Tutorial: Using RIPEstat

Christian Teuschel
Research & Development
Agenda

• Basics
  - 3 layers of RIPEstat
  - RIPEstat Web
    • Result page
    • Customize the result page
    • Compare results
    • Handling abuse
    • Work with BGPlay
Agenda

• Advanced
  • Work with widgets
    • Create a dashboard
  • RIPEstat Data API
    • Using Javascript to work with raw data
  • Nagios check
RIPEstat

Basics
What is RIPEstat

- Information system for Internet number resources

Data

- Routing data
  - Collected by RIS: [http://ris.ripe.net](http://ris.ripe.net)
  - Registration data (whois)
    - RIPE Database & other RIR databases
  - MaxMind’s geolocation data
  - Blacklist data
  - And many more: [https://stat.ripe.net/data-sources](https://stat.ripe.net/data-sources)
Introduction to RIPEstat

- RIPEstat Web
  http://stat.ripe.net

- RIPEstat Widget API

- RIPEstat Data API / RIPEstat Text API
  - https://stat.ripe.net/data/routing-status/data.json?resource=...
• Finding information on RIPEstat
• Information structure on RIPEstat
1. Querying for a Resource (Web)

- Task: Enter “AS333” in the search box
1. Querying for a Resource (Web)

- Result page

Widgets grouped into thematic tabs

Search box

Widgets display different types of information
1. Querying for a Resource (Web)

Tasks

• What network announces 140.78.50.90?
• Is 193.3.4.2 routed?
• In which country is 91.229.42.0/23 used?
• What is its corresponding INETNUM object?
• What widget provides real-time routing status?
• By what percent did the number of prefixes announced within Poland increased over the last two years?
• How would you share interesting network events with a colleague?
2. MyViews (Web)

- Create custom views
  - Click the “MyView” button
  - Drag and drop a widget onto the “MyView” button
MyViews are only visible to you. An option to share your views will be available soon!
2. MyViews (Web)

- Customise MyViews
  - Rename
  - Re-order
  - Control visibility
  - Remove

Re-order widgets as you like
Create SSO Account

• Tasks
  • Go to “https://stat.ripe.net” and click on “Login”
  • “…click here to create one.”
2. MyViews (Web)

• Tasks
  • Create a MyView for a prefix containing the following widgets:
    • Routing Status
    • Looking Glass
    • Routing History
  • Create another MyView with at least two widgets and give it a meaningful name
3. Compare Resources (Web)

- Compare results in different widgets

   Go to “Use Cases” > “Compare Results”

   Select a widget
3. Compare Resources (Web)

- Compare results in different widgets
  - Select the “Prefix Size Distribution” widget
  - Enter “AS1205”
3. Compare Resources (Web)

- In-widget comparison
  - Country Routing Statistics
3. Comparing Resources (Web)

- Tasks
  - Compare the number of announced prefixes for two networks over the past two years using the widget comparison page
  - How does the Internet in Poland compare to the UK? Use in-widget comparison!
4. Handling Abuse (Web)

- Take action in an abuse case with the Abuse Contact Finder

  Go to “Use Cases” > “Looking For Abuse Information”

  Enter the IP address

  In-depth information about abuse
4. Handling Abuse (Web)

- **Rating of the contact**
- **Email contact to report abuse to**
4. Handling Abuse (Web)

Details about the resource and abuse contact:

- Abuse Contact Finder (2001:87c:2e8://48)

  Email-Contact: abuse@ripe.net

  Contact-Quality-Rating: ⭐⭐⭐⭐ (5/5)
  This contact can be used to report abuse.

- Results for 193.0.18.0-193.0.21.255
  - abuse@ripe.net from abuse-contact role

- Special Network Resource Information
  This resource has been identified to be related to this information:
  RIPE NCC PI Allocation
  Held by: n.a.

