

Forum:[Hack Forums](#)

Thread: [\[Development Thread\] PsiProtector - Polymorphic C# Crypter](#)

PsiSolutions - May 25, 2020, 5:12 pm

Hello everyone,

I have decided to start a thread following the development of my new, coded from scratch polymorphic crypter in C#. I will be detailing every step of my process of development of the crypter, until it is eventually fully functioning and released for sale (or abandoned lol). I am doing this to get people interested in this project, as well as to maybe help somebody out in the process of software development. I am no expert by any means, but I have been coding for 10 years and I feel like this is a project I can handle. If you have any questions or comments feel free to post them! Thank you

Images:

<https://imgur.com/a/zTx9ESk>

Planned Features:Unique stub generation

Polymorphic method utilization

Anti-virtualizationAnti-Sandbox

Anti-VMWare

Anti-VirtualPC

ExecutionCustom delay

Don't run until restart

Delete original file (file melt)

RunPE/LoadPE

Native/.NET Support

File PropertiesStartup

Spoof Extension

Clone Icon/Assembly Info

Change file size

Bind to another file

Process persistence

File persistence

Startup persistence

Silent UAC elevation

Bypass 1909 WD (hopefully) (and other AVs)

5 Algorithms PlannedAES

XOR

Blowfish

Triple DES

RSA

Current progress(5/25):

Quote:Completed UI design and controls

Added functionality to all UI elements (file load, assembly information grabber, expected file size calculation)

Implemented custom licensing system

Setup C# CodeDom compiler

Implemented basic USG system (can generate unique .exes with specified code)

More to come! I plan to be developing daily, but I will be starting work back again so it may go slow for a while. Hope you enjoy and let me know what you think!

axel87 - May 26, 2020, 12:19 am

a great job, congratulations

VisualSoft - May 26, 2020, 12:43 am

Good Job on this, Just make sure you bypass all AV on runtime as well as scantime.

Jean-Jacques Rousseau - May 26, 2020, 8:57 am

Polymorphic executables don't work in the malware game. Any unknown executable, even not malicious, will be heur/ml detected. Even if you're not detected, the behavioural engine is tuned against your unknown file.

PsiSolutions - May 26, 2020, 7:08 pm

(May 26, 2020 08:57 AM)Supercut Wrote: [\[CITE\]](#) Polymorphic executables don't work in the malware game. Any unknown executable, even not malicious, will be heur/ml detected. Even if you're not detected, the behavioural engine is tuned against your unknown file.

Interesting, so all programs that make use of polymorphism get flagged as malicious? Or is it just easier to detect a program that is polymorphic in nature? I will try and see what I can do but I may scrap the polymorphism if it gives me problems.

Changelog 5/26

Quote:Implemented delay

Implemented extension spoofer

Implemented file assembly info generation

Implemented icon changer

VisualSoft - May 26, 2020, 7:13 pm

(May 26, 2020 08:57 AM)Supercut Wrote: [\[CITE\]](#) Polymorphic executables don't work in the malware game. Any unknown executable, even not malicious, will be heur/ml detected. Even if you're not detected, the behavioural engine is tuned against your unknown file.

dude let the guy try out, eventually he will know.

Jean-Jacques Rousseau - May 26, 2020, 11:14 pm

(May 26, 2020 07:08 PM)PsiSolutions Wrote: [\[CITE\]](#) Interesting, so all programs that make use of polymorphism get flagged as malicious? Or is it just easier to detect a program that is polymorphic in nature? I will try and see what I can do but I may scrap the polymorphism if it gives me problems.

Changelog 5/26

I used polymorphic LLVM passes before, they worked fine as long as you do the regular anti-heur/ml things on your exe. Using that I was capable of producing 96-98% unique stubs (the pass works on each basic block, inserts and replaces assembly instruction, please read the LLVM documentation if you want to know more).

The problem is that:

Heur/ml is NOT trained to detect malware

It is trained to detect goodware, and flags anything else.

It might work for you, however, I was working with polymorphism for some time, now I have changed method completely where the crypter is not even an exe file. Unknown executables are simply fucked by all engines.

All of this might be false for .NET though, but I would not code a crypter in .NET to begin with.

(May 26, 2020 07:13 PM)VisualSoft Wrote: [\[CITE\]](#) dude let the guy try out, eventually he will know. I am just trying to share my experience, I would love to see true polymorphic binaries in .NET.

