The Brief Glory of Cabassous/FluBot — a private Android banking botnet

medium.com/csis-techblog/the-brief-glory-of-cabassous-flubot-a-private-android-banking-botnet-bc2ed7917027

Aleksejs Kuprins March 16, 2021

Aleksejs Kuprins

Mar 16, 2021

.

12 min read

Introduction

A novel piece of banking malware for Android OS has surfaced around December 2020, attacking users in Spain. In this article, we will discuss the timeline of its existence, the targeted apps, the current size and the spread of the botnet as well as the techniques and capabilities, which the authors have employed, and speculate about how effective they are.

We have found neither indications of its name/alias, nor any sales threads within the underground forums. The latter fact suggests that the malware is likely a private bot, which means that it will have a small client base in the future and no advertisement campaigns.

Timeline of Events

We have discovered the first in-the-wild samples during late December 2020. Since then, the ‘version’ field in the malware’s configuration file was being incremented rather frequently with new features and fixes added every week. The first piece of analysis about Cabassous that we can find came on January 06 from ThreatFabric.

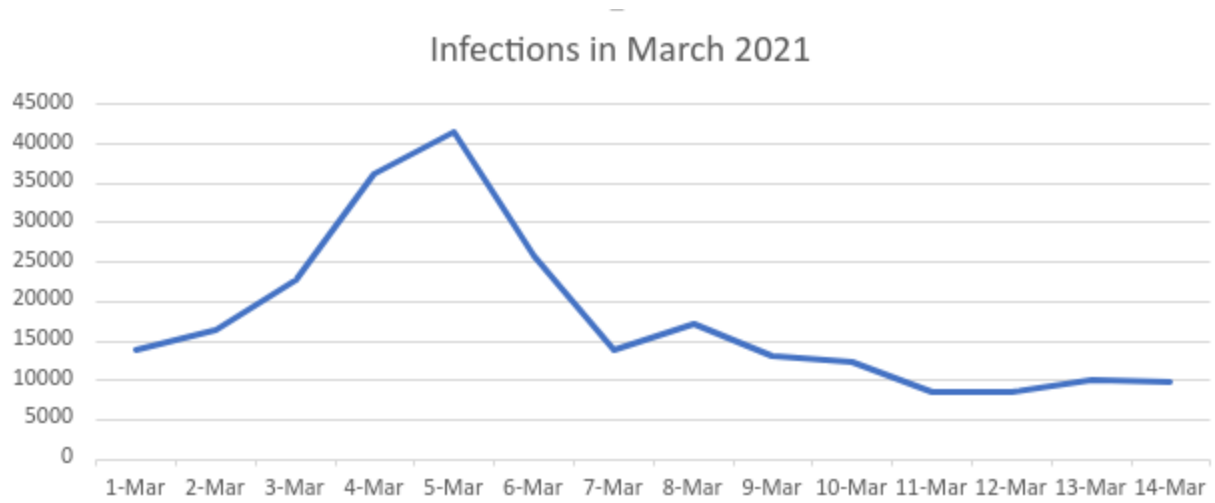
Moving onward, several researchers have been publicly documenting the malware’s spreading campaign on Twitter. Our sinkhole statistics were showing that Cabassous was growing very fast, until it reached around 60,000 infections. By that moment, it suffered a major blow from the Spanish law enforcement authorities on the day of publication of analysis by PRODAFT.

The Spanish authorities coordinated with PRODAFT and arrested 4 suspected members of the criminal group on March 05.

Catalan police released the video of the arrest
Unfortunately, the arrest did not bring the operation to a complete halt. The C&C server was offline during March 05, however, the campaign was restored a couple of days later. On the day of writing, Cabassous is alive and well, spreading the malware, defrauding its victims and even continuing the development of the bot, although the rate of new infections has slowed down.

Alberto Segura reports on the changes in the malware, days after the arrests in Spain
The same week, the malware has started to spread in Poland with the similar phishing theme. The list of targeted apps did not change yet, the latest known version at the moment of writing is 3.6.

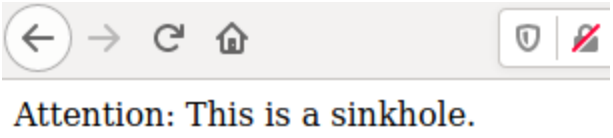
Our sinkhole stats indicate that there has indeed been a fall in the spreading of Cabassous after 5th of March, when the number of infections dropped from ~40k unique connections per day down to ~15k in two days. The statistics continue to fall after that for a different reason — the authors have updated the domain generation algorithm in the newly distributed samples.



