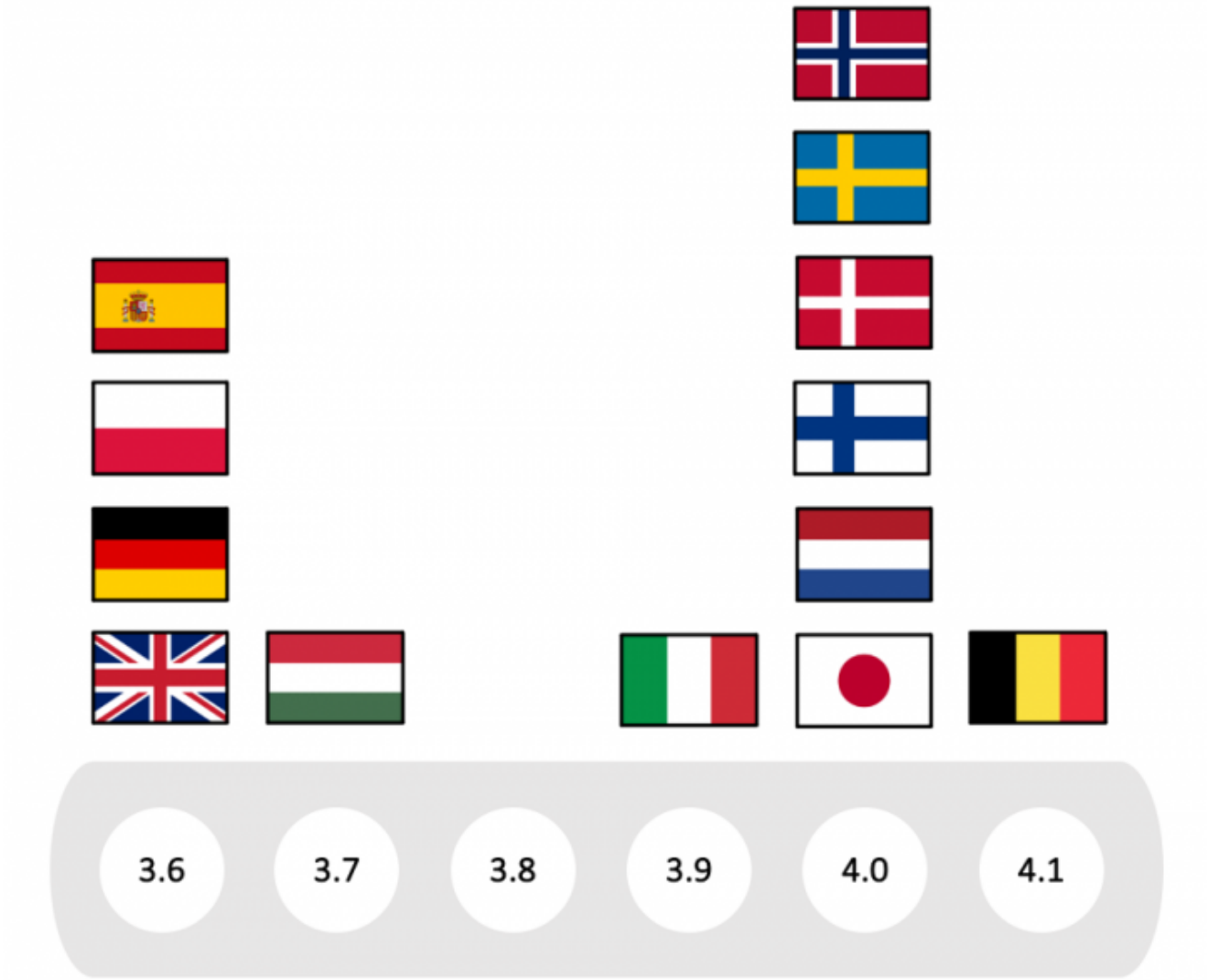
The Flubot malware does not exploit any vulnerabilities in the Android OS or targeted devices but prompts the user to manually grant two powerful system permissions. These permissions allow the attackers to steal credentials from banking and cryptocurrency apps on infected devices, using overlays and then exfiltrate one-time-password (OTP) and two-factor authentication (2FA) codes.

Campaigns and versions

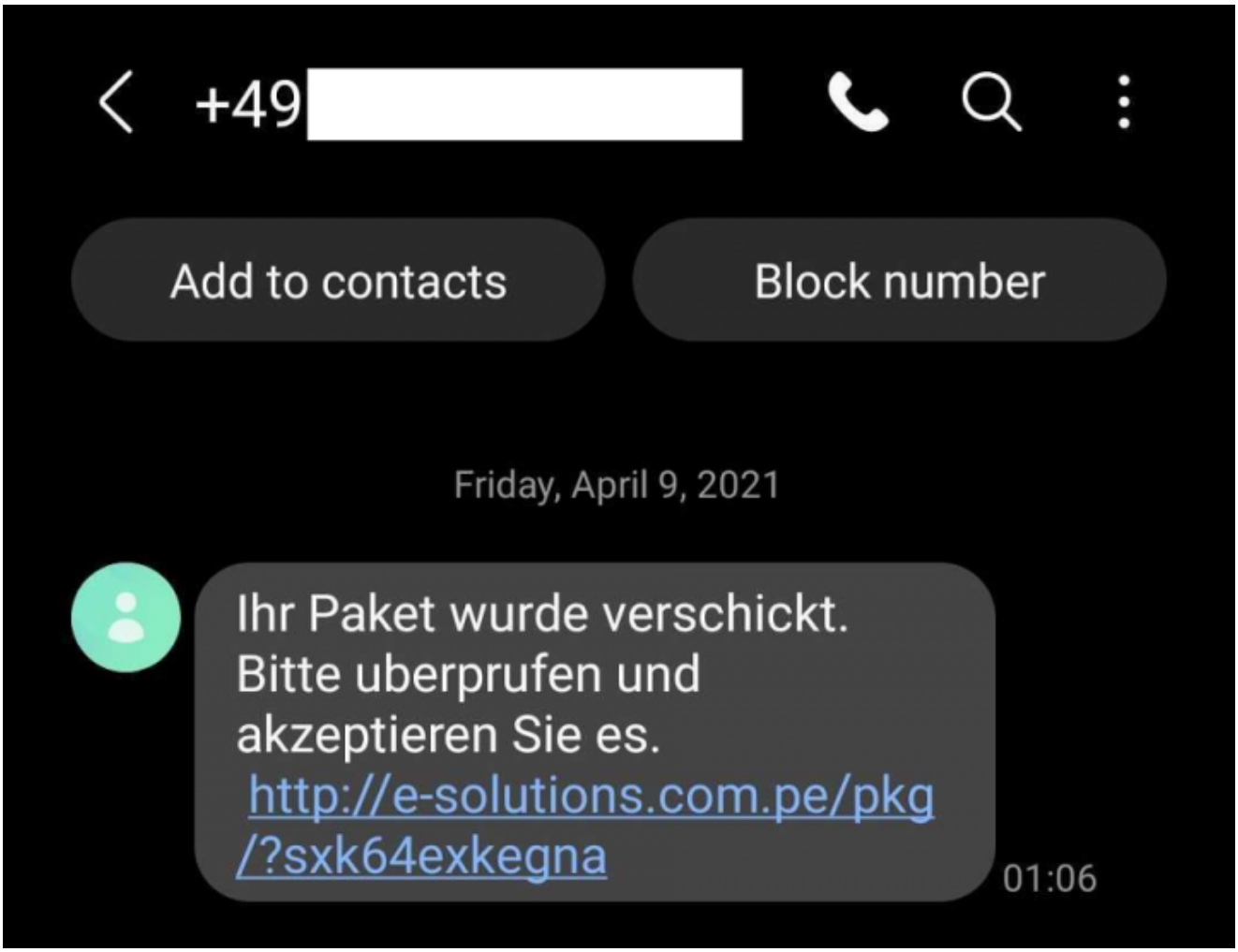
From late 2020 and into early 2021, an initial Flubot campaign hit Spain and reportedly infected more than 60,000 Android devices. Spanish police investigated and eventually arrested four men suspected of being involved in the campaign.

