

ZLAB

Malware Analysis Report

HeroRAT: Analyzing the Telegram based Android malware



Cyber Security Strategists

02/08/2018



Cyber Security Strategists

CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

Table of Contents

Introduction	3
Analysis of the Samples	4
“63a22d065e16ac022910fe1cad6360ecc8539c0b.apk”	4
“3605476181c935413436f0a1cd0e4ecaca72dc7d.apk”	4
“a155e06cb4890e6d4f4802278f5408335395f39c.apk”	4
The malware stealthiness	6
Writing the malware with Xamarin	8
RAT capabilities and features	10
Yara Rules	13



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

Introduction

In June, researchers from security firm ESET discovered a new family of Android Remote Administration Tool (RAT), dubbed HeroRAT, that leverages the Telegram BOT API to communicate with the attacker.

The use of Telegram API can be considered a new trend in Android RAT landscape, because other RAT families implementing the same functionalities, such as [TeleRAT](#) and IRRAT, were discovered in the wild before HeroRAT.

HeroRAT appeared very active in Iran where it was spread through third-party app stores, through tainted social media and messaging apps.

ESET experts speculate that the HeroRAT borrows the source code of a malware that appeared in the hacking community in March 2018, however, HeroRAT has some characteristics that distinguish it from IRRAT and TeleRAT. One of these features is the usage of the Xamarin Framework and TeleSharp Library for the development of the RAT.

HeroRAT is offered for sale on a dedicated Telegram channel, the author offers three different variants depending on its functionalities: bronze (25 USD), silver (50 USD) and gold panels (100 USD). The malware author also released a demo video in which explains the RAT functionalities; below we have a screenshot from this demo video, showing the differences between the three variants.

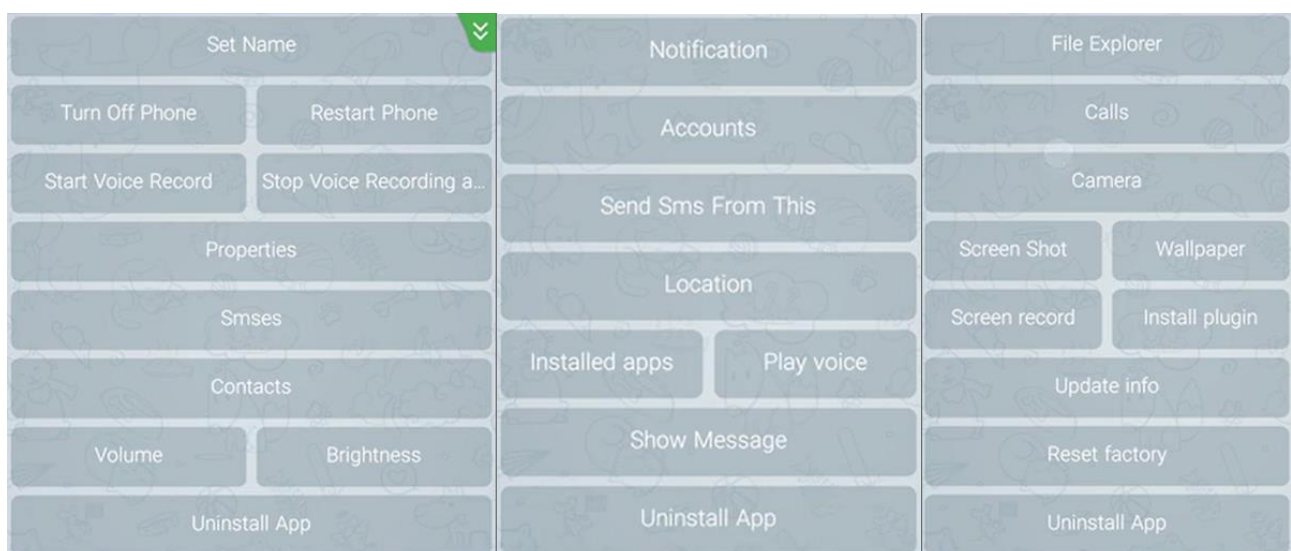


Figure 1 - Differences between the RAT variants




CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com


Analysis of the Samples

In this section we reports all the analyzed sample:


“63a22d065e16ac022910fe1cad6360ecc8539c0b.apk”

