UNIVERSITY OF AMSTERDAM

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

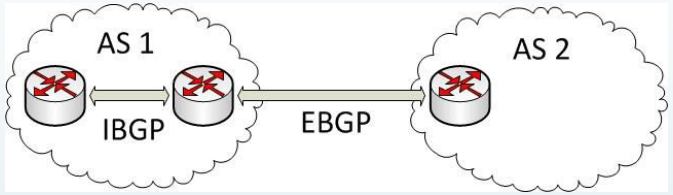
- World peace
- Issues with BGP:
  - BGP is the only protocol, not designed with security
  - A simple, flexible and **secure** automation required
  - Current solutions might be outdated

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)

Background
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)

Background - RPSL 2/4

- 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

Background - RPSL 3/4

- IPv6 policies

  mp-export: to AS6777
  2001:7F8:1::A500:6777:2 at
  2001:7F8:1::A500:3333:1
  announce { 2001:67c:2e8::/48 }

- Route6 object

  route6: fc00:600::/32
  origin: AS3333

Background - RPSL 4/4

```
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 };
            }
```

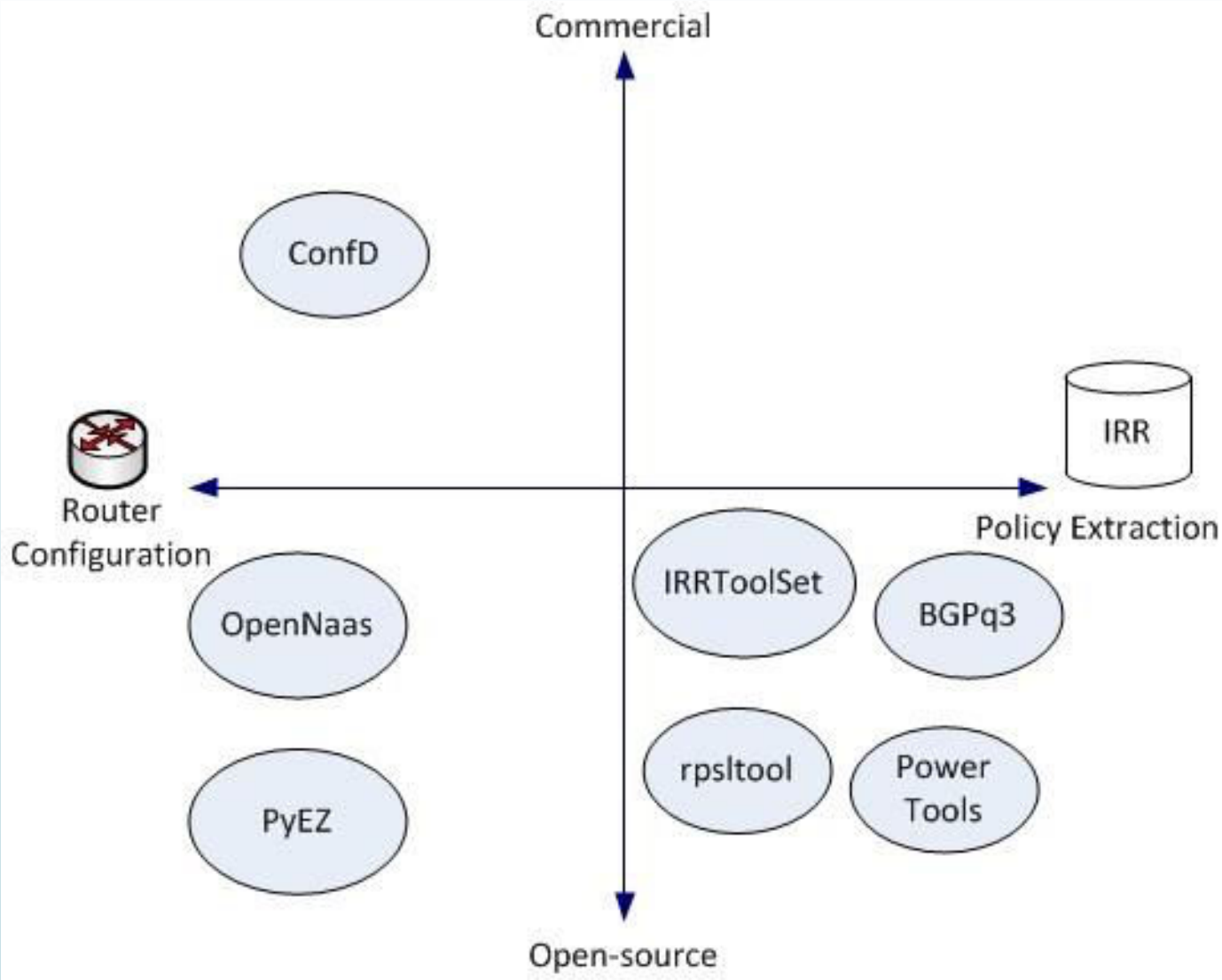
Background
Resource Public Key Infrastructure (RPKI)