SIM-3149568 - May 27, 2020, 12:01 am

Looks very interesting but dude RSA is not used for data encryption in common case. RSA is used for private encryption tunnel like VPN, telegram private chats, etc. And silent UAC elevation.. I want to see this if this means uac bypass

VisualSoft - May 27, 2020, 1:59 am

(May 27, 2020 12:01 AM)tree8069 Wrote: [\[CITE\]](#) Looks very interesting but dude RSA is not used for data encryption in common case. RSA is used for private encryption tunnel like VPN, telegram private chats, etc. And silent UAC elevation.. I want to see this if this means uac bypass you can find many uac bypass around, he is going to use one of those methods.

SIM-3149568 - May 27, 2020, 2:16 am

(May 27, 2020 01:59 AM)VisualSoft Wrote: [\[CITE\]](#) you can find many uac bypass around, he is going to use one of those methods.

Is there a method even working on latest windows like 1909 build? I've knew about uac bypass exploit using event viewer. But it is not working at all.

VisualSoft - May 27, 2020, 3:52 am

(May 27, 2020 02:16 AM)SF Tree Wrote: [\[CITE\]](#) Is there a method even working on latest windows like 1909 build? I've knew about uac bypass exploit using event viewer. But it is not working at all. yes, there are some methods still working.

event viewer is old method and also its runtime detected.

PsiSolutions - May 27, 2020, 8:46 am

(May 27, 2020 02:16 AM)SF Tree Wrote: [\[CITE\]](#) Is there a method even working on latest windows like 1909 build? I've knew about uac bypass exploit using event viewer. But it is not working at all.

There are many exploits and methods for obtaining admin privileges and startup in 1909 publicly available, and I plan to utilize them based on operating system vulnerability. Whether or not these methods add detections is a different story. Being in the very early stages of development, a lot is still unknown to me and in the planning phase. Mostly I started this project to keep me busy in my spare time since I love to code and felt it would be a nice challenge, so in the end if it doesn't work out completely (considering the entire dynamics of the crypting world have changed since I was last active in malware) I won't be too dissatisfied

danny34 - May 27, 2020, 11:38 am

Good luck with your projects once you done polymorphic you should try metamorphic.

PsiSolutions - May 27, 2020, 12:58 pm

(May 27, 2020 11:38 AM)danny34 Wrote: [\[CITE\]](#) Good luck with your projects once you done polymorphic you should try metamorphic.

No joke I was thinking of this this morning. Would be a fun thing to try and implement, I just think it will take a while to code

danny34 - May 27, 2020, 3:57 pm

For asm first thing is to recalculate integers and defeat static signature ex. push dword, mov reg32,dword etc.

With working metamorphic encoder will be

antiscan: 0/26

Vir*sT**al : 0/72

Then add anti emulation for client side and cloud.

Real world example:

Metasploit x86 payload:

```
add edi,[ebp-08h]
```

```
sub edi,05709C146h restore real hash dword
```

```
cmp edi,[ebp+024h]
```

```
jnz AErh
```

```
push blah
```

```
push blah
```

```
push blah
```

```
push 0B01CB606h new hash (dword old + random int)
```

```
call ebp
```

VisualSoft - May 27, 2020, 5:45 pm

(May 27, 2020 08:46 AM)PsiSolutions Wrote: [\[CITE\]](#) There are many exploits and methods for obtaining admin privileges and startup in 1909 publicly available, and I plan to utilize them based on operating system vulnerability. Whether or not these methods add detections is a different story. Being in the very early stages of development, a lot is still unknown to me and in the planning phase. Mostly I started this project to keep me busy in my spare time since I love to code and felt it would be a nice challenge, so in the end if it doesn't work out completely (considering the entire dynamics of the crypting world have changed since I was last active in malware) I won't be too dissapointed

if you dont mind, why dont you link some of those public available bypass over here which works with 1909, many users will find it useful.

PsiSolutions - May 27, 2020, 6:17 pm

(May 27, 2020 05:45 PM)VisualSoft Wrote: [\[CITE\]](#) if you dont mind, why dont you link some of those public available bypass over here which works with 1909, many users will find it useful.

<https://github.com/rootm0s/WinPwnage>

Here is a good summary of currently working exploits in 1909. Also, wikileaks has some good information on exploits on the newest windows used by the CIA and NSA.