Stats from the sinkholed domain nfiuerwtftasnuk[.]com

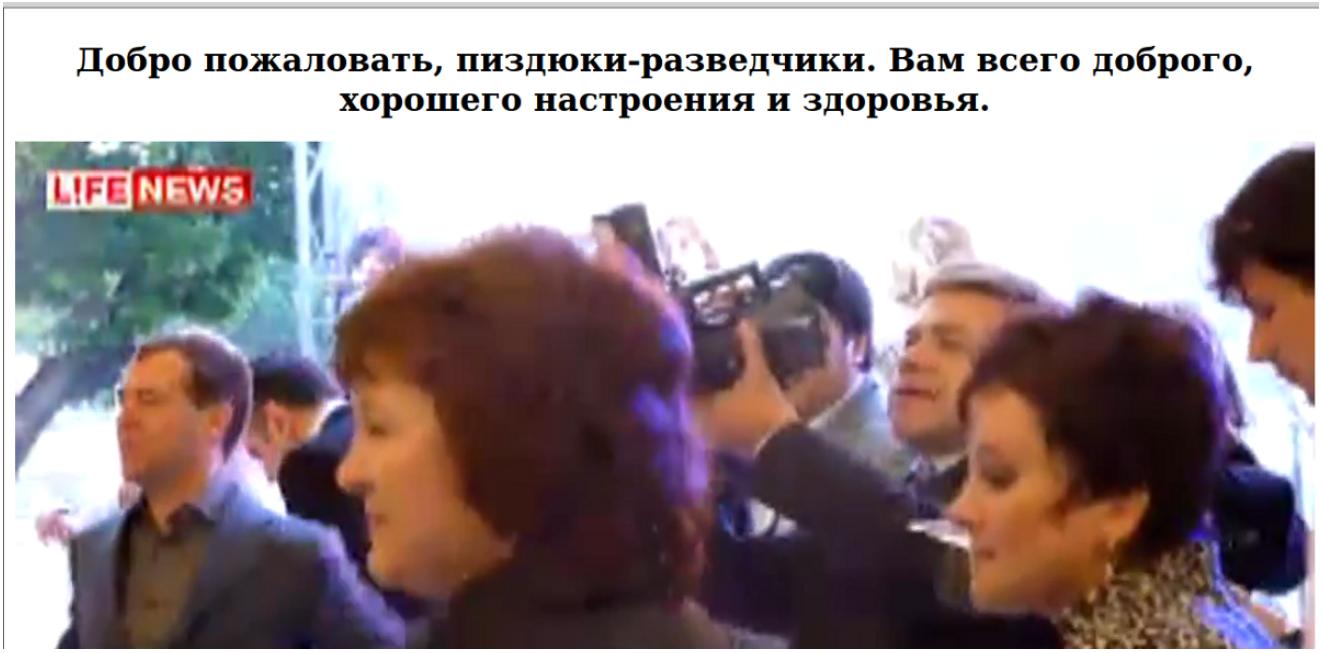
Attention: This is a sinkhole

Speaking of sinkholes, the C&C server's proxy used to display a curious message back in January, which was intended to either deter researchers from looking at the server or just an attempt at humor. Likely the latter. It is quite simple — the server's HTTP response to the root "/" URI was the following text: "Attention: this is a sinkhole". Sinkhole is a term used by security researchers and describes a web server, which was set up to have malicious domains pointing at it. This is done in order to monitor or hijack a botnet, as the malware would connect to it, believing that it is the legitimate C&C server. However in this case, it is just a fake message that means nothing and it is in fact the real C&C server's proxy into its backend.



Index page of the C&C server

Around the end of February 2021, the above message was changed to a strongly worded greeting phrase for researchers in Russian. Also a quote by Dmitriy Medvedev and a video montage of him dancing to an old Russian pop song about "American boy" and "Balalaika" from the 90s. Obviously, this page is reserved for the "public relations" and the original message about the sinkhole is more of a joke rather than a legitimate attempt to conceal the server's location.



“Greetings, researcher-****s. We wish you well, good mood and health!”

Spreading

The main target and most of the current bot pool is the mobile banking and cryptocurrency users from Spain. The spreading campaign is the widely-tested scheme of impersonation of postal and delivery services. A given victim first receives an SMS message about a package delivery with a link to the fake page of the postal service. Upon visiting, the victim is prompted to download the package tracking app, which in fact is the malware. The majority of infections likely come from the bot’s SMS spam: upon infection, the bot sends the local contact list over to the C&C server, then requests the spam message and receives it with the number to be sent to.

