Monitor BGP Updates using BGPalerter

Saw Yan Paing

CCIE #57007

MMNOG 8 (MDY) 2026

Network Monitoring Tools for ISPs

- SNMP Based Monitoring System
- Flow collector
- Route Collector
- Looking Glass
- syslog server

Core Metrics Monitored for ISPs

- **Prefix Hijacking & Visibility Loss**
- **Upstream/Downstream AS-Path Changes**
- **ROA Status Changes**

BGPalerter

WHOIS & WEBSITE ADVANCED WHOIS MAKE A PAYMENT Your IP address: 45.199.180.94

APNIC

Get IP Manage IP Training Events Insights Community Blog Help Centre About Contact

Easy BGP Monitoring with BGPalerter

By Massimo Candela on 27 Jul 2020

Category: Tech matters

Tags: BGP, Guest Post, security

1 Comment

Share on LinkedIn

Blog home



We all know by now that monitoring the correctness of the Border Gateway Protocol (BGP) is a fundamental activity for any actor operating on the public Internet. Monitoring BGP is not only useful for identifying hijacks committed by other Autonomous Systems (ASes) but especially for timely identification of what your AS is doing.

Indeed human error is the first cause of network incidents, and being able to immediately identify a prefix you were not supposed to announce (for example, a typo) or a loss of visibility due to a wrong just-deployed configuration can drastically reduce the impact on the availability of both your and other operators' services.

