



How Open Source Supports Your BGP Research

Massimo Candela
Senior Software Engineer
Global IP Network
massimo@ntt.net
@webrobotics

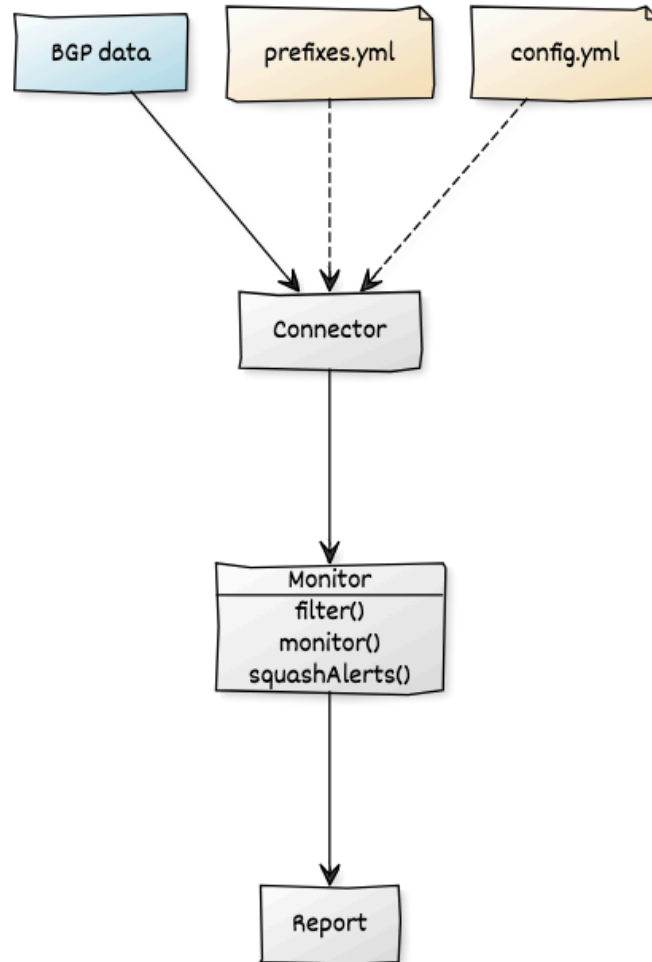
BGPalerter is a tool for analyzing streams of BGP data

- We developed it for monitoring NTT prefixes
 - hijacks, visibility loss, RPKI, and more
- We released it open-source (BSD-3-Clause)
 - <https://github.com/nttgin/BGPalerter>
- **It can be used for research!**

Example Use Case

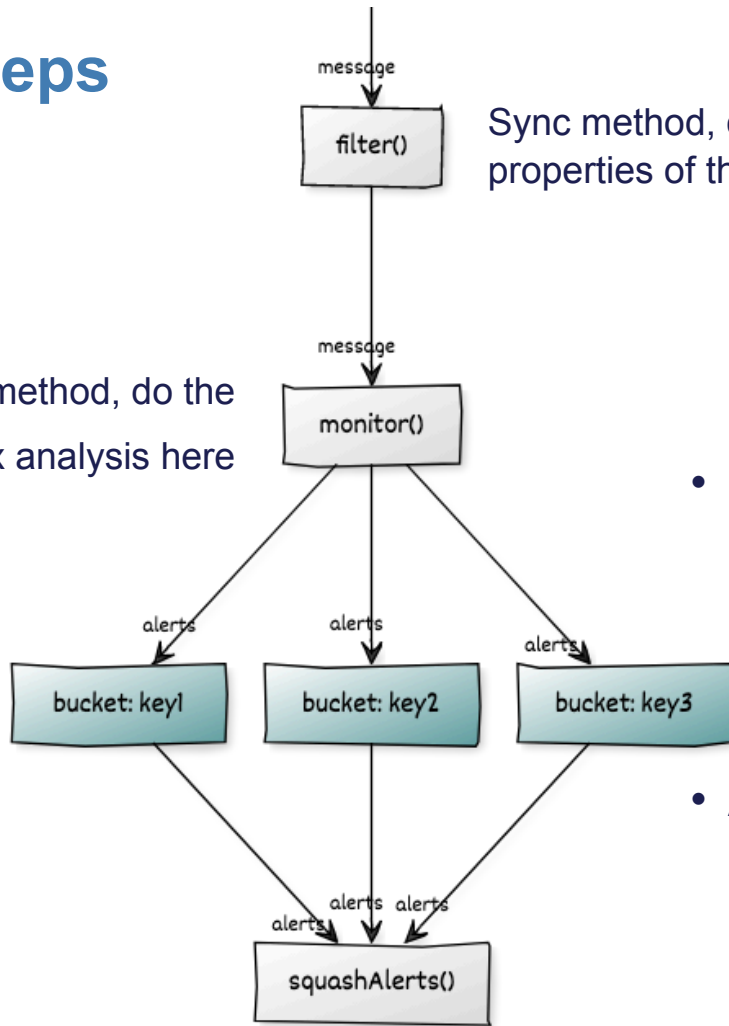
- We want to monitor BGP announcements (*filter condition*)
- Detect bursts of RPKI invalid announcements (*monitor condition*)
 - We need to validate BGP announcements
- Every time a burst happen, we want to report the AS number (*group condition*), the number of RPKI invalids, and the BGP announcements
- We define as “burst” a number of invalid announcements ≥ 20 in 6 minutes (*squash condition*)