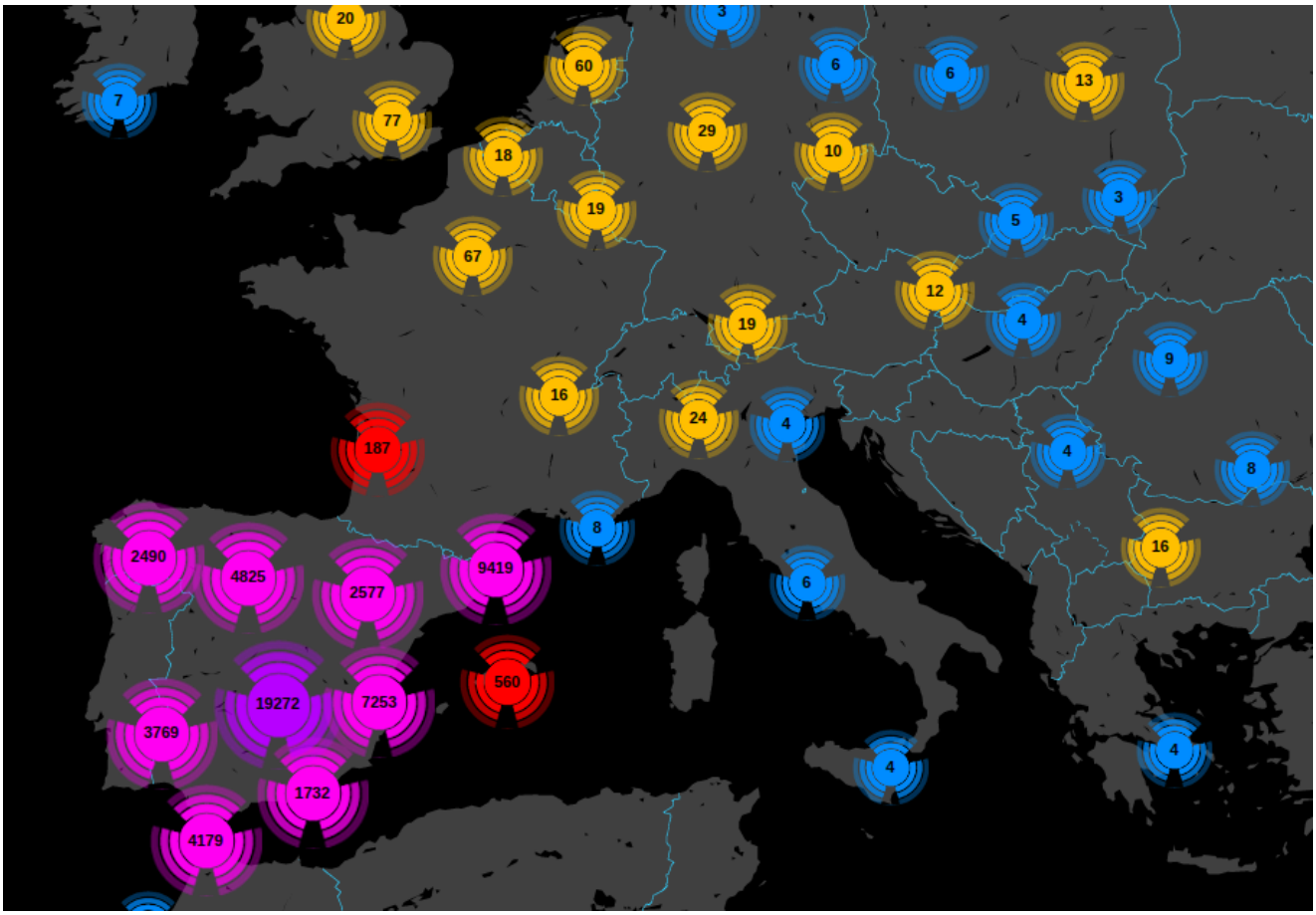
```
1   6[redacted]87,Hola, su envio se entrego el 26/02-2021 en el punto de
    entrega. Vea donde puede recoger su paquete aqui:
    http://[redacted].com/web/?9zk28o1ssj
2   6[redacted]355,Hola, su envio se entrego el 26/02-2021 en el punto de
    entrega. Vea donde puede recoger su paquete aqui:
    http://[redacted]in.com/web/?4n6rhsau7g
3   6[redacted]082,[redacted]: Estimado Antonio Mayan, su paquete ha sido
    detenida en nuestro almacen, para resolver el problema pincha aqui:
    http://[redacted].com/web/?laocnze4wn
4   6[redacted]75,Raquel Casting Miss Real Model, No pudimos entregar su
    paquete. Siga el enlace para programar una nueva fecha de entrega.
    http://[redacted]in.com/web/?y5fntkxp8m
5   6[redacted]437,No hemos podido entregar su paquete. Entre en este enlace
    para programar una nueva fecha de entrega o recogida:
    http://[redacted].com/web/?52c14h1vbg
```

