CRYPTER HANDBOOK

2017 – written by Carrotinblack

INTRODUCTION

Welcome to the crypter handbook, written by me, Carrotinblack, a 17 year old french student, and also big fan of all what is encryption and computer security. I created some crypters like the "carrotcrypter" you can get on HF or on http://carrotnet.cf

In this e-book you will learn how encryption and crypters are working, all in a easy-understanding way. I think you don't know how a crypter is working, so we are going to learn it from the beginning. I hope you will enjoy this e-book, and learn some things about encryption. So, let's begin...

A little introduction about crypters

-What are crypters?

Crypters are programms created in order to "obfuscate" (hide) the source code from a file. They can encrypt the binary code of a file with an encryption algorithm like DES, AES ... You can learn more about that on wikipedia.

-What sort of crypters does exists?

There are two big differences between crypters: The standart crypters who only encrypts the source of a file and an obfuscator, the crypter we will learn to make, who encrypts the file and adds some code to decrypt the file when openned. In this family there are two more families :

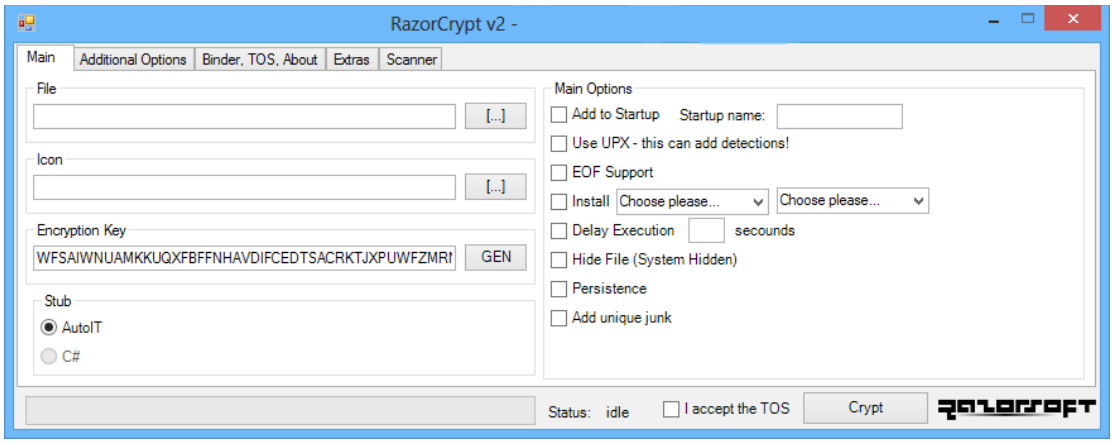
The scantime crypter: This crypter encrypts a file and adds a part to decrypt it on the hard disk when openned. It is named scantime because of the capacity to bypass AntiViruses when they are scanned.

The runtime crypter: The runtime crypter is like the scantime crypter but can also bypass AntiViruses when the programm is ran. It decrypts the programm in memory (RAM) and encrypts it again when closed.

-Some examples:

Here are some examples of different crypters:

The carrotCrypter, created by me



RazorCrypt

I think I don't teached you much when I said that, but now we will learn some more things about this crypters, and over all: the theroy to make your own!

