NBIP nationale beheersorganisatie internet providers

Processing BGP updates with RabbitMQ – more in depth

virtualUKNOF#52

September 7th 2020

“Pim van Stam” <pim@nbip.nl>

NBIP nationale beheersorganisatie internet providers

whois STAM-RIPE

whois ORG-SN292-RIPE

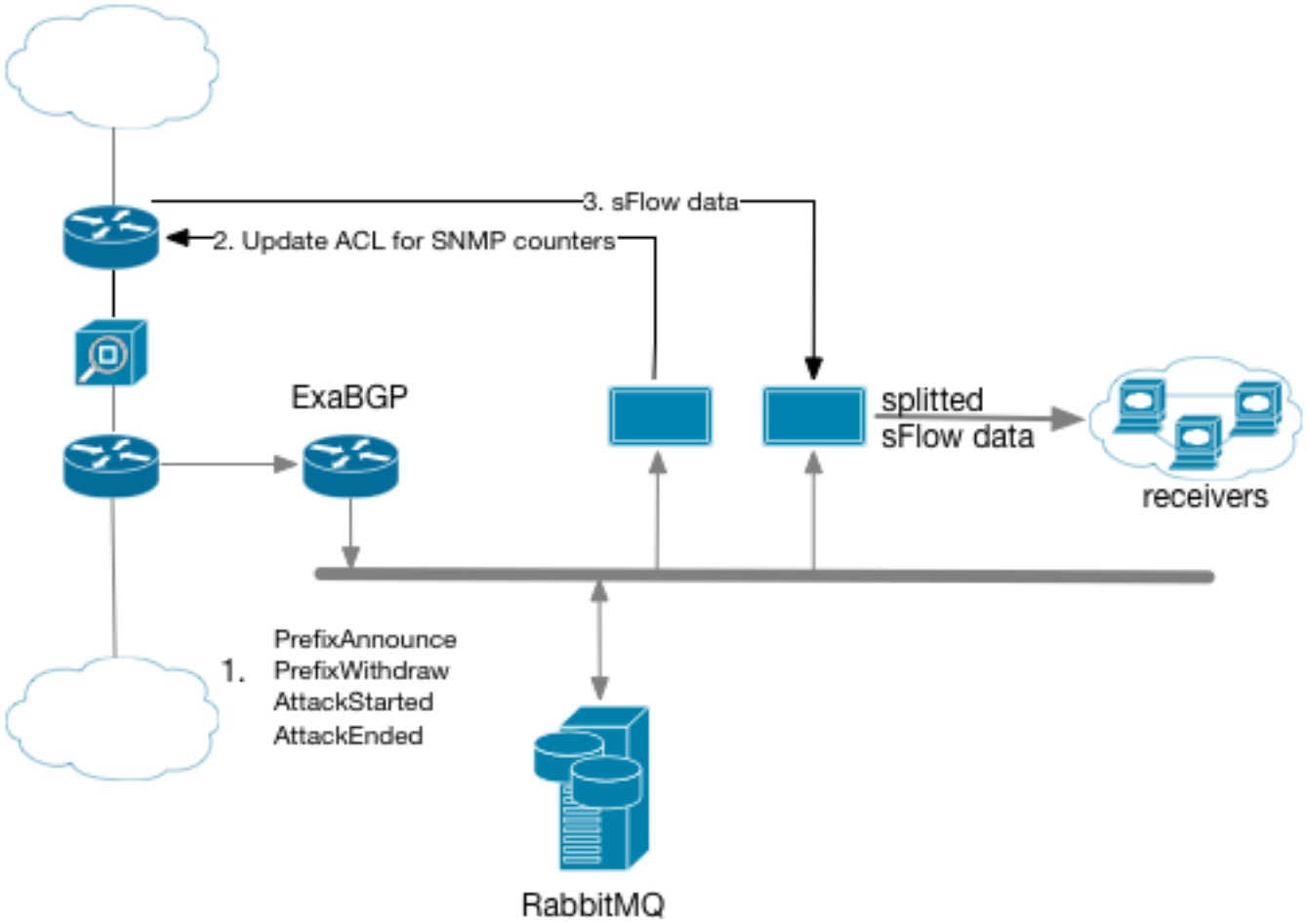
Idea

NBIP nationale beheersorganisatie internet providers

