Route Leaks: Status Update

Alexander Azimov, Qrator Labs
<aa@qrator.net>

Definition

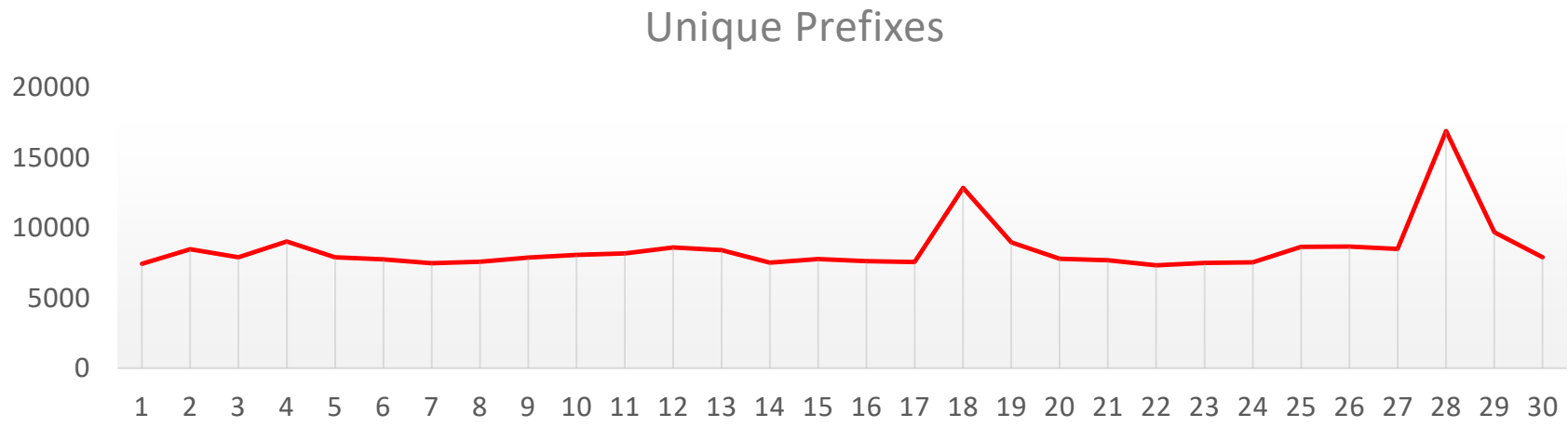
Route Leaks are propagation of BGP prefixes which violate assumptions of BGP topology relationships; e.g. passing a route learned from one peer to another peer or to a transit provider, passing a route learned from one transit provider to another transit provider or to a peer.

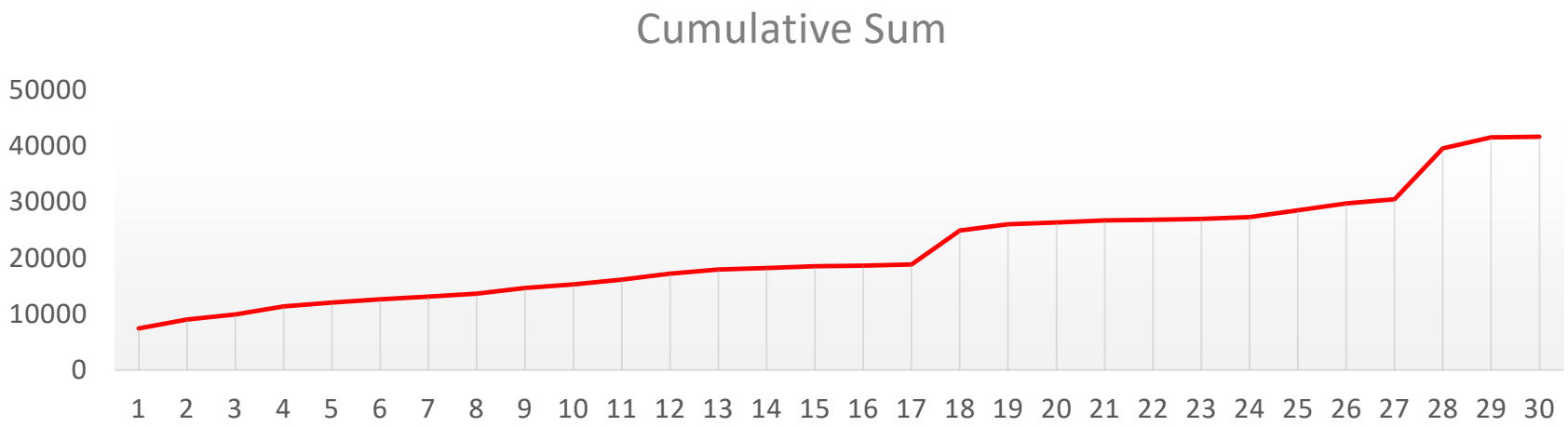
Leaked Prefixes

If your prefixes are leaked:

