Flow Monitoring in bwNET and BelWü

RIPE84 - MAT WG - 18.05.2022

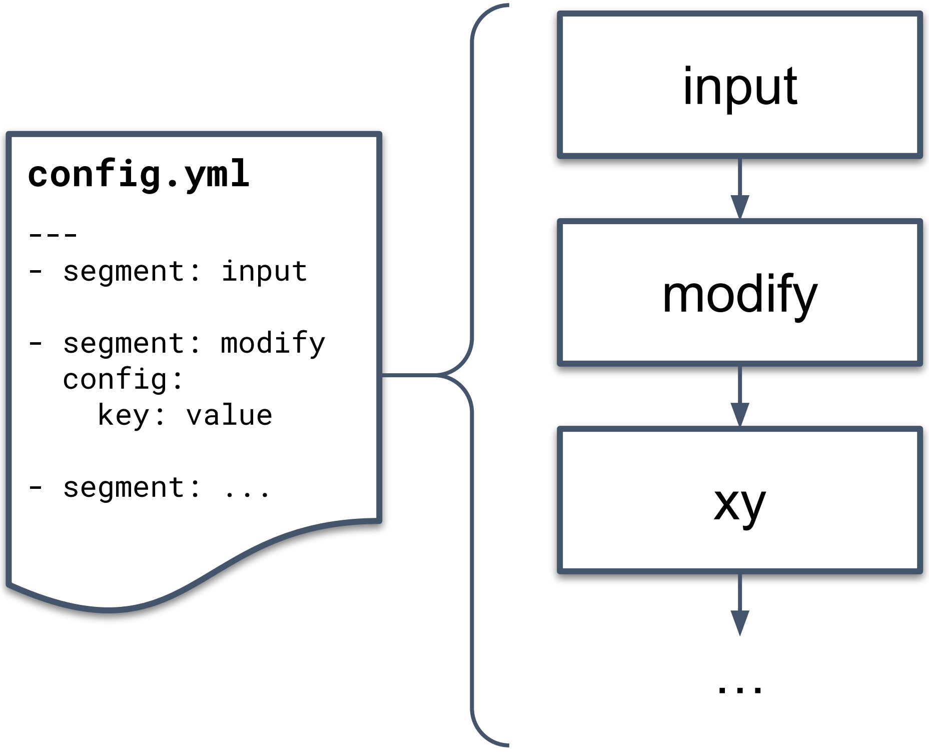
Daniel Nägele

naegele@belwue.de

Map by Users David Liuzzo and Nordwestern, commons.wikimedia.org

flowpipeline Tooling

- completely configuration-defined
- flowpipelines process any form of network flows
- segments act on single flows and pass them along
- many different segments are available:
  - multiple inputs and outputs
  - modification (extension using external sources, annotation, anonymization, …)
  - filtering (statistical, query language, …)
  - exporters, dataset generation



```
config.yml
---
- segment: input

- segment: modify
  config:
    key: value

- segment: ...
```

Code available here:
https://github.com/bwNetFlow/flowpipeline

Input segments

eBPF

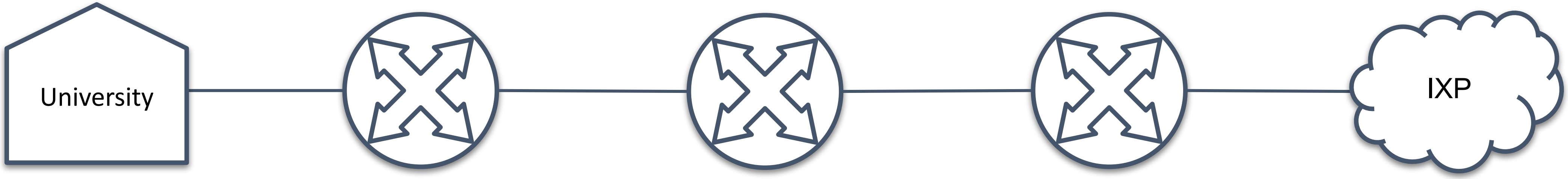
- use eBPF to dump packet headers
- match packets to flows in custom cache using 5-tuple
- export to pipeline

GoFlow 2

- use Goflow v2 to listen for flows in raw format
- supports network devices with sFlow, IPFIX or Netflow v9

