TeleScope-CF XML Content Filtering Engine Library Language Specification

Kirill Belyaev, Palo Alto, CA, USA, 94306

October 27, 2014

1 Library Description and Query Composition

Current TeleScope-CF Java library implementation employs XML parsing and specific pattern-matching that provides standard logical operator constructs to construct the query over the values of XML elements and attributes applied to the XML message on the fly as it is provided in the form of a String object.

The code base has been adopted from TeleScope CQ XML stream broker code base written in C. C code has been re-factored into Java with minor modifications.

This general-purpose library could be used by any Java applications that are involved in the XML message content filtering. Example application scenarios could be intrusion detection, selective rule engines, targeted database insertions during the ETL process and various business logic scenarios. The library could also be used in XML routers and various web services for XML content filtering where XML is a common message passing format.

The engine library could accept either a single query or a set of queries.

The query could be constructed out of a number (two and more) of simple sub-queries (expressions) connected via either a logical | OR operator or a logical & AND operator.

A query can not have both logical operators in it at the same time. Therefore queries could be composed either of:

- sub-queries chained via | OR logical operators
- sub-queries chained via & AND logical operators

This model provides the ability to perform query decomposition and construct a complex query in the form of individual sub-queries submitted to the engine as a query set. This approach allows to eliminate the need for parentheses () between sub-queries for the sake of query simplification.

For example the query

- "(type = STATUS | ORIGIN = EGP) | (type = UPDATE & MULTI_EXIT_DISC = 100 & SRC_AS = 6447 & SRC_PORT = 4321 & ORIGIN = EGP) | (MULTI_EXIT_DISC

= 10 & SRC_AS = 6447 & SRC_PORT = 4321 & ORIGIN = EGP) | (type = KEEPALIVE & DST_AS >3200)"

will be decomposed into several sub-queries each of which will be presented as a separate query to the engine in the form of a query set:

- type = STATUS | ORIGIN = EGP
- type = UPDATE & MULTI_EXIT_DISC = 100 & SRC_AS = 6447 & SRC_PORT = 4321 & ORIGIN = EGP
- MULTI_EXIT_DISC = 10 & SRC_AS = 6447 & SRC_PORT = 4321 & ORIGIN = EGP
- type = KEEPALIVE & DST_AS >3200

Here we provide the sample use cases of valid and invalid query expressions inputs:

- "type = STATUS | type = UPDATE" - valid query expression chained via a logical | OR operator
- "type = STATUS | type = UPDATE | type = KEEPALIVE" - valid query expression chained via a logical | OR operator
- "type = STATUS & type = UPDATE & SRC_AS = 6447" - valid query expression chained via a logical & AND operator
- "type = UPDATE & MULTI_EXIT_DISC = 100 & SRC_AS = 6447 & SRC_PORT = 4321 & ORIGIN = EGP" - valid query expression chained via a logical & AND operator
- "type = STATUS | type = UPDATE & SRC_AS = 6447" - invalid query expression - usage of both logical AND/OR operators is ambiguous and therefore prohibited in a single query. This query could be submitted to the engine in the form of two separate sub-queries - one with OR operator logic and second with AND operator logic

The TeleScope-CF Library Language operators are presented in the Table 1 with sample use cases:

Equality and negation operators (= and !) could be used in expressions involving both string and integer values and change the semantics of operation depending on operand type. Full text search functionality is not yet incorporated in the current release of the library. A subset of this functionality is included through % substring match operator.

TeleScope-CF provides a set of operators designed specifically for processing network prefixes including CIDR ranges. The operators follow the designated network prefix element (in our example it is the PREFIX attribute within the BGP XML Message)

Table 1: TeleScope-CF Library Language operators

Operator	Description	Example use
=	equality operator	ORIGIN = EGP
!	not-equal operator	SRC_AS ! 6447
<	relational less than operator	MULTI_EXIT_DISC <10
>	relational greater than operator	MULTI_EXIT_DISC >10
&	logical AND	ORIGIN = EGP & value = 0
|	logical OR	ORIGIN = EGP | value = 1
%	substring match	type % AS_S

Table 2: Network prefix (CIDR) operators

Operator	Description	Example use	Semantics
e	exact prefix match operator	PREFIX e 211.64.0.0/8	exactly defined network prefix range.
l	less specific prefix match operator	PREFIX l 211.64.0.0/8	less specific network prefix range.
m	more specific prefix match operator	PREFIX m 211.64.0.0/8	more specific network prefix range.

with the subsequent network range value. These are ordinary English letters having special meaning when used within the expression.