HOW DOES A CRYPTER WORK

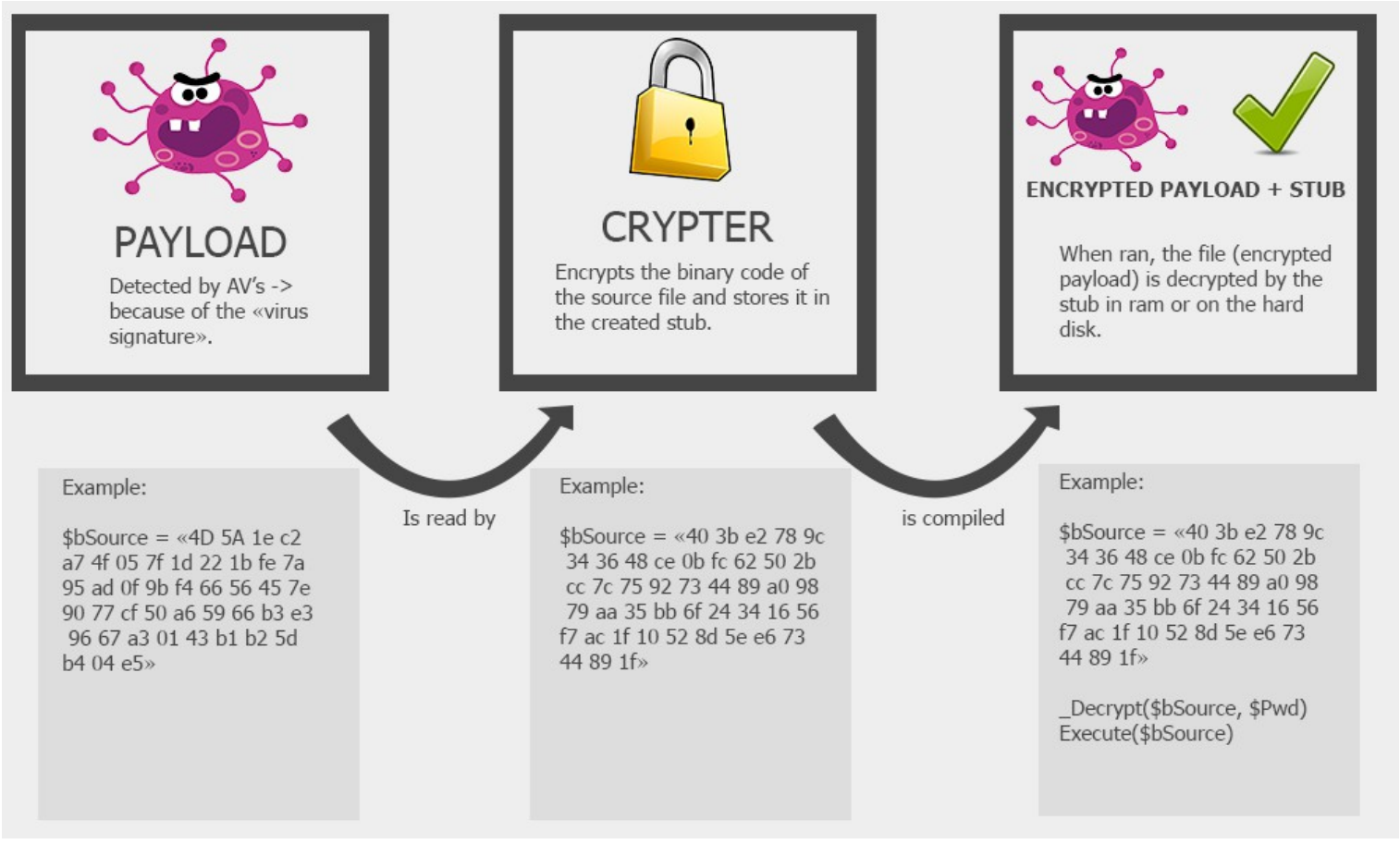
Every crypter works different, we are learning how a very-basic crypter works:
First of all we need some vocabulary:

Crypter GUI: This is the interface of the crypter, with all options etc...

Stub: This is the part that is generated by the crypter and that is for decrypting the payload when ran. It's a part of the final encrypted file, but isn't encrypted itself.

Virus signature: A part of the binary code of a file that is detected by an Antivirus.

So, now we can see the basis of a crypter with this explicative drawing of me :



To resume: the payload is encrypted by the crypter and compiled together with the stub, which will decrypt the encrypted source and run it (runtime) or store it (scantime).

Ok, now that you know the theory working of a crypter you can begin to manage your working space. It's the new chapter of this book.

CREATE YOUR OWN CRYPTER

In order to create your own crypter you have to know some things: Crypters first functionality is to encrypt files to bypass Av's so if your Stub is detected by av's as a threat, your crypter won't be a "crypter" anymore because it will be detected.

In order to prevent this, you must know what follows:

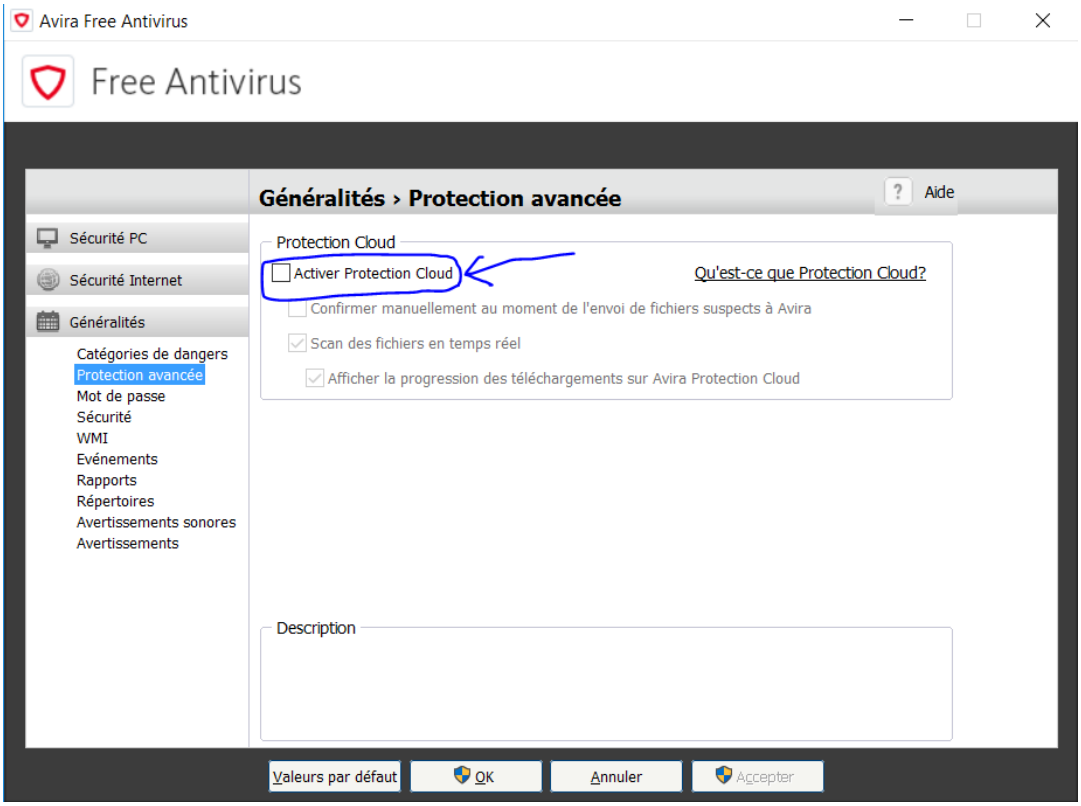
- You will have to try your crypter often on MultiAV scan pages to see if it's undetectable, the problem is that some of these scan websites distribute your file to the AV's companies.
So don't try your encrypted files on VirusTotal or anything you found on google after a simple research, use:

-http://Pscan.xyz

-http://NoDistribute.org

-Your own AV can distribute copies of your stubs, because it is on your computer, if you don't know that it would be useless to create a crypter.

You will need to disable this function on your antivirus, I let you search on google :) (here is for avira)



In our days, scantime is useless, because every AV can stop your payload when ran, so I will teach you to create your own Runtime crypter.

The main function of a runtime stub is the RunPE function. You will need this function for every runtime crypter you create unless someone (or you) found another way to bypass AV's in RAM.

RunPE is a function that creates a process in memory for your programm or just injects your programm into an existing process. In our example above, the RunPE function is the "Execute" function, who reads the binary and executes it in RAM.

With these information, you will be able to create your first crypter, I would just recommend you some languages to create your crypter:

-AutoIT : Is very easy to learn and quite good for crypters because you got many tutorials, sources and some good existing functions.

-vb.net : every existing crypter is written in vb.net (Not "every" but at least most of them), its quite easy to learn.

-ASM : If you know ASM you can do everything :)

-C++/C/C# : Good too because of the disponibility

I wont give you some sources, you should just search by your own if you really want, but I think that you should think a little to learn. And you know, the best sensation on the whole world is to finish your first working crypter, even if it's totally detected, because once you have this basis, you can add what ever you want to make your crypter FUD, what brings us to the next part :

MAKING YOUR CRYPTER FUD

FUD: Stands for "Fully undetectable". That means that NO AV (0) can detect your encrypted files and stub.

Once you finished your first basis crypter, it's UD : undetectable but not FUD.

In order to make your crypter FUD you have some options to add to your crypter :

- -Junk code
- -Informations and Icon changer
- -File bumper
- -Random stub
- -Private stub
- -etc...

