Toctou attack:

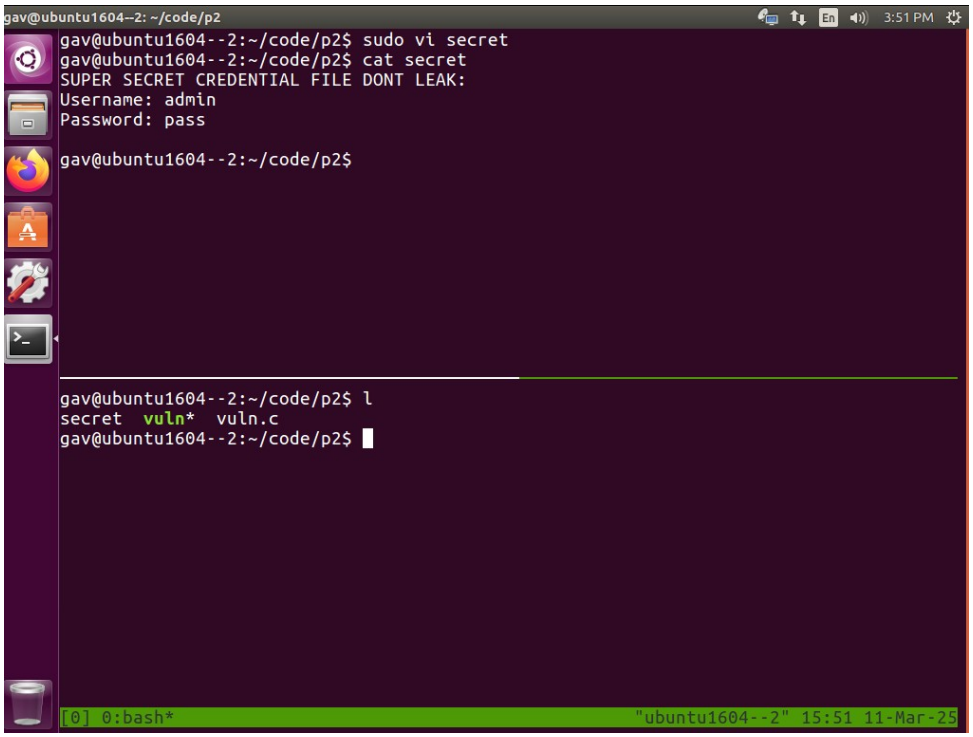
Gavin Stankovsky

Attack Overview:

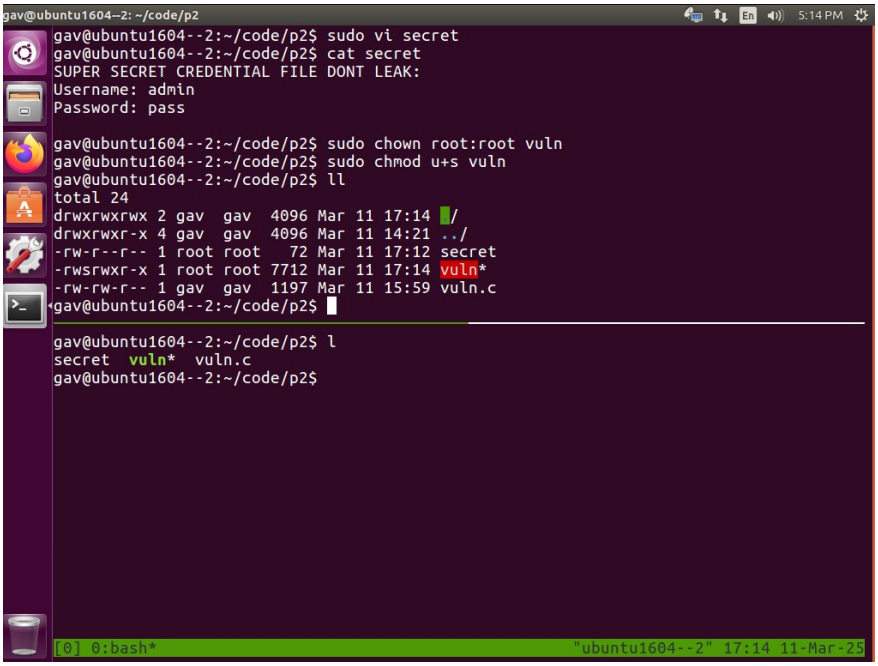
The TOCTOU attack revolves around abusing race conditions in a program with elevated privilege to attack with elevated privilege, specifically attacking after the Time Of Check and before the Time Of Use. The vulnerable program allows a user to enter a file name that will be written to with elevated privilege. After the file name is input the program it does a check to see if the inputted file name is “secret” as it has sensitive information not writable to non-root users this is the Time Of Check (TOC). The vulnerable program then allows the user to input information that will be written into the provided file name, this is the Time Of Use (TOU).