As mentioned earlier, the Flubot operators had been renting out the botnet, so the suspects arrested in Spain are likely just "customers" of the actual Flubot operators, trying to monetize access to victims' mobile phones. This is further evidenced by the fact that new campaigns against additional countries started to emerge soon after.

The Flubot developers have been busy adding support for new target countries to the malware as shown for version 3.6 to 4.1 in the visualization below.



The screenshot below shows an example of a SMS message targeting German users to lure them into following a link to track a fictitious package shipment:

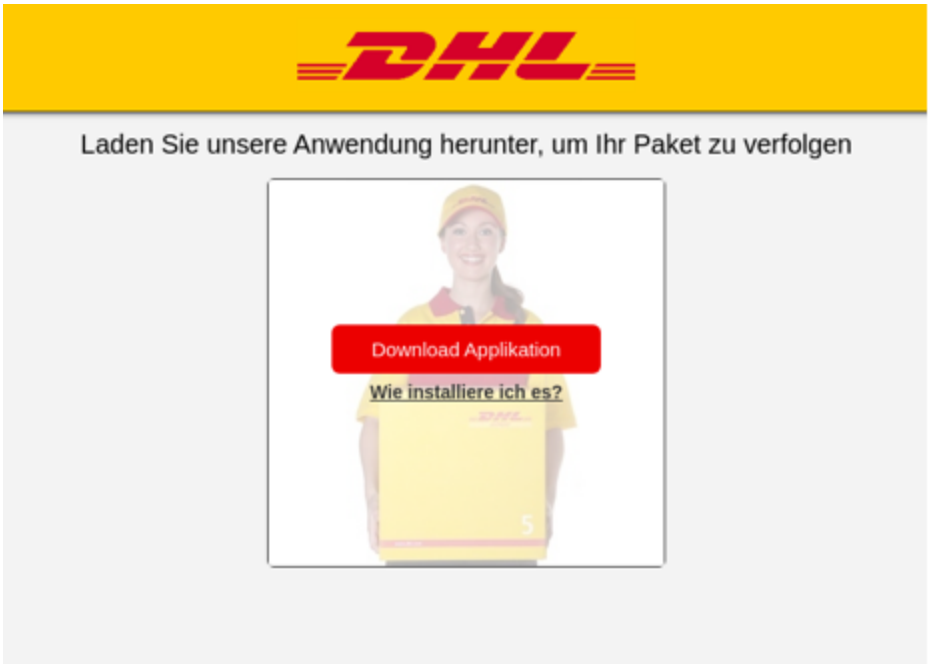


In some cases, the messages include the name of the recipient-— likely crafted using stolen contact-list data from previous victims.

The respective links lead to country-specific landing pages, as can be seen in the comparison of DHL-themed download sites below:

Slide 1 of 5

Slide 2 of 5

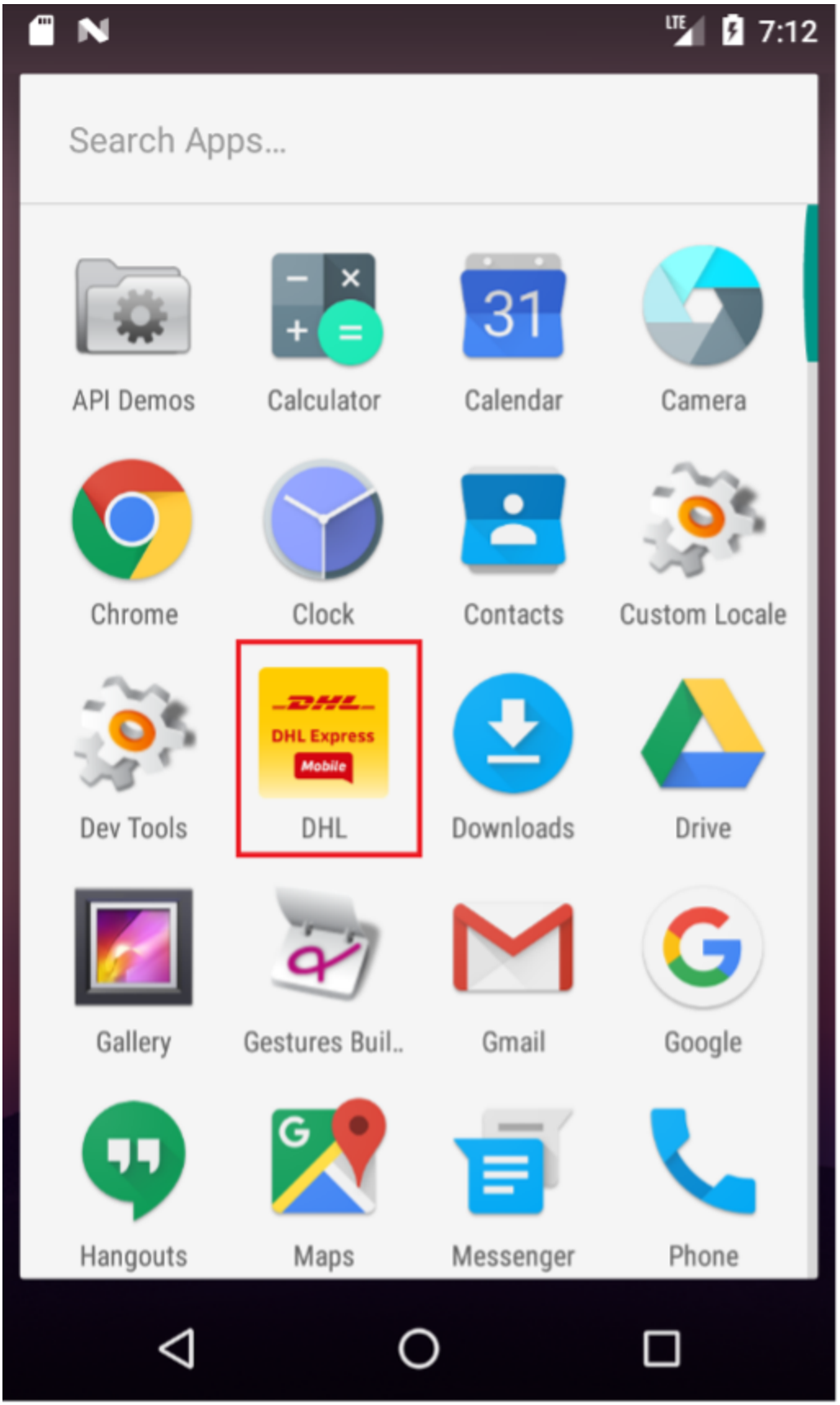


Slide 3 of 5

Slide 4 of 5

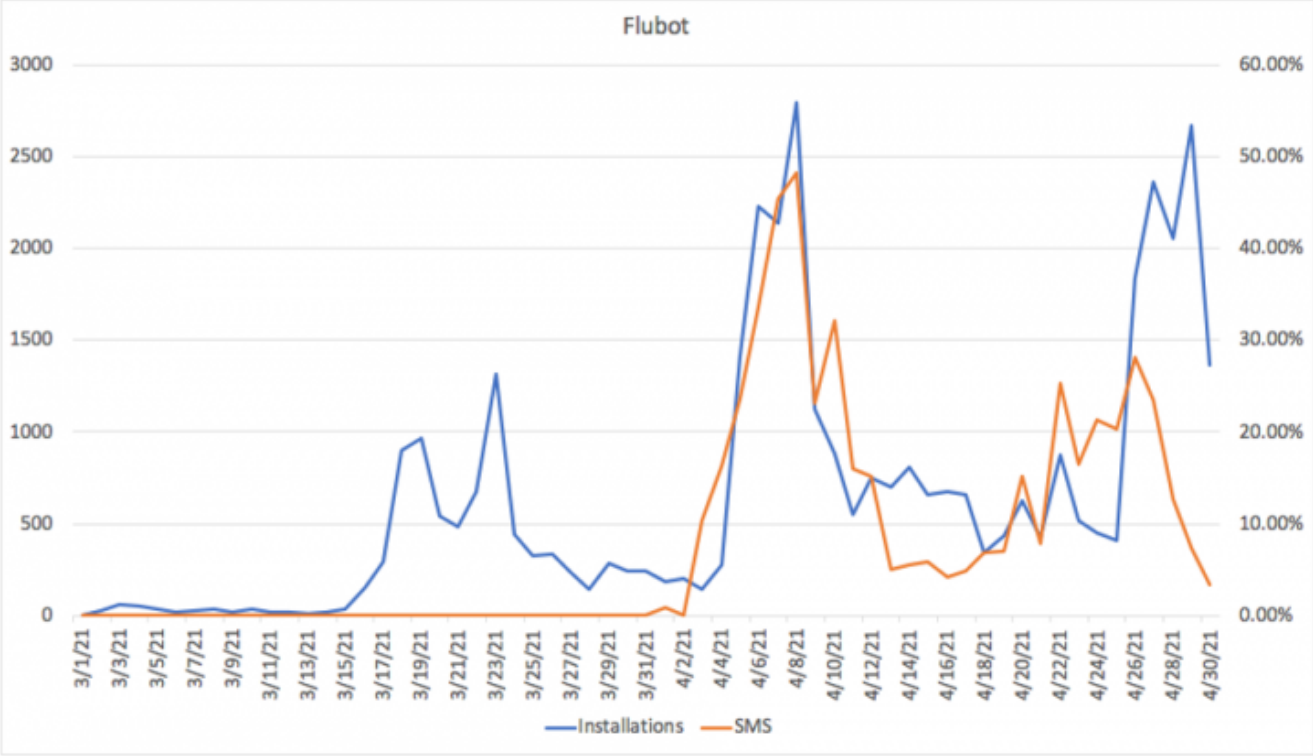
Slide 5 of 5

There are different variants of the Flubot app that disguise themselves by using the brand names of package delivery services like DHL, FedEx, UPS, or the Spanish postal service Correos.


7:12
Search Apps...
API Demos
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Custom Locale
Dev Tools
DHL Express
Mobile
DHL
Downloads
Drive
Gallery
Gestures Buil..
Gmail
Google
Hangouts
Maps
Messenger
Phone

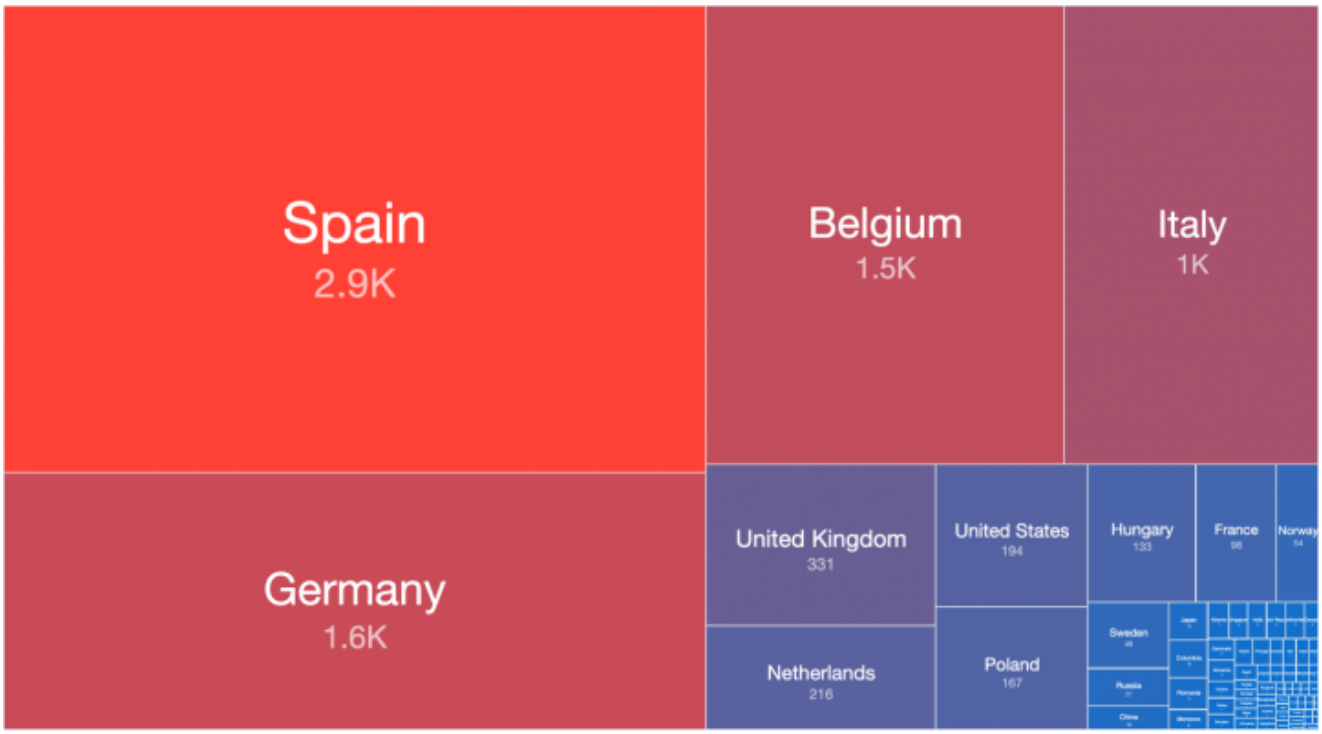
Statistics

Norton Labs researchers analyzed the telemetry of our mobile protection technologies to measure the prevalence of Flubot affecting our customers. The chart below shows the total number of attempted installations of Flubot in blue. The orange line represents Flubot lure SMS messages as a percentage of total blocked SMS messages from unknown senders for the respective day.



The second wave of attempted Flubot installations in our statistics from around April 4 to April 11 coincides with the campaign in Germany. The third wave of attempted installations at the end of April corresponds with the start of the Flubot campaign targeting mobile phone numbers in the UK. The related smishing attacks (orange line) peaking a few days earlier.

