

ECLA Inc.

Regular Expression Compiler RegExComp

The Regular Expression Compiler (RegExComp) compiles a set of regular expression into either a binary image or into C-code for a Deterministic State Machine. The binary image can be executed either by the CI FPGA hardware device or by the CI Software Emulator. The C-code can be compiled with a C Compiler and executed on a target system.

RegExComp state machines can recognize any byte sequences as defined by regular expressions.

Features

- Support PERL Regular Expression operators (See the table below)
- Accepts as input a text file with all the regular expressions. The input regular expressions can be separated in one or multiple categories.
- A category could be port number, an IP address, an application, a type of exploit or any other user defined category.
- The compiler is completely agnostic at what the category is.
- Each category can contain a large number of regular expressions.
- Generates an optimized Deterministic State Machine per input category usable by hardware and software engines.
- Compiler output
 - A file with the state machines encoded in binary for the CIP.
 - An optional file with the state machines as C code to be compiled by a C compiler and run.
 - An optional statistics file as well as a visualize dot file.

Supported Regular Expression Grammar

Symbol	Name	Description
None	Concatenation	Stringing together characters
	Alternation	OR operation
()	Grouping	Parentheses to group elements together
+	+	1 or more repetitions of the previous construct
{ <i>min, max</i> }	{ <i>min, max</i> }	Minimum of <i>min</i> , maximum of <i>max</i> repetitions of the previous construct. If

© 2008-2009 ECLA Inc.

www.ecla.com

sales@ecla.com

		only one argument exists between the parenthesis it is interpreted as a minimum
.	Dot	Matches any character. Note: Different from Perl. A newline character is treated as a standard character and is included in the match results. In Perl, the '.' does not include a newline.
\$	\$	End of line - Anchors to the newline character. Note: Different from Perl. A newline character is treated as a standard character and is included in the match results. In Perl, the newline character is matched but not reported in the results
^	^	Beginning of line – Anchors to the newline character. Note: Different from Perl. A newline character is treated as a standard character and is included in the match results. In Perl, the newline character is matched but not reported in the results.
\n		Newline character
\t		Tab
\r		Carriage Return
\d		Any of the ten decimal digits
\D		Any character other than the ten decimal digits
\s		Any character in the set {' ', '\t', '\n', '\r', '\f'}, where '\f' is a form-feed
\S		Any character <i>not</i> in the set defined by \s
\w		Any character in the ranges a-z, A-Z and 0-9 as well as the underscore () character
\W		Any character not in the set defined by \w
[...]		Positive set - elements in the set are either individual characters or character ranges, e.g., [a-zA-Z0-9]
[^...]		Negative set – Negations of the positive set (containing all characters not in the positive set), e.g. [^a-zA-Z0-9]