- 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 1/2

- Policy extraction:
  - Whois command
  - IRRToolSet
  - RPSLtool
  - IRR PowerTools
  - BGPq3
- Router configuration
  - OpenNaaS
  - ConfD
  - PyEZ

Current Automation Tools 2/2



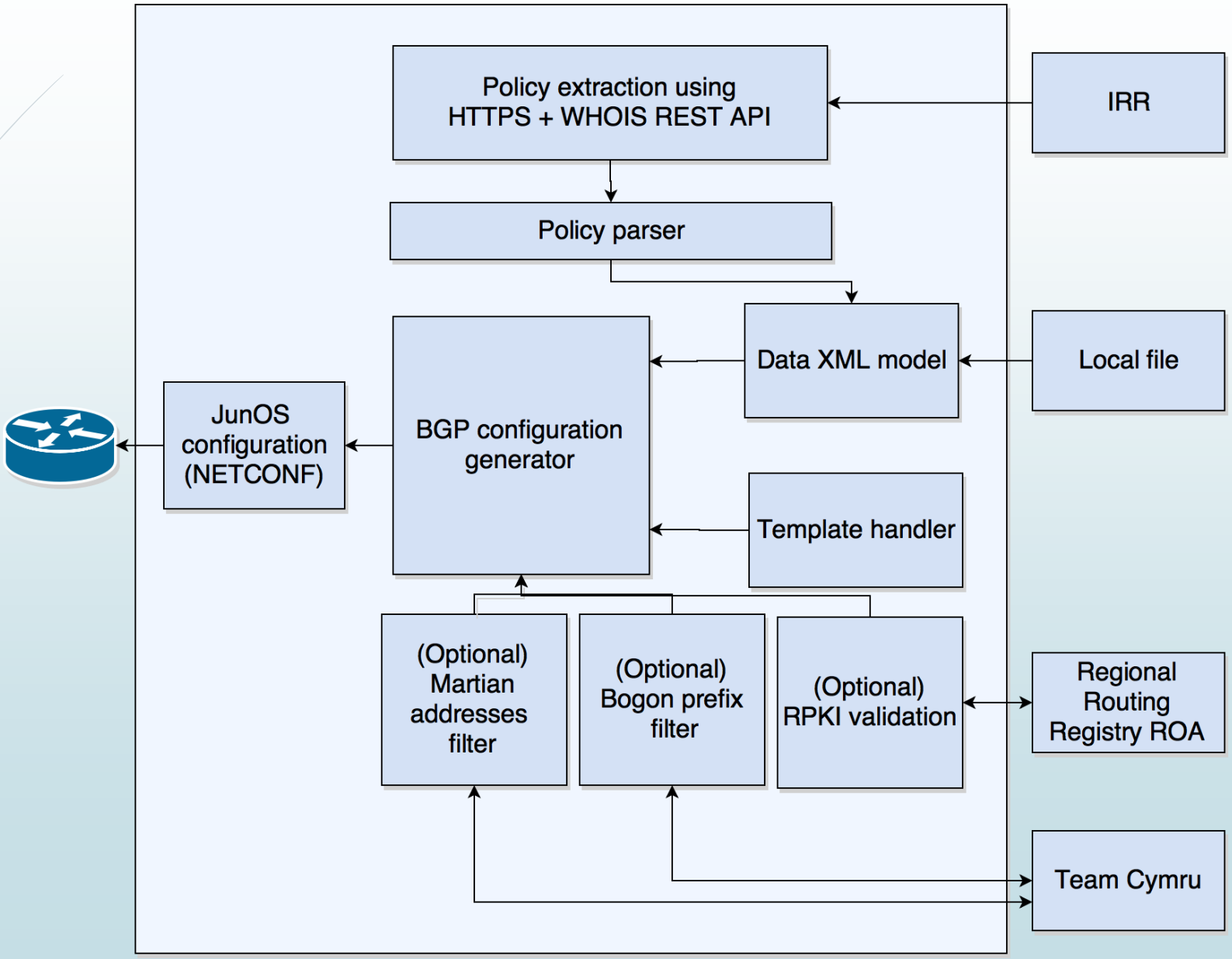
Analysis of current tools

Tool	Advantages	Disadvantages
IRRToolSet	- Full RPSL support - RPSLng support - 32-bit ASN support - Full BGP config generation	- No AS-SET query support - Manual peering configuration - Does not compile - Hard to understand
IRR Power Tools	- Route aggregation - AS-SET queries	- No RPSLng support - No 32-bit ASN support
BGPq3	- RPSL support - RPSLng support - 32-bit ASN - AS-SET queries - Easy to use	- Generates only prefix-list (or route filter) - Cannot extract peering relations
rpsltool	- 32-bit ASN - AS-set queries	- No RPSLng support - Too many conf files

