

CSC 262

Homework 4

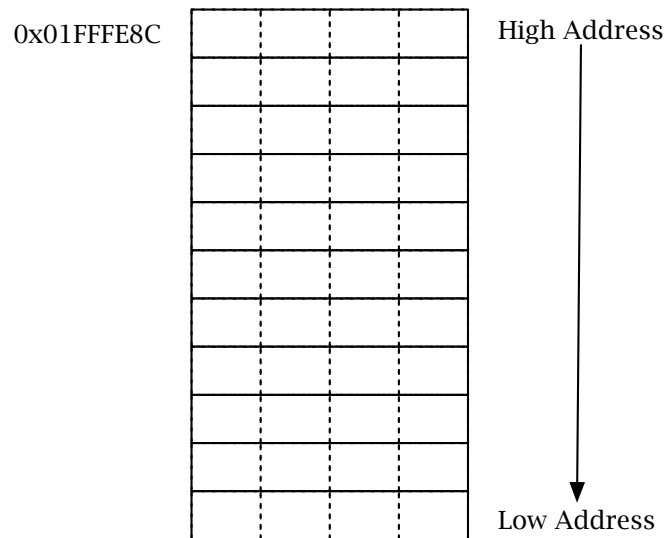
Due Oct. 28, 2008

1 Stack Layout

The following questions all refer to items on the stack. The stack as illustrated is for a call to the `parseArgs` function, defined as:

```
int parseArgs(char** args){  
    int num_args = 0;  
    char buf[16];  
  
    scanf("%s", buf);  
    ...  
}
```

- A) Fill out the stack diagram by labeling each slot with its contents (e.g. write “num_args” rather than the value). Assume that this is after `parseArgs` has been called, but just before the `scanf` has been executed.



- B) Assuming that the top of this frame is located at the address `0x01FFFE8C`, at what address is the return address located?
- C) Assuming that the top of this frame is located at the address `0x01FFFE8C`, at what address is `buf[0]` located?

- D) How many bytes would you have to overwrite in order to corrupt the return address (assuming that the overflow was in `buf`)?

2 ACLs and Capabilities

UNIX systems use a form of access control list associated with each file. The `rwxr-xr--` string is an ACL for the file's owner, the file's group and the rest of the universe. For the purpose of this question, assume that a new UNIX-style operating system is being developed that associates capabilities with users. In this scheme the string `rwxr-xr--` would refer to the capabilities of the user with respect to files they own (`rwX`), files their group owns (`r-x`) and all other files (`r--`).

- A) Assuming that there are 256 users on the system and 16 million (10^6) files, and that each ACL took up 4 bytes of storage in the file's inode; how much space does the capabilities system save?
- B) Assume that the capabilities system is just as fast as the old-style ACL system. Is the capabilities scheme more flexible or less flexible than the old ACL scheme? Please use specific examples in your answer.