2010

WhiteHats

Amit Malik
(DouBle_Zer0)

"There is no security on this earth, there is only opportunity."

-- General Douglas MacArthur

[INJECTOR MASK OR A TOOL]

Injector is a little tool that will inject your code into your target process. The code injection concept is not new but the thinking behind injector is new and powerful. The purpose of this document is to explain the concept behind injector and to provide a powerful technique to bypass anti viruses

Introduction:

Injector is a little tool that will inject your code into your target process. The code injection concept is not new but the thinking behind injector is new and powerful. The purpose of this document is to explain the concept behind injector and to provide a powerful technique to bypass anti viruses. Current state of technique is only valid with post exploitation phase, social engineering (eg. Execute SFX archive in silent mode with proper parameter to injector with little variation in interface –PID issues) or physical access in a Penetration test. (viruses and worms can use with variation in interface☺).

Antivirus:

Antivirus (or **anti-virus**) Software is used to prevent, detect, and remove, malware, including computer viruses, worms, and Trojan horse. Such programs may also prevent and remove adware, spyware, and other forms of malware. – Wikipedia

Antivirus play a major role for the security of a system. But for hackers/ pentesters , it creates some big problems. During the post exploitation phase we have some sort of excess on the victim machine and generally we want to upload some tools on the victim machine for batter control but here antivirus play with our tools and detect them as a malicious file and delete them. Now it may be possible if you are using your own tool then antivirus may fail to detect it but this is rare situation during a pentest. All in all if we use publicly exposed tools then there is higher probability of getting caught by antivirus.

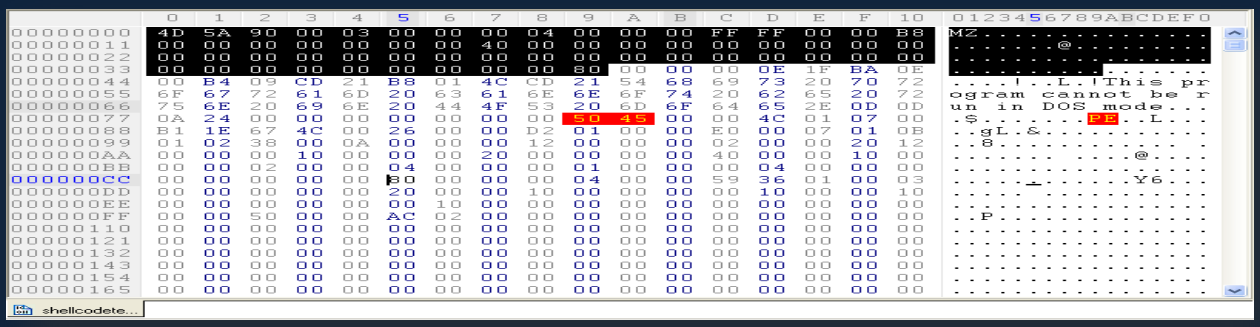
A simple example: suppose we have a reverse shell now we know that this is a single connection to our machine from our victim and we don't want to loose it at any condition. So we upload some tools on the victim machine like netcat or other executable (may be generated from metasploit) to get a backup connection etc.. may be to upgrade our shell to merepreter shell. But our victim is using a anti virus and anti virus delete the uploaded files. Now we need something else means either bypass AV or write our own code.

But the question How antivirus detect our executables (.exe).