Samples of the SMS spam

The less frequently seen versions of the malware lure impersonate the DHL and FedEx delivery services:

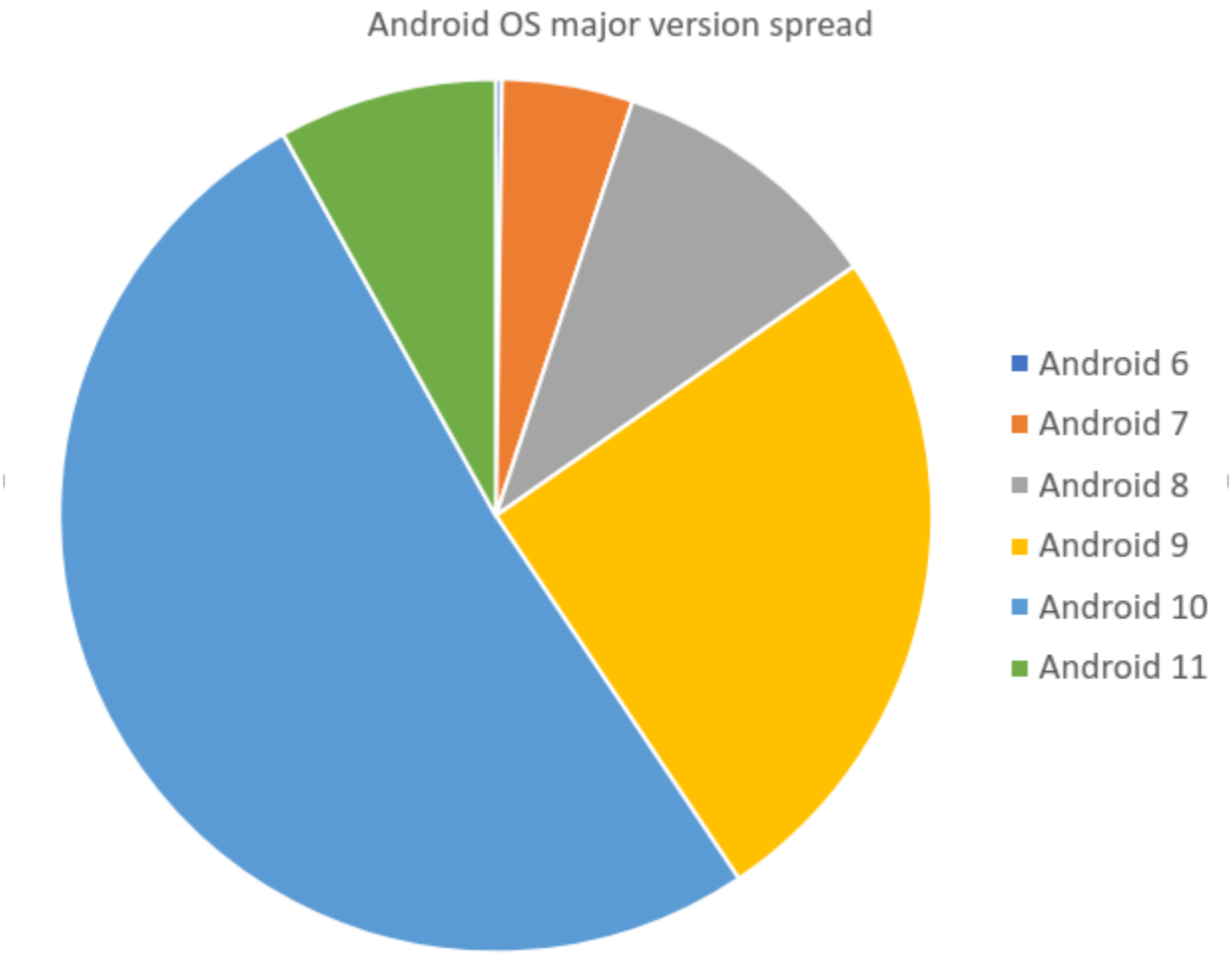
Delivery services themed lures

We have estimated the botnet to currently be of the size of around 60k unique infections in total.