VisualSoft - May 27, 2020, 6:20 pm

(May 27, 2020 06:17 PM)PsiSolutions Wrote: [\[CITE\]](#) <https://github.com/rootm0s/WinPwnage>

Here is a good summary of currently working exploits in 1909. Also, wikileaks has some good information on exploits on the newest windows used by the CIA and NSA.

Nice share Buddy, Its definitely better then UACME

PsiSolutions - May 27, 2020, 8:08 pm

5/27

Quote:Implemented AES encryption of executable file
Implemented custom resource storing method

Testing if decryption works with current method tomorrow. Might implement a RunPE as well

peligro - May 28, 2020, 5:34 pm

a great job, congratulations

BruteForce ? - May 29, 2020, 12:13 am

I am liking this one, let us know how it goes.

Ampere - May 29, 2020, 2:03 am

Looks like a nice project. Any chance of implementing Code Signing on the stub to bypass SmartScreen also? I know that might be a pain to do but could be very useful down the road.

PsiSolutions - May 31, 2020, 6:57 pm

(May 29, 2020 02:03 AM)Ampere Wrote: [\[CITE\]](#) Looks like a nice project. Any chance of implementing Code Signing on the stub to bypass SmartScreen also? I know that might be a pain to do but could be very useful down the road.

I plan to, seems to be a little difficult and I will have to do some more reading on it for sure.

5/31

Quote:Implemented decryption mechanism in stub
Increased stub uniqueness

Next plan is to add a junk code generator in which I can parse my code through and add unique junk code everywhere. Currently the stub is 3/26 because of resource storage, I am going to try a couple of different approaches to get rid of these detections and am hoping the junk code will make the files more unique.

danny34 - June 1, 2020, 4:13 am

(May 31, 2020 06:57 PM)PsiSolutions Wrote: [\[CITE\]](#) I plan to, seems to be a little difficult and I will have to do some more reading on it for sure.

5/31

Next plan is to add a junk code generator in which I can parse my code through and add unique junk code everywhere. Currently the stub is 3/26 because of resource storage, I am going to try a couple of different approaches to get rid of these detections and am hoping the junk code will make the files more unique.

Self signed cert will reduce detection, its simple really. You could write alot to do code signing but why bother.

ShellExecuteWait should pevent any race conditions. Any proper cert is better, but will be blocked. FileVersion Info and digital signature can be randomly generated, but should match.

Autolt:

```
Local $cname = "payload.exe"
```

```
ShellExecuteWait(StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\openssl\openssl.exe",  
"req -x509 -newkey rsa:4096 -config  
&StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\openssl\openssl.cnf -keyout  
&StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\openssl\output\key.pem -out  
&StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\openssl\output\cert.pem -days 3285  
-passout pass:12341234", "", "", @SW_HIDE)
```

```
ShellExecuteWait(StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\osslsigncode-fork\osslsig  
ncode.exe", "-certs  
&StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\openssl\output\cert.pem -pass 12341234  
-key &StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\openssl\output\key.pem -n Microsoft  
-i https://www.Microsoft.com -t http://timestamp.comodoca.com/authenticode -in  
&StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\sign\"&$cname&" -out  
&StringTrimRight(@AutoltExe,12)&"disasm\osslsigncode\sign\"&StringTrimRight($cname,4)&"_sig  
ned.exe", "", "", @SW_HIDE)
```

PsiSolutions - June 9, 2020, 12:20 pm

Lots and lots of coding done recently. See changelog below.

6/9

Quote:Implemented RunPE

Implemented new resource storage mechanism, hiding API calls

Implemented unique double encryption mechanism/RunPE to stub

Currently 3 independent, unique keys are used for encryption/decryption routine

Next on list is junk code generation, am planning on writing a generation class and adding places in stub for junk code to be generated. Current scan results (1/26): <https://imgur.com/a/opzJmjN>

VisualSoft - June 9, 2020, 12:50 pm

(Jun 9, 2020 12:20 PM)PsiSolutions Wrote: [\[CITE\]](#) Lots and lots of coding done recently. See changelog below.

6/9

Next on list is junk code generation, am planning on writing a generation class and adding places in stub for junk code to be generated. Current scan results (1/26): <https://imgur.com/a/opzJmjN>
why dont you post scan link in place of scan image?
Also the detection you getting is really easy to bypass.

PsiSolutions - June 9, 2020, 1:10 pm

(Jun 9, 2020 12:50 PM)VisualSoft Wrote: [\[CITE\]](#) why dont you post scan link in place of scan image?
Also the detection you getting is really easy to bypass.