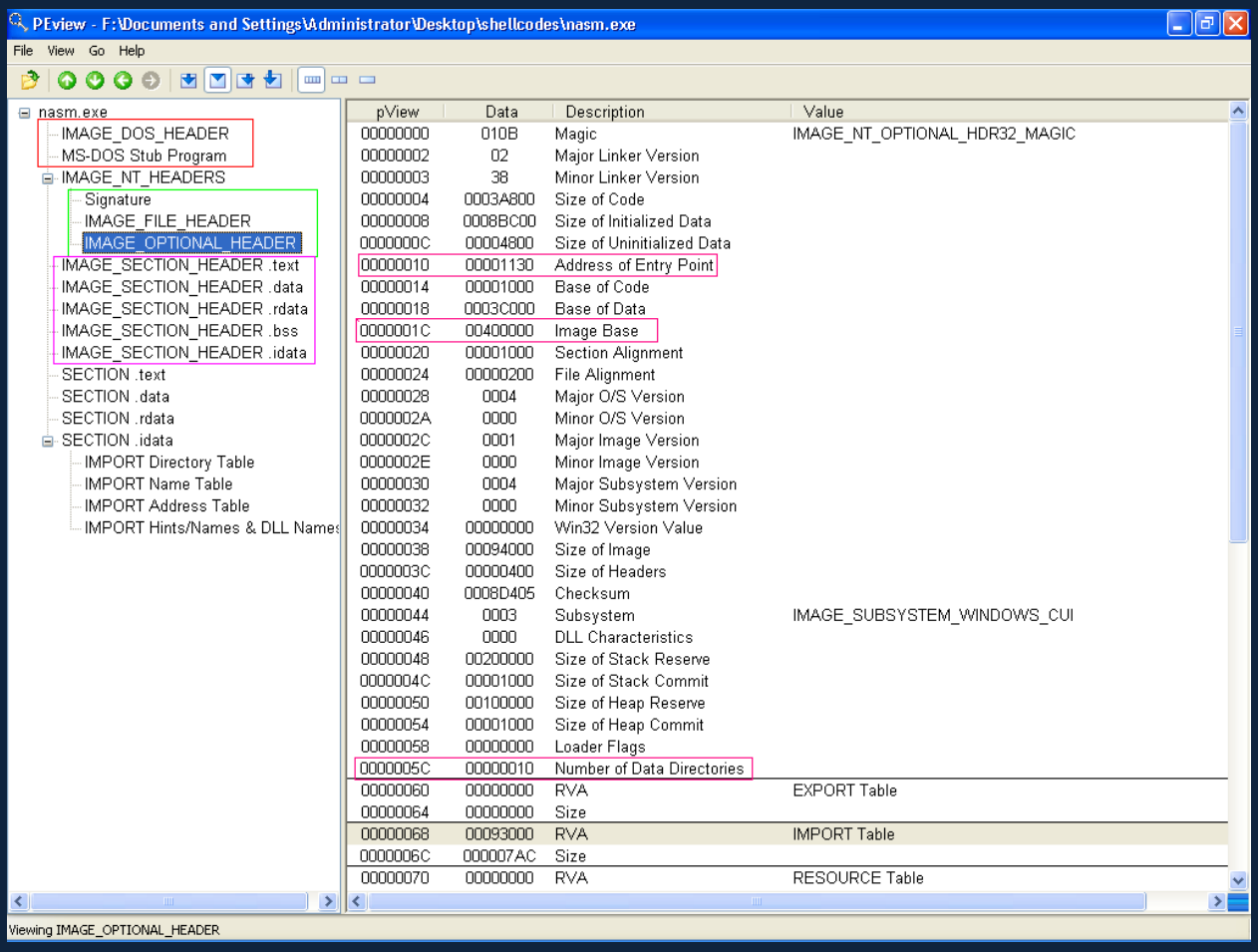
Executables (.EXE):

Basically exe files not only contain your code but also contain some important data that is only meaningful for a Operating system loader.

And technical name for that data is PE(portable executable) file header/ PE file format. I am not explaining pe file format here. But here is snapshot:

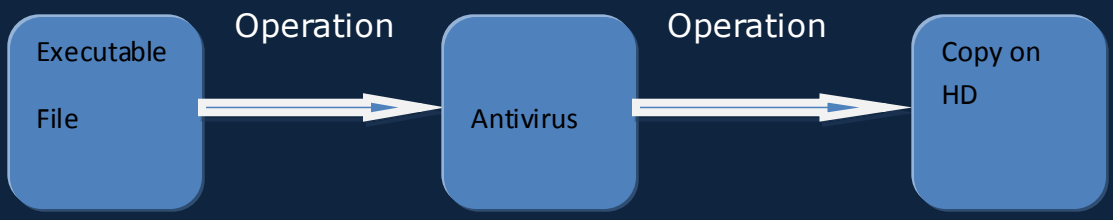


PEview:



So when we execute exe, windows loader first read the PE header and on the basis of header information loader loads the file into memory.