Attack Start:

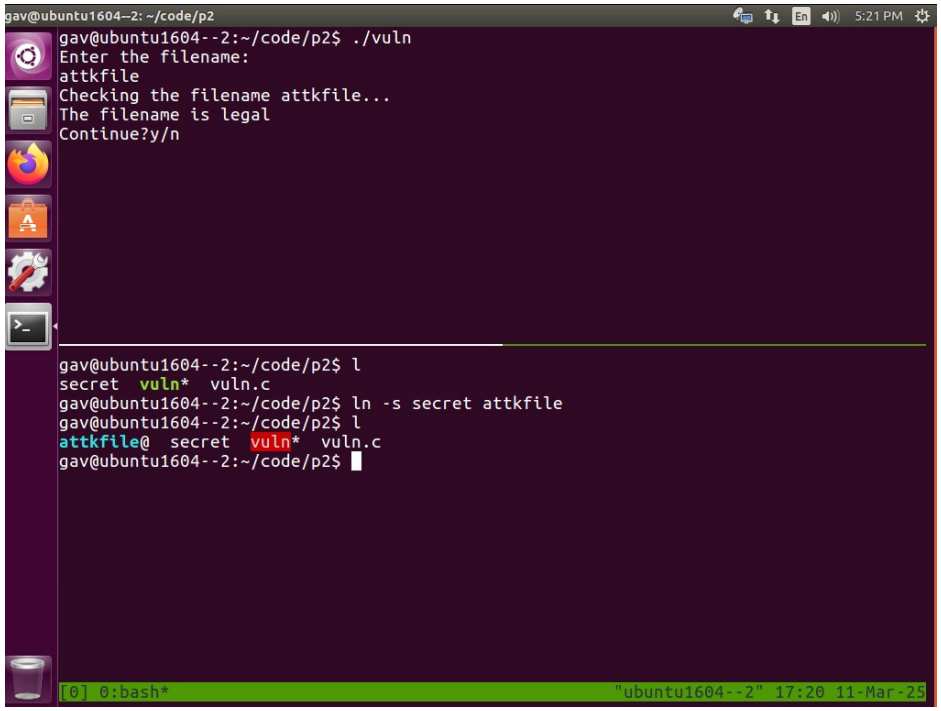
Compile with gcc -o vuln vuln.c and creation of secret file: sudo vi secret and input some fake data:



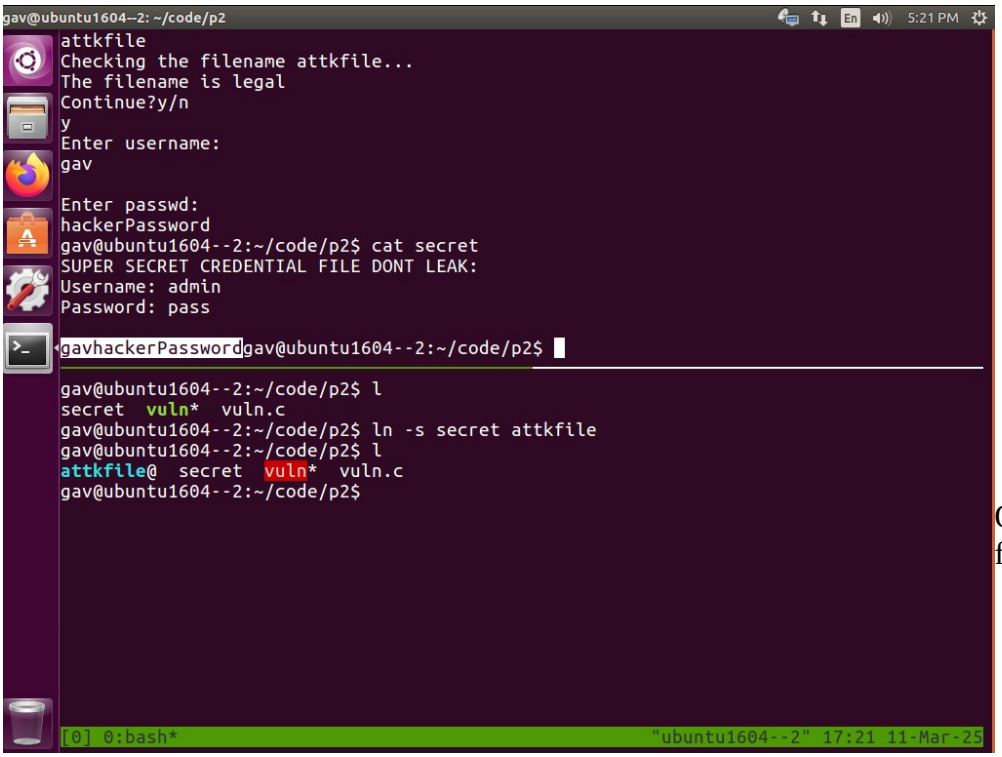
Then to ensure vuln binary has root privilege, change the owner group to root with sudo chown root:root vuln then we can set the SUID bit by doing sudo chmod u+s vuln:



After setting up the two files we can begin the attack. First by executing the program with ./vuln which prompts the user for a filename, we input ‘attkfile’ and the program will CHECK if inputted file is “secret”. Immediately after this we must link ‘attkfile’ to secret BEFORE the program creates ‘attkfile’ as an actual file, we can do this by creating a soft link with ln -s secret attkfile:



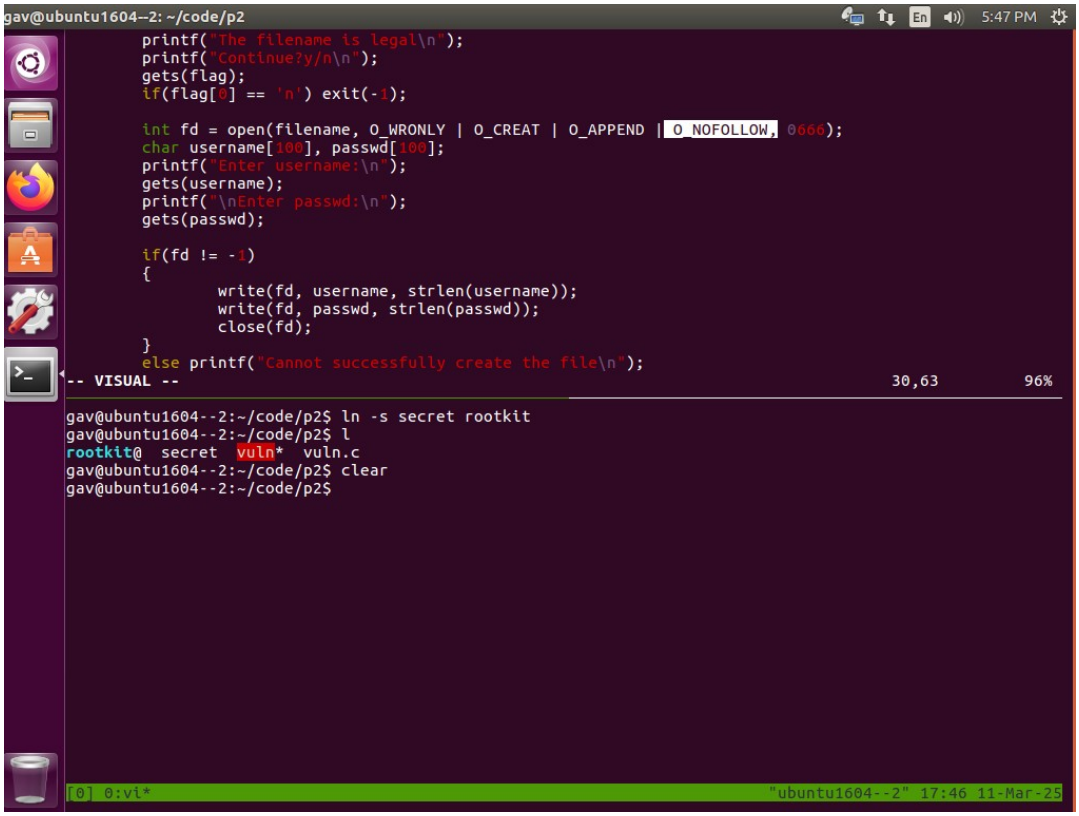
Then we can input ‘y’ to continue the program. NOTE before pressing ‘y’ we have our race condition, meaning we must create a link before the program creates/uses the file. Now we enter the username and password which will be written to attkfile which is linked to “secret” so subsequently it will be written to the “secret” file!



