Forum:Hack Forums

Thread: ReadProcessMemory and WriteProcessMemory

ProtectedOverride - *October 15, 2019, 3:49 am*

Hi,

I have been playing around reading and writing to the PEB of an external process using ReadProcessMemory and WriteProcessMemory which is part of kernel32.dll. Whilst I am able to read values, such as CommandLine, I do not seem to be able to then change this by using WriteProcessMemory. Does anyone have an example of this that would help me do this?

Thanks
James

HFDone - *October 15, 2019, 4:46 am*

you need to use unicode string

ProtectedOverride - *October 15, 2019, 4:59 am*

This is what I am trying:

```
byte[] dataBuffer = Encoding.Unicode.GetBytes("NEW STRING");
IntPtr bytesWritten = IntPtr.Zero;
int cnt = 0;

WriteProcessMemory(handle, us.Buffer, dataBuffer, dataBuffer.Length, ref cnt)
```

HFDone - *October 15, 2019, 5:09 am*

search for peb patch and see how its done

GraeskMytologi - *October 15, 2019, 11:27 am*

(Oct 15, 2019 03:49 AM)ProtectedOverride Wrote: *[CITE]* Hi,

I have been playing around reading and writing to the PEB of an external process using ReadProcessMemory and WriteProcessMemory which is part of kernel32.dll. Whilst I am able to read values, such as CommandLine, I do not seem to be able to then change this by using WriteProcessMemory. Does anyone have an example of this that would help me do this?

Thanks
James
You need to change the protection level with VirtualProtectEx

Paskowsky - *October 15, 2019, 12:41 pm*

(Oct 15, 2019 03:49 AM)ProtectedOverride Wrote: *[CITE]* Hi,

I have been playing around reading and writing to the PEB of an external process using ReadProcessMemory and WriteProcessMemory which is part of kernel32.dll. Whilst I am able to read values, such as CommandLine, I do not seem to be able to then change this by using WriteProcessMemory. Does anyone have an example of this that would help me do this?

Thanks

James
What error WriteProcessMemory shows you?

If i remeber correctly in PEB strings are stored as UNICODE_STRING
Code typedef struct _UNICODE_STRING {
    USHORT Length;
    USHORT MaximumLength;
    WCHAR *Buffer;
} UNICODE_STRING, PUNICODE_STRING;so to replace a string with another you can allocate a new RW memory page, write your custom string and edit the PEB CommandLine pointer and lenght to point to allocated value.

Regards,
Paskowsky

Skofos - *October 16, 2019, 5:49 am*
always using @Amaa game trainer. WriteProcessMemory and ReadProcessMemory : GitHub - ama6nen/project-autism: Project-autism (old growtopia trainer) 2017 source code

ProtectedOverride - *October 16, 2019, 7:45 am*
@HFDone I tried the example back from 2011:

pastebin dot com/8mujr2ps

However I find I am taking this path:

```
    if (!WriteProcessMemory(hProcess, rtlUserProcParamsAddress + 0x70,
(IntPtr)(&Current_ImageBase), Marshal.SizeOf(typeof(UNICODE_STRING)), out intPtrOutput))
            {
                Console.WriteLine("ERROR: Failed to reflect change back to PEB.\n");
                return false;
            }
```

Could this be that the code has some "32Bit" specifics that need changing for working with 64Bit? I have not worked at this low of a level when programming before so just trying to understand.

Thanks
James

Further update, I had a play with someone else implementation:

pastebin dot com/Z9hdggtM

However this part:

status = NtProtectVirtualMemory(hProc, ref lpRemoteCommandLine, ref dwSizeUnicodeString, 0x04 /* read, write */, ref dwOldProtections);

Causes the applicaiton to exit with an "Access Violation"..Is this just .NET doing something I can bypass?

HFDone - *October 16, 2019, 9:18 am*
are you using v2 using System; using System.Linq; using System.Collections.Generic; using Syst -

Pastebin.com

you should enable safe code in vs to use it
you have wrong value in the second param should be + 0x40

```
Current_ImageBase.buffer = pNewAddr;
if (!WriteProcessMemory(hProcess, rtlUserProcParamsAddress + 0x40,
(IntPtr)(&Current_ImageBase), Marshal.SizeOf(typeof(UNICODE_STRING)), out intPtrOutput)) {
    Console.WriteLine("ERROR: Failed to reflect change back to PEB.\n");
     return false;
}
```

ProtectedOverride - *October 16, 2019, 9:37 am*

is that the "Allow unsafe code: enabled" in Build properties? I have that ticked. Tried with 0x70 , as I noted 0x40 was for windows title..oops but getting the same error, the bool returns false when using this example... hmmm

HFDone - *October 16, 2019, 10:02 am*

create new project c&p the pastebin code run notepad and try it

ProtectedOverride - *October 16, 2019, 10:32 am*

ok so, if I comiple as x64 and run it against an x64 process this is when I get the above error. If I compile as x32 and run against x32 process I get no error. However I do not see the change reflected in Task Manager under "Command Line". I do close and reopen to get it to refresh it's values. Any ideas?

Update:

It seems it is working for 32Bit against 32Bit when I check in process explorer but Task manager does not reflect the change, any ideas if I can mitigate this? Regarding the 64Bit problem I am guessing I need to look at the variable types used maybe?

