Beyond errno: Error Handling in C

David Svoboda

SEI | CMU

This tutorial covers error handling in C. In many modern languages the recommended mechanism for error handling is exceptions, but C does not support exceptions. Due to its age, C has acquired a plethora of technologies for handling errors. These range from return codes, errno, and abort() to more esoteric techniques like goto chains, setjmp()/longjmp(), and runtime constraint handlers. With the myriad techniques for handling errors, how then is a C programmer expected to handle errors securely? In this tutorial, we will examine the technologies available to the C developer for handling errors. Each technique has its own pros and cons, and we will examine correct and incorrect uses of each, with an eye towards software security and maintainability. We will briefly touch on C++ exceptions, but we intend to focus on platforms where exceptions are not available. The audience is expected to have a beginner's understanding of C, and they will be given small code examples to understand throughout the tutorial in order to increase their understanding. They can expect to become aware of the most common error-handling technologies widely used and employed by C programmers.

In this tutorial you will learn the various techniques of handling errors in C. These range from return codes, errno, and abort() to more esoteric techniques like goto chains, setjmp()/longjmp(), and runtime constraint handlers. We will briefly touch on C++ exceptions, but we intend to focus on platforms where exceptions are not available. You are expected to possess at least a beginner's understanding of C and be able to understand simple C code examples.

This tutorial is planned as a slide presentation and discussion. The audience is expected to participate in discussions about techniques, and answer questions like “What does this code do?” or “How could an attacker make this program crash?”.