

Forum:[Hack Forums](#)

Thread: [ReFUD Crypter Source's RunPE](#)

Venture - August 4, 2019, 12:56 pm

So as I understand it helps to restructure the algorithm, change up variable names, and values.

Is there any tutorial's on ReFUDing a stub for a crypter? I've searched far and wide to no avail.

Oxycodone - August 5, 2019, 1:10 am

I think theres more to it than just changing variables etc.

I remember back in the day they basically ruined most public runpe's due to everyone and their mother doing this & using public ones released (which was awesome. but a big mistake due to such a great method being ruined by idiots.)

I'm not into the scene much anymore, so my advice is probably not that great.

but look at this:

<https://www.adlice.com/runpe-hide-code-behind-legit-process/>

Quote:"RunPE: Detection

As this trick is simple, it's also simple to detect. We can assume safely (except for .NET assemblies) that a PE Header will be 99% the same in memory and in the disk image of a process.

Knowing that, we can then compare in each process the PE header of the file on disk with the image in memory. If there's too much differences, we can safely assume the process is hijacked. RogueKiller in version 10.8.3 is able to detect RunPE injection."

So maybe you can have your stub in another language, and maybe thats the best way to go about this?

or find your own private method / a private method.

Sorry for double posting, I can't figure out how to edit the last post. I meant to say research why the headers will be 99% the same in most applications. and why .NET they can't safely assume it'll be the same header.

Figure out how anti-viruses are detecting the runpe, and go from there. See what they're looking for or whats giving your application away.

gigajew - August 13, 2019, 5:29 pm

Here's a 32-bit only RunPE I made earlier maybe it'll help!

gigajew - August 22, 2019, 8:05 pm

Change the call stack and signatures of your methods. Use different apis. There are quite a few to choose from. But beware, some are not available across all windows versions and editions.

The standard signature "void Run(byte[], string)" and the variations thereof are most likely in most

anti viruses

API example:

You want to write memory and currently use WriteProcessMemory

Solution: change it to NtWriteVirtualMemory

Callstack example:

Change the order and count of which your methods are executing in

```
Code VirtualAllocEx(...);  
WriteProcessMemory(...);
```

Becomes

```
Code MsgBox(...);  
WriteProcessMemory(...);  
VirtualAllocEx(...); // original func  
MsgBox(...);  
RtlZeroMemory(...);  
WriteProcessMemory(...); // original func
```

Finally call your api functions dynamically or even manual map functions

equity - August 22, 2019, 8:20 pm

(Aug 22, 2019 08:05 PM)gigajew Wrote: [\[CITE\]](#) Change the call stack and signatures of your methods. Use different apis. There are quite a few to choose from. But beware, some are not available across all windows versions and editions.

The standard signature "void Run(byte[], string)" and the variations thereof are most likely in most anti viruses

API example:

You want to write memory and currently use WriteProcessMemory

Solution: change it to NtWriteVirtualMemory

This is quite useful, thank you.

Facade - August 23, 2019, 8:27 am

Change up functions/variables till you find out what the antivirus actually detects. Since the antivirus has some signatures into a database, you'll need to be a little creative to bypass them all.

Venture - August 23, 2019, 8:40 am

(Aug 22, 2019 08:05 PM)gigajew Wrote: [\[CITE\]](#) *snip*

When I'm looking at your runpe, where did you get specific memory addresses to write to.

gigajew - August 23, 2019, 9:22 am

(Aug 13, 2019 05:29 PM)gigajew Wrote: [\[CITE\]](#) Here's a 32-bit only RunPE I made earlier maybe it'll

(Aug 23, 2019 08:40 AM)Skully Wrote: [\[CITE\]](#) When I'm looking at your runpe, where did you get specific memory addresses to write to.

A debugger. I use windbg with loaded ntdll symbols

OldHolBorn - March 29, 2021, 2:08 pm

Change All Methods Done!, no need to know coding or anything for this, just make a little research, you can change all detected methods easily.