Pipeline



Monitor steps

Async method, do the complex analysis here



Sync method, only check for intrinsic properties of the BGP message

- Each alert is composed of:
 - Description of the verified monitored condition
 - BGP messages that verified the condition
- Alerts stay in the system for “fadeOffSeconds”

Example Use Case

- We want to monitor BGP announcements (***filter condition***)
- Detect bursts of RPKI invalid announcements (***monitor condition***)
 - We need to validate BGP announcements
- Every time a burst happen, we want to report the AS number (***group condition***), the number of RPKI invalids, and the BGP announcements
- We define as “burst” a number of invalid announcements ≥ 20 in 6 minutes (***squash condition***)

Example Use Case

- We want to monitor BGP announcements (***filter condition***)
- Detect bursts of RPKI invalid announcements (***monitor condition***)
 - We need to validate BGP announcements
- Every time a burst happen, we want to report the AS number (***group condition***), the number of RPKI invalids, and the BGP announcements
- We define as “burst” a number of invalid announcements ≥ 20 in 6 minutes (***squash condition***)

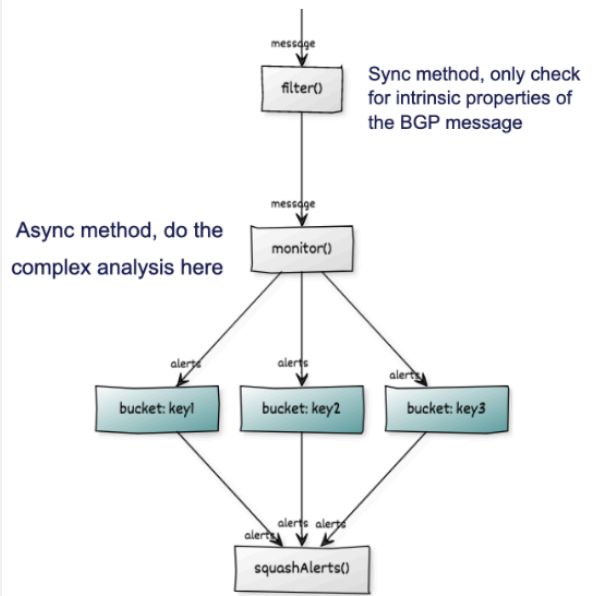
Project

- BGPalerter -/Documents/work/BG
 - .github
 - bin
 - docs
 - node_modules library root
 - src
 - config
 - connectors
 - inputs
 - monitors
 - monitor.js
 - monitorAS.js
 - monitorHijack.js
 - monitorNewPrefix.js
 - monitorPassthrough.js
 - monitorPath.js
 - monitorPathNeighbors.js
 - monitorROAS.js
 - monitorRPKI.js
 - MonitorRpkIBursts.js**
 - monitorSwUpdates.js
 - monitorVisibility.js
 - processMonitors
 - reports
 - utils
 - connectorFactory.js
 - consumer.js
 - env.js
 - generatePrefixesList.js
 - model.js
 - worker.js
 - tests
 - .babelrc
 - .eslintrc.json
 - .gitignore
 - .hound.yml
 - .npmignore
 - .npmrc
 - AUTHORS
 - bgpalerter.pid
 - build.sh
 - config.yml
 - config.yml.example
 - Dockerfile
 - export.json

```

1  import Monitor from "../monitor";
