CSU0033 Operating Systems, Homework 5

Department of Computer Science and Information Engineering
National Taiwan Normal University

May 22, 2025

The purpose of this homework assignment is to guide you through a series of steps that is typical in science/engineering discovery. You will begin with some observation of certain phenomenon, make some hypothesis, then figure out a way to re-generate such phenomenon in a repeatable fashion, to confirm with your hypothesis. Then after some study and analysis, you will create some solution that can handle the phenomenon. Finally, you will try to exploit the system, to seek how you could do some interesting things with a maybe-not-so-interesting system. Along the way, you will also learn more about the file system and will sharpen your hands-on skill.

As always, you are encouraged to ask and help answer questions on Moodle. If you used AI to help you in answering any of the questions, tell us in which specific ways you used it. Knowing that you used AI will not affect our grading. It is just that we would like to know how you used it.

Study Chapter 8 in the xv6 book, by which you will learn many valuable information that will help you tackle this homework.

Contents

1 Inode leak and its kernel remedy (50 points)

Watch this demo of an *inode leak*: https://youtu.be/fd_sIBMAGzI. Just like memory leak and space leak, inode leak will prevent a system from fully utilize its resource.

Task-1 (30 points) Create your user-space *ileak.c* program that will cause an inode leak if the system crashes before the program ends. For this task, you are not allowed to modify the kernel nor create new system calls. Write a user-space program which uses the default system calls. Submit both your code and a screen recording of your demo showing the occurrence of inode leak (i.e., the newly created file did not use the smallest inode number not shown in *ls*). For your interest, take a look at xv6's implementation of *ls* in *user/ls.c*. 15 points for your working code *ileak.c*, and 15 points for your video.

Task-2 (20 points) Now, implementation a solution that fixes the inode leak after system reboot. Section 8.9 in the xv6 book sketches two possible solutions. You can implement either one, or you can try to come up with your own solution. The solution must be implemented in the kernel. Submit both (1) your solution as a kernel patch, with respect to the original kernel version that you downloaded in the beginning of the semester, and (2) a video showing that your *ileak.c* will no longer cause inode leak after you patched your system with your solution. 10 points for each submission.

2 Information leak, a user-level remedy, and a little hack (50 points)

Now, watch the following video, which shows you that in some cases, people may still get data from a deleted file: https://youtu.be/BMgGr2CAAVk. Note that in this case, the information leaks not from memory but from disk.

Task-1 (10 points) Explain what could cause the *grep* show two pattern matches for the information in your file. Note that the issue is not in *grep* but in the file system design.

Task-2 (10 points) Explain what could cause the *grep* still show a pattern match even after we've deleted the file via *rm*. Again, the issue is not in *rm* but in the file system design.

Task-3 (10 points) Explain why, as in the video, the *grep* only found one match of the pattern NTNU. Compare it with what you've observed in Task-1.

Task-4 (10 points) Now, while we may implement a kernel-space solution to prevent such information leak, from a user's perspective we will want some user-space solution. Because, for example, we might not have access to the kernel code, nor may we have the privilege to recompile the kernel. In this part of the assignment, you are asked to figure out a way to prevent information leak from a deleted file. Your solution must not involve any coding. Submit a video to demonstrate that your solution will really clean

the information of the deleted file from your *fs.img*, so no tool will be able to get the information back.

Task-5 (10 points) Now, put on a hat as a curious hacker. Suppose there is a user who did not know the solution you've found in Task-4, and suppose that user indeed has had put a password in a file with pattern `MyPass=******`, where `******` is a six-digit password (for example, a birth date 061208), but the user has deleted the file already. Now, given the user's *fs.img* and the fact that the password is six-digit long, use `MyPass=` as a clue to get back that six-digit password. Submit a video recording that shows how you set up the stage for this hacking and how you can get the password. For this task, you are allowed to write some code or use some tools to achieve the goal of password-hunting.