HFDone - *October 16, 2019, 10:54 am*

the patch is in memory and probably the task get the cmd from the file
for 64 bit you need 64 bit address or you can use anycpu option to work with both

ProtectedOverride - *October 16, 2019, 11:10 am*

but the command is only in memory? Lets sya I execute SomeProcess.exe -t SuperSecretArgument ... Are you saying I can not "overwrite" the "SuperSecretArgument" from task manager?
Thanks!

HFDone - *October 16, 2019, 11:26 am*

task just show you the process info
anyway you dont really want the task show you change it

ProtectedOverride - *October 16, 2019, 12:02 pm*

in my scenario I do want the task manager to reflect the change :) Anyway I have a way around that. If I can get it working on 64bit I have a workable solution. Tried with "AnyCPU" compiling, no luck, fails to write back change as before. When you say need 64 bit address, what part of the code is 32 bit specific?

Thanks

Paskowsky - *October 16, 2019, 12:44 pm*

(Oct 16, 2019 12:02 PM)ProtectedOverride Wrote: *[CITE]* in my scenario I do want the task manager

to reflect the change :) Anyway I have a way around that. If I can get it working on 64bit I have a workable solution. Tried with "AnyCPU" compiling, no luck, fails to write back change as before. When you say need 64 bit address, what part of the code is 32 bit specific?

Thanks
What changes from x86 to x64 is offsets from PEB base address since different data types are used and so IntPtr size is 8 instead of 4

like here :

Code if (!ReadProcessMemory(hProcess, peb + 0x10, (IntPtr)(&rtlUserProcParamsAddress), Marshal.SizeOf(typeof(IntPtr)), out intOutput))
Take a look at this website :
https://www.geoffchappell.com/studies/windows/win32/ntdll/structs/peb/index.htm

peb + 0x10 would become peb + 0x20

Regards,
Paskowsky ?

ProtectedOverride - *October 16, 2019, 2:52 pm*
(Oct 16, 2019 12:44 PM)Paskowsky Wrote: *[CITE]* What changes from x86 to x64 is offsets from PEB base address since different data types are used and so IntPtr size is 8 instead of 4

like here :

Code if (!ReadProcessMemory(hProcess, peb + 0x10, (IntPtr)(&rtlUserProcParamsAddress), Marshal.SizeOf(typeof(IntPtr)), out intOutput))
Take a look at this website :
https://www.geoffchappell.com/studies/windows/win32/ntdll/structs/peb/index.htm

peb + 0x10 would become peb + 0x20

Regards,
Paskowsky ?
Thank you @Paskowsky, this has helped a lot, made some progress but having trouble allocting memory in 64bit process, presumably I need to change this:

```
/* allocate memory for Input /
        Current_ImageBase.Length = (ushort)(input.Length  2);
        pNewAddr = VirtualAllocEx(hProcess, IntPtr.Zero, Current_ImageBase.Length, /*
AllocationType.Commit / 0x1000 | / AllocationType.Reserve / 0x2000, /
MemoryProtection.ExecuteReadWrite */ 0x40);
        if (pNewAddr == IntPtr.Zero)
        {
            Console.WriteLine("ERROR: Could not allocate memory in target process.\n");
            return false;
        }
```

Paskowsky - *October 16, 2019, 11:02 pm*
(Oct 16, 2019 02:52 PM)ProtectedOverride Wrote: *[CITE]* Thank you @Paskowsky, this has helped a

lot, made some progress but having trouble allocting memory in 64bit process, presumably I need to change this:

```
 /* allocate memory for Input /
            Current_ImageBase.Length = (ushort)(input.Length  2);
            pNewAddr = VirtualAllocEx(hProcess, IntPtr.Zero, Current_ImageBase.Length, /*
AllocationType.Commit / 0x1000 | / AllocationType.Reserve / 0x2000, /
MemoryProtection.ExecuteReadWrite */ 0x40);
            if (pNewAddr == IntPtr.Zero)
            {
                Console.WriteLine("_error_: Could not allocate memory in target process.\n");
                return false;
            }
 [code]
 throw new Win32Exception(Marshal.GetLastWin32Error());
 [code]
```

inside the if, it will show what error you are getting from windows.

SKIDROOTKIT - *October 18, 2019, 11:01 am*

(Oct 16, 2019 02:52 PM)ProtectedOverride Wrote: *[CITE]* Thank you @Paskowsky, this has helped a lot, made some progress but having trouble allocting memory in 64bit process, presumably I need to change this:

```
 /* allocate memory for Input /
            Current_ImageBase.Length = (ushort)(input.Length  2);
            pNewAddr = VirtualAllocEx(hProcess, IntPtr.Zero, Current_ImageBase.Length, /*
AllocationType.Commit / 0x1000 | / AllocationType.Reserve / 0x2000, /
MemoryProtection.ExecuteReadWrite */ 0x40);
            if (pNewAddr == IntPtr.Zero)
            {
                Console.WriteLine("_error_: Could not allocate memory in target process.\n");
                return false;
            }
```

Are both your own process & the remote process running as the same architecture? (64 bit, not wow64)

Does hProcess have the PROCESS_VM_OPERATION process access right?

ProtectedOverride - *October 19, 2019, 4:11 am*

My issue was i had changed the memory protection setting on VirtualAllocEx. I made some progress in that CommandLine was being blanked. Away for a couple of days but ill do some more testing when I get back to see if I can figure what I might have missed.

ProtectedOverride - *October 21, 2019, 5:54 am*

Thanks for all the help, got this working, my final problem was I was using 0x78 rather then 0x70. Updated this and got it working on x64 :)