2
3  export default class MonitorRpkIBursts extends Monitor {
4
5      constructor(name, channel, params, env, input){...};
6      updateMonitoredResources = () => {};
7
8      filter = (message) => {
9
10         return message.type === 'announcement';
11     };
12
13     monitor = (message) => {...};
14
15
16     squashAlerts = (alerts) => {...};
17
18 }
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```



message.json

```

1  {
2      "type": "announcement",
3      "prefix": "138.207.66.0/24",
4      "peer": "193.239.118.105",
5      "path": [ 206313, 50629, 174, 3356, 12025 ],
6      "originAS": [ 12025 ],
7      "nextHop": "193.239.118.105",
8      "aggregator": null,
9      "timestamp": 1637547054080,
10     "communities": [[ 174, 21100 ]]
11 }

```


Example Use Case

- We want to monitor BGP announcements (*filter condition*)
- Detect bursts of RPKI invalid announcements (*monitor condition*)
 - We need to validate BGP announcements
- Every time a burst happen, we want to report the AS number (*group condition*), the number of RPKI invalids, and the BGP announcements
- We define as “burst” a number of invalid announcements ≥ 20 in 6 minutes (*squash condition*)

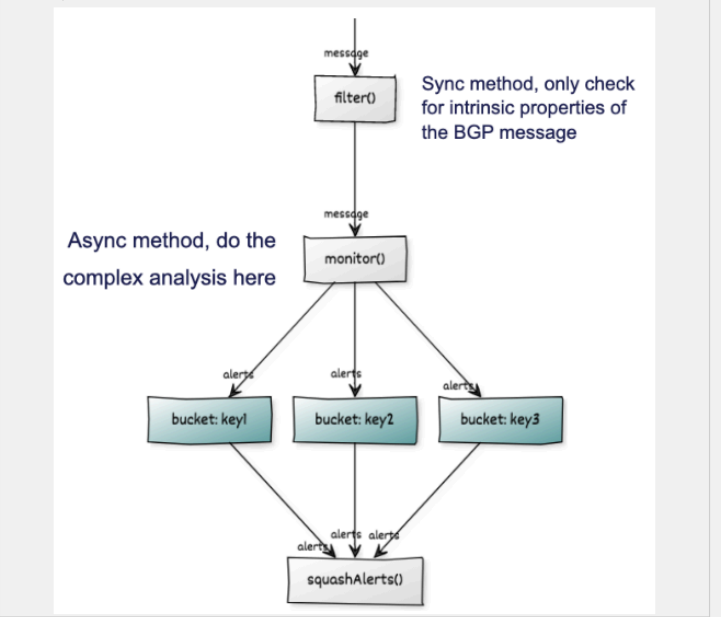
Project

- BGPalerter ~/Documents/work/BGPalerter
 - .github
 - bin
 - docs
 - node_modules library root
 - src
 - config
 - connectors
 - inputs
 - monitors
 - monitor.js
 - monitorAS.js
 - monitorHijack.js
 - monitorNewPrefix.js
 - monitorPassthrough.js
 - monitorPath.js
 - monitorPathNeighbors.js
 - monitorROAS.js
 - monitorRPKI.js
 - MonitorRpkIBursts.js
 - monitorSwUpdates.js
 - monitorVisibility.js
 - processMonitors
 - reports
 - utils
 - connectorFactory.js
 - consumer.js
 - env.js
 - generatePrefixesList.js
 - model.js
 - worker.js
 - tests
 - .babelrc
 - .eslintrc.json
 - .gitignore
 - .hound.yml
 - .npmignore
 - .npmrc
 - AUTHORS
 - bgpalerter.pid
 - build.sh
 - config.yml
 - config.yml.example
 - Dockerfile
 - export.json

