



**RIPE NCC**  
RIPE NETWORK COORDINATION CENTRE

# Routing Information System (RIS)

# Routing Data (RIS)



- 18 BGP collectors and growing
- 600+ peers
- 150+ full-feed peers





# Raw BGP data!

- 15+ years of raw data (5.8 TB) available to download and analyse yourself :)
  - <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data>
- Readable using BGPdump utility
  - open source, maintained by RIPE NCC
  - <https://bitbucket.org/ripenncc/bgpdump>
- ...and by other tools
  - CAIDA BGPStream: <http://bgpstream.caida.org/>

# Exercise



- Go onto virtual machines and use bgpdump or bgpstream to read the raw BGP data files

# RIPEstat Data API



- All these queries are available through an API
- Actually, all those shiny web interfaces use the API anyway
- You can use it too!! Write your own scripts, etc.
- [https://stat.ripe.net/docs/data\\_api](https://stat.ripe.net/docs/data_api)
- There are also some extra API calls which are not yet visualised

# RIPEstat Data API



- Remember this started because looking glasses are instantaneous?
- BGP State
  - [https://stat.ripe.net/docs/data\\_api#BGPState](https://stat.ripe.net/docs/data_api#BGPState)
- This data call returns the state of BGP routes for a resource at a certain point in time, as observed by all the RIS collectors
- This is derived by applying a computation of state to the RIB dump (granularity=8h) that occurred exactly before that time, using the BGP updates observed between the RIB time and the query time

# RIPEstat Data API – BGP State – Selectel case



- Show me who and how was announcing this prefix at that time
- <https://stat.ripe.net/data/bgp-state/data.json?resource=188.93.16.2&timestamp=2017-07-16T00:51:23Z>

```
"data": {
  "bgp_state": [
    {
      "source_id": "00-178.255.145.243",
      "path": [50304, 42708, 2854, 49505],
      "community": [],
      "target_prefix": "188.93.16.0/22"
    },
    {
      "source_id": "00-193.0.0.56",
      "path": [3333, 1136, 24785, 24785, 24785, 24785, 20562, 2854, 49505],
      "community": [],
      "target_prefix": "188.93.16.0/22"
    },
    ...
  ],
  "query_time": "2017-07-16T00:51:23",
  "resource": "188.93.16.0/22",
  "nr_routes": 44
}
```