Current spread of Cabassous around Europe

About half of the infected devices are running Android 10. Supporting the latest versions of Android OS seems to be working out very well for Cabassous:



Distribution of infected operating systems

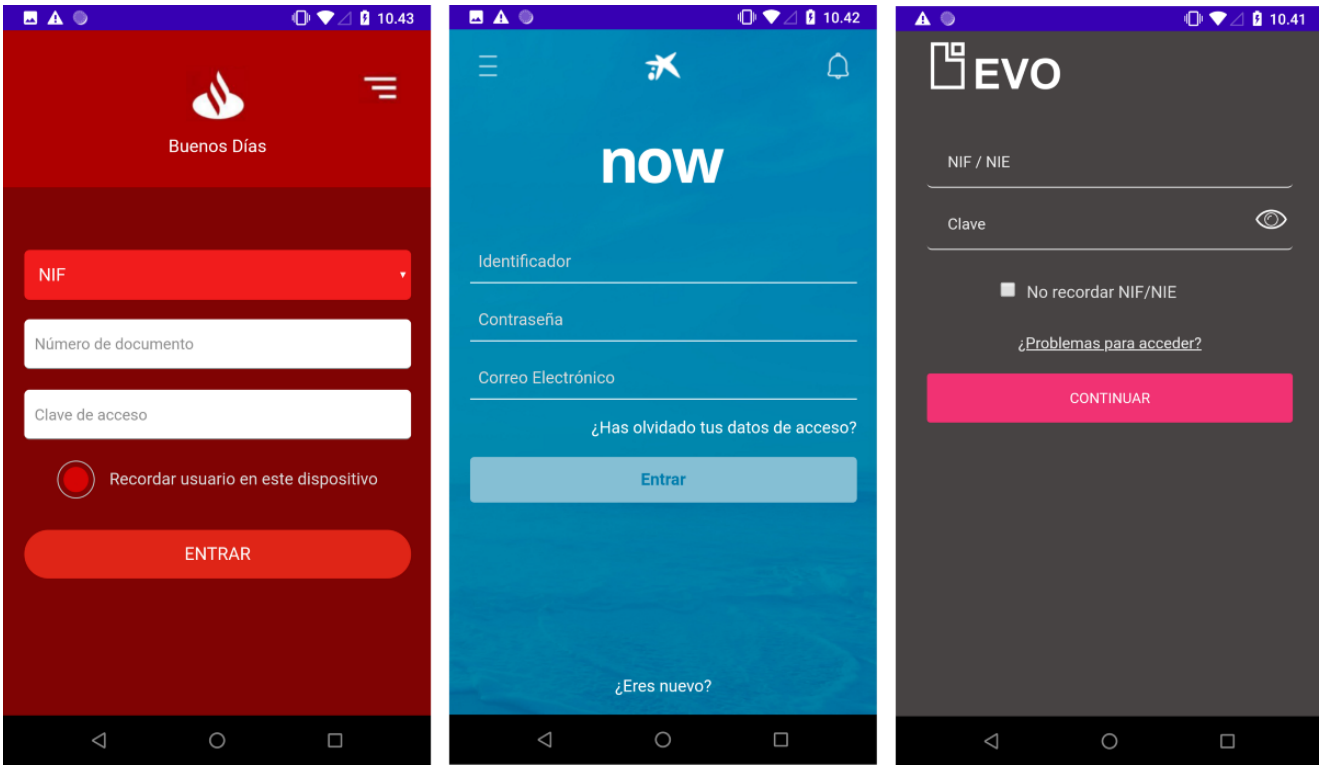
The following applications are targeted by overlay attacks (aka injects). Spanish banking apps:

```
“Bankinter Móvil” – com.bankinter.launcher“BBVA Spain | Online banking” –
com.bbva.bbvacontigo“Cajasur” – com.cajasur.android“Grupo Cajamar” –
com.grupocajamar.wefferent“Imagin. Much more than an app to manage your money” –
com.imaginbank.app“Kutxabank” – com.kutxabank.android“ruralvía” – com.rsi“Banca Móvil
Laboral Kutxa” – com.tecnocom.cajalaboral“Santander” – es.bancosantander.apps“Bankia”
– es.cm.android“EVO Banco móvil” – es.evobanco.bancamovil“Ibercaja” –
es.ibercaja.ibercajaapp“CaixaBankNow” – es.lacaixa.mobile.android.newwapicon“Banca
Digital Liberbank” – es.liberbank.cajasturapp“Openbank – banca móvil” –
es.openbank.mobile“Pibank” – es.pibank.customers“UnicajaMovil” –
es.univia.unicajamovil“Banco Sabadell App. Your mobile bank” –
net.inverline.bancosabadell.officelocator.android“ING España. Banca Móvil” –
www.ingdirect.nativeframe
```

Cryptocurrency apps:

```
“Binance: Bitcoin Marketplace & Crypto Wallet” – com.binance.dev“Coinbase – Buy &
Sell Bitcoin. Crypto Wallet” – com.coinbase.android“Blockchain.com Wallet – Buy
Bitcoin, ETH, & Crypto” – piuk.blockchain.android
```

Here are a few examples of the inject screens:



Webinjects targeting Spanish speaking users

Capabilities

As mentioned above, the Cabassous's main business is banking fraud. It provides the following functionality:

