## Summary

The goal of this laboratory exercise is to obtain hands-on practice with using the Ghidra application. We will be using Ghidra to analyze the famous Carnegie-Mellon “bomb” binary. This is a Linux executable binary (built using gcc and C-language source code) that consists of six “phases.” Each phase expects students to input a particular string on stdin. If the expected string is entered, then that phase is “defused.” Otherwise, the bomb “explodes” by printing “**BOOM!!!**”. The goal is to defuse all the phases. We will analyze the binary using Ghidra to reverse engineer the expected strings for each of the six phases. I have made your task easier by rebuilding the binary as a 32-bit \*.elf, with the optimizer disabled and debugging information included.

Prerequisites

* Please group into 3-4 students so we don’t overload the network
* Install Ghidra and latest Java SE JDK per the instructions on Blackboard in your Ubuntu VM

## Details

1. Download the bomb binary from the GitLab server
2. Start Ghidra ($ ghidra)
3. Import the bomb binary in Ghidra
4. View the strings
5. View the functions
6. Locate the main function
7. Navigate the control flow using the graph views
8. Locate the first phase and identify where the bomb explodes
   1. Work backwards from there!
9. Once you have solved this phase, locate the next phase and repeat the process until you have successfully defused all the phases

## References and useful resources

* For obtaining the bomb binary, I will email the file.
* Bryant and O’Hallaron, Carnegie Mellon University, <http://csapp.cs.cmu.edu/2e/labs.html>