Executable Detection (top level view):

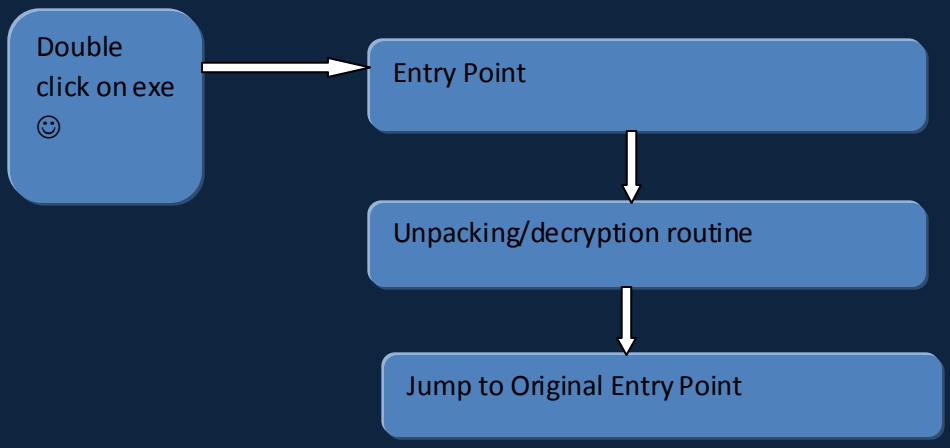


(Antivirus scan the file if(malicious) then delete/etc.)

So when we copy the file on system then antivirus scan the file (check for known signatures) and if the file is malicious then delete the file.

Encrypter/packer/protector:

The above technologies (Encrypter/packer/protector) are not only to evade anti viruses. People also use them for ethical work. Basically encryptor/packers encrypt/pack the file and add the decryption/unpacking routine into the file. So when we execute the file windows loader loads the file on the basis of PE header (encryptor/packer does not pack the PE header of a file they only change the necessary values in PE header. Eg. Address of entry point, sections etc..). so the unpacking work something like this:



So unpacking/decryption routine unpack the code and then jump on the Original Entry Point (on our real code freshly recovered from unpacking routines).

So if we pack the exe with any packer then Antivirus should not be able to detect the exe?. Ok now its time to do some practical.

Pack any malicious file with UPX (a freeware packer) and then test again with Antivirus. You should see that your AV is again popping up the alert and telling that hey don't try to fool me it is malicious.. But our file was packed how AV still know that it is a malicious file and the answer is AV has also signature for UPX, so what AV is doing it loads the file detect the packer then decrypt/unpack(AV also know how to unpack UPX packed files ☺) the file and then test the file, And this is the reason that AV still know that file is malicious. But hey where is proof that AV is doing all this shit to test file.

Ok to prove just open the same packed file in a hex editor and just overwrite the starting bytes with some garbage values and then again test it with your AV. Now you should see that AV is saying that file is clean (Reason: Because file is corrupted and AV have no option to unpack/decrypt it) fantastic that's all we want..

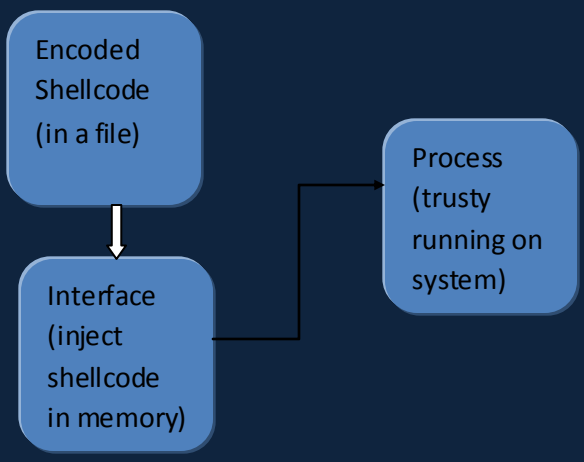
The Concept of Injector (Ingeneric way to bypass AV):

Because exe files are going to detected by AVs(at least if you pack them with the publicly exposed packers/encryptors). So we have to think in a another way..

And the another way is: split the exe into two parts (not physically)

1) **The core code (the actual code that performs a specific task for eg. Bind shell)**
2) **The interface – a mechanism that will inject the code into memory and execute that code.**

So the functioning is something like this:



Note that from the above explanation we know that shellcode/code into a file is not going to be detected by AV because AV don't know how to decode shellcode. (Don't talk about Heuristic, I know AV vendors are joking ☺)

Important Note: you may be thinking that why I am saying encoded shellcode because if you use metasploit shellcodes there signatures may be in AVs. If you encode the shellcode with any available encoder in metasploit then AVs not able to decode it in a file and not able to detect it (if you don't understand it read the whole stuff again ☺).