```

8  updateMonitoredResources = () => {};
9
10
11 filter = (message) => {
12   return message.type === 'announcement';
13 };
14
15
16 monitor = (message) => {
17   const { prefix, originAS } = message;
18
19   return this.rpki
20     .validate(prefix, originAS)
21     .then(result => {
22
23       if (result.valid === false) {
24
25         const key = originAS;
26
27         this.publishAlert(key,
28           prefix,
29           matchedRule: {},
30           message,
31           extra: {});
32       }
33     });
34 };
35
36
37
38 squashAlerts = (alerts) => {...};
39
40
41
42
43
44
45
46
47
48
49
50

```



```

message.json x
1  {
2    "type": "announcement",
3    "prefix": "138.207.66.0/24",
4    "peer": "193.239.118.105",
5    "path": [ 206313, 50629, 174, 3356, 12025 ],
6    "originAS": [ 12025 ],
7    "nextHop": "193.239.118.105",
8    "aggregator": null,
9    "timestamp": 1637547054080,
10   "communities": [[ 174, 21100 ]]
11 }

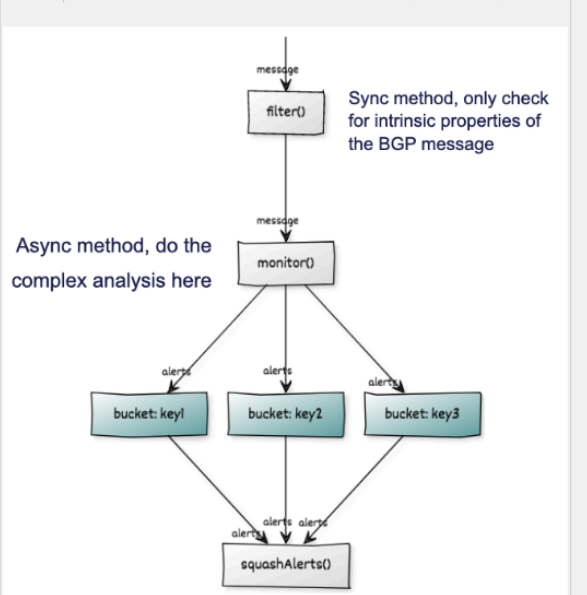
```

Example Use Case

- We want to monitor BGP announcements (*filter condition*)
- Detect bursts of RPKI invalid announcements (*monitor condition*)
 - We need to validate BGP announcements
- Every time a burst happen, we want to report the AS number (*group condition*), the number of RPKI invalids, and the BGP announcements
- We define as “burst” a number of invalid announcements ≥ 20 in 6 minutes (*squash condition*)

- Project
- ▼ BGPalerter ~\Documents\work\BC
 - > .cache
 - > .github
 - > bin
 - > docs
 - > node_modules library root
 - ▼ src
 - > config
 - > connectors
 - > inputs
 - ▼ monitors
 - monitor.js
 - monitorAS.js
 - monitorHijack.js
 - monitorNewPrefix.js
 - monitorPassthrough.js
 - monitorPath.js
 - monitorPathNeighbors.js
 - monitorROAS.js
 - monitorRPKI.js
 - MonitorRpkIBursts.js**
 - monitorSwUpdates.js
 - monitorVisibility.js
 - > processMonitors
 - > reports
 - > utils
 - connectorFactory.js
 - consumer.js
 - env.js
 - generatePrefixesList.js
 - model.js
 - worker.js
 - > tests
 - .babelrc
 - eslintrc.json
 - .gitignore
 - .hound.yml
 - .npmignore
 - .npmrc
 - AUTHORS
 - bgpalter.pid
 - build.sh
 - config.yml
 - config.yml.example
 - Dockerfile
 - export.ison