- RIR Information
  RIPE NCC
  RIPE NCC’s Whois: https://whois.ripe.net/search/ripe.net/
4. Handling Abuse (Web)

- **Tasks**
  - What is the abuse contact for 193.0.20.22 or the hotel network?
  - Check an IP address from your home network

- **Discussion:**
  - What can you do in these cases?
    - No abuse contact found
    - No response on an abuse report
5. Let’s BGPlay! (Web)

- Use BGPlay to see how your network is routed
  - BGPlay is a tool that show the routing state in an animated and highly-interactive manner
  - Go to: https://stat.ripe.net/widget/bgplay
5. Let’s BGPlay! (Web)

- Control panel:
  - Covered time period
  - RRC selection

- Interactive graph visualisation

- Control timeline

- Selection timeline

- BGP event, ASN or ASN path details
5. Let’s BGPlay! (Web)

Examples:
- Prefix with announcements & withdrawals: 84.205.64.0/24
- Check IPv6 connectivity: 2001:67c:2e8::/48
- Multi-homed prefix: 199.7.80.0/24
5. Let’s BGPlay! (Web)

• Task
  • Find the up-stream provider for AS1205
  • Is AS3333 multi-homed?
  • Check the IPv6 connectivity of your own network
RIPEstat

Advanced
6. Embedding Widgets On Your Site

- Visualising raw data
  - What is the concept?
    - The RIPEstat Widget API uses the RIPEstat Data API (or any other RESTful-API) to create visualisations helping to make the data more understandable
    - Build on top of open web standards (HTML, CSS, Javascript)
  - RIPEstat widget API documentation: https://stat.ripe.net/docs/widget_api
6. Embedding Widgets On Your Site

This ISP embedded widgets on its page.

- Prefix Count widget
- AS Path Length widget
6. Embedding Widgets On Your Site

• Task
  • Create a simple dashboard page to monitor your network
  • Create a simple HTML page
    • Download the sample page: https://stat.ripe.net/widgets/demo/widget_home.html
  • Embed three widgets of your choice
  • Make the widget smaller and remove the RIPEstat logo
• Using raw data output
  • What is the concept?
    • The RIPEstat Data API provides the lowest level of access to data. This data can be fed into custom applications or just used for scripting.
    • RIPEstat data API documentation: https://stat.ripe.net/docs/data_api
• Tasks
  • Use your browser to retrieve all announced prefixes for AS6714
Tasks

- A simple check on a resource of your choice if it is seen on the Internet
- Create a simple script using resource-overview or routing-status

https://stat.ripe.net/widgets/demo/script_me.html
9. Create a Nagios Check

- Create a Nagios check using the RIPEstat Data API
  - Nagios is a monitoring tool that supports the development of custom checks (like other monitoring tools)
  - Result codes for a check show the state:

<table>
<thead>
<tr>
<th>State</th>
<th>Result Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ok</td>
<td>0</td>
</tr>
<tr>
<td>Warning</td>
<td>1</td>
</tr>
<tr>
<td>Error</td>
<td>2</td>
</tr>
</tbody>
</table>
9. Create a Nagios Check

Example of a Python based check:

```python
if args.transit:
    transit = args.transit.split(',')[0]
origin = args.origin.split(',')[0]

nagios_status = 0
origin_errors = 0
transit_errors = 0

nagios_message = ""
url = "https://%s/data/looking-glass/data.json?resource=%s" % (args.statserver, urllib2.quote(args.prefix, ''))
lookingglass_raw = urllib2.urlopen(url)
lookingglass_json = json.load(lookingglass_raw)
if lookingglass_json['data_call_status'] != 'supported':
    print "WARN: %s is under maintenance" % (args.statserver)
sys.exit(1)
for rrc in lookingglass_json['data']['rrcs']:
    for peer in lookingglass_json['data']['rrcs'][rrc]['entries']:
        aspath = peer['as_path'].split()
        details = peer['details'][0].split()
        peer_addr = details[0]
        nexthop = details[2]
        router_id = details[3]

        if len(aspath) > args.minpath:
            if aspath[-1] not in origin:
                origin_errors += 1
                nagios_message = nagios_message + ("Origin mismatch %s (%s): %s; " % (rrc, peer_addr, aspath[-1]))
            if args.transit:
                if aspath[-2] not in transit:
                    transit_errors += 1
                    nagios_message = nagios_message + ("Transit mismatch %s (%s): %s; " % (rrc, peer_addr, aspath[-2]))

if origin_errors == 0 and transit_errors == 0:
    nagios_message = "%s OK: %s Origin is %s " % (args.prefix, args.origin)
    if args.transit:
        nagios_message = "%s and all transits match %s" % (nagios_message, args.transit)
    else:
        nagios_message = "%s Origin is %s " % (nagios_message, args.origin)

elif origin_errors >= args.crit or transit_errors >= args.crit:
    nagios_status = 2
    nagios_message = "ERROR: " + nagios_message

elif origin_errors >= args.warn or transit_errors >= args.warn:
    nagios_status = 1
    nagios_message = "WARN: " + nagios_message
```

Christian Teuschel - RIPE68 - 12 May 2014