APACHE kafka

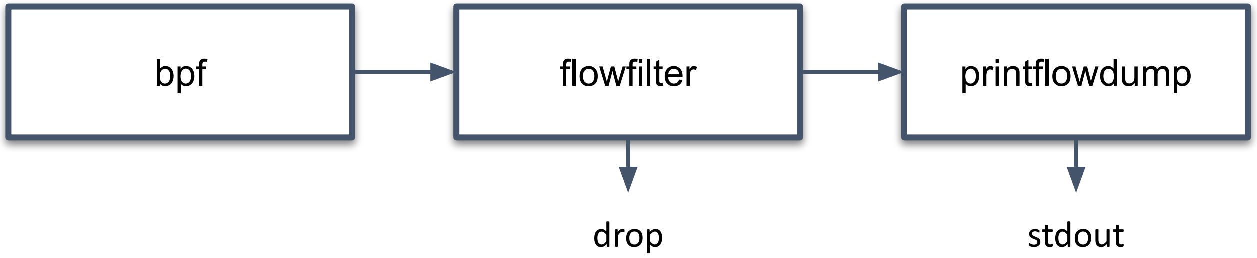
- receive flows generated by another flowpipeline from a Kafka cluster
- flows can come pre-filtered or enriched

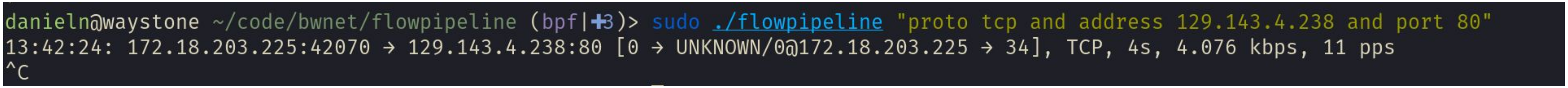


Flow-level tcpdump from eBPF

config.yml

```
---
- segment: bpf
  config:
     device: eth0

- segment: flowfilter
  config:
    filter: $0

- segment: printflowdump
```



