

picoCTF{ov3rflows\_ar3nt\_that\_bad\_ef01832d}

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <signal.h>


#define FLAGSIZE_MAX 64


char flag[FLAGSIZE_MAX];


void sigsegv_handler(int sig) {

    printf("%s\n", flag);

    fflush(stdout);

    exit(1);

}


void vuln(char *input){

    char buf2[16];

    strcpy(buf2, input);

}


int main(int argc, char **argv){

    FILE *f = fopen("flag.txt", "r");

    if (f == NULL) {

        printf("%s %s", "Please create 'flag.txt' in this directory with your",

            "own debugging flag.\n");

        exit(0);

    }

    fgets(flag, FLAGSIZE_MAX, f);

    signal(SIGSEGV, sigsegv_handler); // Set up signal handler


    gid_t gid = getegid();

    setresgid(gid, gid, gid);


    printf("Input: ");

    fflush(stdout);

    char buf1[100];
```

```

gets(buf1);

vuln(buf1);

printf("The program will exit now\n");

return 0;
}

```

```

void vuln(char *input){

    char buf2[16];

    strcpy(buf2, input);

}

```

We can see from this function that it is copying the input to buf2 variable which has a size of 16 so we have to overflow it

The screenshot shows the picoCTF website interface. The challenge 'buffer overflow 0' is displayed, with a description that asks the user to overflow a buffer. A terminal window is overlaid on the challenge page, showing a netcat listener on saturn.picoctf.net:50958. The input 'picoCTF{ov3rfl0ws\_ar3nt\_th4t\_b4d\_ef01832d}' is entered, which causes the program to crash with a segmentation fault. The challenge has been solved by 24,128 users and has an 88% like rating.