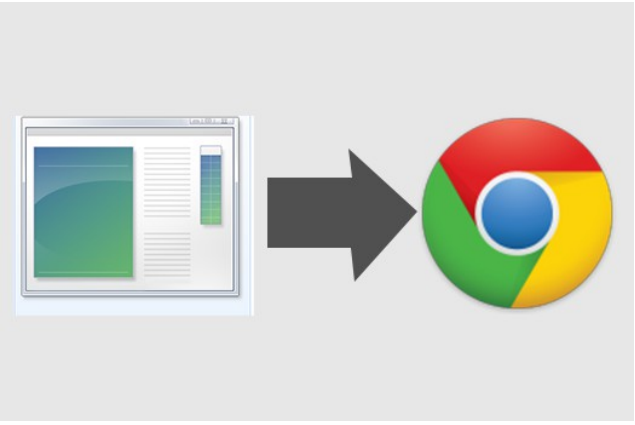
1)Junk code:

Junk code is like the name is saying "non-sense code" to add to your stub. Often, you will have to create a script that creates random strings, functions, variables, calls etc... This code won't change the working of your programm but will add some code more to analyse for AV's (normal code, who isn't detected by AV's and seems to be legit)
Here is an example:

```
1   $Sm8mPg9O9R = "mmqmeB9oj6"
2   If "5OueIXdq6izBm" == "i8ZoEs9e4z" Then
3      $5l7C0Ek4Jz81i6 = "5lt208C43WD1i6"
4   EndIf
5   For $WV194D676G987 = 1 To 12.5517175956629
6      $sLviAJYASV = "0nvVAhYASh"
7   Next
8   Func _h96qFh3($4B9qWm7899G, $OfiaMJR3nNm56q)
9   For $DYApaB7K17KsX = 1 To 9.4455069093965
10     $j2Bw1Ie6xsSz95 = "jEBXO6YLxsvN9N"
11  Next
12  For $c4xmApsmkoM5N1 = 1 To 8.00440416298807
13     $v0gAty511cknE = "vd6yVy5Jg7knE"
14  Next
15  For $88I3xrkmD2x = 1 To 13.8329028906301
16     $8kmhEaZ3VI45 = "Ok4hENbzieD5"
17  Next
18
19  EndFunc
```

You can see that there are variables (l.1), functions etc... with random names. They are nonsense.

2)Information and Icon changer



This can help you bypass some AV's more : You can create random file versions, copyrights, descriptions etc...
The icon changer is important, because some AV's detect suspicious icons, they should be in high resolution.

3)File bumper

A file bumper bumps your file to a number of Kb. Sometimes it helps bypass AV's.

4)Random Stub

Often, the RunPE module is overused because the crypter developpers copy/paste the sources. This makes the module detected by AV's.
This let's you 2 choices :

- You create your own RunPE (recommended)
- You change the RunPE module you have

Once you have done that, it's possible that some AV's still detects your RunPE. To avoid that you need to randomize them : by creating random variable names, function names etc...
Change everything you can change with a function that randomizes your RunPE module.

Example => "_RunBinary()" to "_Rgd55YHjst5rfsqM()"

5)Private stub

This is the best alternative : Never create the same stubs twice, up to you to find how (there are many options).

This helps to avoid scantime detection, but runtime detction is another thing. The crypter would not be able to assure runtime detection if the file is a rat with every possible settings.

Ok, now that you have your crypter, maybe FUD or not, you should add some functionalities to it.

CRYPTER FUNCTIONALITIES

Now I will talk about some functionalities you should build into your crypter in order to make them FUD, or to make the encrypted file better.

USG : Is the part of the crypter that generates a random and private stub.

Delay : You can add a delay before the execution of your script. That helps to bypass Runtime detections from some AV's and there buit-in sandboxes.

Anti memory scan: Another way to bypass runtime detection is Anti memory scan, wich denies the acess to the process your file is running in.

EOF : To avoid file-corruption, you'll need sometimes to preserve the “End of file”. So add it !

Startup : This function adds your file to the programms windows starts at the start of the pc.

LeftToRight : A simple way to create another extension for you .exe file.

Binder : Very usefull function, that allows you to bind your encrypted file with another, like a jpg.

Custom msgBox : I let guess you what it is. It's a simple msgBox at the execution of your file.

Antis : Anti sandboxie, wireshark etc...

Once your crypter is FUD you won !

If your crypter isn't FUD anymore, don't worry it doesn't take long to reFUD your crypter, just change some things and try things! It will work.

To avoid detection of your Crypter :

-Don't scan it on distributed scanners

-Don't put it on skype, mediafire, google drive, or dropbox.

This is the end of this e-bbok ! I hope you enjoyed learning and creating your own crypter!
If you have any questions or other requests, just ask me on HackForums.net, I'm CleanCarrot.

END

TOS: This e-Book is for learning purpose only, don't do illegal things.
I'm not in responsible in anyway for what you are doing with this e-book.
The redistribution or copying of this e-book is prohibited.
No refunds.