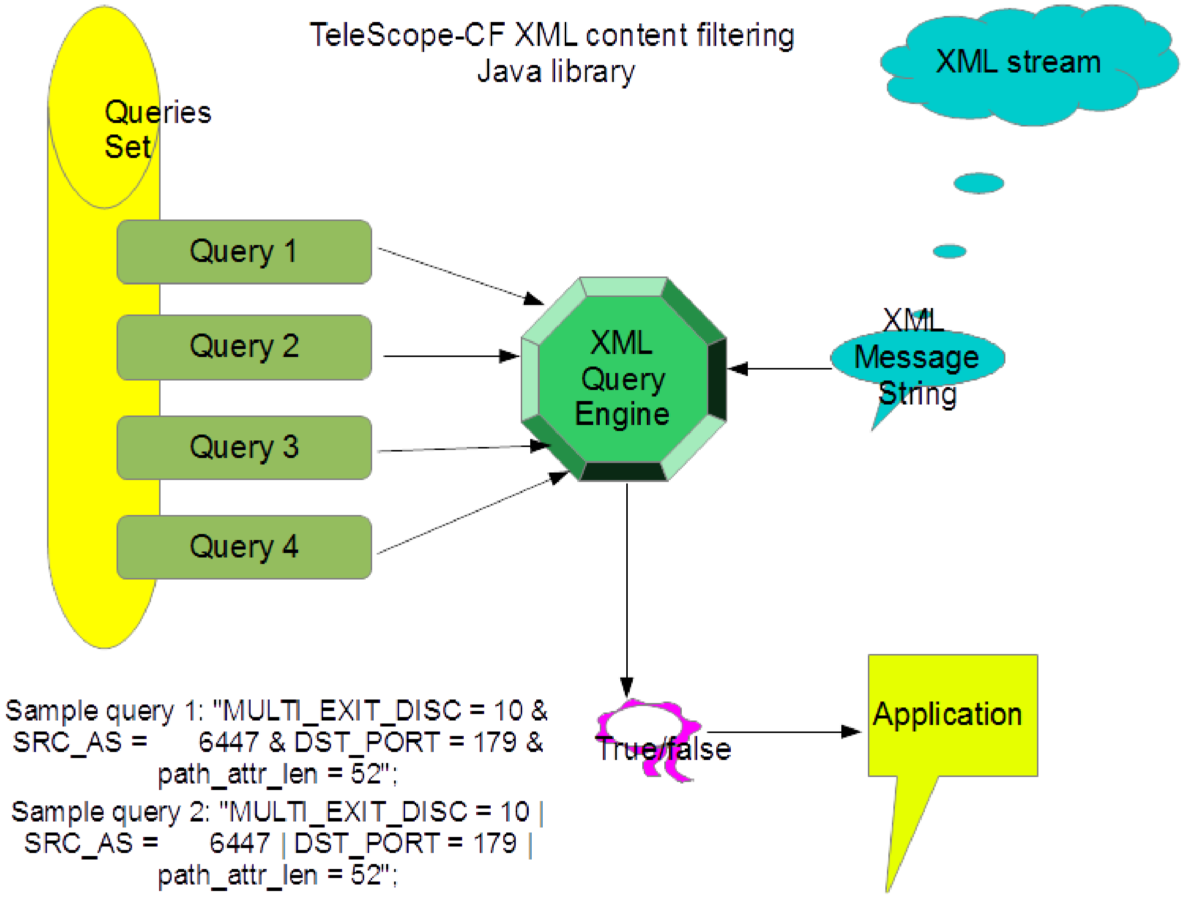
The network prefix processing operators are not included in the current release of the library.

2 Example Java library usage interface

Java library is provided as a jar file that should be imported by the targeted application for XML content filtering:

```
import static iface.Constants.DefaultArraySize;
import implementation.*;
String Query = ”MULTI_EXIT_DISC = 10 & SRC_AS = 6447 & DST_PORT = 179 & path_attr_len = 52”;
Engine engine = new Engine();
engine.addQuery(Query);
boolean result = engine.runQueries(XMLmessage);
System.out.println(”query returned ” + result);
engine.deleteQueries();
```

More then one Query could be added via the addQuery() method for a specified XML message string. Any query that evaluates to true would make engine terminate



Sample query 1: "MULTI_EXIT_DISC = 10 & SRC_AS = 6447 & DST_PORT = 179 & path_attr_len = 52";

Sample query 2: "MULTI_EXIT_DISC = 10 | SRC_AS = 6447 | DST_PORT = 179 | path_attr_len = 52";

Figure 1: TeleScope-CF Library Architecture

and return true. If none of the registered queries evaluates to true the engine will return false to the calling application.