```
22
23   if (result.valid === false) {
24
25     const key = originAS;
26
27     this.publishAlert(key,
28       prefix,
29       matchedRule: {},
30       message,
31       extra: {});
32   }
33 };
34
35
36
37
38 squashAlerts = (alerts) => {
39
40   const thresholdInvalids = 20;
41
42   if (alerts.length >= thresholdInvalids) {
43     const { originAS } = alerts[0].matchedMessage;
44
45     return `${originAS} announced ${alerts.length} invalid prefixes`;
46   }
47 };
48
49
50
```



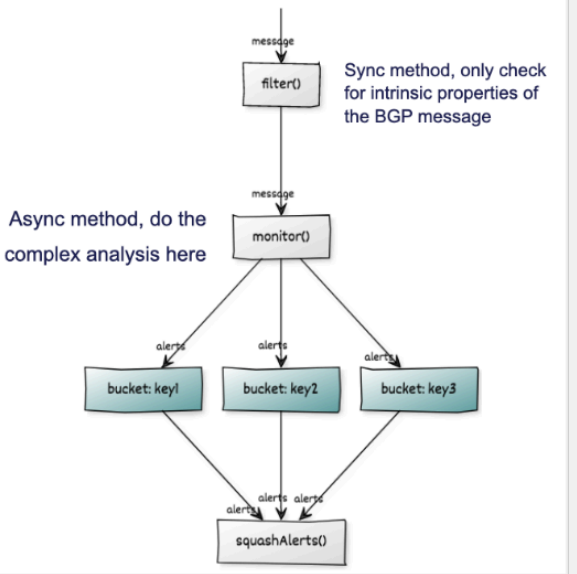
```
message.json
1 {
2   "type": "announcement",
3   "prefix": "138.207.66.0/24",
4   "peer": "193.239.118.105",
5   "path": [ 206313, 50629, 174, 3356, 12025 ],
6   "originAS": [ 12025 ],
7   "nextHop": "193.239.118.105",
8   "aggregator": null,
9   "timestamp": 1637547054080,
10  "communities": [ [ 174, 21100 ] ]
11 }
```

Project Structure:

- Project
 - BGPalerter ~/Documents/work/BC
 - .cache
 - .github
 - bin
 - docs
 - node_modules library root
 - src
 - config
 - connectors
 - inputs
 - monitors
 - monitor.js
 - monitorAS.js
 - monitorHijack.js
 - monitorNewPrefix.js
 - monitorPassthrough.js
 - monitorPath.js
 - monitorPathNeighbors.js
 - monitorROAS.js
 - monitorRPKI.js
 - MonitorRpkIBursts.js
 - monitorSwUpdates.js
 - monitorVisibility.js
 - processMonitors
 - reports
 - utils
 - connectorFactory.js
 - consumer.js
 - env.js
 - generatePrefixesList.js
 - model.js
 - worker.js
 - tests
 - .babelrc
 - .eslintrc.json
 - .gitignore
 - .hound.yml
 - .npmignore
 - .npmrc
 - AUTHORS
 - bgpalterter.pid
 - build.sh
 - config.yml
 - config.yml.example
 - Dockerfile

```

1077 # ALLOW TO RUN BGPALERTER BEHIND AN HTTP/HTTPS PROXY.
1088 # You can also specify which module can bypass the proxy.
1089 # More information here: https://github.com/nttgin/BGPalerter/blob/main/docs/http-proxy.md
1110
1111 # httpProxy: http://username:password@127.0.0.1:9000
1112
1113 #####
1114 # RPKI settings:
1115 # Global RPKI settings shared across all monitors requiring RPKI data
1116 # More information here: https://github.com/nttgin/BGPalerter/blob/main/docs/rpki.md
1117 #
1118 # 29/06/2021: rpkiClient has been set as default vrpProvider in this example since is the only one s
1119 # ROAs expiration data. See what expiration data enables you to do: https://github.com/nttgin/BGPale
1120
1121 rpki:
1122   vrpProvider: rpkiClient
1123   preCacheROAs: true
1124   refreshVrpListMinutes: 15
1125   markDataAsStaleAfterMinutes: 120
1126
1127 #####
1128 # Advanced settings (Don't touch here!)
1129 # Please, refer to the documentation to know the meaning of the following parameters.
1130
1131 alertOnlyOnce: false
1132 fadeOffSeconds: 360
1133 checkFadeOffGroupsSeconds: 30
1134 pidFile: bgpalterter.pid
1135 maxMessagesPerSecond: 10000
1136 multiProcess: false
1137 environment: research
1138 configVersion: 2
    