Shown below are metrics from sinkholing efforts (as provided by Shadowserver on May 11 (and previously on April 27) for seeds 1136, 1642, 1813, 1905, 1949, and 2931. These efforts show a decline in infections (Refer to the table of countries and seed values below). As with many sophisticated malware families, it's reasonable to expect an ebb and flow in the infection (and detection) rates as sinkholing and mitigation efforts are leapfrogged by the operator's improvements.



Analysis

The Flubot malware continues to evolve through active development since its introduction to the cybercrime underground. Especially since March of 2021, several versions were released ranging from 3.5 to 4.1. Even for samples with the same version number, our analyses revealed differences and the addition of new features. The research described in this section is the result of analyzing variants of Flubot version 4.0 that only differed in the list of targeted countries but not in functionality.

Permissions

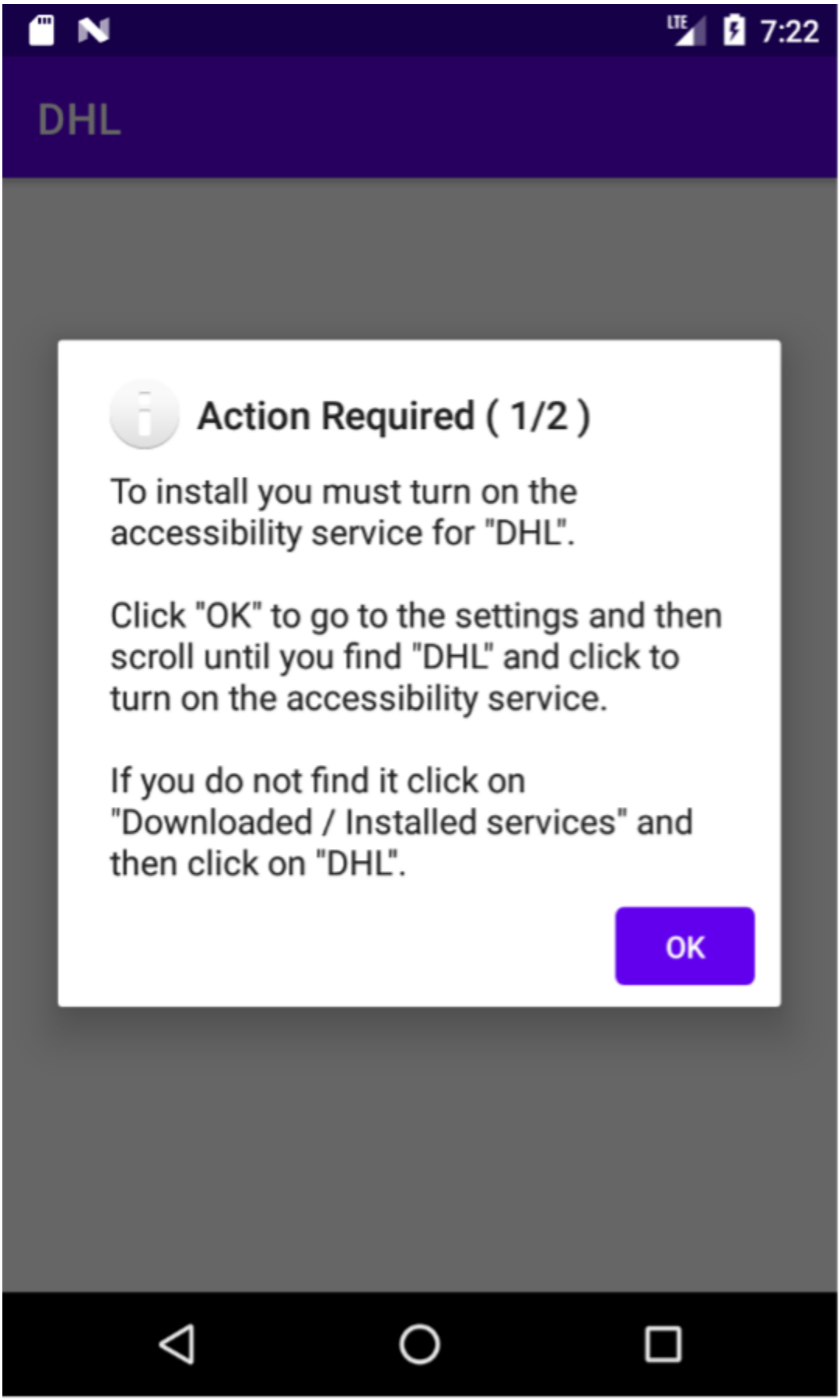
Flubot requests an extensive list of Android permissions through the system dialog upon installation:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.eg.android.AlipayGphone">
    <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="28"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_SMS"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.REQUEST_DELETE_PACKAGES"/>
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

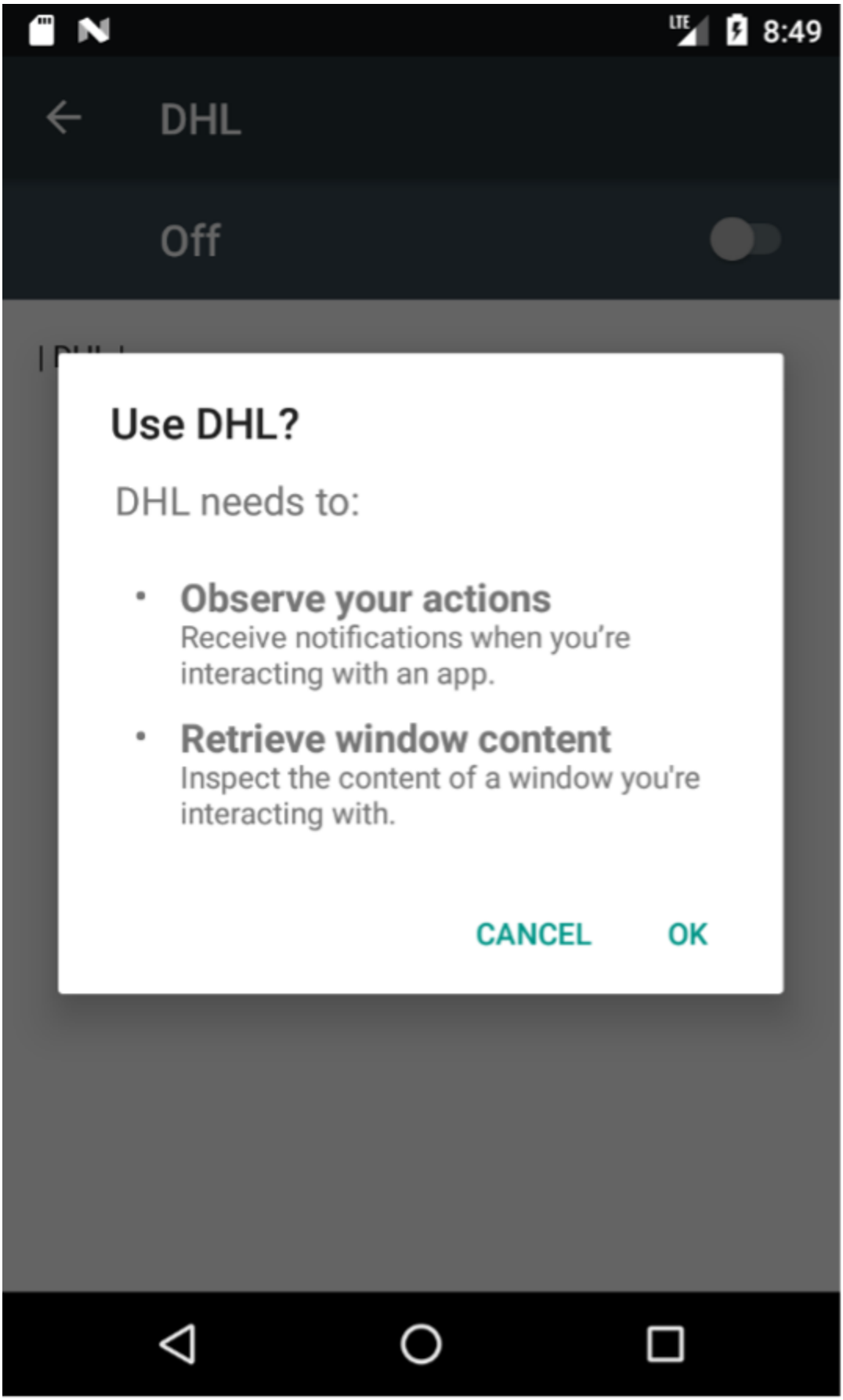
Once the app is executed, it prompts the user to grant two additional permissions by following instructions to manually allow “Accessibility Service” and “Notification Access” through the phone’s system settings.

