RPSL Dictionary

Craig Labovitz
Merit Network, Inc.

33rd IETF, July 1995

RPSL Dictionary Goals

- Extensibility — suppport for new policy attributes/mechanisms
- Multi-protocol support
- Programming language independent
- Auto-reconfiguration for some tools as library changes

Danvers Proposed RPSL Dictionary

rpsl-attribute: <name> <type> <default-value> <comment>

dictionary: IRR

rpsl-attribute: pref int 0 "preference"

rpsl-attribute: MED int 0 "multi-exit discrimination"

rpsl-attribute: color int0 "color attribute"

rpsl-attribute: AS_PATH_LENGTH int 1 "as_path length"

Problems

- Ambiguity

  What does "community= 10" mean?

- Not enough functionality

  Perform arbitrary function on aspath?

New RPSL Dictionary Proposal

- Object-oriented —attribute classes and methods
- Dictionary has reference implementation and information for syntax checking
- Also can have C,Perl, etc. library implementations

RPSL Dictionary

rpsl-attribute: <attribute-name> [<base classes>] [<protocols>] [<comment>] <methods>

rpsl-attribute: pref "preference" integer bgp,idrp
pref.syntax_check ();
pref.match ();
pref.greater_than ();
pref.set ();

RPSL Dictionary Methods

rpsl-attribute-method: <attribute_object>.<method>
<language> [<comment>] <reference code>

```
rpsl-attribute-method: pref.match perl
"Test if preference matches"
sub pref.match {
  local ($pref) = @_;
  $_pref = get_attribute ("pref");
  return ($_pref == $pref);
}
```


RPSL Libraries

- Dictionary/Library available for anonymous ftp
- Perl,C,C++, etc. libraries with API
- Tools reload/recompile when libraries change
- Each language has internal representation for objects (OOB)

RPSL Usage

- Referenced as object_name.method (arguments)

```
import filters community.matches ("HEPNET") and
    aspath.regexp_match ("[690I179]*")
parameters AS2 AS56
parameters AS78 *
do community.set ("EUROPE"), import;
```


Tools and Dictionary

- Dictionary changes slowly
- Tools may need to recompile when dictionary changes
- Not all tools can use dictionary implementations directly

Possible Tool/Dictionary Interactions

- Prior to execution tools check RPSL for changes
- Whois query when encounteer unknown attribute
- Administrator ftps new libraries when changes announced

Defining Dictionary Attributes

- How are new RPSL terms added?
- Can anyone register attributes? Security problem?
- Local/experimental attributes?

Security

- Allow automatic execution of code?
- "Safe" languages? Safe C?
- Limit who can register
- Initially propose use flat files available from IRR

Sample C++ Attribute Class

```
class aspath_c: attribute_c {
   private
     aspath_t aspath;
   public
     int match_exactly ();
     int contains_as ();
}


clas integer_c : attribute_c  {
   private
     int integer;
   public
     int set ();
}
```