Although in some cases (Eg. Avast may be with others also) AV not alert if you use shellcodes that are not encoded because AV think that txt file are lame files. But if you force fully scan the file than AV alert.

Second part of the concept is the interface that will inject the code into a process. Code injection is not a new concept (dll injection is one of the most popular example).

Note: All the things are generic and are not specific to any tool or shellcodes. Metasploit and shellcodes are used only to demonstrate the concept. You can also inject your codes "that are detectable to AV in exe mode" with this method and can bypass AV.

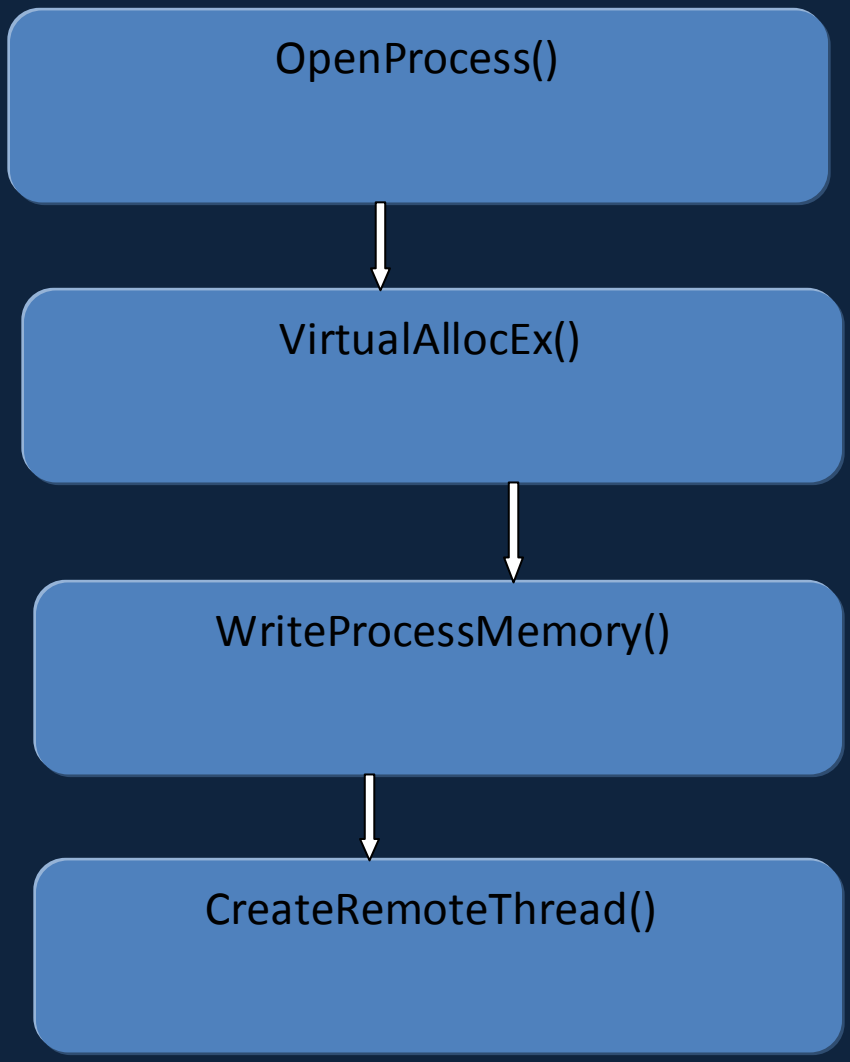
Things that you can do with this method:

1) Can backdoor a process
2) Can provide many backup shells (every type)

Key Benefit: we can use publically available tool (malicious) without fear.

And many-2 others things (that depends on imagination).

Work Flow of Injector



Note: With variations we can accomplish many things ☺

[Download Injector]

https://sites.google.com/site/mamit30/home/injector

[Video Demonstration]

http://vimeo.com/14139105

References:

1. Metasploit
   http://www.metasploit.com
2. UPX
   http://upx.sourceforge.net/
3. PEview
   http://www.magma.ca/~wjr/
4. An In-depth Look into the Win32 Portable Executable File Format
   http://msdn.microsoft.com/en-us/magazine/cc301805.aspx
5. Three Ways to Inject Your Code into Another Process
   http://www.codeproject.com/KB/threads/winspy.aspx
6. Fsecure Malware Analysis Course (free)
   https://noppa.tkk.fi/noppa/kurssi/t-110.6220/luennot