1. Increased delays;
2. DoS;
3. MiTM attack.

Leaked Prefixes



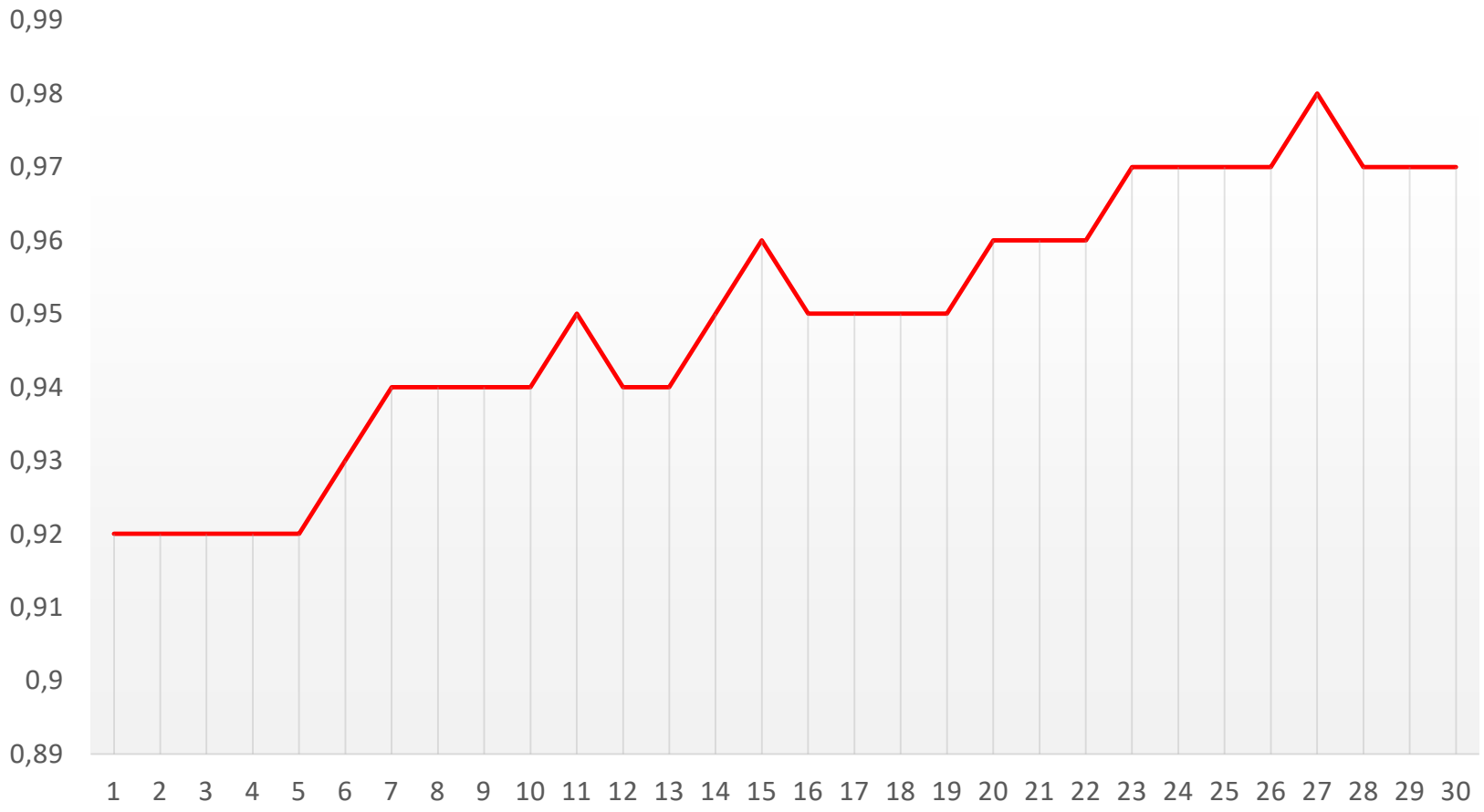


Accepting Leaked Prefixes

If your AS accepts leaked prefixes:

1. Increased delays;
2. DoS;
3. MiTM attack.

Accepting Leaked Prefixes

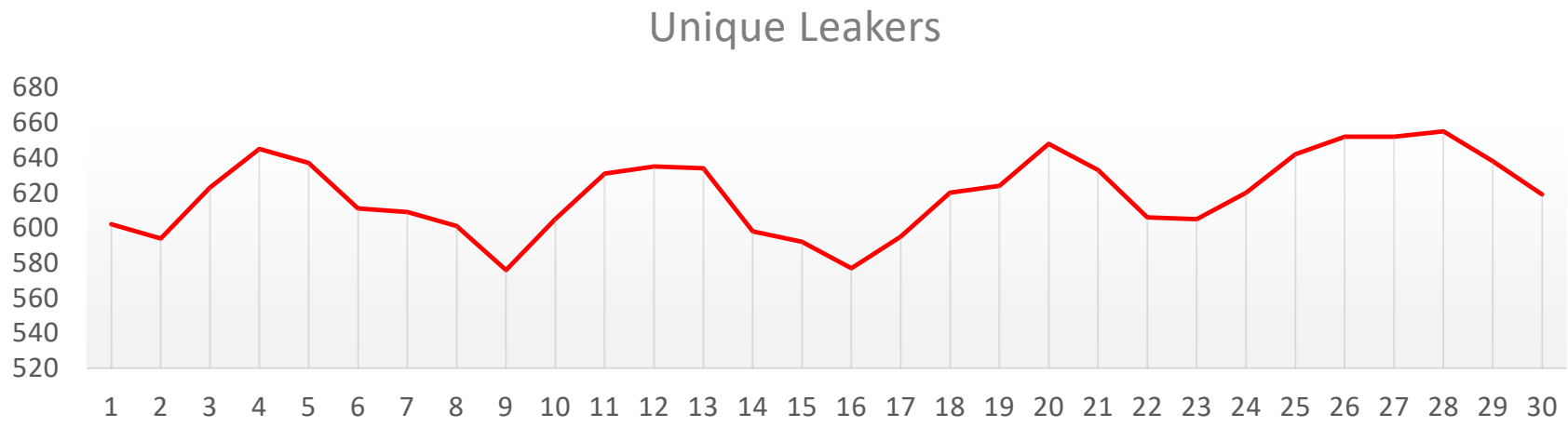


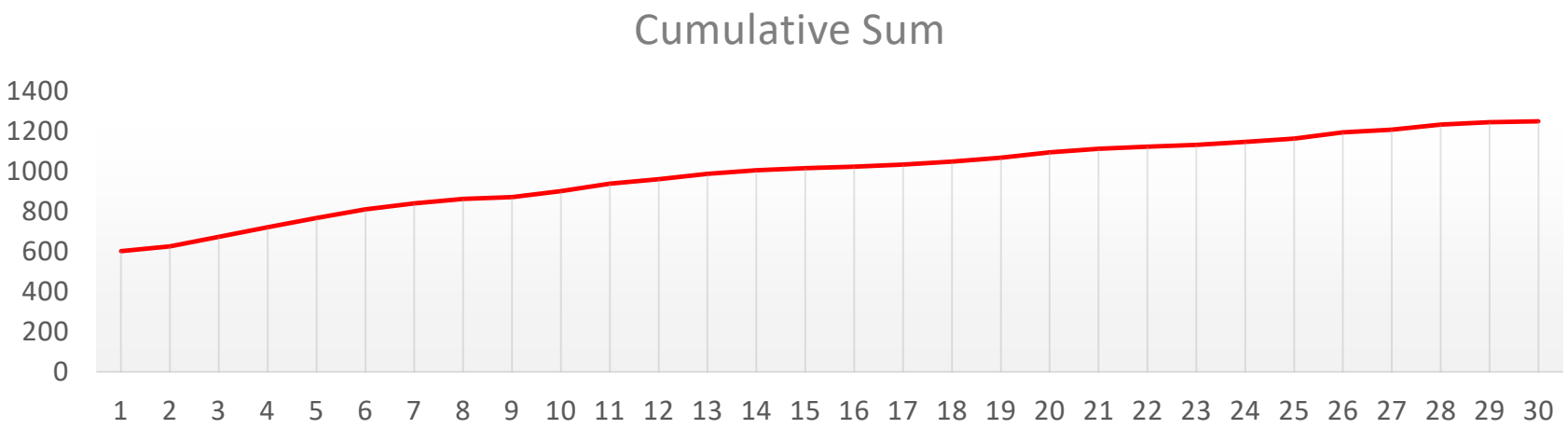
Leakers

If your AS leaks prefixes:

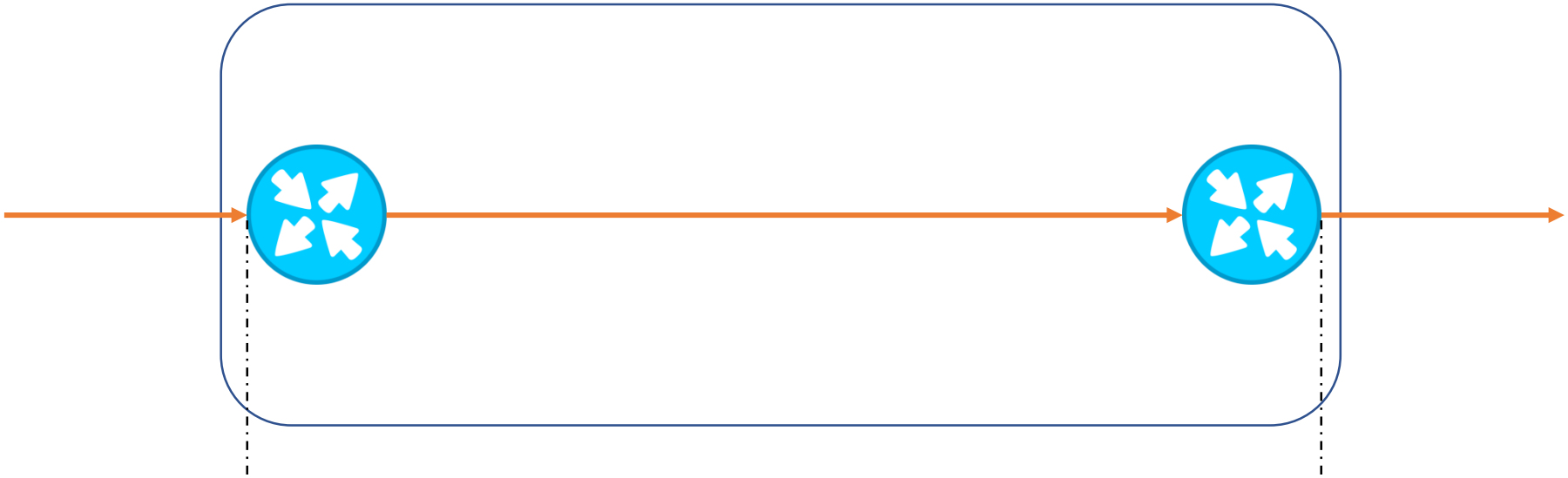
1. DoS attack, was it your goal?
2. MiTM attack, was it your goal?
3. If not, money loss, packet loss, reputation loss.

Leakers


Unique Leakers
680
660
640
620
600
580
560
540
520
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30


Cumulative Sum
1400
1200
1000
800
600
400
200
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Communities