MD5	896ffa6cb6d7789662acedc3f9c024a0
SHA-1	63a22d065e16ac022910fe1cad6360ecc8539c0b
SHA-256	92edbf20549bad64202654bc51cc581f706a31bd8d877812b842d96406c835a1
File Size	7,02 MB
Package name	System.OS
Icon	

“3605476181c935413436f0a1cd0e4ecaca72dc7d.apk”

MD5	0e6fdbdf1fb1e758d2352407d4dbf91e
SHA-1	3605476181c935413436f0a1cd0e4ecaca72dc7d
SHA-256	a002fca557e33559db6f1d5133325e372dd5689e44422297406e8337461e1548
File Size	7,01 MB
Package name	FreeInterNet.OS
Icon	

“a155e06cb4890e6d4f4802278f5408335395f39c.apk”

MD5	e16349e8bb8f76dcff973cb71e9ea59e
SHA-1	a155e06cb4890e6d4f4802278f5408335395f39c
SHA-256	3b40b5081c2326f70e44245db9986f7a2f07a04c9956d27b198b6fc0ae51b3a2
File Size	9,49 MB
Package name	Andro.OS
Icon	



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

The malware stealthiness

The three variants refer to the same RAT, but are repackaged using different package name, icon and botmaster's username. For this reason we have analyzed just one samples in depth.

Obviously, the application required all permissions on the target as reported in the following image:

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.CAPTURE_AUDIO_OUTPUT"/>
<uses-permission android:name="android.permission.MANAGE_ACCOUNTS"/>
<uses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.DEVICE_POWER"/>
<uses-permission android:name="android.permission.REBOOT"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.READ_FRAME_BUFFER"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.READ_PROFILE"/>
<uses-permission android:name="android.permission.WRITE_SECURE_SETTINGS"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.DELETE_PACKAGES"/>
<uses-permission android:name="android.permission.INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.READ_SOCIAL_STREAM"/>
```

Figure 2 - RAT permissions

Once installed, the application will show the message “This application can’t run on your device” and displays a fake uninstalling process.



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

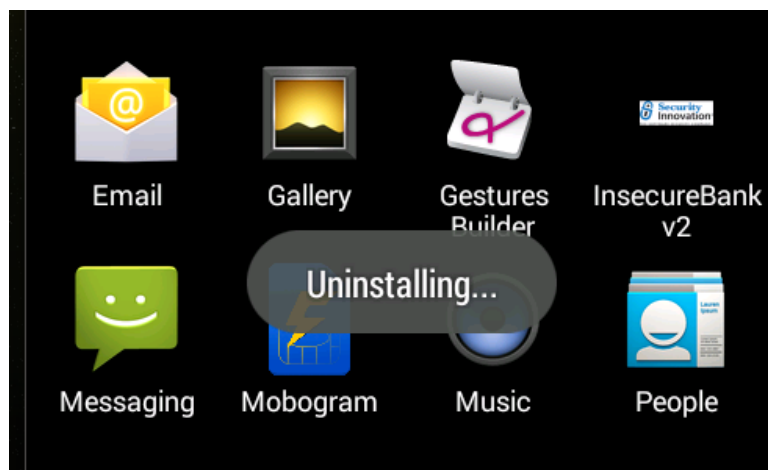


Figure 3 - RAT fake uninstalling message

Then the icon of the app is removed from the home panel even if the RAT is still active in service-mode, as shown below.



Figure 4 - The RAT is active after the fake uninstalling

In order to hide the app's icon and make the RAT stealthy, the attacker uses the method `setComponentEnabledSetting` setting to `Disabled` but specifying the parameter `DontKillApp`, as shown below:



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

