a1-milestone documentation
Boris Kravchenko
bkravche
20332359

sploit1.c

Vulnerability: The vulnerability exploited is the buffer overflow vulnerability *(submit.c 36-41).* There is no check for whether or not the source file is larger than the buffer, therefore a file larger than 512 characters will overflow the buffer.

Exploit: My program creates a file for submission that is 612 characters long, and it pads the first half of the file with noops, puts the shellcode in the middle, and the rest is filled with the return address. The submit command then overwrites it's own stack upon reading this file. The return address is overwritten with an address that's in the noop region of the buffer, therefore control is transferred to the code inside the buffer upon function return, and the shellcode is executed.

Repair: This exploit could be repaired by adding a simple check in *(submit.c 36-41)* to ensure the source file isn't larger than the buffer.

sploit2.c

Vulnerability: The vulnerability exploited is the TOCTTOU vulnerability *(submit.c 126 - 154)*. It's possible to change the symlink to the log file between check and write, which allows me to append a string to any file in filesystem (since submit has root privileges).

Exploit: My program creates a symlink between /home/user/submit.log and a file my user has permission to write to. Then my process forks and runs the submit command, with the following message "hacker::0:0:hacker::/bin/bash", which, if appended to /etc/passwd, creates a new root user. While submit is running, my process changes the symlink of /home/user/submit.log to /etc/passwd. My exploit keeps retrying the above steps until success (the passwd file gets changed). Upon success, it spawns a new shell with the hacker account.

Repair: This exploit could be repaired by using file descriptors instead of file paths, and by locking the log file during the execution of submit.