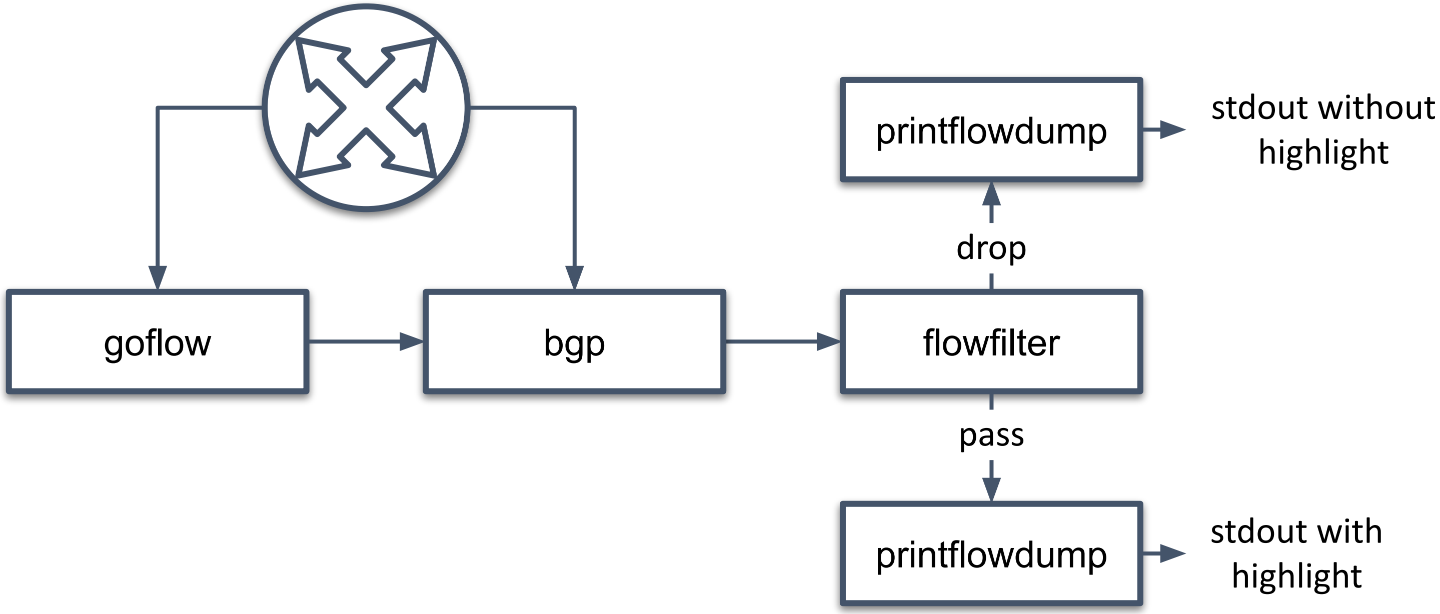


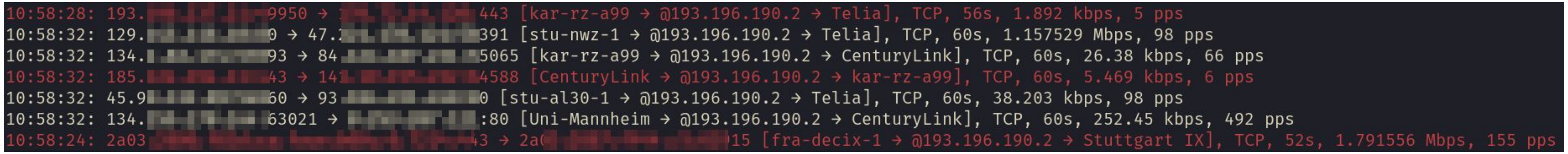
```
danieln@waystone ~/code/bwnet/flowpipeline (bpf|✚3)> sudo ./flowpipeline "proto tcp and address 129.143.4.238 and port 80"
13:42:24: 172.18.203.225:42070 → 129.143.4.238:80 [0 → UNKNOWN/0@172.18.203.225 → 34], TCP, 4s, 4.076 kbps, 11 pps
^C
```

Checking for RPKI Invalids

config.yml

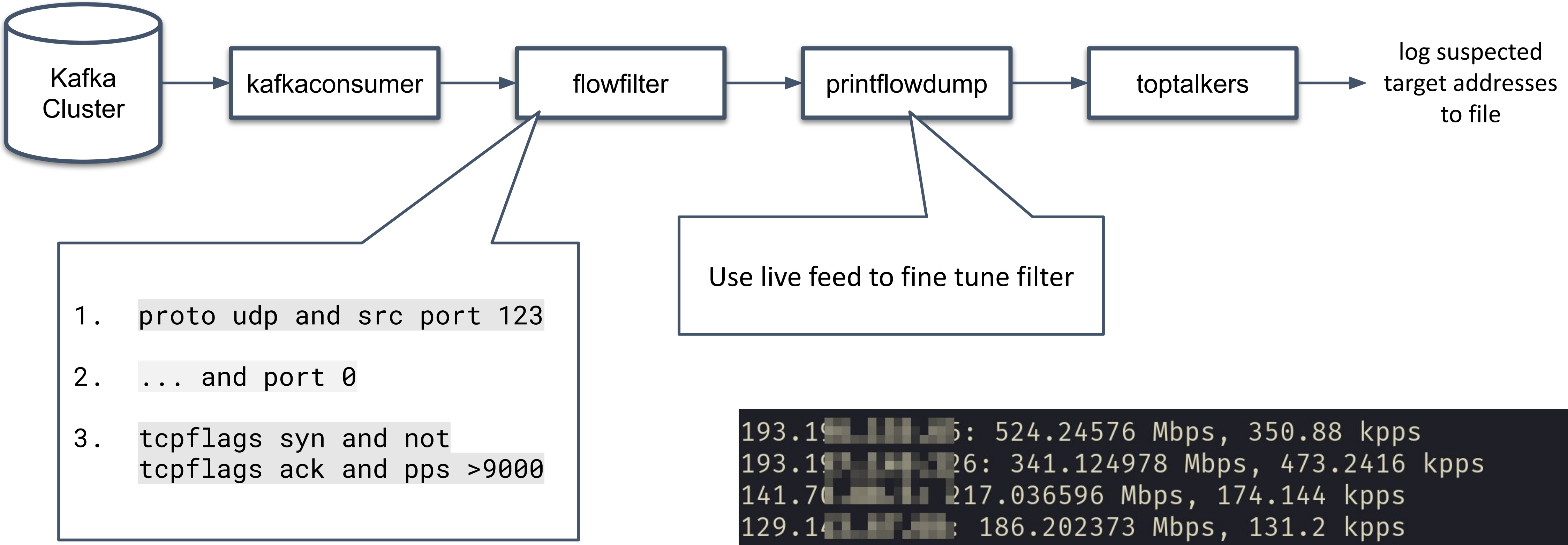
```
---
- segment: goflow

- segment: bgp
  config: ...

- segment: branch
  if:
  - segment: flowfilter
    config:
      filter: rpki invalid
  then:
  - segment: printflowdump
    config:
      highlight: true
  else:
  - segment: printflowdump
```