```

inf.Values.C = base.Application.BaseContext;
inf.OpenSetting(Application.Context);
PackageManager arg_41_0 = this.PackageManager;
ComponentName componentName = new ComponentName(this, base.Class);
arg_41_0.SetComponentEnabledSetting(componentName, ComponentEnabledState.Disabled, ComponentEnableOption.DontKillApp);
this.MoveTaskToBack(true);
inf.Values.RuningCamera = false;
if (inf.Values.sharep.GetBoolean("ftr", true))
{
    inf.Values.sharep.Edit().PutBoolean("ftr", false).Commit();
    string[] array = new string[]
    {
        "This Application Can't Run On Your Device",
        "Uninstalling...",
        "Uninstall finished"
    };
};
if (!Locale.Default.Language.Equals("en"))
{
    array[0] = "این نرم افزار قادر به اجرا بر دستگاه شما نمیباشد";
    array[1] = "درحال حذف نصب";
    array[2] = "حذف نصب پایان یافت";
}
}

```

Figure 5 - Code used to hide the app's icon

The above image shows the application also checks the device Local and displays its messages in the appropriate language.

Writing the malware with Xamarin

Unpacking the apk, we noticed the presence of a folder, called “assemblies”, containing some DLLs:

Nome	Dimensione	Dimensione co...	Ultima modifica	Creato	Ultimo accesso
TeleSharp.dll	66 048	66 048	2017-10-03 10:56	2017-09-16 20:22	2017-09-16 20:22
System.Xml.Linq.dll	50 688	50 688	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
System.Xml.dll	1 047 552	1 047 552	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
System.ServiceModel.In...	227 840	227 840	2016-03-01 23:01	2017-09-16 20:22	2017-09-16 20:22
System.Runtime.Serializ...	6 144	6 144	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
System.IO.Compression...	9 216	9 216	2017-10-03 10:56	2017-09-16 20:22	2017-09-16 20:22
System.IO.Compression...	56 320	56 320	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
System.dll	729 600	729 600	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
System.Core.dll	288 256	288 256	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
RestSharp.dll	167 936	167 936	2017-10-03 10:56	2017-09-16 20:22	2017-09-16 20:22
mscorlib.dll	1 992 704	1 992 704	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
Mono.Android.dll	899 072	899 072	2017-10-03 10:56	2017-09-16 20:26	2017-10-03 10:56
android.os.dll	121 856	121 856	2017-10-03 10:56	2017-09-16 20:22	2017-09-16 20:22

Figure 6 - DLLs from Xamarin



CSE CyberSec Enterprise SPA
 Via G.B. Martini 6, Rome, Italy 00100, Italia
 Email: info@csecybsec.com
 Website: www.csecybsec.com

These libraries result from the usage of Xamarin Framework that allows the malware writers to develop the Android application using the C# language.

In the above list, we can see the presence of the TeleSharp library, a C# implementation of Telegram Bot API, publicly available on GitHub. Some RAT functionalities are based directly on the TeleSharp APIs (such as “Send Location”), so the writing of the malware was particularly simple for the author.

<https://github.com/MojtabaTajik/TeleSharp>

TeleSharp

TeleSharp is C# implementation of Telegram Bot API

It support most of the features of Telegram Bot API including :

- Get Bot information
- Read received messages
- Send text message
- Forward message
- Replay message
- Send location
- Send & receive sticker
- Send & receive photo
- Send & receive video
- Send & receive audio
- Send & receive document
- Set current status of bot like(Typing, Upload photo, Upload video and etc)
- Support Inline Bots
- Support ReplyKeyboardMarkup (Custom keyboards)

It's very light and simple to use, enjoy it ;)

Figure 7 - The TeleSharp library



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

Due to the usage of Xamarin Framework, all the Java source code files, extracted using specific tool, contain only the wrappers of the real code which is contained into “android.os.dll” file. It was simple to decompile the dll to access the actual source code by using some tools specifically designed for the analysis of .NET executable.

RAT capabilities and features

Exploring the source code, we found all the capabilities of a classic RAT such as read SMSs and calls log, get location, turn off or on, download other files, etc:

