BGP Configuration
Automation on Edge Routers

System and Network Engineering Msc. Research Project
Stella Vouteva & Tarcan Turgut

Supervisor: Stavros Konstantaras, NLNetLabs
Introduction

- Big Internet
  - Depletion of IPv4 addresses
  - Deaggregation
  - 594,000 globally routable prefixes
  - Complex filtering

- Big issues:
  - Misconfiguration
  - Security
  - February 2008 (Pakistan Telekom) - Youtube
  - April 2010 (Chinese Telecommunications) - %15 of the Internet
Motivation

- World peace
- Issues with BGP:
  - BGP is the only protocol, not designed with security
  - A simple, flexible and secure automation required
  - Current solutions might be outdated
Research Questions

- To what extent current technologies can be used efficiently to automate the BGP configuration?
  - What are the existing public tools used to collect BGP policy information?
  - Are those tools reliable enough to provide the necessary information?
  - Do current technologies adapt to the security trends in BGP?
  - What are the limitations of automatic BGP configuration?
Background
Border Gateway Protocol (BGP)

- Inter-AS Routing
- Policy based routing decision
  - Depends on trust between organizations
  - BGP attributes
- Upcoming Security Features
  - BGPsec (not implemented yet)
  - RPKI (in the wild)
Internet Routing Registry (IRR)

- Database of routing policies
- Publicly available
- 34 databases: RIPE, RADB, APNIC etc.
- Only RIPE has authentic data
- Contains RPSL objects
Background
Routing Policy Specification Language (RPSL) 1/4

- Used to specify routing policies in IRR
- Defined in RFC 2622, RFC 2650 (1999)
- RPSLng RFC 4012 (2005)
Aut-num object:
  aut-num: AS1103
  import: from AS3333 accept ANY
  export: to AS3333 announce AS-SURFNET

AS-set object:
  as-set:  AS-SURFNET
  members: AS1101, AS1102, AS1103, AS1104, AS1124 etc.

Route object:
  route:  145.100.0.0/15
  origin: AS1103
IPv6 policies

mp-export: to AS6777

announce { 2001:67c:2e8::/48 }

Route6 object

route6: fc00:600::/32
origin: AS3333
Background - RPSL

```rpsl
import:         {   from AS-ANY     action community .= {3239:1000}; accept ANY;
} refine {
from AS8342   action community .= {3239:201}; accept ANY;
from AS29304  accept NOT ANY;
} refine {
from AS3239:AS-UPSTREAM action pref=25;
accept AS3239:RS-PREF:PeerAS
OR <AS3239:AS-PREF:PeerAS$>;
from AS29648  action pref=26; accept ANY;
from AS3239:AS-UPSTREAM action pref=30; accept ANY;
from AS-ANY action pref=15; accept AS3239:RS-PREF:PeerAS;
accept (<AS3239:AS-IN:PeerAS$> OR <PeerAS$>)
AND NOT { 0.0.0.0/0 }; 
}
```
Background
Resource Public Key Infrastructure (RPKI)

- IETF Standard published 2012
- Origin Validation using X.509 PKI Certificates
- Consists of ROAs:
  - Origin AS - Prefix
- To avoid prefix hijacking
- Only %5 of prefixes are signed
Current Automation Tools

- Policy extraction:
  - Whois command
  - IRRToolSet
  - RPSLtool
  - IRR PowerTools
  - BGPq3

- Router configuration
  - OpenNaaS
  - ConfD
  - PyEZ
## Analysis of current tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRRToolSet</td>
<td>- Full RPSL support</td>
<td>- No AS-SET query support</td>
</tr>
<tr>
<td></td>
<td>- RPSLng support</td>
<td>- Manual peering configuration</td>
</tr>
<tr>
<td></td>
<td>- 32-bit ASN support</td>
<td>- Does not compile</td>
</tr>
<tr>
<td></td>
<td>- Full BGP config generation</td>
<td>- Hard to understand</td>
</tr>
<tr>
<td>IRR Power Tools</td>
<td>- Route aggregation</td>
<td>- No RPSLng support</td>
</tr>
<tr>
<td></td>
<td>- AS-SET queries</td>
<td>- No 32-bit ASN support</td>
</tr>
<tr>
<td>BGpq3</td>
<td>- RPSL support</td>
<td>- Generates only prefix-list (or route filter)</td>
</tr>
<tr>
<td></td>
<td>- RPSLng support</td>
<td>- Cannot extract peering relations</td>
</tr>
<tr>
<td></td>
<td>- 32-bit ASN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- AS-SET queries</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Easy to use</td>
<td></td>
</tr>
<tr>
<td>rpsltool</td>
<td>- 32-bit ASN</td>
<td>- No RPSLng support</td>
</tr>
<tr>
<td></td>
<td>- AS-set queries</td>
<td>- Too many conf files</td>
</tr>
</tbody>
</table>
## Features and Functionality Requirements

<table>
<thead>
<tr>
<th>Features</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv6 support</td>
<td>Query IRR &amp; parse RPSL</td>
</tr>
<tr>
<td>32-bit ASN support</td>
<td>Local file for extra information</td>
</tr>
<tr>
<td>AS-SET query support</td>
<td>Automatic peering configuration</td>
</tr>
<tr>
<td>Route aggregation</td>
<td>Push config to router</td>
</tr>
<tr>
<td>Vendor independent</td>
<td>Security</td>
</tr>
</tbody>
</table>
Architecture - Design BGPwizard

Policy extraction using HTTPS + WHOIS REST API

Policy parser

Data XML model

Template handler

(Optional) Martian addresses filter

(Optional) Bogon prefix filter

(Optional) RPKI validation

JunOS configuration (NETCONF)

BGP configuration generator

IRR

Local file

Regional Routing Registry ROA

Team Cymru
  name: KPN
  username: root
  password: password.1
  neighbors:
    - 80.249.208.71:
      as: 3333
      group: Group1
      import_policy:
        use_RPSL: True
        name: POL_IMF1
      export_policy:
        use_RPSL: True
        name: POL_EXP1
        lprefix: 300
        med: 100
        community:
          name: KPNtoRIPE
          string: 206:3333
          policy_name: POLNAME2
      RPKI: False
      Bogon: True
      Martians: True
      logical_system: A
Implementation

- Python
- Whois REST API
- NETCONF (NCClient)
- RIPE RPKI Validator
- Team Cymru bogon and full bogon lists
Testbed

- GNS3
- 3x VirtualBox VM's running JunOS Olive on top of FreeBSD
- Ubuntu Desktop VirtualBox VM
Test Scenario
Limitations and future work

- Only Juniper, but can easily be extended
- Private peering
- Does not support complex RPSL (show examples)
- No local file syntax check
- Limit to the policy-statement size (not tool's issue)
- Missing proper error handling
- Trusts data from RIPE
- Replaces policy-statement, should compare first and if same, ignore
- No management IP
- No IPv6 yet
Conclusion

- RPSL does not keep up with BGP security trends (yet)
- Existing tools are unreliable
- BGP automation is limited to using different tools for router configuration and data query
- Current technologies can only be partially used for BGP configuration
Demo
Questions