# Code Safety: memory errors in C

**Description** Illustrate how SAW is used to show code weakness in C programs due to memory failures. Considered are incorrectly dereferencing an object, being able to access private data that should not be accessible, changing data on the stack, heap overflow, format errors.

**Purpose** Getting more familiar with constructs that can be used in SAW scripts and how they are used.

**Audience** This module is intended for:

1 The general public
2 K-12 and college classes on Cyber Defense and Math Logic
3 preparation for proficiency in the use of tools and a computing environment suitable for the study of cyber defense

**Objectives** After completing the module:

1 You will know how to run clang, the C language compiler to llvm
2 You will know how to write a llvm specification intended for a SAW script
3 You will be able to write Cryptol specs enabling safety checks on C code

**Keywords** Cryptol, SAW, Yices, ABC, Z3, CVC4, Boolector, stdint.h, primitive data types, buffer overflow, stack overflow, heap overflow, dangling pointer, dereferencing.

**Category** cybersecurity > education

**Delivery** java applets and written documentation in pdf format

**Team** John Franco and Ethan Link

**Assessment** The applets provide the means for experimentation. Questions are asked in the documentation that help with the set up of experiments. The ideas that learners come up with is evidence that the module was successful.

**Workflow** No particular schedule was established

**Environment** All materials are contained in a single jar file. The jar file can be run on any computer where java version 11 or higher and some pdf reader such as acroread or evince are available. The jar file may be executed in the cyber range or learners may download the jar file (which is considered to be an executable file) and run it on their personal computers.