```
AreYouSure(string, string, string, string) : SendMessageParams @060000A7
callnotify(string, string) : SendMessageParams @060000A8
Calls(string, string) : SendMessageParams @060000AC
Camera(string, string) : SendMessageParams @06000097
contacts(string, string) : SendMessageParams @06000098
downloadcancel(string, string) : SendMessageParams @06000095
FileDialog(string, string) : SendMessageParams @060000A1
filesandfolders(string, string) : SendMessageParams @06000094
FolderDialog(string, string) : SendMessageParams @060000AE
inl(string) : InlineKeyboardMarkup @060000A9
inlbtns(string, string) : List<List<InlineKeyboardButton>> @06000096
inlbtnsfolderdialog(string, string) : List<List<InlineKeyboardButton>> @06000097
List(string) : InlineKeyboardMarkup @06000095
List(string, string) : SendMessageParams @060000AA
listforme(string) : List<InlineKeyboardButton> @06000089
listforme2(string) : List<InlineKeyboardButton> @0600008A
listforme3(string) : List<InlineKeyboardButton> @0600008B
listforme4(string) : List<InlineKeyboardButton> @0600008C
listforme5(string) : List<InlineKeyboardButton> @0600008D
listforme6(string) : List<InlineKeyboardButton> @0600008E
Location(string, string) : SendMessageParams @06000091
lockme(string, string) : SendMessageParams @06000096
Locsett(string, string) : SendMessageParams @06000090
modify(string, string, string, UserInformation) : SendMessageParams @06000092
NewCall(CallNotify) : SendMessageParams @0600009B
NewLocation(LocationNotify) : SendMessageParams @0600009C
NewSms(SmsNotify) : SendMessageParams @0600009A
notify(string, string) : SendMessageParams @060000A6
OkCancel(string, string, string) : InlineKeyboardMarkup @06000094
Private(string, string) : SendMessageParams @06000092
prop(string, string) : SendMessageParams @06000093
Sendwitfile(string, string, string) : SendMessageParams @06000094
setpath(string, string) : SendMessageParams @0600009C
showORwrite(string, string) : SendMessageParams @060000A2
Smses(string, string) : SendMessageParams @060000AD
smsnotify(string, string) : SendMessageParams @060000A7
turnOFForON(string, string, string, string, string) : InlineKeyboardMarkup @06000094
usecamera(string, string, string) : SendMessageParams @06000094
```

Figure 8 - RAT capabilities

The main token of the Telegram bot, used as Command and Control (C2C) of the RAT, appears in a source code of a class:



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

```
// TOKEN: 6X04000004 RID: 100
public static string MainToken = "471395041:AAEIXJaagcNgYUZ1I4-8o89KLDJ7BNDpyLs";
// ...
```

Figure 9 - Main token of the Telegram Bot

So, we are able to query the Telegram REST API in order to obtain the BOT's username and try to send commands to it:

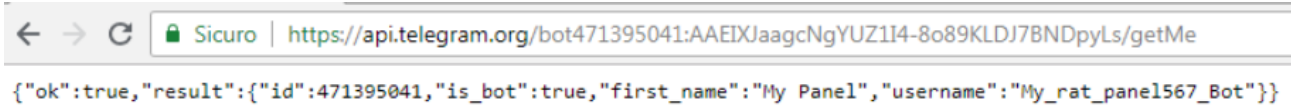


Figure 10 - Bot information

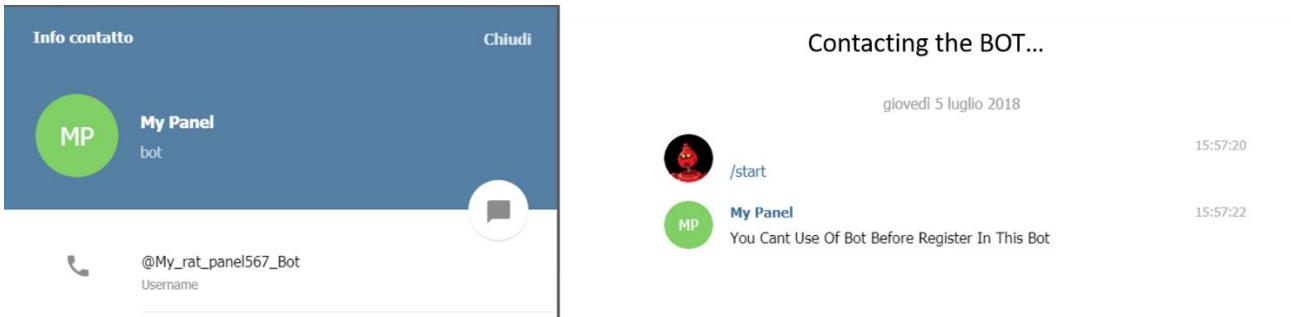


Figure 11 - Contacting the bot

Unfortunately, the BOT doesn't accept our commands, but replies with the message "You Cant Use Of Bot Before Register In This Bot". This message suggests the bot only accepts commands from a specific source (i.e. a user associated with a specific username).

The source code confirmed this thesis, it include the comparation between the ID of the sender and the ID of the "Manager". The "Manager" ID and the BOT's MainToken are the only differences between the three samples we have analyzed.



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com