Set communities depending on policy

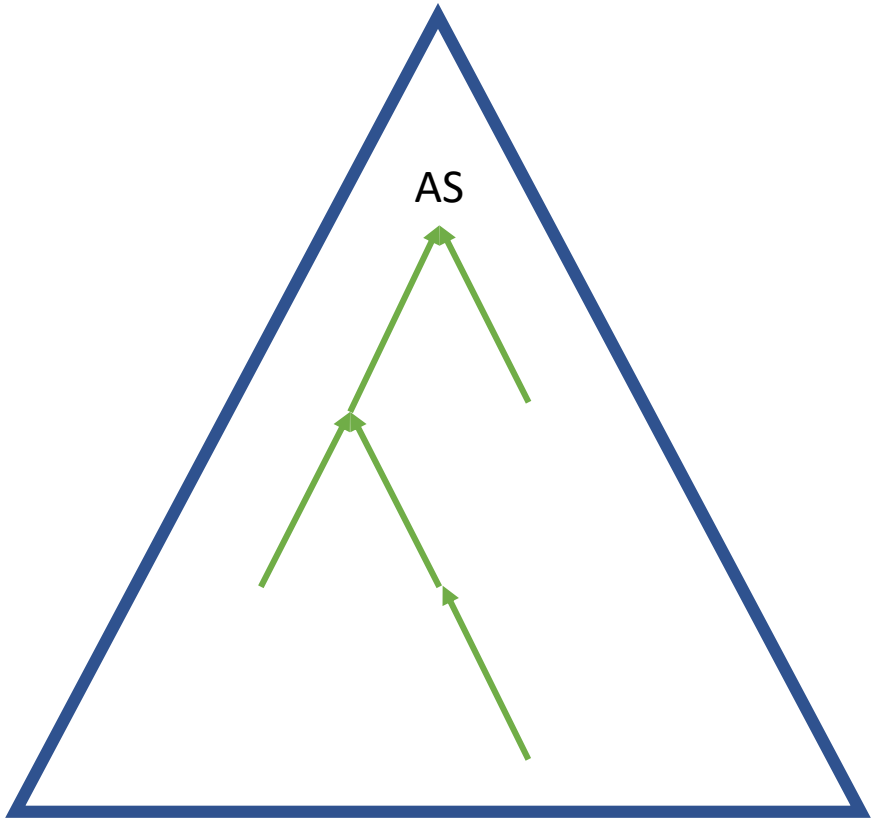
Filter routes depending on set communities

No enforcement of policy existence and its correctness

Proactive Approach

Build filters using AS cone.

Can we fully rely on AS-SET?

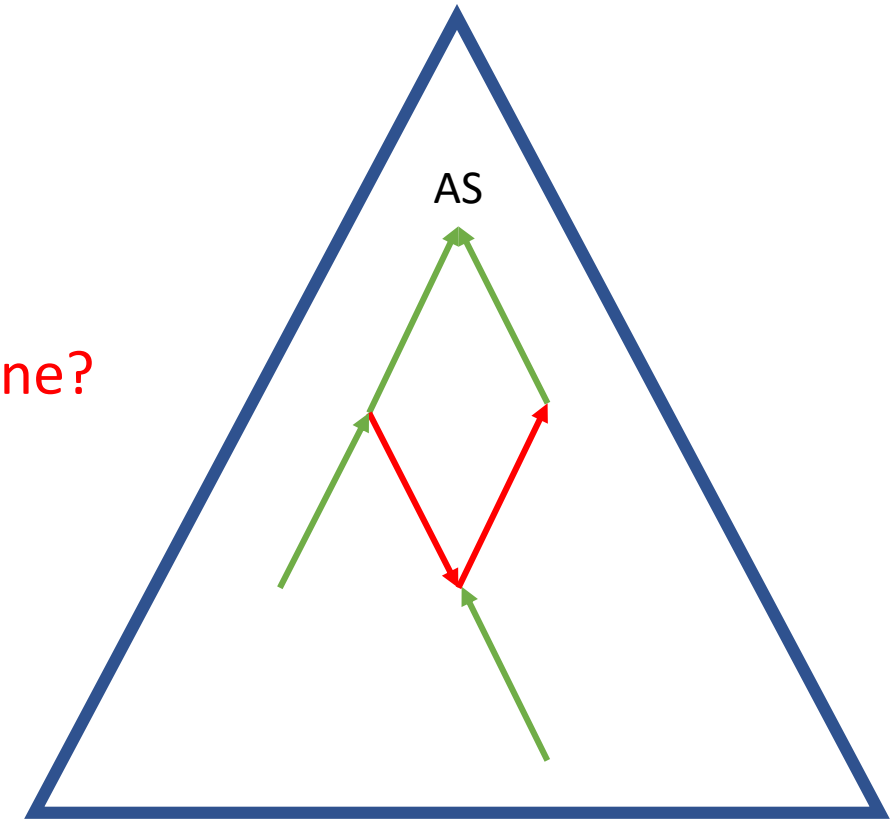


Proactive Approach

Build filters using AS cone.

Can we fully rely on AS-SET?

What if leak happens inside AS cone?



Monitoring

- BGPStream + Caida AS Relations;
- DYN/Renesys;
- Radar by Qrator.