- Intercept SMS messages
- Send SMS messages and automated contact list spam
- Display overlays/injects for banking and cryptocurrency apps, as well as a generic credit card phishing screen
- Steal contacts
- Open URLs
- Disable PlayProtect
- Run USSD commands
- Uninstall App
- SOCKS proxy

The overlays are displayed using the standard WebView.

```
private void LoadHtml() {
    String GetInject = Bot.GetInject(packageName);
    if (GetInject == null) {
        finishAndRemoveTask();
    } else {
        this.webView.loadDataWithBaseURL(null, GetInject,
            Deobfuscator$app$Release.getString(-22587301149671),
            Deobfuscator$app$Release.getString(-2301679787927L),
            null);
    }
}
```

Cabassous loading an overlay

Techniques

There are two techniques that are worthy of noting in this case. The obfuscation and the C&C communication.

Most of the samples are packed with Tencent's Legu Packer application. This solution encrypts the APK file and hides its content from the simple sandbox services. However, it is also trivial to unpack using either software like Frida, Xposed Framework, or the apklab.io service.

The second layer of obfuscation is the string encryption. The software package for that was taken from a public github repository. This means that after the first layer (Tencent Legu) is unpacked — you are still looking at the malware code with all of its strings packed into a single class in an encrypted state, while any uses of the strings are replaced with calls to the obfuscator's de-obfuscation routine, which only allows the strings to be decrypted during runtime. This obfuscation is also trivial to bypass using any runtime hooking method, such as Frida or Xposed Framework.

README.md

build passing

Paranoid

String obfuscator for Android applications.

Usage

In order to make Paranoid work with your project you have to apply the Paranoid Gradle plugin to the project. Please notice that the Paranoid plugin must be applied **after** the Android plugin.

Public Github repository of the string obfuscation module

Communication

The specialty about the C&C communication of this malware is the rare use of a DGA module (Domain Generation Algorithm). Normally a piece of any malware comes with a domain name or an IP address of the C&C server embedded into it. The use of DGA is a different approach — instead of connecting to a predefined address, the malware carries a block of code, which would manipulate text and/or digits in a specific way. The output of this operation is a list of possible domain addresses, typically a lot of them — 2000 (5000 as of version 3.6) in case of Cabassous.

Every time the malware starts up, it generates a set of possible domain names of the C&C server, tries to resolve them and then connects to the first one that resolves to an IP address and is not offline. The strength of this technique is the resilience against the domain takedowns. In the event of the current C&C server domain getting seized and taken down by either the law enforcement or the domain registry service, the malware still has a lots of other domains to use. The threat actors can then simply register a new domain from the list and all of the infected devices will automatically connect to it, thus being completely unaffected by the takedown.

Usually a malware with a DGA module will begin its domain generation from a given seed value, like a short string. This value can be frequently updated by the criminals in order to prevent the investigators from being able to predict their future domains and blocking them all before they are even used. However, the authors of Cabassous took a different approach: the seed of their algorithm is the combined string of the digits of the current year and month. They have also not added any mechanism to update the seed, or the malware itself. Basically this means that anyone can run their malware on an analysis device with a system date set to the next month and predict all of the future generated domains. The can then block all of these domains on the DNS level within their organization.

```
private static void GetSeed() {
    int i = Calendar.getInstance().get(Calendar.YEAR);
    int i2 = Calendar.getInstance().get(Calendar.MONTH);
    long j = (long) ((i ^ i2) ^ 0);
    seed = j;
    j *= 2;
    seed = j;
    j *= ((long) i) ^ j;
    seed = j;
    j *= ((long) i2) ^ j;
    seed = j;
    j *= ((long) null) ^ j;
    seed = j;
    seed = j + 1136;
}
```

The DGA seed is based on the current year and month

The order of the generated domains in this case is also intentionally randomized, but not to a great extent. It appears that the generated lists of domains are very similar with just a few differences. In our tests we observed a certain degree of predictability. To measure it, we ran

the malware on a device with the date set to the first day of the next month, then we wrote down the first 10 out of 2000 generated domains. Then we ran the test 10 more times. Around 5 domains from the first run's first 10 domains would appear within the first 10 domains in the following 10 tests, usually on almost identical position in the order of generation.

```
public static void FindHost() {
    lock.lock();
    GetSeed();
    ThreadPoolExecutor threadPoolExecutor = (ThreadPoolExecutor) Executors.newFixedThreadPool(50);
    AtomicBoolean atomicBoolean = new AtomicBoolean(false);
    AtomicReference atomicReference = new AtomicReference();
    atomicReference.set(null);
    Random random = new Random(seed);
    for (int i = 0; i < MAX_HOSTS; i++) {
        String string = Deobfuscator$app$Release.getString(-4260184874903L);
        for (int i2 = 0; i2 < 15; i2++) {
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append(string);
            stringBuilder.append((char) (random.nextInt(25) + 97));
            string = stringBuilder.toString();
        }
```

DGA order randomization feature

The version 3.6 of Cabassous has introduced an actual full randomization of the domain order, using the standard Java library — Collections.shuffle().