Get Updates

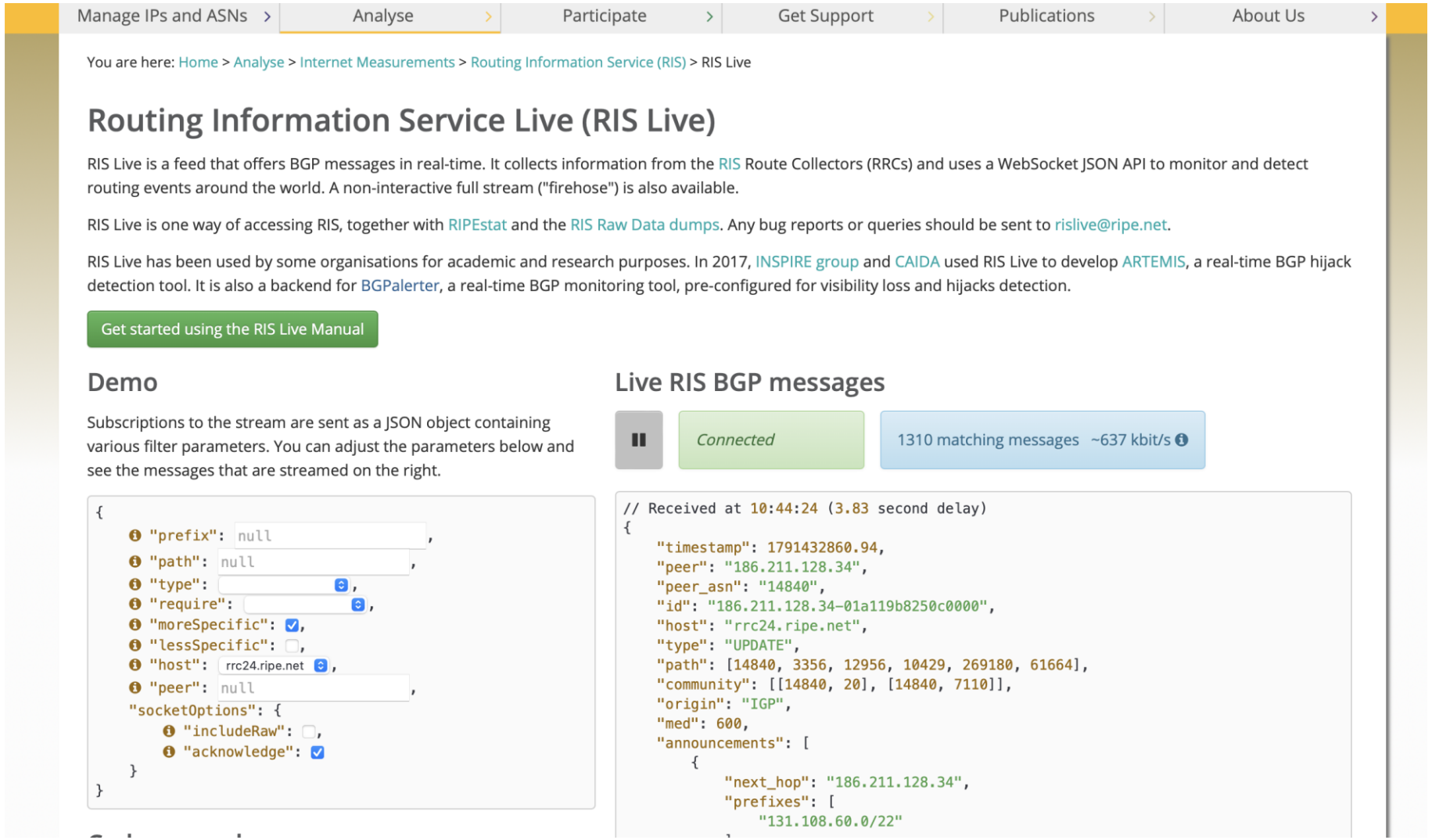
Email *

Subscribe! Show options

Authors

- Adli Wahid
- Aftab Siddiqui
- Geoff Huston
- George Michaelson
- Jen Linkova

RIPE RIS



- 600+ Peer (Global)

BGPalerter

Self-configuring BGP monitoring tool, which allows you to monitor in real-time if:

- Developed by NTT
- any of your prefixes loses visibility;
- any of your prefixes is hijacked;
- your AS is announcing RPKI invalid prefixes (e.g., not matching prefix length);
- your AS is announcing prefixes not covered by ROAs;
- any of your ROAs is expiring;
- ROAs covering your prefixes are no longer reachable;
- RPKI Trust Anchors malfunctions;
- a ROA involving any of your prefixes or ASes was deleted/added/edited;
- your AS is announcing a new prefix that was never announced before;
- an unexpected upstream (left-side) AS appears in an AS path;
- an unexpected downstream (right-side) AS appears in an AS path;
- one of the AS paths used to reach your prefix matches a specific condition defined by you

Alerta Server

- **Prerequisites**
  - “super user” privileges
  - MongoDB 4.0
  - Python 3

Step 1: Install Packages

To install MongoDB 6.0 run the following commands:

- sudo apt-get update
- sudo apt-get install -y mongodb-org

Step 1: Install Packages

Start the MongoDB server, check it is running and set it to start on reboot:

- sudo systemctl start mongod
- sudo systemctl status mongod
- sudo systemctl enable mongod

To run Alerta we need to ensure all Python 3 dependencies are installed:

- sudo apt-get install -y python3 python3-setuptools python3-pip python3-dev python3-venv
- sudo apt-get install -y nginx uwsgi-plugin-python3

Step 1: Install Packages

To install the Alerta server and alerta command-line tool into a Python 3 virtual environment run:

- cd /opt
- python3 -m venv alerta
- alerta/bin/pip install --upgrade pip wheel alerta-server alerta pymongo<4.0

Step 1: Install Packages

Create /etc/alerta.conf:

sudo nano /etc/alerta.conf

```
PROFILES = {
        'default': {
            'DATABASE_URL': 'mongodb://localhost:27017/alerta',
            'SECRET_KEY': 'your-super-secret-key-change-this',
            'CORS_ORIGINS': ['*']
        }
    }
```

Step 1: Install Packages

To install the web console run:

- wget https://github.com/alerta/alerta-webui/releases/latest/download/alerta-webui.tar.gz
- tar zxvf alerta-webui.tar.gz
- cp -a dist /var/www/html

Step 2: Configuration

Create a wsgi python file, uWsgi configuration file and systemd script:

- sudo vi /var/www/wsgi.py

```
from alerta import create_app


app = create_app()
```

Step 2: Configuration

The uwsgi server mounts the Alerta API on /api, logs to syslog and uses a unix socket to communicate with the nginx web server:

- sudo vi /etc/uwsgi.ini

```
[uwsgi]
chdir = /var/www
mount = /api=wsgi.py
callable = app
manage-script-name = true
env = BASE_URL=/api

master = true
processes = 5
logger = syslog:alertad

socket = /tmp/uwsgi.sock
chmod-socket = 664
uid = www-data
gid = www-data
vacuum = true

die-on-term = true
```