# Live stream demo

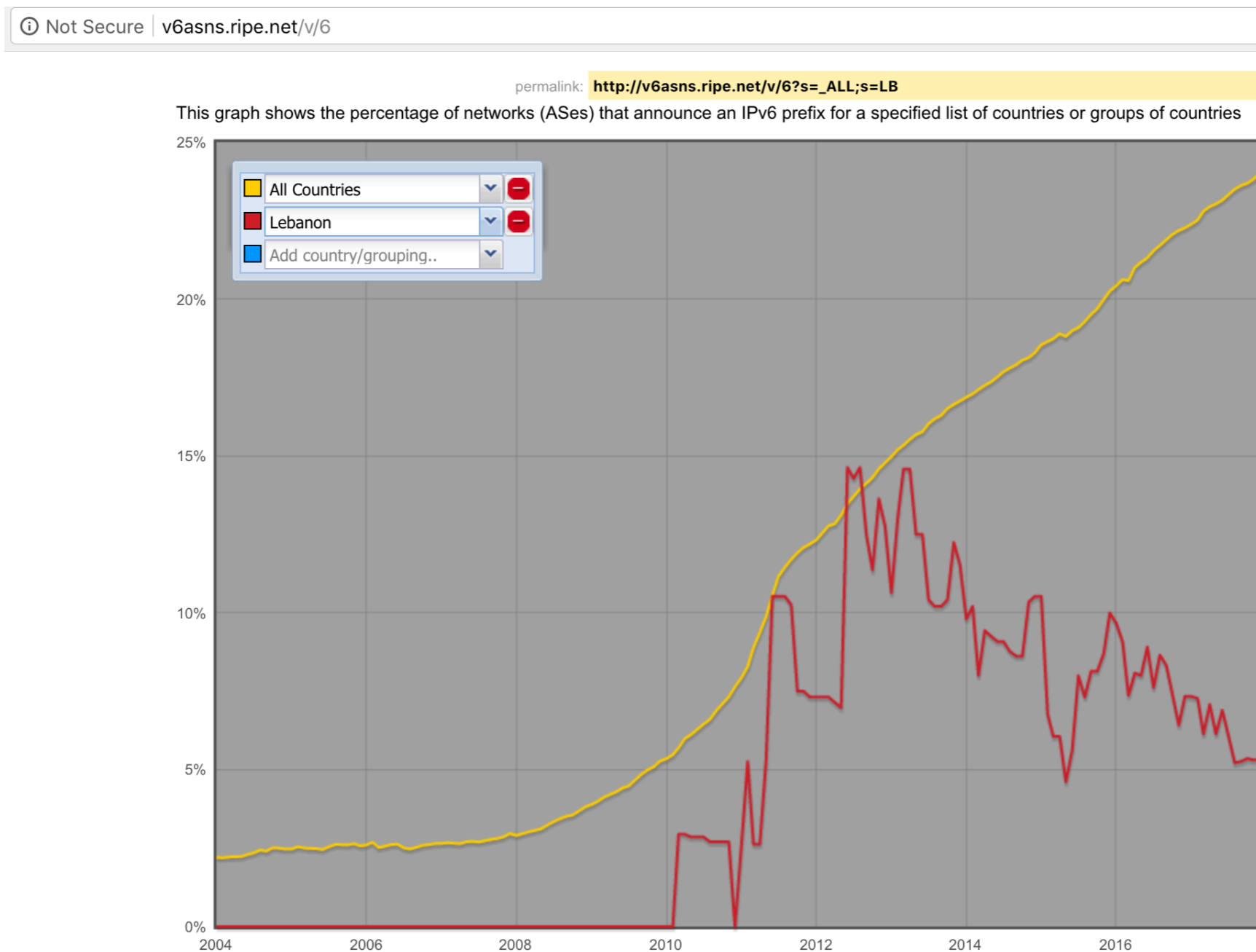
- Prototype!!
- Let's see if it works
- <http://stream-dev.ris.ripe.net/demo>
- python: <https://github.com/sdstrowes/ris-streamer>
- Live stream enables new applications
  - BGP hijack detection
  - Real time anomaly analysis
  - Live monitoring of your routes