Preliminary Results

- Well managed communities will prevent you from leaking;
- Well defined policy can filter some leaks;
- Monitoring can assist you in tracking route leaks;

No opportunity to stop leak propagation in automated way

Peering Relations/Roles

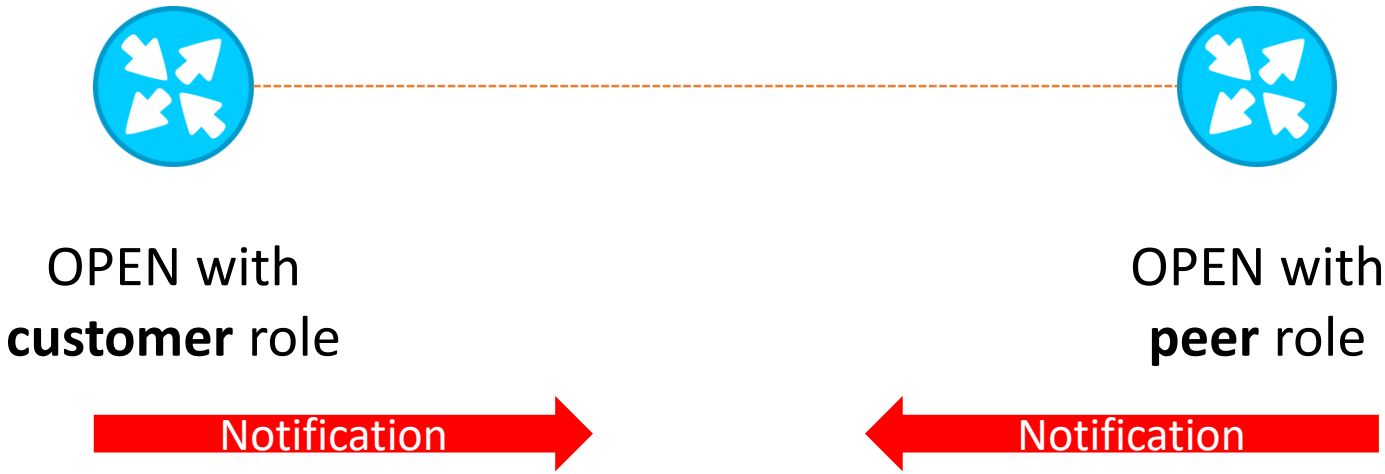
Provider: sends their own routes and (possibly) a subset of routes learned from their other customers, peers, and transit providers to their customer.

Customer: accepts 'transit routes' from its provider(s) and announces their own routes and the routes they have learned from the transitive closure of their customers to their provider(s).

Peer: announces their routes and the routes from their customer cone to other Peers.

Internal: announces all routes, accepts all routes.

BGP Roles



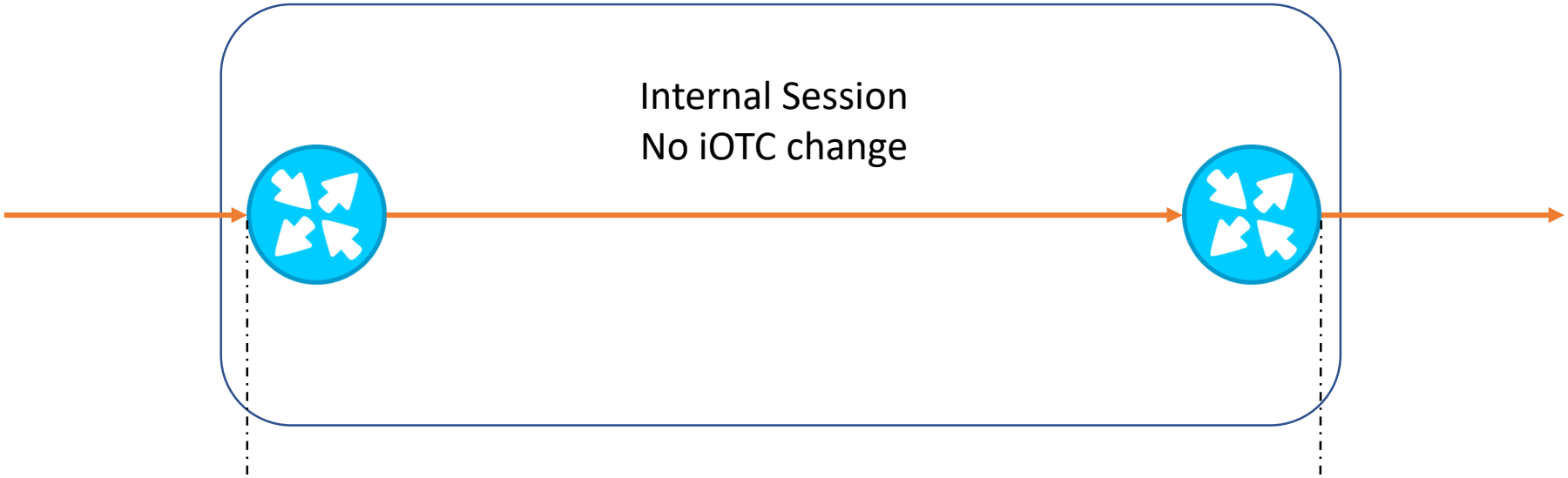
3 pairs of non-conflict roles:

1. Peer <---> Peer
2. Customer <---> Provider
3. Internal <---> Internal

Considerations

- Roles are native;
- Roles are not revealing any sensitive data to other parties;
- Roles have a number of applications.

Route Leak Prevention: iOTC

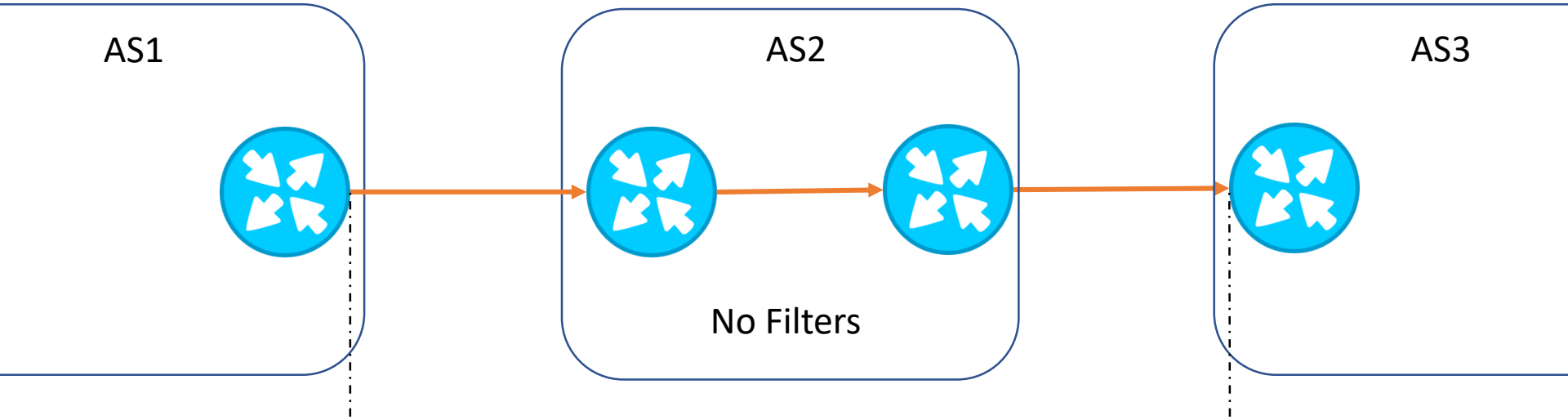


Set iOTC if role
is customer or peer

Filter routes if iOTC is set and
role is customer or peer

If route was learned from a provider or peer it should not be
announced to another provider or peer

Route Leak Detection: eOTC



If role is provider or peer
eOTC=AS1

If role is provider or peer
eOTC is set and eOTC!=AS2

If route was learned from a customer or peer and eOTC is set
and eOTC != neighbor AS then route was leaked

What should we do with Route Leak?

What should we do with Route Leak?

- What if there is no alternatives?
- What if somebody violated eOTC value?