Below screenshots show the app prompting the user to enable the “Accessibility Service” and its respective setting screen. The Android “Accessibility Service” is supposed to allow apps to assist users with disabilities. This app setting grants powerful permissions to control what is shown on the phone display and to intercept user input. Several Android malware families have integrated prompts to socially engineer the user into willfully enabling the accessibility setting to then use it for keylogging of credentials and overlay app screens (see below).

Slide 1 of 2

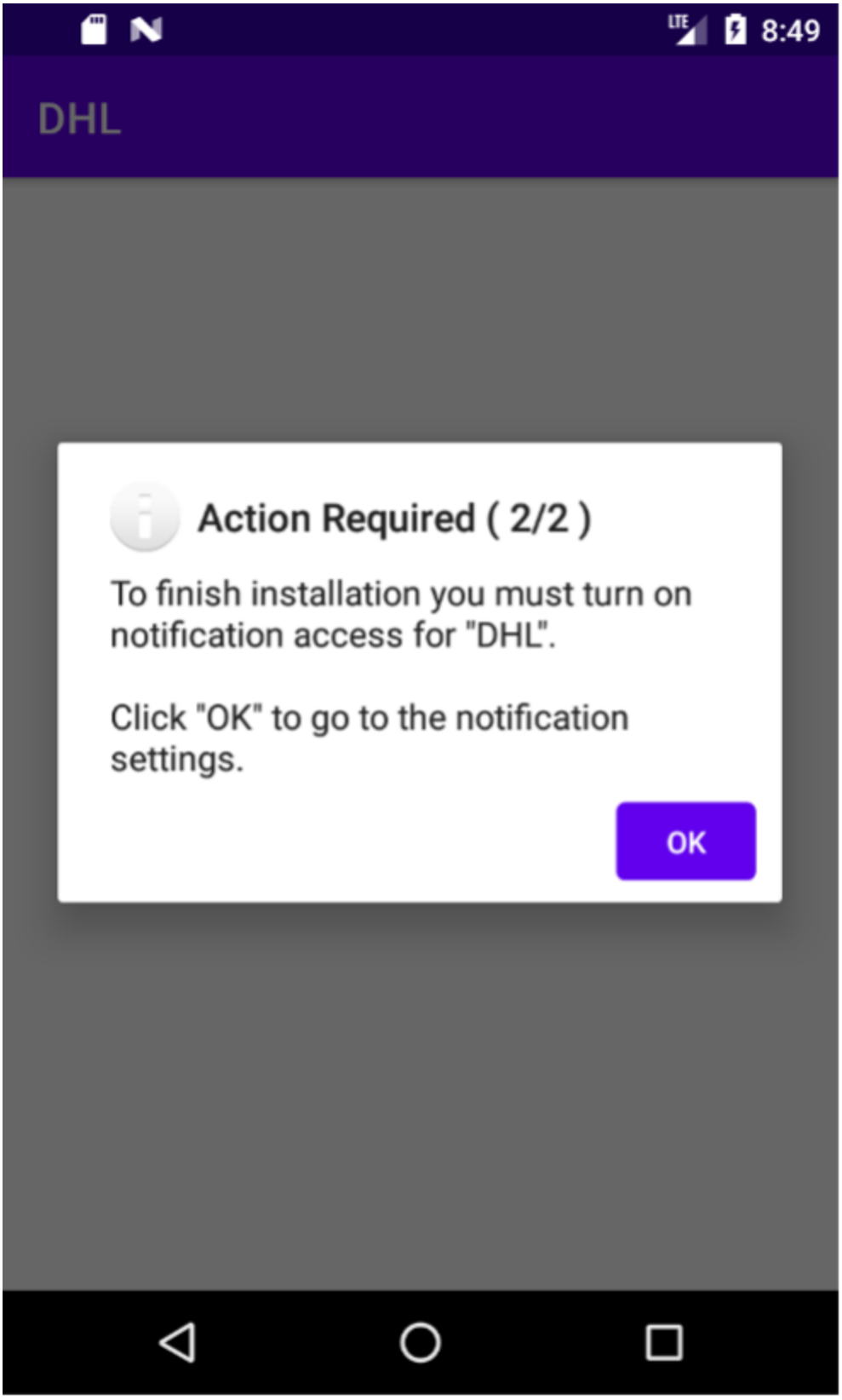

7:22
DHL
Action Required (1/2)
To install you must turn on the accessibility service for "DHL".
Click "OK" to go to the settings and then scroll until you find "DHL" and click to turn on the accessibility service.
If you do not find it click on "Downloaded / Installed services" and then click on "DHL".
OK