Step 2: Configuration

Create a systemd configuration file for the uwsgi server:

- sudo vi /etc/systemd/system/uwsgi.service

```
[Unit]
Description=uWSGI service

[Service]
ExecStart=/opt/alerta/bin/uwsgi --ini /etc/uwsgi.ini

[Install]
WantedBy=multi-user.target
```

Start the uwsgi server, check the current status and enable it to start on reboot:

- sudo systemctl start uwsgi
- sudo systemctl status uwsgi
- sudo systemctl enable uwsgi

Step 2: Configuration

Configure nginx to serve Alerta as a uWsgi application on /api and the web console as static assets.

- sudo vi /etc/nginx/sites-enabled/default

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        location /api { try_files $uri @api; }
        location @api {
            include uwsgi_params;
            uwsgi_pass unix:/tmp/uwsgi.sock;
            proxy_set_header Host $host:$server_port;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        location / {
                root /var/www/html;
                try_files $uri $uri/ /index.html;
        }
}
```

Step 2: Configuration

Restart nginx so that it picks up the new configuration:

- sudo service nginx restart

Modify the existing web console config.json configuration file to set the endpoint to /api:

- sudo vi /var/www/html/config.json

  {"endpoint": "/api"}

Step 3: Customization

Firstly, generate a random string:

```
$ cat /dev/urandom | tr -dc A-Za-z0-9_\!\@\#\$\%\^\&\*\(\)-+= | head -c 32 && echo
```

Assign the random string to the `SECRET_KEY` sever setting:

```
$ vi /etc/alertad.conf
```

```
SECRET_KEY='<INSERT_RANDOM_STRING>'
```

Step 3: Customization

Add a new severity level called “Fatal” as the highest possible severity and remove some unwanted severity levels:

```
$ vi /etc/alertad.conf
```

```
SEVERITY_MAP = {
    'fatal': 0,
    'critical': 1,
    'warning': 4,
    'indeterminate': 5,
    'ok': 5,
    'unknown': 9
}
```

The default “normal” severity is `normal` however it is no longer a valid severity following the changes made above so you must explicity set the default severity to `ok` by adding:

```
DEFAULT_NORMAL_SEVERITY = 'ok'
```

Step 3: Customization

Configure the default “reject” plugin to allow the additional alert environment called “Code” and not just “Production” or “Development”:

```
$ vi /etc/alertad.conf
```

```
PLUGINS=['reject']
ALLOWED_ENVIRONMENTS=['Production', 'Development', 'Code']
```

Make sure you restart uwsgi so that the Alerta API picks up the new severity and plugin configurations:

```
$ sudo service uwsgi restart
```

Step 3: Customization

To use the command-line tool to submit a test alert you first need to create a configuration file that defines what API endpoint to use:

```
$ vi $HOME/.alerta.conf
```

```
[DEFAULT]
endpoint = http://localhost/api
```

Send a test “fatal” alert and confirm it has been received by viewing it in the web console:

- /opt/alerta/bin/alerta send -r net01 -e down -s fatal -E Code -S Network -t 'net01 is down.’
- /opt/alerta/bin/alerta query

Integrations

Core

There are a few core integrations which have been developed to showcase how easy it is to get alerts or events from other tools into Alerta. They are:

- Nagios Event Broker - forward host/service check results with suppression during downtime
- InfluxData Kapacitor - forward alerts for metric anomalies and dynamic thresholds
- Zabbix Alert Script - forward problems, acknowledged and OK events
- Sensu Plugin - forward Sensu events
- Riemann Plugin - generate alerts from thresholds defined against metric streams
- Kibana Logging - log alerts to Elasticsearch for historical visualization of alert trends

Integrations

Contrib

There are several more integrations available in the contrib repo which may be useful. They are:

- Amazon SQS - receive alerts from SQS that were sent using the SNS core plugin
- E-mail - send emails after a hold-time has expired (requires the `AMQP`_ message queue core plugin)
- Opsweekly - query Alerta to generate Opsweekly reports
- Pinger - generate ping alerts from list of network resources being pinged
- SNMP Trap - generate alerts from SNMPv1 and SNMPv2 sources
- Supervisor - trigger alerts and heartbeats based on process deamon events
- Syslog Forwarder - receive **RFC 5424**, **RFC 3164** syslog and Cisco syslog messages
- URL monitor - trigger alerts from web service query responses

Integrations

External

Some third-party monitoring tools have built-in support for Alerta. They are:

- elastalert - alerting on anomalies, spikes, or other patterns of interest from data in Elasticsearch
- netdata - a system for distributed real-time performance and health monitoring
- Tick Stack - designed to handle metrics and events using Telegraf, InfluxDB, Chronograf, and Kapacitor

Integrations

Webhooks

Webhooks are a way of integrating with other systems by triggering HTTP callbacks to the Alerta server API when an event occurs.

Built-in Webhooks

- AWS CloudWatch
- Grafana
- Graylog
- New Relic
- PagerDuty
- Pingdom
- Prometheus Alertmanager
- Riemann
- SeverDensity
- Slack
- Google Stackdriver
- Telegram

BGPalerter

Self-configuring BGP monitoring tool, which allows you to monitor in real-time if:

- Developed by NTT
- any of your prefixes loses visibility;
- any of your prefixes is hijacked;
- your AS is announcing RPKI invalid prefixes (e.g., not matching prefix length);
- your AS is announcing prefixes not covered by ROAs;
- any of your ROAs is expiring;
- ROAs covering your prefixes are no longer reachable;
- RPKI Trust Anchors malfunctions;
- a ROA involving any of your prefixes or ASes was deleted/added/edited;
- your AS is announcing a new prefix that was never announced before;
- an unexpected upstream (left-side) AS appears in an AS path;
- an unexpected downstream (right-side) AS appears in an AS path;
- one of the AS paths used to reach your prefix matches a specific condition defined by you

Running BGPalerter from binaries

Linux

Download the binary:
wget https://github.com/nttgin/BGPalerter/releases/latest/download/bgpalerter-linux-x64

Download config.yml.example as config.yml (in the same directory of the binary)

Make the binary executable (e.g., chmod +x bgpalerter-linux-x64)

Auto-configure it: ./bgpalerter-linux-x64 generate -a _YOUR_ASN_ -o prefixes.yml -i -m

Run it: ./bgpalerter-linux-x64
Or use nohup ./bgpalerter-linux-x64 & to leave it running after you close the terminal

Create systemd service file

Linux

sudo vi /etc/systemd/system/bgpalerter.service

```
[Unit]
Description=BGPalerter
After=network.target

[Service]
Type=simple
Restart=on-failure
RestartSec=30s
User=bgpalerter
WorkingDirectory=/home/mmix
ExecStart=/home/mmix/bgpalerter-linux-x64

[Install]
WantedBy=multi-user.target
```

Create systemd service file

Reload systemd

systemctl daemon-reload

Enable and start the service

systemctl enable bgpalerter

systemctl start bgpalerter

Automatic Updates

vi upgrade.sh

```
#!/usr/bin/env bash

# Change the directories if needed
DIR=/home/bgpalerter
LOGS=$DIR/logs

# If log file does not exist, create it
if [ ! -f $LOGS/upgrade.log ]; then
  touch $LOGS/upgrade.log
  chown bgpalerter:bgpalerter $LOGS/upgrade.log
fi

# Delete log file if larger than 5MB
find $LOGS -type f -name "upgrade.log" -size +5M -delete

exec 1>> $LOGS/upgrade.log 2>&1

cd $DIR

# Download the latest version and save it to a temp file
wget --no-verbose -O bgpalerter-linux-x64.tmp
https://github.com/nttgin/BGPalerter/releases/latest/downloa
d/bgpalerter-linux-x64

# Set permissions to execute the file
chmod +x bgpalerter-linux-x64.tmp

# Set variables to compare versions
if [ -f bgpalerter-linux-x64 ]; then
  # If a file exists already
  v1=$(./bgpalerter-linux-x64 -v)
  v2=$(./bgpalerter-linux-x64.tmp -v)

else
  v1=$"0"
  v2=$(./bgpalerter-linux-x64.tmp -v)
fi
```

Automatic Updates

vi upgrade.sh