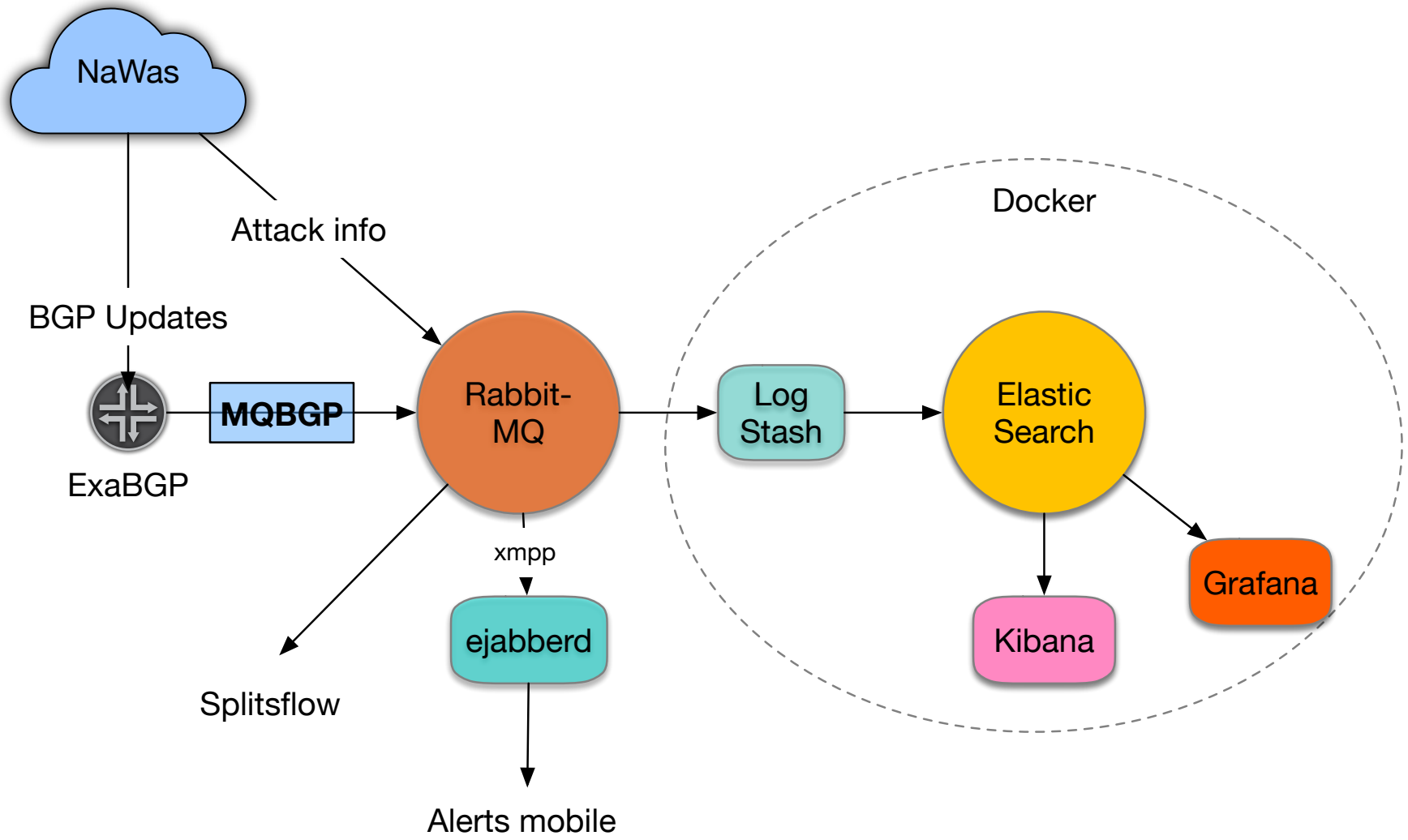
- Prefix updates needed for several tasks:
  - Split sFlow data based on destination AS number
  - Modify filtering based on AS-number of customer
  - Activate filtering
- Building blocks for process BGP updates and several tasks using it
- Scalable
- Hook into framework for pattern recognition project
- Easy integrations for alerting on smartphone