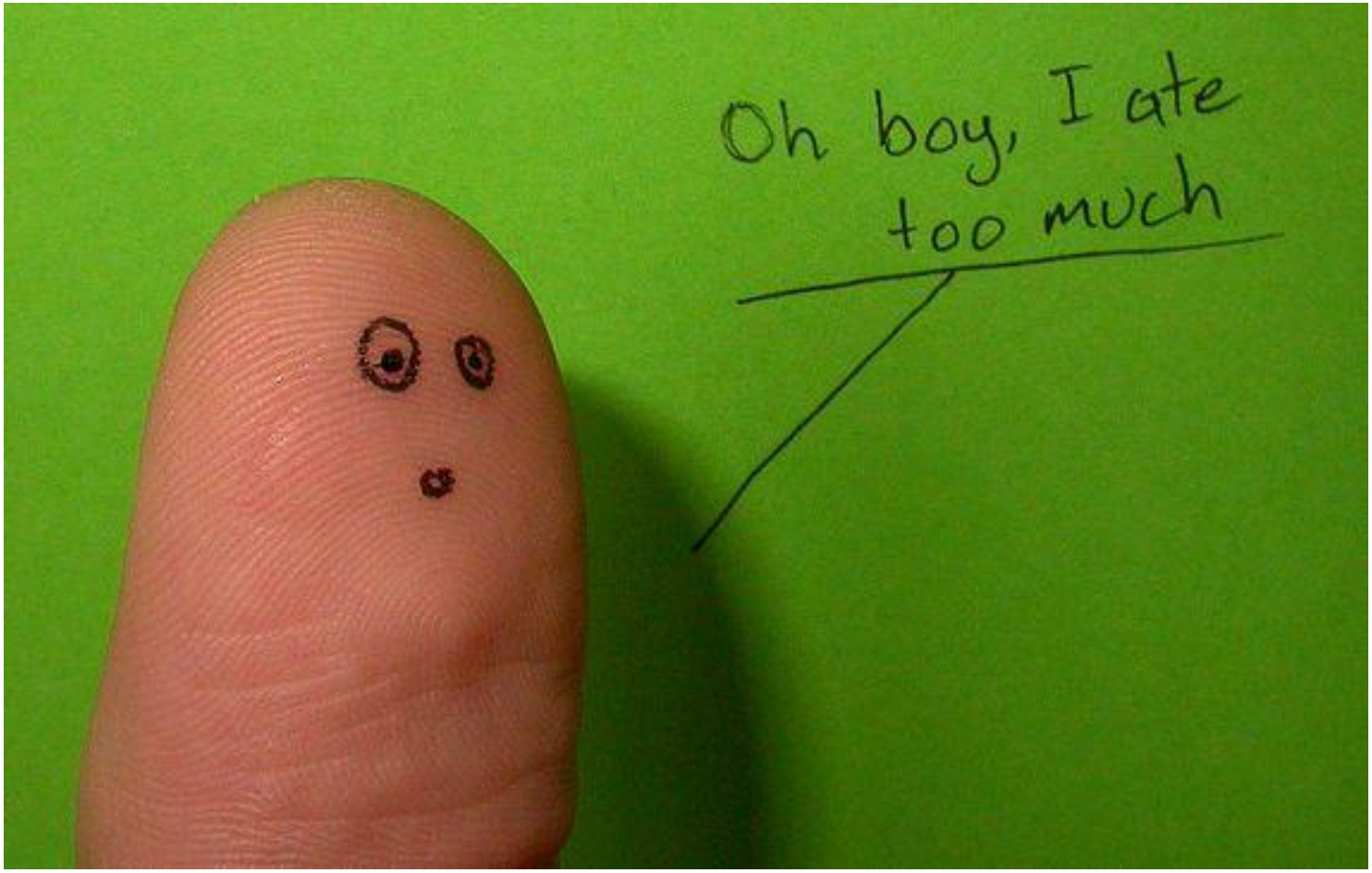
Deprioritization instead of filtering!

Implementation

```
protocol bgp IAMOPERATOR {
      local as MY_AS;
      neighbor X.X.X.X as AS_PROVIDER;
      role customer;
}
```

Github: https://github.com/QratorLabs/bird

No Fat Fingers Inside



IETF: Slow Motion

March 2016: OTC attribute and roles;

October 2016: OTC functionality split between eOTC and iOTC;

March 2017: clarification of peering relations, eOTC is moved to a separate draft.

Current version:

https://www.ietf.org/id/draft-ymbk-idr-bgp-open-policy-03

https://tools.ietf.org/html/draft-ymbk-idr-bgp-eotr-policy-00

IETF: Slow Motion

https://tools.ietf.org/html/rfc7908

Problem Definition and Classification

https://tools.ietf.org/html/draft-ietf-idr-route-leak-detection-mitigation

Alternative to eOTC

https://tools.ietf.org/html/draft-ietf-grow-bgp-reject

Change of BGP default behaviour

Results

- Well managed communities will prevent you from leaking;
- Well defined policy can filter some leaks;
- Monitoring can assist you in tracking route leaks;
- Roles + iOTC + eOTC can solve the general problem of route leaks that are result of mistake;
- Collaborate with IETF!!!
- Give us feedback: init.qrator.net/details/route-leak-mitigation