sys_write(char *buff){
    char *tmp;
    int len;
    len=strlen(tmp);
    tmp=malloc(len+1)
    (------conten switch----)
    strcpy(temp,buff);
}
Content switch may put something in the buffer and causes an over flow

Solution
• Use TLB to move the buffer to kernel, so that a second thread or something could not modify the buffer.
• Check the length of buff

*Time-of-check-to-Time-of-use (race condition)
*Pass by values between trust domains

Wuftp
Void loguser(char *user){
    char buff(512);
    sprintf(buf,user);
}
Suppose 513 char is write to buff, return address is overwritten and could lead to execution of attacker’s code.

NX bit (hardware) – mark pages of stack as non-executable would solve this problem since attacker could not run their code.
However, a new kind of attack, arc injection, aka “return-to-libc” counters it.