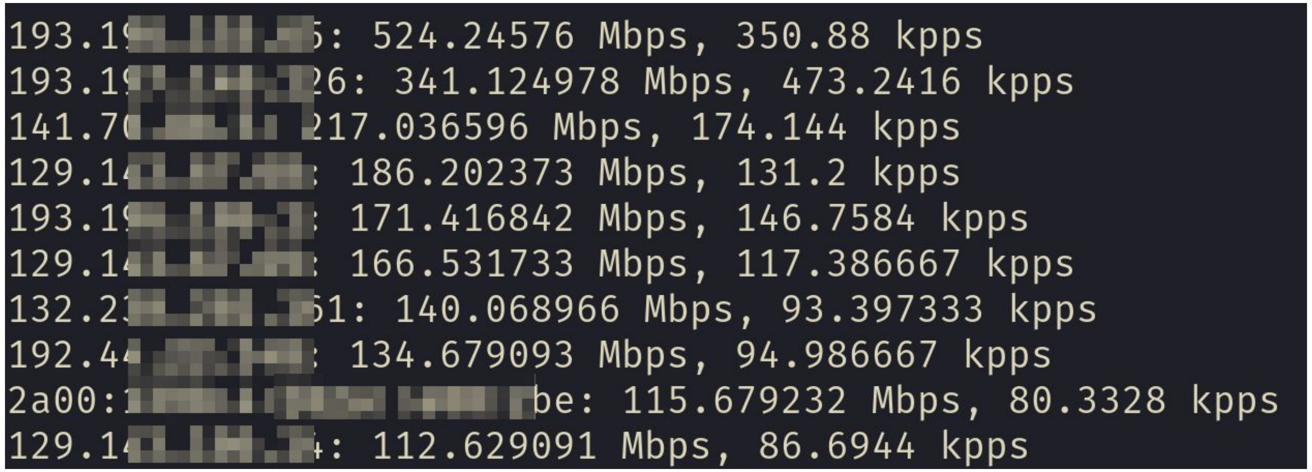
```
10:58:28: 193.[redacted]9950 → [redacted]443 [kar-rz-a99 → @193.196.190.2 → Telia], TCP, 56s, 1.892 kbps, 5 pps
10:58:32: 129.[redacted]0 → 47.[redacted]391 [stu-nwz-1 → @193.196.190.2 → Telia], TCP, 60s, 1.157529 Mbps, 98 pps
10:58:32: 134.[redacted]93 → 84[redacted]5065 [kar-rz-a99 → @193.196.190.2 → CenturyLink], TCP, 60s, 26.38 kbps, 66 pps
10:58:32: 185.[redacted]43 → 14[redacted]4588 [CenturyLink → @193.196.190.2 → kar-rz-a99], TCP, 60s, 5.469 kbps, 6 pps
10:58:32: 45.9[redacted]60 → 93[redacted]0 [stu-al30-1 → @193.196.190.2 → Telia], TCP, 60s, 38.203 kbps, 98 pps
10:58:32: 134.[redacted]63021 → [redacted]:80 [Uni-Mannheim → @193.196.190.2 → CenturyLink], TCP, 60s, 252.45 kbps, 492 pps
10:58:24: 2a03[redacted]3 → 2a[redacted]15 [fra-decix-1 → @193.196.190.2 → Stuttgart IX], TCP, 52s, 1.791556 Mbps, 155 pps
```

Basic DDoS Detection



```
1.  proto udp and src port 123

2.  ... and port 0

3.  tcpflags syn and not
    tcpflags ack and pps >9000
```

Use live feed to fine tune filter



```
193.1[redacted]: 524.24576 Mbps, 350.88 kpps
193.1[redacted]26: 341.124978 Mbps, 473.2416 kpps
141.7[redacted]217.036596 Mbps, 174.144 kpps
129.1[redacted]: 186.202373 Mbps, 131.2 kpps
193.1[redacted]: 171.416842 Mbps, 146.7584 kpps
129.1[redacted]: 166.531733 Mbps, 117.386667 kpps
132.2[redacted]51: 140.068966 Mbps, 93.397333 kpps
192.4[redacted]: 134.679093 Mbps, 94.986667 kpps
2a00:[redacted]be: 115.679232 Mbps, 80.3328 kpps
129.1[redacted]: 112.629091 Mbps, 86.6944 kpps
```

Thanks for your time!

Questions?

naegele@belwue.de - @debugloop (on social networks)