Requirements for dissector generator

Introduction

This document contains requirements for a utility that allows Wireshark to interpret the binary representations of C-language structs.

While C structs seldom are exchanged across networks, they are sometimes used in inter-process communication. The purpose of the utility described here is to provide Wireshark with the capability of automatically dissecting the binary representation of a C struct, as long as its definition is known. This would provide Wireshark with one of the components necessary to analyze e.g. IPC communication.

The part of Wireshark responsible for decoding data are the dissectors. The easiest way to extend Wireshark with new dissectors is to implement them as Lua scripts. This allows new dissectors to be added without recompiling Wireshark.

The expected work flow for the utility is to read one or more C header files with struct definitions, and output dissectors for these. A configuration file or source scode annotations in the header files may be used when additional configuration is required.

Requirements

REQ-001 The utility shall be able to read basic C language struct definitions, and generate a Wireshark dissector for the binary representation of the structs.

REQ-002 The utility shall support structs with any of the basic data types (e.g. int, boolean, float, char) and structs.

REQ-003 The utility shall be able to follow #include <...> statements. This allows parsing structs that depend on structs or defines from other header files.

REQ-004 Each struct may be connected to one or more references (integer value). For instance, a member parameter 'type' can have names for a set of values.

REQ-005 The dissector shall be able to recognize invalid values for a struct member. Allowed ranges should be specified by configuration. An example is an integer that indictates a percentage between 0 and 100.

REQ-006 A struct may have a header and/or trailer (other registered protocol). This must be configurable.

REQ-007 The dissector shall be able to display each struct member. Structs within structs shall also be dissected and displayed.

REQ-008 It shall be possible to configure special handling of specific data types. E.g. a 'time_t' may be interpreted to contain a unixtime value, and be displayed as a date.

REQ-009 An integer member may indicate that a variable number of other structs (array of structs) are following the current struct.

REQ-010 Integers may be an enumerated named value or a bit string.

REQ-011 The dissectors produced shall be able to handle binary input from at least Windows 32bit and 64bit, Solaris 64bit and Sparc. Example: BOOL is 1 byte on Solaris and 4 bytes on Win32. Endian and alignment also differs between the architectures.