# IPv6 ness



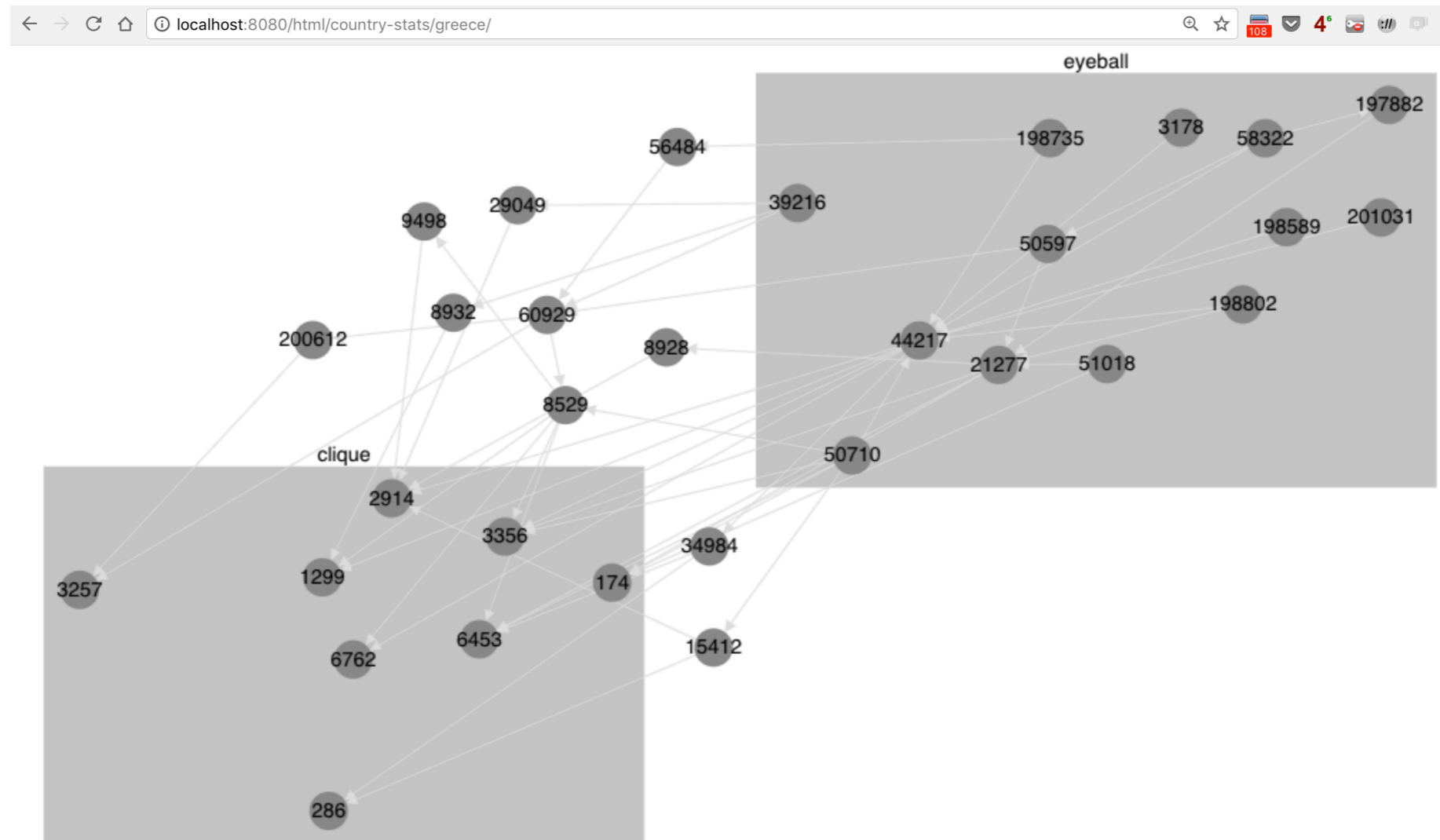
- [http://v6asns.ripe.net/v/6?s=\\_ALL;s=LB](http://v6asns.ripe.net/v/6?s=_ALL;s=LB)



# LB users -> Internet core routes?



- Example code for Greece, can we do LB?



# BGP Outages in Lebanon?



- Correlate known events (outages, weather?) with BGP data
- Was the Lebanese population affected?
  - Add in per-network user-population estimates
  - early prototype at: <https://github.com/emileaben/country-outage-pop-viz>



# Monitor important prefixes?

- How are your ccTLD netblocks routed?

```
e1000$ dig lb ns +short
zeina.aub.edu.lb.
fork.sth.dnsnode.net.
ns1.dns.aq.
rip.psg.com.
e1000$ dig zeina.aub.edu.lb +short
193.188.128.14
e1000$ dig fork.sth.dnsnode.net +short
77.72.229.254
e1000$ dig ns1.dns.aq +short
203.119.56.132
e1000$ dig rip.psg.com +short
147.28.0.39
e1000$ ip2as 193.188.128.14
12812
e1000$ ip2as 77.72.229.254
8674
e1000$ ip2as 203.119.56.132
45181
e1000$ ip2as 147.28.0.39
3130
```

- #hackathon project?
- Also RTTs with RIPE Atlas?



# Monitor important prefixes?

- How are your news-sites routed?
- How are your banks routed?
  - How are they protected (/24, IRR, RPKI, all of the above?)
- Are they doing IPv6 already?
- BGP hijackability?
  - Longer paths are easier to hijack #hackathon project?
  - Other protections in place? #hackathon project?
  - IRR, RPKI?