I believe once I add junk methods and some anti-scantime logic it will be FUD, but I am happy with it since this is with RunPE included.

<https://antiscan.me/scan/new/result?id=CiWFopjmMAhg>

Seems there are two more detections now, though this is a different stub... either its due to lack of logic other than load and execute or the site is distributing. Anyone know if this site is legit, or any legit sites?

VisualSoft - June 9, 2020, 1:13 pm

(Jun 9, 2020 01:10 PM)PsiSolutions Wrote: [\[CITE\]](#) I believe once I add junk methods and some anti-scantime logic it will be FUD, but I am happy with it since this is with RunPE included.

<https://antiscan.me/scan/new/result?id=CiWFopjmMAhg>

Seems there are two more detections now, though this is a different stub... either its due to lack of logic other than load and execute or the site is distributing. Anyone know if this site is legit, or any legit sites?

Avira uses cloud scan and the detection looks to be coming from runpe.

PsiSolutions - June 9, 2020, 1:16 pm

(Jun 9, 2020 01:13 PM)VisualSoft Wrote: [\[CITE\]](#) Avira uses cloud scan and the detection looks to be coming from runpe.

The RunPE and API calls are all encrypted, without encryption stub is ~12/26. I will mess around with the code and try to get it FUD in its current state before progressing

GammaDune - June 9, 2020, 3:21 pm

This project looks awesome! If you need a beta tester or someone to bounce some ideas off of I have a few years of .NET experience and could definitely assist. Shoot me a PM!

PsiSolutions - June 12, 2020, 1:36 pm

6/12

Quote:Added junk code generator

Added additional lightweight loader and new silent EXE dropping mechanism

Loading wrapper is only 40 lines of code, does nothing malicious, and has no resources

<https://antiscan.me/scan/new/result?id=qpawYCoDWR14>

Still getting picked up by Avira and F-Secure for dropper, not quite sure why but maybe it is due to the size of the data in the file. Junk code does nothing to change that detection, any help would be appreciated

SIM-3149568 - June 15, 2020, 4:48 am

(Jun 12, 2020 01:36 PM)PsiSolutions Wrote: [\[CITE\]](#) 6/12

<https://antiscan.me/scan/new/result?id=qpawYCoDWR14>

Still getting picked up by Avira and F-Secure for dropper, not quite sure why but maybe it is due to the size of the data in the file. Junk code does nothing to change that detection, any help would be appreciated

Maybe due to an storage method. Change storage method or make lower storage entropy.

Takehito Morokuzu - June 15, 2020, 5:10 am

(May 26, 2020 08:57 AM)Supercut Wrote: [\[CITE\]](#) Polymorphic executables don't work in the malware game. Any unknown executable, even not malicious, will be heur/ml detected. Even if you're not detected, the behavioural engine is tuned against your unknown file.

Not sure where you're getting this from. I've never had a piece of software be detected by an antivirus just because it's unknown.

Jean-Jacques Rousseau - June 15, 2020, 6:42 am

(Jun 15, 2020 05:10 AM)Takehito Morokuzu Wrote: [\[CITE\]](#) Not sure where you're getting this from. I've never had a piece of software be detected by an antivirus just because it's unknown.

My own tests. An unknown executable is more easily detected by reputation or cloud engines and their machine learning (WD Wacatac, Avira, Sophos for example), whereas recognized stubs are less easily detected, for example, an MSI installer.

Takehito Morokuzu - June 15, 2020, 9:49 am

(Jun 15, 2020 06:42 AM)Supercut Wrote: [\[CITE\]](#) My own tests. An unknown executable is more easily detected by reputation or cloud engines and their machine learning (WD Wacatac, Avira, Sophos for example), whereas recognized stubs are less easily detected, for example, an MSI installer.

Well I mean I've made plenty of software that hasn't been seen by any av and it's only ever detected by securage apex

Jean-Jacques Rousseau - June 15, 2020, 11:02 am

(Jun 15, 2020 09:49 AM)Takehito Morokuzu Wrote: [\[CITE\]](#) Well I mean I've made plenty of software that hasn't been seen by any av and it's only ever detected by securage apex

Of course, my current cryptor doesn't use recognized stubs. Just be prepared that behaviour engine will be tuned against exes without a recognized stub.

mshar1 - June 17, 2020, 4:45 am

a great job, congratulations