

Forum:[Hack Forums](#)

Thread: [Read/Write Process Memory as another process?](#)

Meterpreter - December 30, 2017, 1:45 am

Hey guys, I know this is possible, and I don't want to be babied, merely want to know the technical name for it so I can research.

Attach to process -> Attach to another process from that process -> tunnel read/write back and forth.

So my program attaches to chrome.exe, then attaches to bobthebuilder.exe, reads and writes data. Bobthebuilder.exe thinks chrome.exe is reading/writing data.... I'm over explaining...

Any help would be much appreciated, haven't coded in a little while.

Again, don't want to be bottle fed how to do it, just the name of it, and perhaps a helpful link if you can be bothered.

Thank you <3

_Pr0xy - December 30, 2017, 6:05 am

I'm not sure if this is what you mean, but in C you can hook a functions API calls and "listen" for them, which a lot of banking trojans such as Zeus take advantage of in order to view plaintext bank credentials - <https://blog.didierstevens.com/2008/03/19/bpmtk-spying-on-ie/> - that site gives an example on how you can hook Internet Explorer 8, but I am sure it can be applied to other applications, so one program listens for the API calls and passes it onto another. If I misunderstood what you are trying to ask then completely ignore this post ?

jammy47 - December 30, 2017, 6:11 am

is this something to do with handles or a debugger?

You need to create 2 processes. 1 will read data from chrome and write to its own space and the 2nd one will read from the process that is reading from chrome. You'll need handles. Openprocess can be used to get handle.

SIM-3475829 - December 30, 2017, 7:10 am

I am pretty sure he is trying to make cheats?
Correct me if i am wrong

movsx - December 30, 2017, 1:27 pm

Yes, look into interprocess communication. It is a standard technique for bypassing anticheat ObRegisterCallbacks handle stripping (BattlEye, EAC)- as they don't strip handles from system processes like lsass, csrss, svchost, etc.

Inject code into the system process, then use interprocess communication to do reading/writing.

Meterpreter - December 30, 2017, 4:14 pm

I think I found it, I believe it is CreateRemoteThread.... Putting a segment of my code into another processes memory with MemoryAloc function, then from there my segment runs inside the other

process.

Please correct me if I am wrong, don't want to create a useless program :')

aeiro - December 30, 2017, 8:45 pm

try process hollowing or forking.

morph - January 2, 2018, 8:00 am

My suggestion is that you allocate remote memory in Chrome.exe and call Read/WriteProcessMemory through creating a thread in Chrome.exe. I'll wildly assume that it's CS:GO you are creating a cheat for, and if that's true, don't bother doing this if it's for "safety". Won't help.

Meterpreter - January 2, 2018, 3:57 pm

(Jan 2, 2018 09:00 AM)n0ctua Wrote: [\[CITE\]](#) My suggestion is that you allocate remote memory in Chrome.exe and call Read/WriteProcessMemory through creating a thread in Chrome.exe. I'll wildly assume that it's CS:GO you are creating a cheat for, and if that's true, don't bother doing this if it's for "safety". Won't help.

Yeh I have recently looked up and studied multiple methods. Found RunPE methods to be useless.

Guess I'd be moderately safe considering my code is actually mine with zero public headers.

Next step would be to great a ring0 unsigned driver? But is that necessary for VAC? From what I've found ring0 is harder to accomplish on Windows 10? Not many resources and such

Vyxn - January 2, 2018, 4:04 pm

(Jan 2, 2018 04:57 PM)Meterpreter Wrote: [\[CITE\]](#) Yeh I have recently looked up and studied multiple methods. Found RunPE methods to be useless.

Guess I'd be moderately safe considering my code is actually mine with zero public headers.

Next step would be to great a ring0 unsigned driver? But is that necessary for VAC? From what I've found ring0 is harder to accomplish on Windows 10? Not many resources and such

So you want your cheat to be internal? I'd say that's one thing VAC love to find on their process but good luck. ?

Meterpreter - January 2, 2018, 5:23 pm

(Jan 2, 2018 05:04 PM)Vyxn Wrote: [\[CITE\]](#) So you want your cheat to be internal? I'd say that's one thing VAC love to find on their process but good luck. ?

My cheat is currently external.

I was thinking of making a driver, not injecting a DLL. I want to operate at system level (ring0), meaning I have direct access to memory. Which would keep me safe from any level 3 anti-cheat protection.

I'm not too sure about the specifics of VAC, but from what I have read VAC operates at user level (3).

Regardless, I'm still new to C++, highly doubt I'll be able to make a fully functioning driver to pass memory info back to a fully functioning program. :')

morph - January 3, 2018, 4:34 am

Man.. I'm getting A LITTLE tired on people giving advice about VAC when they don't know shit. No

offense Vyxn, but you are completely fucking wrong. When it comes to VAC, it doesn't matter at all if the cheat is external or internal. If VAC has access to the cheats memory (which, SURPRISE, it has in both cases) it can get detected.

