Reversing Challenge: Package Mismanager

F&A

FROG & ASSOCIATES

You are a team of new hires at Frog & Associates, a management-focused organization. With the onboarding of a visionary and competitive CEO, the company is expanding its horizons to become the sole leader in providing management solutions for ALL situations that require managing.

Recently, your new supervisors discovered a forgotten and incomplete package managing program for Linux-based operating systems that was written by a long retired developer as a side project. Some cryptic notes about the functionality of the program were found as well. However, the source code itself is lost, the software definitely contains serious bugs, its operations are password protected, and, most importantly, the original developer is not interested in coming out of retirement to help out in any capacity.

Your managers expressed interest in recovering the source code so the program could continue to be developed and become a final product that could be sold and eventually make Frog & Associates a household name. Your team immediately accepts this challenge - you all are nearing the end of a course on reverse engineering, after all.

Management and the other developers worked together to compile these comments about the program's behavior, information about the previous developer, and usable details from the former developer's notes that may be helpful during the reversing process:

"The program was compiled so that it only works when it is extracted to ***/opt/pkgmis****. For instance, the binary path should be at* ***/opt/pkgmis/bin/pkgmis****."*

“This program depends on libalpm. We found the source code for this Highly® Proprietary™ library, but it should not be necessary to solve our issues. Use the header file at ***/opt/pkgmis/include/alpm.h*** *to get an idea of the API and functions available.”*

“The executable is very large, but the program does not seem to do a lot of things.”

“We do not know the password for the program. It was likely something common though, since the former developer notoriously had a poor memory.”

“We have some actual source code from another project the developer was working on before they retired. We can see that they liked to leave comments in the code that would provide some explanation for its functionality. However, a lot of the code was missing these comments.”

“The program uses a ton of memory when it runs. This is confusing since again, it does not seem to have much functionality.”

“The program seems to get hung up on certain functions. We don’t think this is a problem with the system that the executable is running on, but we can’t be sure.”

“The executable gets a segmentation fault when the sync-refresh command is run. We don’t have any idea why this is happening.”

“We keep getting an ‘unable to lock database’ message and found that we can get it to work by deleting the file at ***/opt/pkgmis/var/lib/pkgmis/db.lck****. It seems like this has to be done after almost every command.”*

Your team must reverse, reconstruct, and identify (and correct if possible!) any problems you find in this application for Frog & Associates. Good luck!