```



```

11 message.json
12 {
13   "type": "announcement",
14   "prefix": "138.207.66.0/24",
15   "peer": "193.239.118.105",
16   "path": [ 206313, 50629, 174, 3356, 12025 ],
17   "originAS": [ 12025 ],
18   "nextHop": "193.239.118.105",
19   "aggregator": null,
20   "timestamp": 1637547054080,
21   "communities": [[ 174, 21100 ]]
22 }
    
```

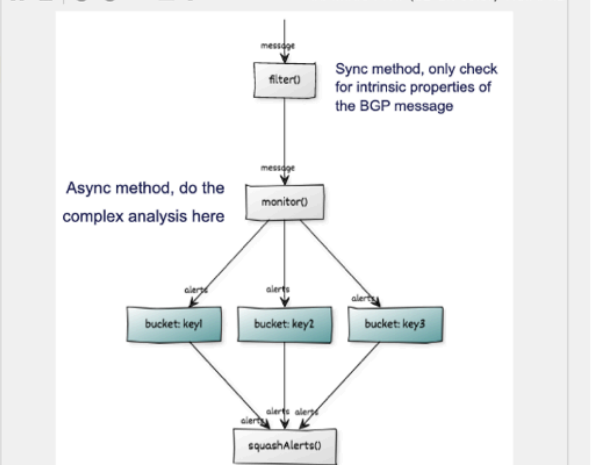
Project

- ▼ BGPalerter ~/Documents/work/BGPalerter
 - ▼ .cache
 - ▼ .github
 - ▼ bin
 - ▼ docs
 - ▼ logs
 - ▼ .error
 - ▼ .reports
 - ▼ error.log
 - ▼ error-2021-11-23.log
 - ▼ reports.log
 - ▼ reports-2021-11-23.log
 - ▼ node_modules library root
 - ▼ src
 - ▼ config
 - ▼ connectors
 - ▼ inputs
 - ▼ monitors
 - ▼ processMonitors
 - ▼ reports
 - ▼ utils
 - ▼ connectorFactory.js
 - ▼ consumer.js
 - ▼ env.js
 - ▼ generatePrefixesList.js
 - ▼ model.js
 - ▼ worker.js
 - ▼ tests
 - ▼ .babelrc
 - ▼ .eslintrc.json
 - ▼ .gitignore
 - ▼ .hound.yml
 - ▼ .npmignore
 - ▼ .npmrc
 - ▼ AUTHORS
 - ▼ hgpalerter.nid

```

1 2021-11-23T16:18:06+00:00 verbose: AS2024 announced 32 invalid prefixes
2 2021-11-23T20:11:11+00:00 verbose: AS5022 announced 24 invalid prefixes
3 2021-11-23T21:14:40+00:00 verbose: AS2024 announced 21 invalid prefixes
4

```



```

1 {
2   "type": "announcement",
3   "prefix": "138.207.66.0/24",
4   "peer": "193.239.118.105",
5   "path": [ 206313, 50629, 174, 3356, 12025 ],
6   "originAS": [ 12025 ],
7   "nextHop": "193.239.118.105",
8   "aggregator": null,
9   "timestamp": 1637547054080,

```

```

> bgpalerter@1.28.4 serve
> babel-node index.js

★ Loaded config: /Users/massimocandela/Documents/work/BGPalerter/config.yml
BGPalerter, version: 1.28.4 environment: research
Monitoring everything

```

Contribute!



- Source code on GitHub
 - <https://github.com/nttgin/BGPalerter>

Thank you.

Massimo Candela

Senior Software Engineer, Network Information Systems Development

Global IP Network

massimo@ntt.net

@webrobotics

www.gin.ntt.net

@GinNTTnet #globalipnetwork #AS2914