To OP:

As long as you will not be selling this cheat or giving it out to loads of people, you honestly shouldn't care about security. It will stay undetected as long as you built it from scratch.

int main() - January 3, 2018, 8:58 am

What you're trying to do is steal a handle from another process to read & write

I suggest you post your questions on unknowncheats most people on here will lead you astray like all posts besides the one above mine, but before you do make sure you're proficient enough to know what to ask, you seemed to have an error getting the base address of a module which gives me the idea that you're not very experienced, so by all means do not try to do this until you have a good grasp of what you're doing.

annnnnnnnnd no it's not createremotethread

morph - January 3, 2018, 11:14 am

(Jan 3, 2018 09:58 AM)int main() Wrote: [\[CITE\]](#) What you're trying to do is steal a handle from another process to read & write

I suggest you post your questions on unknowncheats most people on here will lead you astray like all posts besides the one above mine, but before you do make sure you're proficient enough to know what to ask, you seemed to have an error getting the base address of a module which gives me the idea that you're not very experienced, so by all means do not try to do this until you have a good grasp of what you're doing.

annnnnnnnnd no it's not createremotethread

You say that I'm the only one posting something that won't lead him astray.. And below that paragraph, you say that you can't use CreateRemoteThread? Eh.. Did you even read what I wrote? You can very well use CreateRemoteThread for what he wants to do.

int main() - January 3, 2018, 1:29 pm

(Jan 3, 2018 12:14 PM)n0ctua Wrote: [\[CITE\]](#) You say that I'm the only one posting something that won't lead him astray.. And below that paragraph, you say that you can't use CreateRemoteThread? Eh.. Did you even read what I wrote? You can very well use CreateRemoteThread for what he wants to do.

He actually can, I'm dumb

morph - January 3, 2018, 3:56 pm

(Jan 3, 2018 02:29 PM)int main() Wrote: [\[CITE\]](#) He actually can, I'm dumb
?

BTW, off topic, are you the guy that's selling an external CS:GO cheat with some Ub3r guy?

Meterpreter - January 3, 2018, 4:05 pm

I mean I never used Visual Studio 2017 in the past and not used to getting pointless errors. With 2015 my code operates perfectly.

To be fair, I went from knowing zero c++ to coding this from scratch... So I thought I was doing pretty good in terms of learning time.

I also created an external IP grabber, PRETTY FUCKING LIT MATE. Nah but in all seriousness, do you guys know if it's difficult to run unsigned drivers on Windows 10? From what I've read every version of Windows treats drivers differently which is weird and annoying. And I couldn't find any modern/new/up-to-date resources on making drivers.

morph - January 3, 2018, 4:18 pm

(Jan 3, 2018 05:05 PM)Meterpreter Wrote: [\[CITE\]](#) I mean I never used Visual Studio 2017 in the past and not used to getting pointless errors. With 2015 my code operates perfectly.

To be fair, I went from knowing zero c++ to coding this from scratch... So I thought I was doing pretty good in terms of learning time.

I also created an external IP grabber, PRETTY FUCKING LIT MATE. Nah but in all seriousness, do you guys know if it's difficult to run unsigned drivers on Windows 10? From what I've read every version of Windows treats drivers differently which is weird and annoying. And I couldn't find any modern/new/up-to-date resources on making drivers.

Firstly you'll need to learn C, because you'll need to make the driver in C. I recommend you learn some more about C/C++ and some more about reverse engineering before continuing.

Meterpreter - January 3, 2018, 6:06 pm

(Jan 3, 2018 05:18 PM)n0ctua Wrote: [\[CITE\]](#) Firstly you'll need to learn C, because you'll need to make the driver in C. I recommend you learn some more about C/C++ and some more about reverse engineering before continuing.

Will I need to use out-dated OS alongside a Windows kernel exploit?

Or is it as simple as allowing unsigned drivers, and then running it?

For sure I will man, just questioning a couple things.

Vyxn - January 4, 2018, 6:31 am

(Jan 3, 2018 05:34 AM)n0ctua Wrote: [\[CITE\]](#) Man.. I'm getting A LITTLE tired on people giving advice about VAC when they don't know shit. No offense Vyxn, but you are completely fucking wrong. When it comes to VAC, it doesn't matter at all if the cheat is external or internal. If VAC has access to the cheats memory (which, SURPRISE, it has in both cases) it can get detected.

To OP:

As long as you will not be selling this cheat or giving it out to loads of people, you honestly shouldn't care about security. It will stay undetected as long as you built it from scratch.

No worries, I'm not offended and I love to be corrected. ?

My defense on my post about internal as VAC really read their own process first rather than hack using external. This was only base on my reverse engineering experience. I'd love a response from you if you want.

(Jan 3, 2018 07:06 PM)Meterpreter Wrote: [\[CITE\]](#) Will I need to use out-dated OS alongside a Windows kernel exploit?

Or is it as simple as allowing unsigned drivers, and then running it?

For sure I will man, just questioning a couple things.

I'd suggest you read more about how rootkit works if you really want to go in your plan. (I have a less idea on this topic and still working on it)

int main() - January 4, 2018, 9:13 am

(Jan 4, 2018 07:31 AM)Vyxn Wrote: [\[CITE\]](#) No worries, I'm not offended and I love to be corrected. ? My defense on my post about internal as VAC really read their own process first rather than hack using external. This was only base on my reverse engineering experience. I'd love a response from you if you want.

I'd suggest you read more about how rootkit works if you really want to go in your plan. (I have a less idea on this topic and still working on it)

<https://www.unknowncheats.me/forum/anti-cheat-bypass/128973-does-vac-exactly-work-noob-version.html>

Worth a read if you're interested in the topic

morph - January 4, 2018, 10:35 am

(Jan 4, 2018 07:31 AM)Vyxn Wrote: [\[CITE\]](#) No worries, I'm not offended and I love to be corrected. ? My defense on my post about internal as VAC really read their own process first rather than hack using external. This was only base on my reverse engineering experience. I'd love a response from you if you want.

I'm quite surprised you didn't get triggered ?

Anyhow.. I don't know if the grammar in your post is fucked up or if I am being mentally disabled. But if you meant that VAC reads from csgo.exe before any other process it is not relevant in this discussion.