```
# If there is no old version
if [ "$v1" == "0" ];then
  #Remove the file
  rm bgpalerter-linux-x64.tmp

  echo "This script upgrades BGPalerter; however, $DIR/bgpalerter-linux-x64 cannot be found. Please, install it first https://github.com/nttgin/BGPalerter/blob/main/docs/linux-service.md"
  exit 1

# The versions are different
elif [ "$v1" != "$v2" ];then
  # Remove the old version
  rm bgpalerter-linux-x64

  # Rename the temp file
  mv bgpalerter-linux-x64.tmp bgpalerter-linux-x64

# Kill the process
  # We use kill because "systemctl restart bgpalerter" works only
  # if you run this script as root. Systemctl will restart the process.
  kill -9 $(cat $DIR/bgpalerter.pid) || true

  echo "A new version of BGPalerter has been installed"

else
  # If the versions are the same, delete the temp file
  rm bgpalerter-linux-x64.tmp
  echo "BGPalerter is up to date"
fi
```

Automatic Updates

chmod +x upgrade.sh

Perform the check weekly

crontab –e

0 0 * * 0 /home/mmix/upgrade.sh

config.yml (Report Setting)

Local report.log

```
reports:
  - file: reportFile
    channels:
      - hijack
      - newprefix
      - visibility
      - path
      - misconfiguration
      - rpki
      - roa
    params:
      persist: true
```

Alerta API

```
reports:
  - file: reportAlerta
    channels:
      - hijack
      - newprefix
      - visibility
      - path
      - misconfiguration
      - rpki
      - roa
    params:
      key: "33mPuwTxX4OCs1TyU4lb9MdTYeyzGlilvjQOh7jY"
      environment: Production
      severity:
        hijack: critical
        newprefix: informational
        visibility: warning
        path: warning
        misconfiguration: major
        rpki: major
        roa: major
```

```
resourceTemplates:
      default: "${type}"
      hijack: "hijack::${prefix}@@${asn}"
      newprefix: "newprefix::${prefix}@@${asn}"
      visibility: "visibility::${prefix}@@${asn}”
      path: "path::${prefix}@@${asn}"
    urls:
      default: http://10.0.2.95/api
      noc: http://10.0.2.95/api
```

API Key test

```
curl -i -X POST http://10.0.2.95/api/alert \
  -H "Authorization: Key 33mPuwTxX4OCs1TyU4lb9MdTYeyzGlilvjQOh7jY" \
  -H "Content-Type: application/json" \
  -d '{
    "resource": "84.205.68.0/24",
    "event": "APITest",
    "environment": "Production",
    "severity": "critical",
    "service": ["BGPalerter"],
    "text": "Testing API key auth"
  }'
```

Path Monitoring

- Each item in the path list is a matching rule. Each matching rule is composed of:
- match, the regular expression that will be tested on each AS path. If the expression tests positive, the BGP message triggers an alert. ASns are comma separated.
- notMatch, the regular expression that will be tested on each AS path. If the expression tests positive, the BGP message will not trigger an alert. ASns are comma separated.
- matchDescription, the description that will be reported in the alert in case the matchin rule results in a match.
- maxLength, the maximum length allowed for an AS path. Longer paths will trigger an alert.
- minLength, the minimum length allowed for an AS path. Shorter paths will trigger an alert.

Path Monitoring (Example)

```
165.254.255.0/24:
  asn: 15562
  description: an example on path matching
  ignoreMorespecifics: false
  path:
   - match: ".*2194,1234$"
     notMatch: ".*5054.*"
     matchDescription: detected scrubbing center
   - match: ".*123$"
     notMatch: ".*5056.*"
     matchDescription: other match
```

Path Monitoring (Example)

```
103.116.192.0/24:
  description: No description provided
  asn:
    - 137955
  ignoreMorespecifics: false
  path:
    - match: ".*174,136168,137955$"
      matchDescription: detected cogent path
    - match: ".*45629,136168,137955$"
      matchDescription: detected Jastel path
    - match: ".*2914,136168,137955$"
      matchDescription: detected NTT path
    - match: ".*10099,149855,137955$"
      matchDescription: detected China Unicom path
  ignore: false
  group: noc
```

Testing rules

- https://lg.as6453.net/