Slide 2 of 2

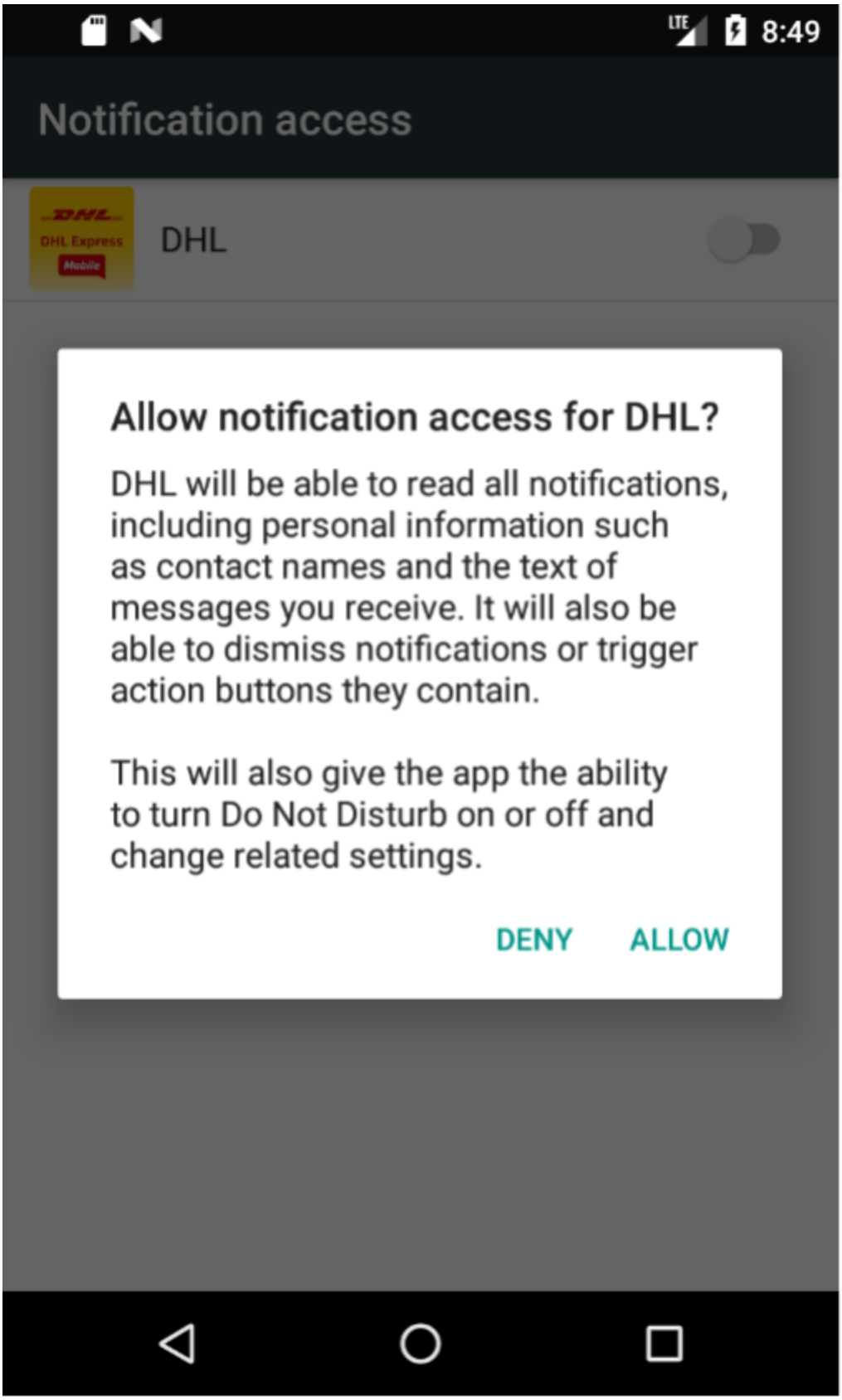

8:49
DHL
Off
Use DHL?
DHL needs to:
Observe your actions
Receive notifications when you're interacting with an app.
Retrieve window content
Inspect the content of a window you're interacting with.
CANCEL
OK

In the next step, the user is prompted to also enable “Notification Access” for the app in the corresponding system dialog.

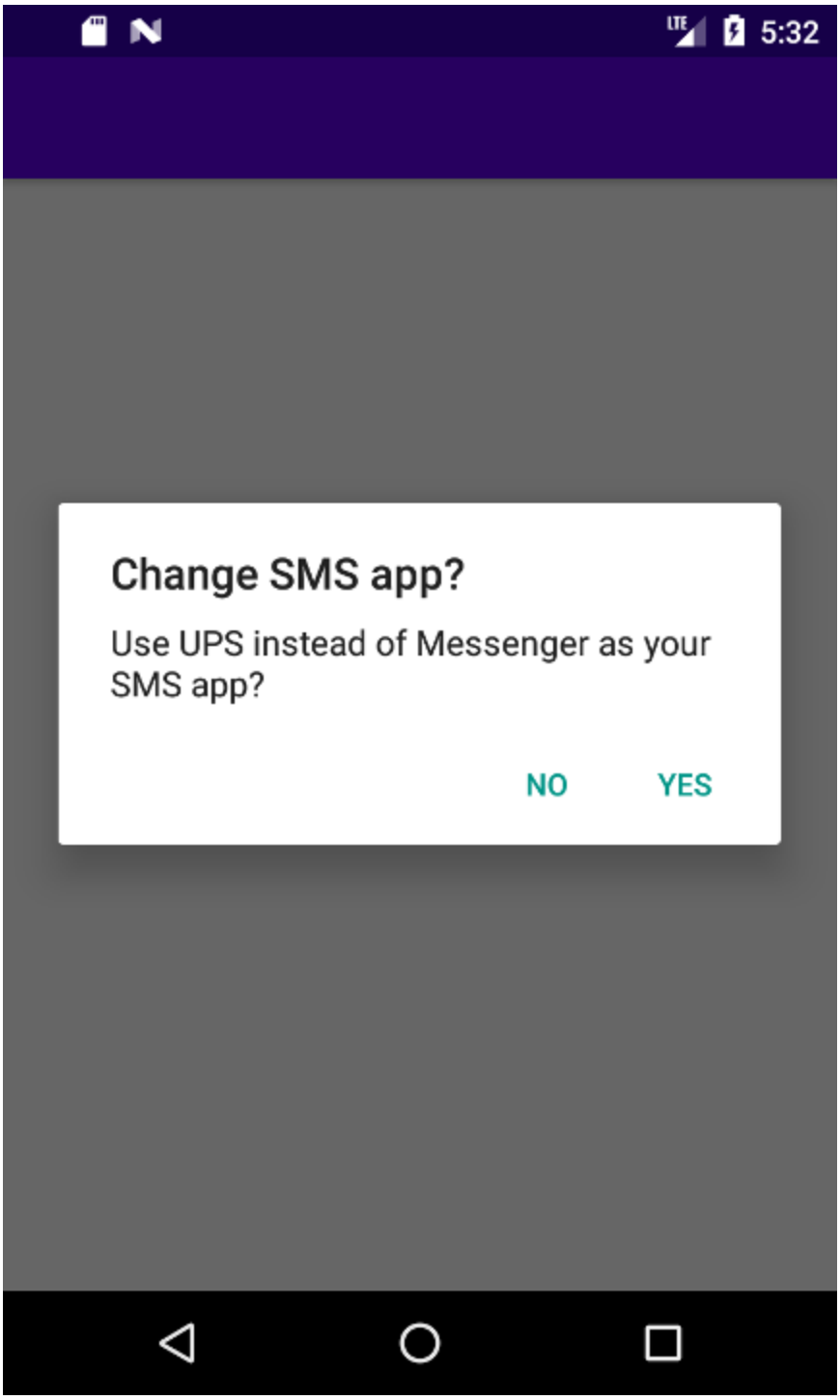
Slide 1 of 2



Slide 2 of 2



Flubot will use the “Accessibility Service” permission to register as the default messaging app on the phone. This allows the malware to quietly send and intercept SMS messages and thus capture OTPs used in online banking.



