Introduction

Extensible programming languages were first discussed in the 1960s. As originally discussed in Douglas McIlroy's 1960 paper, the original concept was to use a small number of macros in order to extend a compiler to accept very general extensions to the original language accepted by the compiler (CITE McIlroy's paper). This has been modernized in various ways, including the idea of extensible syntax, compilers, and even extensible runtimes (CITE? wiki has dead link).

One example of an extensible programming language is `ABLE`C (cite), an extensible version of the C programming language, built on top of the `Silver` (cite) attribute grammar system. Various extensions exist for `ABLE`C that were created to solve various problems (CITE paper containing extensions from MELT).

This paper examines the uses of extensible programming languages in a real-world application: an Internet Relay Chat (IRC) server implementation. The IRC protocol was originally designed for text-based conferencing (cite IRC RFC), and many server implementations are available. For this paper, we consider the specific implementation of `ngIRCd` (next-generation IRC daemon), written in the C programming language. This open-source server allows for end users to clone the server and modify it to suit their specific purposes. These modifications, however, open the server up to various security vulnerabilities. The C programming language, while fast, does not provide many security checks for the programmer. We desire a way of allowing users to modify the server without introducing any security vulnerabilities.

Modifying this C code provides a natural use for the `ABLE`C language. By introducing both new and existing extensions to the `ngIRCd` server, we are able to both eliminate certain existing vulnerabilities in the server that are innate to the C programming language, but also we can make certain aspects of the server easier to implement and understand for end users, as well.