The DGA module also protects the botnet from hijacking. Since the domains can be predicted and anyone can register the future domains of the malware, in theory one should be able to steal all of the bots from the criminals. However, Cabassous has implemented a server verification mechanism. The bot's outgoing messages to the C&C are encrypted with the server's public key. This means that no one will be able to decrypt the hijacked traffic until they obtain the private key from the C&C server. The incoming traffic is encrypted with the simple XOR, which does not matter much.

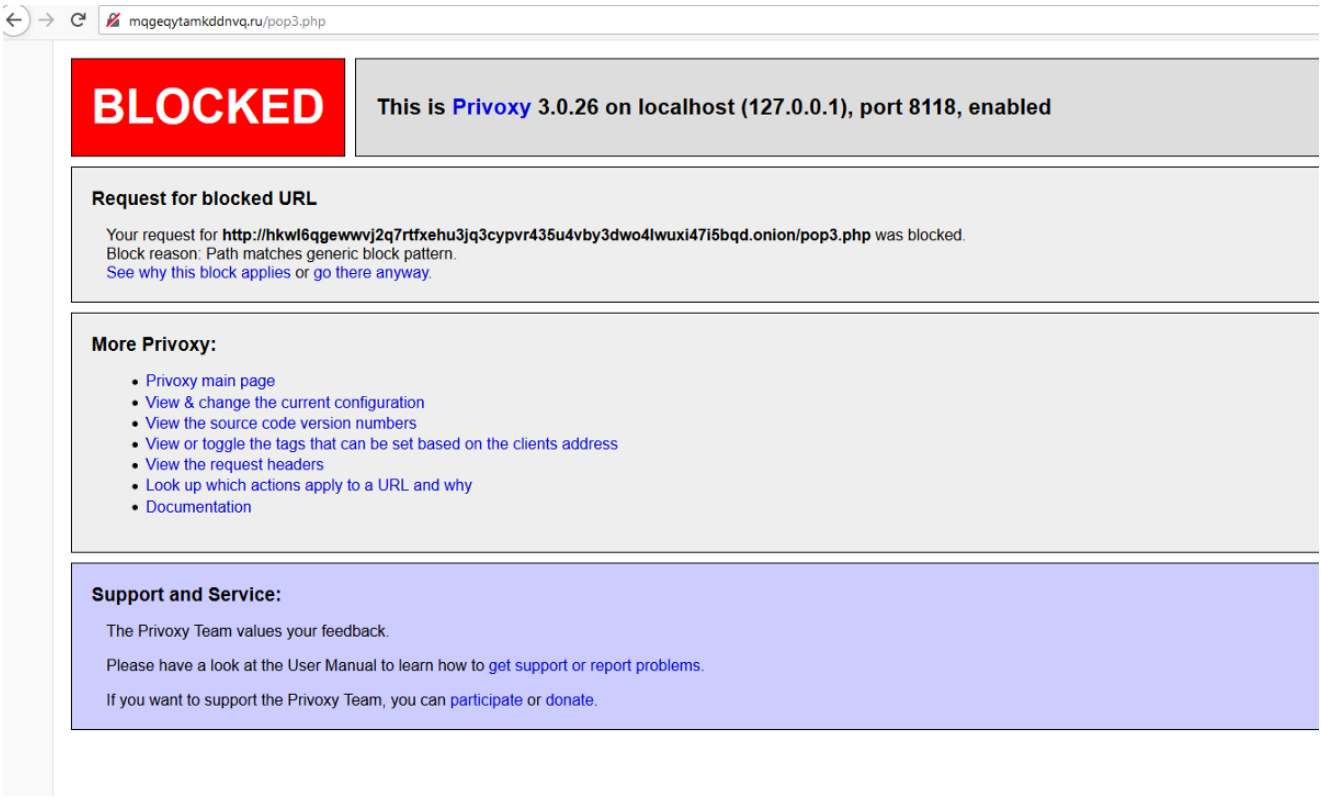
Whenever the bot first connects to a server, it sends it a PKI encrypted bot ID string. It then expects the server to be able to decrypt it and send it back. The DGA module will move on to the next domain if the server does not respond with the same bot ID.

```
private static String EncryptRSA(String str) {
    try {
        PublicKey generatePublic = KeyFactory.getInstance(
            Deobfuscator$app$Release.getString(-38499664158615L))
                .generatePublic(new X509EncodedKeySpec(Base64.decode(
                    Deobfuscator$app$Release.getString(-36811742011287L), 0)));
        Cipher instance = Cipher.getInstance(Deobfuscator$app$Release.getString(-38516844027799L));
        instance.init(1, generatePublic);
        return Base64.encodeToString(instance.doFinal(str.getBytes(StandardCharsets.UTF_8)), 2);
    } catch (Exception unused) {
        return null;
    }
}
```

Outgoing traffic encryption with an embedded public key

The listing of the set of commands, which the server can exchange with the bot are included in the IOC section of this article.