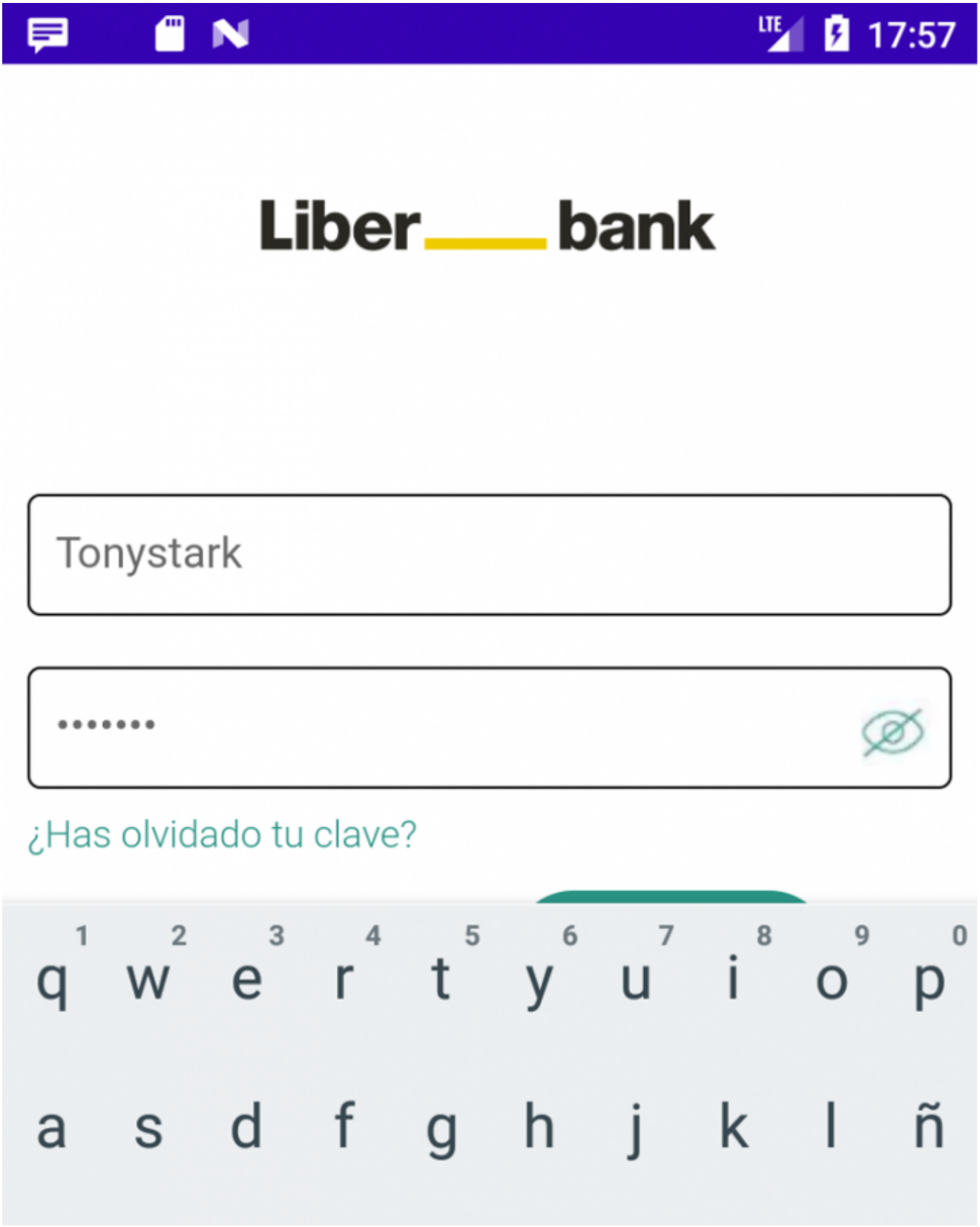
Overlays

The main functionality of the Flubot mobile banking trojan is to steal login credentials for banking and cryptocurrency apps running on the phone. The malware achieves this using the previously mentioned permissions and abusing existing Android APIs.

After installation, it will send a list of installed applications that the C2 server matches to a list of package names targeted by the cybercriminals. Flubot will then download specifically crafted HTML pages that will be displayed as an “overlay” when the respective app is

started. When the user then enters their credentials, they interact with Flubot instead of the banking app and their credentials are harvested.

The following screenshot shows an overlay captured on an infected analysis device that intercepts the login for Spanish Liberbank:



Domain generation algorithm

Instead of using a static list of C2 servers, the developers added a domain generation algorithm (DGA) to Flubot. Early versions Flubot generated a list of 2,000 domains each month, while the most recent version 4.0 can generate up to 5,000 domains. When generating the domains, the algorithm alternates between appending the top-level domains .ru (Russia), .su (Senegal), and .cn (China). This is notable, because it is more difficult for network operators to sinkhole domains in these top-level domains.

```
for (int i = 0; i < 5000; i++) {
    String a2 = a.a("");
    for (int i2 = 0; i2 < 15; i2++) {
        a2 = a2 + ((char) (random.nextInt(25) + 97));
    }
    if (i % 3 == 0) {
        sb = new StringBuilder();
        sb.append(a2);
        sb.append(a.a(".ru"));
    } else if (i % 2 == 0) {
        sb = new StringBuilder();
        sb.append(a2);
        sb.append(a.a(".su"));
    } else {
        sb = new StringBuilder();
        sb.append(a2);
        sb.append(a.a(".cn"));
    }
    arrayList.add(sb.toString());
}
```

The algorithm calculates the initialization value for the random number generator in the function above using two parameters: the year/month and a static seed value.

```
private static void d() {
    int i = Calendar.getInstance().get(1);
    int i2 = Calendar.getInstance().get(2);
    long j = (long) ((i ^ i2) ^ 0);
    f1392a = j;
    long j2 = j * 2;
    f1392a = j2;
    long j3 = j2 * (((long) i) ^ j2);
    f1392a = j3;
    long j4 = j3 * (((long) i2) ^ j3);
    f1392a = j4;
    long j5 = j4 * (((long) 0) ^ j4);
    f1392a = j5;
    f1392a = j5 + ((long) d);
}
```

Until Flubot version 3.7 the seed value remained static, while later versions include different seed values depending on the country code of the infected Android phone as shown in the screenshot below:

```
case '\f':
    k.f1403a = a.a("49");
    k.f1404b = 1945;
    f1398a = e;
    break;
case '\r':
    k.f1403a = a.a("48");
    k.f1404b = 2931;
    f1398a = g;
    break;
case 14:
    k.f1403a = a.a("39");
    k.f1404b = 1813;
    f1398a = d;
    break;
case 15:
case 16:
case 17:
case 18:
    k.f1403a = a.a("34");
    k.f1404b = 1136;
    f1398a = f;
    break;
case 19:
case 20:
    k.f1403a = a.a("44");
    k.f1404b = 1642;
    f1398a = f1399b;
    break;
case 21:
case 22:
    k.f1403a = a.a("81");
    k.f1404b = 1905;
    f1398a = f1400c;
```

The current list of countries and seed values for Flubot in the analyzed variants are as follows:

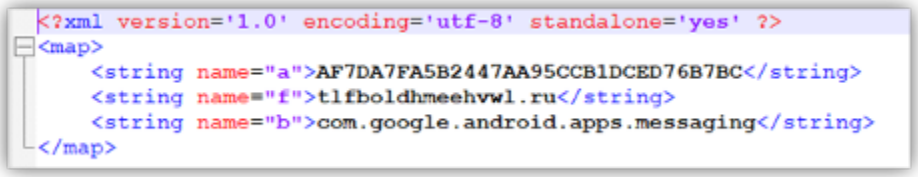
Prefix	Country	Seed Value
+34	Spain	1136
+48	Poland	2931
+49	Germany	1945
+39	Italy	1813
+44	United Kingdom	1642
+81	Japan	1905
+31	Netherlands	2931
+45	Denmark	2931
+358	Finland	2931
+46	Sweden	2931
+47	Norway	2931
+32	Belgium	2931