Infrastructure

NBIP nationale beheersorganisatie internet providers



Integrations



Components

NBIP nationale beheersorganisatie internet providers

- Core infrastructure with Juniper MX routers
- ExaBGP for BGP updates
- Python library for sending and receiving messages - MQBGP
- RabbitMQ message broker
- Definition of messages on the bus
- Receivers
  - Alerts per mail for BGP prefix changes (advertise/withdraw) -> prefix_listener
  - ACL switch voor SNMP data
  - Splitting sFlow on destination AS: splitsflow.py
  - Creating customer profile in mitigation device with REST API
  - Starting tcpdump for analysing data

Mqbgp library

NBIP nationale beheersorganisatie internet providers

- In python
- On github
- For client (receiving) and server (sender) side
- BGP messages (from exabgp for instance)
- Queuing and listener for subscribing to RabbitMQ

Mqbgp (2)

NBIP nationale beheersorganisatie internet providers

- Class PrefixMessage
  - set_message()
- Class PrefixListMessage
  - set_list()
  - get_list()
  - add_prefix()
- Class PrefixListRequestMessage (not implemented yet)
- Class Queue
  - connect() and disconnect()
  - publish_message()
  - subscribe()
- Class Listener
  - listen()
  - convert_to_message_object()

Exabgp

- Connected to MX for receiving partly or full table
- Exabgp config for receiving prefixes
- BGP monitor python script

Exabgp (2)

NBIP nationale beheersorganisatie internet providers

- Junos config like:

```
neighbor 172.30.4.2 {
    description "ExaBGP monitoring and control";
    peer-as 65521;
          export [ rp-bgp-out-nexthopself rp-allow-myas rp-allow-customer rp-any-reject-all ];
          import rp-any-reject-all;
}
```

- Replace export policy for something like advertise_everything to send full table

Exabgp (3) - exabgp.conf

NBIP nationale beheersorganisatie internet providers

```
process bgpmonitor {
          run /etc/exabgp/bgp-monitor4.py;
          encoder json;}

template {
          neighbor test {
                    api {
                              processes [ bgpmonitor ];
                              neighbor-changes;
                              receive {
                                        parsed;
                                        update;
                              }
                              send {
                                        packets;
                              }}}}

neighbor 172.30.4.1 {
          description "Receive route info from members";
          inherit test;
          router-id 172.30.4.1;
          local-address 172.30.4.2;
          local-as 65521;
          peer-as 200020;
          hold-time 1200;
}
```

Exabgp (4) - bgp_monitor4 snippets

```
class exabgp_message()
     decode_json()
class neighbor()
     decode_json()

def get_asn()
def get_community()
def get_prefixes()
def del_prefixes()
def send_bgp_message()
      pm = mqbgp.PrefixMessage(asn, ip_prefix, nexthop, announce, as_path, community)

main:
mq = mqbgp.Queue()
mq.connect()
examsg = exabgp_message()

While True:
     line = sys.stdin.readline().strip()
     jdata = json.loads(line)
     examsg.decode_json(jdata)
     send_bgp_message(examsg.time, asn[0], pr, nexthop, True, asn[1], comm)
```

prefix_listener - snippets

NBIP nationale beheersorganisatie internet providers

```
def send_message(subject, msg):
     emsg = email.mime.text.MIMEText(msg)
     s = smtplib.SMTP(mailserver)
     s.sendmail(mailfrom, mailto, emsg.as_string())

def callback_prefix_updates(message:mqbgp.PrefixMessage):
     msg = message.get_message()
     subject = "Prefix change …"
     body = "Prefix change:"
     body += "\nType: " + msg['type']
     body += "\nPrefix: " + msg['prefix']
     body += "\nAS number: " + msg['asn']
     body += "\nNexthop: " + msg['nexthop']
     body += "\nAS Path: " + str(msg['as_path'])
     body += "\nCommunity: " + str(msg['community'])
     send_message(subject, body)

def mainroutine():
     while True:
          li = mqbgp.Listener("/etc/exabgp/config.yml")
          li.listen(mqbgp.PrefixMessage, callback_prefix_updates)

with daemon.DaemonContext(stderr=fileout, stdout=fileout):
            mainroutine()
```

NBIP nationale beheersorganisatie internet providers

Questions?

Thank You