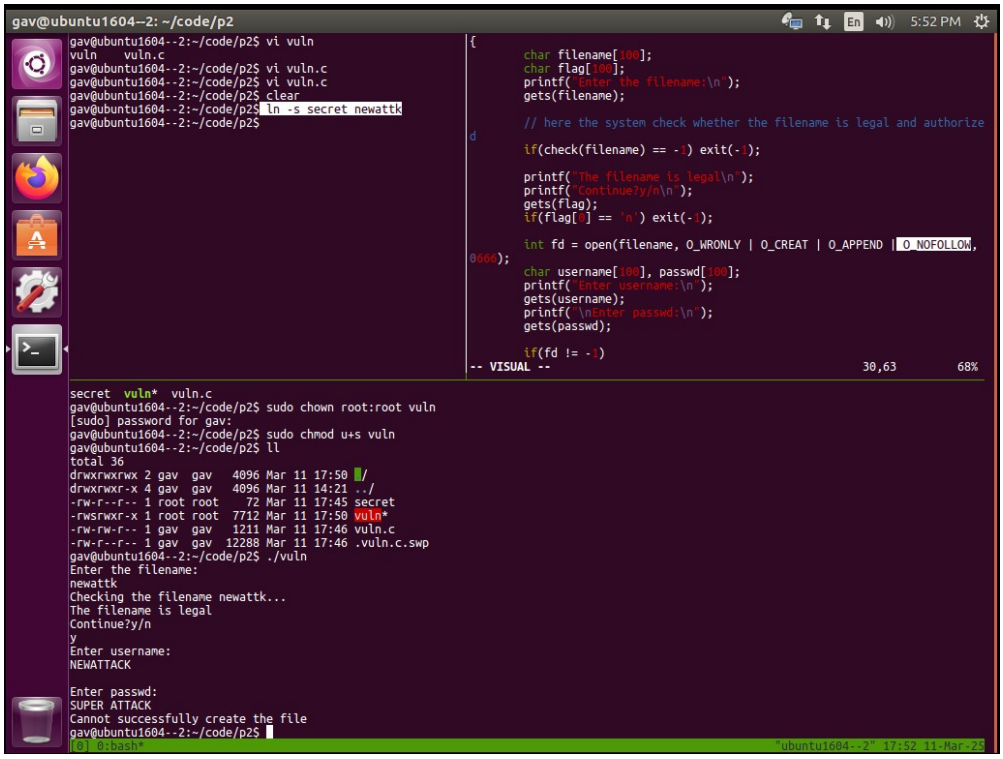
O_
f

Defense Against TOCTOU:

To defend against this type of race condition attack we should check whether or not the file we are opening is a symbolic link to another file, effectively not allowing connections to sensitive files. This can be achieved by adding the option O_NOFOLLOW to our open() call.



To ensure this works I followed the process above with the new setting added to open() which as shown below wont allow the file to be used:



After running with the error I removed the O_NOFOLLOW option and re-tried the attack to ensure the setting was the reason we stopped the attack.