```

}
if (M.From.Id.ToString() == inf.Manager.ChatId)
{
    if (inf.Manager.ChatId != M.From.Id.ToString())
    {
        inf.Manager.ChatId = M.From.Id.ToString();
        inf.Manager.SaveSetting();
    }
    inf.Manager.InitMessage(M, MC);
}
else
{
    UserInformation userInformation = UserInformation.FindUser(M.From.Id, UsersMode.Users);
    if (userInformation != null)
    {
        userInformation.InitMessage(M, MC);
    }
    else
    {
        inf.Values.api.SendMessage(new SendMessageParams
        {
            ChatId = M.From.Id.ToString(),
            Text = "You Cant Use Of Bot Before Register In This Bot"
        });
    }
}
}
}

```

Figure 12 - Sender ID check

The image below shows the “Manager” username found in the analyzed sample:

```

public static Creator Manager = new Creator("mrpishgam");

```

Figure 13 - Manager username

Using the username, we found the account of the “manager” (or botmaster) on Telegram:

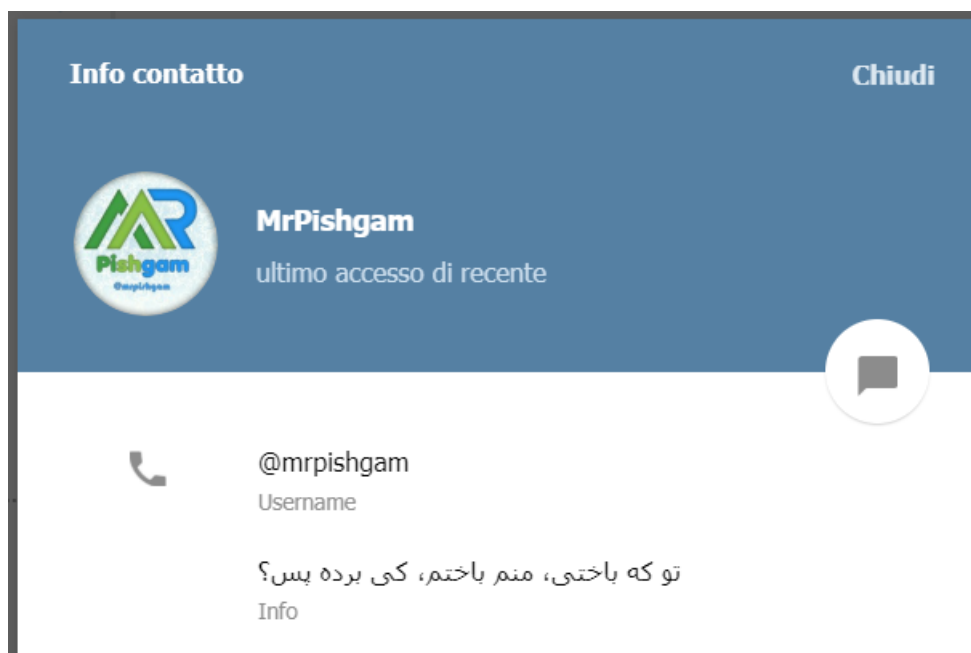


Figure 14 - Manager account



CSE CyberSec Enterprise SPA
 Via G.B. Martini 6, Rome, Italy 00100, Italia
 Email: info@csecybsec.com
 Website: www.csecybsec.com

Yara Rules

```
rule HeroRAT {
  meta:
    description = "Yara Rule to individuate some samples of
HeroRAT Android malware"
    author = "CSE CybSec Enterprise - ZLab"
    last_updated = "2018-07-31"
    tlp = "white"
    category = "informational"
  strings:
    $a = "assemblies/TeleSharp.dll"
    $b = "assemblies/Mono.Android.dll"
    $c = {49 64 00 67 65 74 5F 4D 79 4D 61 6E 61 67 65 72}
    $d = {52 65 70 6C 79 4D 65 73 73 49 64 00 73 65 74 5F 43
68 61 74 49 64}
  condition:
    $a and $b and ($c or $d)
}
```



CSE CyberSec Enterprise SPA
Via G.B. Martini 6, Rome, Italy 00100, Italia
Email: info@csecybsec.com
Website: www.csecybsec.com