SEMESTER PROJECT PART 2: REVERSING THE BINARY SCANNER

Due: 11:59pm 12/3

DUE DATE:

Your completed report and supporting idb are due at 11:59pm on 12/3.

LATE POLICY:

10% per day. We will not be generous with the late policy for this part of the project. Please do not ask for extensions.

FILES:

A file named 'distro.zip' will be placed in your home directory (/home/<your login>/distro.zip) on class VM 2 (enee459b-2.ece.umd.edu). We will make an announcement when they are available. Inside the zip file contains the binary scanner we've selected for you to reverse engineer. All files were built to run on the VMs. Segfaults are normal! Part of the challenge will be reversing the program to get it to behave properly and not segfault. Some students will be reversing the same software so there is ABSOLUTELY no collaboration allowed.

PROJECT DESCRIPTION:

This is the fun part of the semester project (we hope!) You will be reversing one of your fellow students' binary scanner in an effort to document its usage, operation, vulnerabilities, etc.

Capture your findings in a written report (related to goals below). Take screen snippets or screenshots of key discoveries in IDA. Put pseudo-code in appendices for reference. Tell us about any structures/global data you discover. If you wrote code to assist in your analysis, include it in your report. If you are thorough in documenting your findings, we expect you to have no problem reaching 8-10 pages in this report.

GOALS:

The extent of the reversing that you can perform on the binary scanner is largely open ended. There are however several basic goals that we want you to focus on before tackling harder problems:

1. Document how the software operates. We didn't include the User Guide because what fun would that be? Try to extrapolate all of the use cases. How does it work? Does it rely on any special files to be present? What is the normal operation? Are there any 'secret' or administrator modes? How does it fail? This should be your first step in reversing. This is 'Black Box' reversing where you learn as much as you can about the program before loading it up in IDA Pro. Resist the temptation! The more you can learn just by interacting with the program, the less you'll have to discover the hard way in IDA Pro. Remember to use binutils (nm, readelf, strings) and gdb. Reversing is much more than IDA Pro!

2. How does the binary authenticate users for reading analyzed data (if at all)? Can this be defeated?

3. What information about the ELF is parsed, stored, and retrieved?

4. Is the parsed information correct? (i.e. is the program parsing the file correctly)? Does the program parse all possible values for the binary data?

5. Document how the binary scanner stores information about parsed ELF files. What is the format of the information on disk? Recreate a structure to describe each record. **Hint**: Try patching your binaries to bypass obfuscation methods to better understand the storage format (remember to save a backup!)

6. What obfuscation is used and where? How is the information on disk obfuscated? Are usernames/passwords obfuscated?

7. Document how the software is vulnerable, and what can be done (if anything) to secure/protect it. Vulnerabilities can take many forms: intentional backdoors, breakable encryption/obfuscation, stack overflows, etc.

8. Document your reversing findings in a .idb file. This is required as part of your submission and will be graded. You must show a) naming of functions, b) naming of local variables and arguments, and c) comments in-line describing what the code is doing.

OTHER POINTS OF INTEREST:

1. You must submit .idb files for the binary scanner to showing your reversing progress.
2. In your report, tell us not only the findings that you achieved, but how you came to those conclusions.

HAPPY REVERSING!