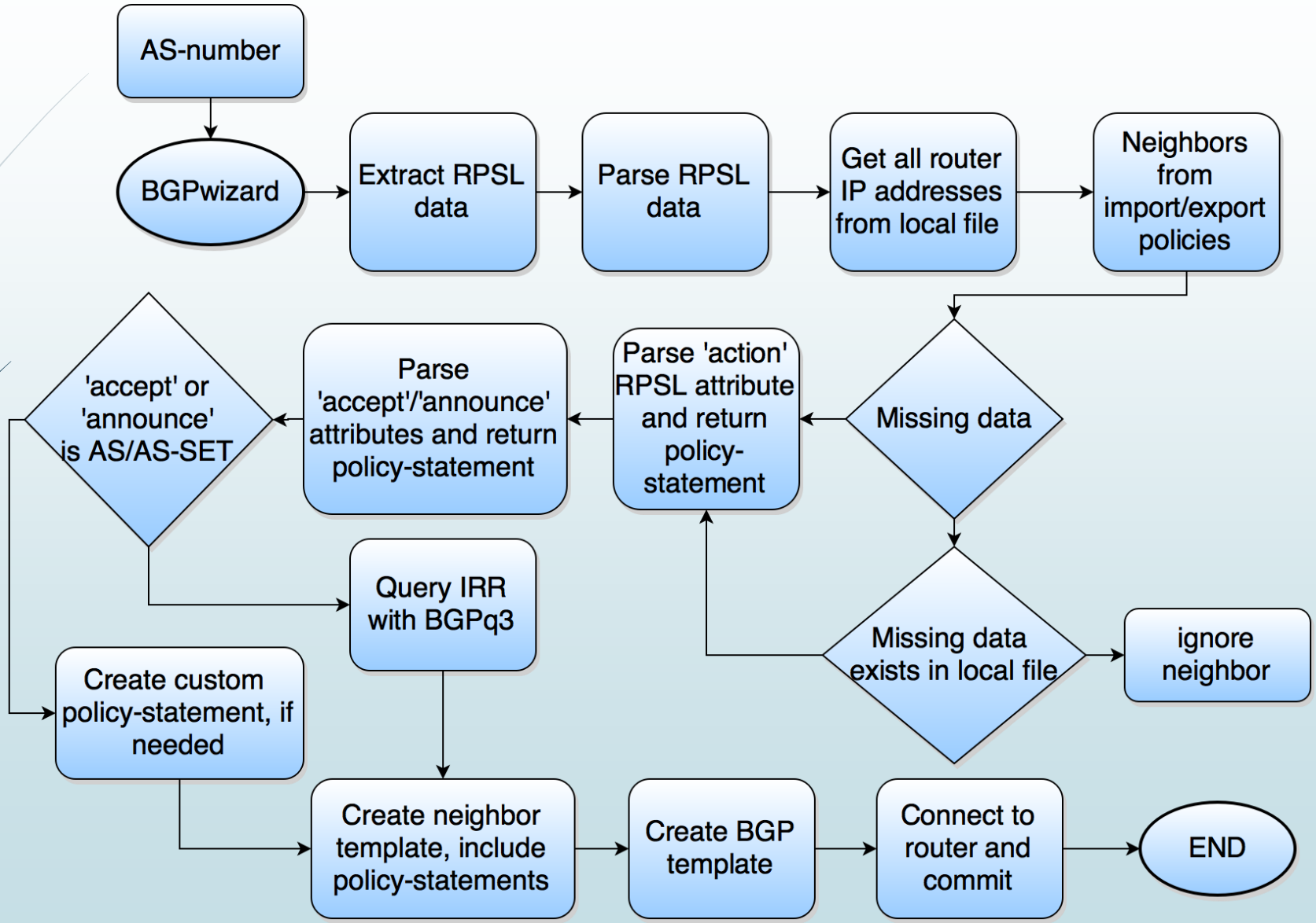
Features and Functionality Requirements

Features	Functionality
IPv6 support	Query IRR & parse RPSL
32-bit ASN support	Local file for extra information
AS-SET query support	Automatic peering configuration
Route aggregation	Push config to router
Vendor independent	Security

Architecture - Design BGPwizard



Architecture - Flow chart



Architecture - Local file design

```
- ip: 80.249.210.108
  name: KPN
  username: root
  password: password.1
  neighbors:
        - 80.249.208.71:
            as: 3333
            group: Group1
            import_policy:
                  use_RPSL: True
                  name: POL_IMP1
            export_policy:
                  use_RPSL: True
                  name: POL_EXP1
                  lpref: 300
                  med: 100
                  community:
                        name: KPNtoRIPE
                        string: 286:3333
                        policy_name: POLNAME2
            RPKI: False
            Bogon: True
            Martians: True
            logical_system: A
```

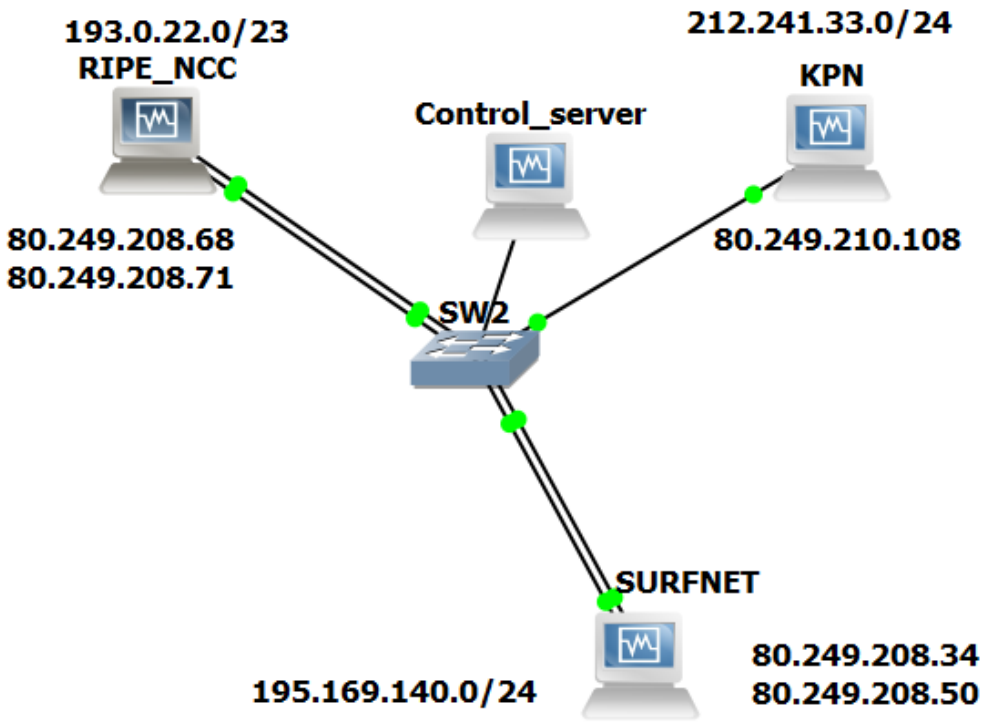
Implementation

- 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