

---

# Tranalyzer2

regex\_pcre



Perl Compatible Regular Expressions (PCRE)

---



Tranalyzer Development Team

Contents

<b>1</b>	<b>regex_pcre</b>	<b>1</b>
1.1	Description . . . . .	1
1.2	Dependencies . . . . .	1
1.3	Configuration Flags . . . . .	1
1.4	Flow File Output . . . . .	5
1.5	Packet File Output . . . . .	5
1.6	Plugin Report Output . . . . .	5

# 1 regex\_pcre

## 1.1 Description

The `regex_pcre` plugin provides a full PCRE compatible regex engine.

## 1.2 Dependencies

### 1.2.1 External Libraries

This plugin depends on the **pcre** library.

<b>Ubuntu:</b>	<code>sudo apt-get install</code>	<code>libpcre3-dev</code>
<b>Arch:</b>	<code>sudo pacman -S</code>	<code>pcre pcre2</code>
<b>openSUSE:</b>	<code>sudo zypper install</code>	<code>pcre-devel</code>
<b>Red Hat/Fedora<sup>1</sup>:</b>	<code>sudo dnf install</code>	<code>pcre-devel</code>
<b>macOS<sup>2</sup>:</b>	<code>brew install</code>	<code>pcre</code>

### 1.2.2 Required Files

The file `regexfile.txt` is required (automatically generated from `scripts/regexfile.txt`). Refer to Section 1.3.4 for more details.

## 1.3 Configuration Flags

### 1.3.1 regfile\_pcre.h

The compiler constants in `regfile_pcre.h` control the pre-processing and compilation of the rule sets supplied in the regex file during the initialization phase of Tranalyzer.

Name	Default	Description	Flags
<code>RULE_OPTIMIZE</code>	1	0: No opt rules allocated 1: Allocate opt rule structure and compile regex	
<code>REGEX_MODE</code>	<code>PCRE_DOTALL</code>	Regex compile time options	
<code>PREIDMX</code>	4	Max number of node predecessors	

### 1.3.2 regex\_pcre.h

The compiler constants in `regex_pcre.h` control the execution and the output the rule matches.

Variable	Default	Description	Flags
<code>EXPERTMODE</code>	0	0: Alarm with highest severity: class type and severity, 1: full info	

---

<sup>1</sup>If the `dnf` command could not be found, try with `yum` instead

<sup>2</sup>Brew is a packet manager for macOS that can be found here: <https://brew.sh>

Variable	Default	Description	Flags
PKTTIME	0	0: no time, 1: timestamp when rule matched	
AGGR	0	1: Aggregate alarms	
SALRMFLG	0	1: enable sending FL_ALARM for pcapd	
MAXREGPOS	30	Maximal # of matches stored / flow	
RGX_POSIX_FILE	"regexfile.txt"	Name of regex file under <i>/tralyzer/plugins</i>	
OVECCOUNT	3	Value % 3	

### 1.3.3 Environment Variable Configuration Flags

The following configuration flags can also be configured with environment variables (ENVCTRL>0):

- RGX\_POSIX\_FILE

### 1.3.4 regexfile.txt

The *scripts/regexfile.txt* file has the following format:

#ID	PreID	Flags	ClassID	Severity	Sel	Regexmode	FlwStat	Proto	srcPort
	dstPort	offset	Regex						
# standalone rule: Alarm, start L7, Regexmode: default, select FlwStat: Req; Proto, dstPort									
1	0	0x10	15	3	0x8000000d	0x00000000	0x00000000	6	0
	80	0	(OPTIONS GET HEAD POST PUT DELETE TRACE CONNECT) [^\r\n]*\u7avi*\bin						
# standalone rule: Alarm, disabled, start L7, select Regexmode: (PCRE_CASELESS PCRE_DOTALL),									
FlwStat: Teredo, IPv6, Vlan, Repl; Proto, srcPort									
3	0	0x10	15	3	0x0800000e	0x00000005	0x00088101	6	80
	0	0	\x31\xDB\x8D\x43\x0D\xCD\x80\x66.*\x31						
# standalone rule: Alarm, start L7, Regexmode: default, FlwStat: IPv4, Rply									
4	0	0x10	15	3	0x8000000c	0x00000000	0x00004001	6	80
	0	20	\x38\x55\x42\x66\xe2\xb5\x34.*\xb5\x95\xbb						
# standalone rule, Alarm, start L7, select Regexmode: (PCRE_CASELESS PCRE_DOTALL)									
100	0	0x10	1	0	0x88000000	0x00000005	0x00000000	6	0
	80	0	^http/1.0						
# root rules to following tree, Reset if leaf fires									
202	0	0x40	10	4	0x80000000	0x00000000	0x00000001	6	0
	80	0	(GET PUT).*update/u7avi177u1705ff.bin						
203	202,4	0x41	20	4	0x88000000	0x00000005	0x00000001	6	0
	80	0	302 (?i)Found						
# successors and predecessors, Reset if leaf fires									
204	202,203	0x41	43	5	0x80000000	0x00000000	0x00000001	6	0
	21	0	(?i)\.exe						
# successors 206 & 205 to 204 AND ruleset, don't reset tree if 205 fires									
205	204	0x16	40	4	0x80000002	0x00000000	0x00000000	6	0
	20	0	^get .*porno.*						
206	204	0x56	35	6	0x8000000c	0x00000000	0x00000001	6	0
	21	0	igfxzoom\.exe						

Lines starting with a '#' denote a comment line and will be ignored. All kind of rule trees can be formed using rules also acting on multiple packets using different ID's and Predecessor as outlined in the example above. Regex rules with the same ID denote combined predecessors to other rules. Default is an OR operation unless ANDPin bits are set. These bits denote the different inputs to a bitwise AND. The output is then provided to the successor rule which compares with the ANDMask bit field whether all necessary rules are matched. Then an evaluation of the successor rule can take place. Thus, arbitrary rule trees can be constructed and results of predecessors can be used for multiple successor rules. The variable Flags controls the basic PCRE rule interpretation and the flow alarm production (see the table below), e.g. only if bit eight is set and alarm flow output is produced. ClassID and Severity denote information being printed in the flow file if the rule fires.

Flags	Description
2 <sup>0</sup> (=0x01)	Predecessor OPS
2 <sup>1</sup> (=0x02)	Predecessor OPS
2 <sup>2</sup> (=0x04)	Leaf
2 <sup>3</sup> (=0x08)	—
2 <sup>4</sup> (=0x10)	Print alarm to flow file
2 <sup>5</sup> (=0x20)	Rule active only in flow boundary
2 <sup>6</sup> (=0x40)	Reset REG_F_MTCH tree if match
2 <sup>7</sup> (=0x80)	Internal: Regex match

Predecessor OPS	OP	Description
0x00	NONE	None, solitary rule
0x01	AND	and(pred1, pred2, ...)
0x02	OR	or(pred1, pred2, ...)
0x03	XOR	xor(pred1, pred2, ...)

The Sel column controls the header selection of a rule in the lower nibble and the start of regex evaluation in the higher nibble. The position of the bits in the control byte are outlined below:

Sel	Description
2 <sup>0</sup> (=0x00000001)	Activate srcPort field
2 <sup>1</sup> (=0x00000002)	Activate dstPort field
2 <sup>2</sup> (=0x00000004)	Activate L4Proto field
2 <sup>3</sup> (=0x00000008)	Activate flowStat field
2 <sup>27</sup> (=0x08000000)	PCRE mode active; otherwise default
2 <sup>28</sup> (=0x10000000)	Header start: Layer 2
2 <sup>29</sup> (=0x20000000)	Header start: Layer 3
2 <sup>30</sup> (=0x40000000)	Header start: Layer 4
2 <sup>31</sup> (=0x80000000)	Header start: Layer 7

Bit 0 - 27 selects the first 32 bit of flowStat, the protocol, source and destination port will be evaluated per rule, all others will be ignored. The flowStat field might contain other bits meaning more selection options in future. The

offset column depicts the start of the regex evaluation from the selected header start, default value 0. The `Regex` column accepts a full PCRE regex term. If the regex is not correct, the rule will be discarded displaying an error message in the Tranalyzer report.

The `regexmode` column denotes the mode of regex compilation and execution, listed below. If **0x00000000** then the default defined by `REGEX_MODE` is used.