The DGA domains point to a server running an instance of PRIVOXY, which is a software that redirects the requests to the real server, as another measure of takedown resilience. Early on in the campaign we have observed the server to leak the real backend address via the PRIVOXY info page: `hkwl6qgewwvj2q7rtfxehu3jq3cypvr435u4vby3dwo4lwuxi47i5bqd[.]onion`. Very soon the misconfiguration was fixed and the onion address was changed.



Misconfigured PRIVOXY info page

Summary

The threat actors in this case are definitely somewhat savvy and hardly beginners in the business. The authors have started out with targeting just one country, knowing fully well that defrauding even one country takes a lot of effort. There is no need to make too much noise in the other countries, when you know up front that you will not have enough time to fraud their victims. While taking those careful steps, they probably felt that their private malware could become unique, so they stuffed it with a DGA algorithm. Domain generation is very rarely used in Android malware.

The authors release updates to the malware code every few days, which is quite frequent. Despite the many successes, the arrests have still taken place and we may see more of them.

The implementation of the DGA is unique and secure. Throwing PKI cryptography on the bot's outgoing messages and only simple XOR on the incoming messages shows that the criminals know about the hijacking risks and how to address the issue. We speculate that the criminals might rent the botnet out for use by other carders, at which point we will start seeing other countries to be targeted by it.

For now, we recommend being very careful about dealing with your package tracking since this infection vector has proven itself incredibly viable and effective. Whenever you expect a package, we recommend that you copy the tracking number from your email and then navigate to the website of the delivery service manually. This way, you avoid clicking the links, which you receive over SMS or email. Also, your delivery service would always have their real tracking app published on the official application stores — that is Apple AppStore for iOS devices and GooglePlay for Android devices. Getting the app only from the official store makes it far less likely that you will download a fake malicious version instead.

References

- — ThreatFabric's initial report on Cabassous
- — PRODAFT's analysis paper
- — Alberto Segura's report on the version 3.4 after the arrests in Spain
- — Daniel López's collection of spam domains
- — OpenSource string obfuscator "paranoid"
- — powerful runtime debugging framework for Android and other platforms
- — Runtime hooking framework Xposed
- — the free web proxying software which the Cabassous criminals have used to hide their real malware backend
- — Analysis of the Tencent's Legu packer

IOC

Bot Commands

```
PREPINGPINGLOGSMS_RATEGET_SMSGET_INJECTGET_INJECTS_LISTCONTACTSSMSINTERCEPTINGINTERCEP
```

C2 proxy IPs

```
8.209.76.918.209.80.738.208.101.253
```

old C2 backend onion address

```
hkwl6qgewwvj2q7rtfxehu3jq3cypvr435u4vby3dwo4lwuxi47i5bqd[.]onion
```

Spam/APK delivery URLs (cheers to @danlopgom on his)

```
7277320[.]ru/app/acoi[.]my/fedex/ailnoir[.]com/app/aminatech[.]net/fedex/amzstudy[.]cc
cost[.]com/app/colegioaugustoribeiro[.]com[.]br/fedex/contornosdesign[.]pt/pkg/cwfplac
network[.]com/web/ganesha[.]com[.]py/web/geeklevi[.]com/fedex/illuminate[.]org/info/in
dh[.]nl/pack/lavozislamica[.]com/www/lgklgklgk[.]com/fedex/magicboximportados[.]com[.]
daikou[.]com/info/prtysh[.]in/app/raeloficial[.]com/pkg/raisegroup[.]it/fedex/rasf[.]s
-thvitstore-
c7a[.]com/pack/yangbin[.]100cuo[.]com/pack/ylem222[.]com/p/yourelectricians[.]co[.]uk/
```

SHA256

```
d22c5db75f6260823e83057721a2d3e90a9821bdbc81ec52683e8cce49d9a49d80260cc2e49e1014d64f87
```

C2 proxy DGA domains (a short batch of them for the sake of example)

```
afhckrfcucjbpln.comaoeaqxivuikhhdp.cnarymabkciiyygmh.combdoefrixkguoivh.rubfsggebsrhig
```