After generating the DGA domains, the malware will iterate through the list and try to resolve each entry to an IP. While early versions of Flubot used the DNS service configured in the phone, recent versions have switched to utilize encrypted DNS-over-HTTPS services from Cloudflare and Google.

```
public static String d(String str) {
    a.a("cloudflare-dns.com");
    a.a("/dns-query?name=%s&type=A");
    try {
        g gVar = new g();
        gVar.c(a.a("cloudflare-dns.com"));
        gVar.f(false);
        gVar.d(String.format(a.a("/dns-query?name=%s&type=A"), str));
        gVar.h(true);
        gVar.e(443);
        gVar.b(new String[][]{new String[]{a.a("Accept"), a.a("application/dns-json")}});
        if (!gVar.i()) {
            return null;
        }
        JSONArray jSONArray = new JSONObject(gVar.a()).getJSONArray(a.a("Answer"));
        return jSONArray.getJSONObject(k.f1405c.nextInt(jSONArray.length() - 1)).getString(a.a("data"));
    } catch (Exception unused) {
        return null;
    }
}
```

The communication from the C2 to the Flubot malware is encrypted using an XOR operation with a static key. Communication in the opposite direction, from the malware to the C2, uses RSA Public Key Infrastructure (PKI) to encrypt the data. This also doubles as a protection mechanism to prevent researchers or competitors from taking over the botnet via C2. While the C2 is protected through this PKI scheme, sinkholing predicted DGA domains is still possible.

The malware also generates a random “Bot ID” and retrieves device information such as Device ID and manufacturer details which is then submitted to the botnet during communication with the C2 server. The “Bot ID” and the first working DGA domain will then be stored in an XML configuration file:

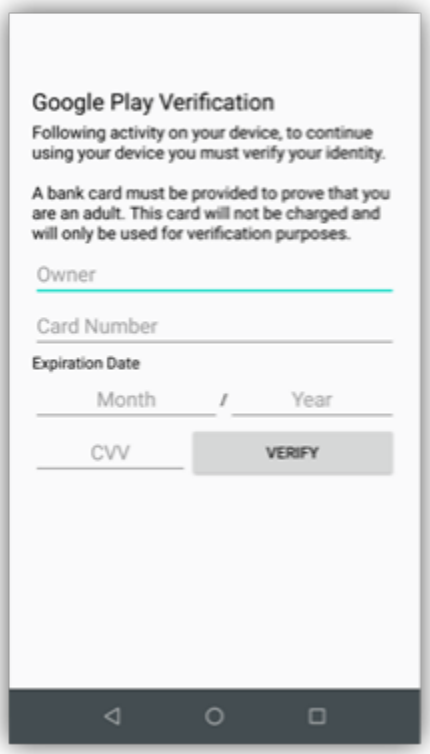


Botnet commands

When communicating with the C2 server, the botnet can respond with the following list of commands:

Command	Description
UNINSTALL_APP	Uses the accessibility service to uninstall specified apps.
CARD_BLOCK	Displays a fake “Google Play” verification asking for credit card details.
SMS_INT_TOGGLE	Toggle interception of SMS messages.
BLOCK	Blocks notifications from being displayed.
SOCKS	Opens a network port as a SOCKS proxy.
UPLOAD_SMS	Exfiltrate SMS messages.
OPEN_URL	Opens a specified URL via the browser.
NOTIF_INT_TOGGLE	Toggle interception of notifications.
RUN_USSD	Execute the specified USSD code
DISABLE_PLAY_PROTECT	Disables the Google Play Protect service.
RELOAD_INJECTS	Triggers reloading the list of applications to be injected with overlays.
SEND_SMS	Sends a specified text to a recipient phone number.
GET_CONTACTS	Retrieves and sends the phone’s contact list to the C2.
RETRY_INJECT	Retries an inject that was already shown to the user.

Besides the overlays targeting specific applications, the malware also has a generic mechanism to steal credit card information through social engineering. This is triggered by a “CARD_BLOCK” command from the C2 server and displays a “Google Play Verification Window” as shown below:



Evasion

The Flubot developers are using a custom packer and evasion mechanisms to complicate detection and analysis of the malware. The main application is packed and stored under the filename “assets/classes-v1.bin.” Once the app is executed, it unpacks the malware into “app_apkprotector_dex/classes-v1.dex” of the application’s data directory. In addition, text strings in the unpacked file are obfuscated likely using the opensource string obfuscator framework Paranoid1.

The screenshots below show a list of strings in Flubot before and after manual de-obfuscation:

Slide 1 of 2

```
static {
    a.a(-561307959114901);
    a.a(-561479757806741);
    a.a(-561909254536341);
    a.a(-561995153882261);
    a.a(-562081053228181);
    a.a(-562166952574101);
    a.a(-562252851920021);
    a.a(-562338751265941);
    a.a(-562424650611861);
    a.a(-579303872085141);
    a.a(-579647469468821);
    a.a(-579862217833621);
    a.a(-580034016525461);
    a.a(-580420563582101);
    a.a(-580764160965781);
```

Slide 2 of 2

```
a.a("4.0");
a.a("/poll.php");
a.a("a");
a.a("b");
a.a("c");
a.a("d");
a.a("e");
a.a("f");
a.a("MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAiQ3
a.a("PREPING");
a.a("PING");
a.a("LOG");
a.a("SMS_RATE");
a.a("GET_SMS");
a.a("GET_INJECT");
a.a("GET_INJECTS_LIST");
a.a("CONTACTS");
```

Version 4.1

Shortly before the publication of this report, a version 4.1 of Flubot was discovered. However, the analysis described above still apply to the new version as well.

Summary & Outlook

Mobile banking trojans like Flubot are becoming more popular among cybercriminals as more consumers than ever before rely on mobile devices for internet banking, access to financial services, and e-commerce. The Flubot malware is in active development and is likely to continue to spread throughout the world as support for new regions and functionality is implemented in future versions.

Flutbot can circumvent security controls by tricking the end-user into giving powerful system permissions. Better user awareness training is key to combatting this type of attack:

- Only install applications from trusted sources like the Google Play store.
- Only grant permissions to trusted applications and understand what capabilities that permission is granting before accepting them.

Indicators

Example hashes of analyzed Flubot variants:

- 74183f6454d2aaa44fcb363eb71beb33f04845c7fe4b402d06a87bab7b99e235
- 5c3384bfeb479db3f1ed98578c80d1e3859640ea7cbfe62fbaa9634118cf4636
- 5c9057d6d19f82fbba255d58e9b0da7102fed08ee25e548e08f0a5b22efc42a2

Active C2 domains:

- iixoqoiphdhkdbq[.]ru
- pvvbjvedsmjphil[.]ru
- tlfboldhmeehvwl[.]ru

Package names used in early Flubot versions:

- com.iqiyi.i18n
- com.eg.android.AlipayGphone
- com.tencent.mm
- com.tencent.mobileqq
- com.taobao.taobao

About the Author

Armin Buescher

Sr. Principal Security Researcher, Norton Protection Labs

Armin Buescher is a security researcher and software engineer for Norton Protection Labs and is focused on the analysis of attack trends and development of novel detection technologies. He has more than 10 years of experience working in the security industry.

linkedin

About the Author

Gokulakrishnan S

Sr. Threat Analysis Engineer

Gokulakrishnan is Sr. Threat Analysis Engineer at Nortonlifelock with 11 years of experience in Malware Analysis, Incident Response & Digital Forensics. He is an expert in monitoring and researching emerging threats.