<b>regexmode</b>	<b>Name</b>	<b>Description</b>
2 <sup>0</sup> (=0x00000001)	PCRE_CASELESS	Compile
2 <sup>1</sup> (=0x00000002)	PCRE_MULTILINE	Compile
2 <sup>2</sup> (=0x00000004)	PCRE_DOTALL	Compile
2 <sup>3</sup> (=0x00000008)	PCRE_EXTENDED	Compile
2 <sup>4</sup> (=0x00000010)	PCRE_ANCHORED	Compile, DFA exec
2 <sup>5</sup> (=0x00000020)	PCRE_DOLLAR_ENDONLY	Compile
2 <sup>6</sup> (=0x00000040)	PCRE_EXTRA	Compile
2 <sup>7</sup> (=0x00000080)	PCRE_NOTBOL	Exec, DFA exec
2 <sup>8</sup> (=0x00000100)	PCRE_NOTEOL	Exec, DFA exec
2 <sup>9</sup> (=0x00000200)	PCRE_UNGREEDY	Compile
2 <sup>10</sup> (=0x00000400)	PCRE_NOTEMPTY	Exec, DFA exec
2 <sup>11</sup> (=0x00000800)	PCRE_UTF8	Compile
2 <sup>12</sup> (=0x00001000)	PCRE_NO_AUTO_CAPTURE	Compile
2 <sup>13</sup> (=0x00002000)	PCRE_NO_UTF8_CHECK	Compile, DFA exec
2 <sup>14</sup> (=0x00004000)	PCRE_AUTO_CALLOUT	Compile
2 <sup>15</sup> (=0x00008000)	PCRE_PARTIAL_SOFT	Exec, DFA exec
2 <sup>16</sup> (=0x00010000)	PCRE_DFA_SHORTEST	DFA exec
2 <sup>17</sup> (=0x00020000)	PCRE_DFA_RESTART	DFA exec
2 <sup>18</sup> (=0x00040000)	PCRE_FIRSTLINE	Compile
2 <sup>19</sup> (=0x00080000)	PCRE_DUPNAMES	Compile
2 <sup>20</sup> (=0x00100000)	PCRE_NEWLINE_CR	Compile, DFA exec
2 <sup>21</sup> (=0x00200000)	PCRE_NEWLINE_LF	Compile, DFA exec
2 <sup>22</sup> (=0x00400000)	PCRE_NEWLINE_ANY	Compile, DFA exec
2 <sup>23</sup> (=0x00800000)	PCRE_BSR_ANYCRLF	Compile, DFA exec
2 <sup>24</sup> (=0x01000000)	PCRE_BSR_UNICODE	Compile, DFA exec
2 <sup>25</sup> (=0x02000000)	PCRE_JAVASCRIPT_COMPAT	Compile
2 <sup>26</sup> (=0x04000000)	PCRE_NO_START_OPTIMIZE	Compile, DFA exec
2 <sup>27</sup> (=0x08000000)	PCRE_PARTIAL_HARD	Exec, DFA exec
2 <sup>28</sup> (=0x10000000)	PCRE_NOTEMPTY_ATSTART	Exec, DFA exec
2 <sup>29</sup> (=0x20000000)	PCRE_UCP	Compile

## 1.4 Flow File Output

The regex\_pcre plugin outputs the following columns:

Column name	Type	Description	Flags
rgxCnt	U16	Number of regex alarms	
rgxRID_cType_sev	R(U16_U8_U8)	Regex ID, class type and severity	EXPERTMODE=0

If EXPERTMODE=1, the following columns are displayed:

rgxRID_cType_sev_	R(U16_U8_U8_	Regex ID, class type, severity,	PKTTIME=0
pktN_bPos	U32_U16)	packet number and byte position	
rgxRID_cType_sev_	R(U16_U8_U8_	Regex ID, class type, severity,	PKTTIME=1
pktN_bPos_time	U32_U16_TS)	packet number, byte position and time)	

## 1.5 Packet File Output

In packet mode (-s option), the regex\_pcre plugin outputs the following columns:

Column	Type	Description	Flags
rgxCnt	U16	Number of regex alarms	
rgxRID_cType_sev	R(U16_U8_U8)	Regex ID, class type and severity	

## 1.6 Plugin Report Output

The following information is reported:

- Number of alarms in number of flows with max severity