Homework 2 Austin Strobel

1.

a)

Since the buffer is 128 bytes in the example given, a string larger than 128 bytes could be assigned to "str," which is then copied into "buf." This would then cause a buffer overflow. The return address in the function do_something() can then be overwritten. The address of the shellcode in the malicious code will then overflow into the original return address on the stack. The return statement at the end of the function will return the shellcode, which is where the attacker's code is.

b)

An attacker could overflow the user variable with an input greater than 128 bytes. In the strcpy() call, the buf variable would be overflowed and could overwrite the tmp variable with a value less than 100, like 75, in order to enter the first if statement.

2.

a)

A symlink attack is certainly possible in this scenario, as when alice-write and alice-read are being executed, an attacker could replace the user's intended file with his/her own file. Given that alice-write is the target of this attack, the symlink will lead the program to the attacker's file as if it were actually being led to the intended file. The attacker then gains write access to the user's file.

In a denial of service attack, the attacker could bombard the system by executing SetUID scripts with different input files. As the permissions need to be checked after each execution, the CPU becomes overloaded, limiting available resources. Since the system is operating slowly due to the flood of input caused by the attacker (by continuously executing alice-write), legitimate users are not able to access the file that is being attacked.

b)

The admin of this system could set up logs of which users are accessing the system and what programs they are executing. This way, if a user is suddenly executing multiple programs in a short amount of time, the authorities can respond to an attack quicker. If the rate is limited by the admin, the user can only make requests every so often, not bogging down the system.

In order to avoid a symlink attack, the user could use the "stat" command to see if the path is a symlink or not. Using absolute paths, rather than relative paths, is also an option to prevent a symlink attack, as a relative path can be manipulated to make it seem like it points to a legitimate file.