-We ssh into vsftpd for easier access of the files.

-Within dejavu.c, it initialized door[8], which only allocated 8 bytes for “door”.

-However, gets(door) prompts the user for input but did not limit the size of the user’s input.

-Therefore, a user input of larger than 8 bytes can overflow the buffer and change the rip address to point to the shell code to open up a shell script.

-We ran the program in gdb and set a breakpoint at deja\_vu.

-From the gdb, we used “print &door” to get the address of the door array, which is 0xbffffb58.

-We also used “info register ebp” to get the address of %ebp, which is 0xbffffb68.

-This shows that the ebp is 16 bytes up the stack from door array.

-By inserting 16 bytes of random junk, we will reach the ebp. Since the rip is directly above the ebp, we need to insert another 4 bytes of junk before reaching the rip.

-Next, we need the return address of the rip to point to the start of the shell code. Therefore, we need to add 8 bytes from the ebp address, to point directly after the rip, where we started inserting the shell code.

-After some calculations, our input is (20 bytes of junk, which we used 20 ‘a’) + (return address for rip to point to the start of the shell code, which is ebp address + 8 bytes) + (shell code)

-This causes the rip to point to the shell code we inserted and spawns a shell script