ASPA: Status Update

Alexander Azimov, Yandex

a.e.azimov@gmail.com

Fighting Route Leaks with AS-SET filters

Autonomous System Provider Authorization

ASPA

- customer_asn – signer
- provider_asns – authorized to send routes to upper providers or peers
- AFI – IPv4 or IPv6

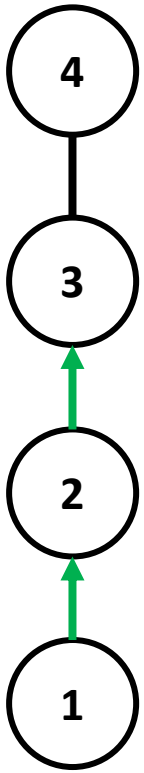
ASPA Pair Verification

1. Retrieve all cryptographically valid ASPAs in a selected AFI with a customer value of AS1. The union of SPAS forms the set of "Candidate Providers.“
2. If the set of Candidate Providers is empty, then the procedure exits with an outcome of **“Unknown.”**
3. If AS2 is included in the Candidate Providers, then the procedure exits with an outcome of **“Valid.“**
4. Otherwise, the procedure exits with an outcome of **“Invalid."**

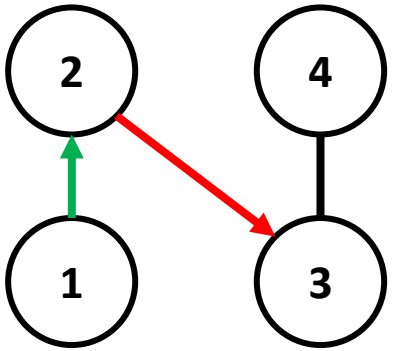
Terms

- Line goes up – route is announced from customer to provider;
- Line goes down – route is announced from provider to customer;
- Line goes straight – route is announced from peer to peer;
- The arrow shows the order of the ASPA check, not the route advertisement!

Route Received from Customer

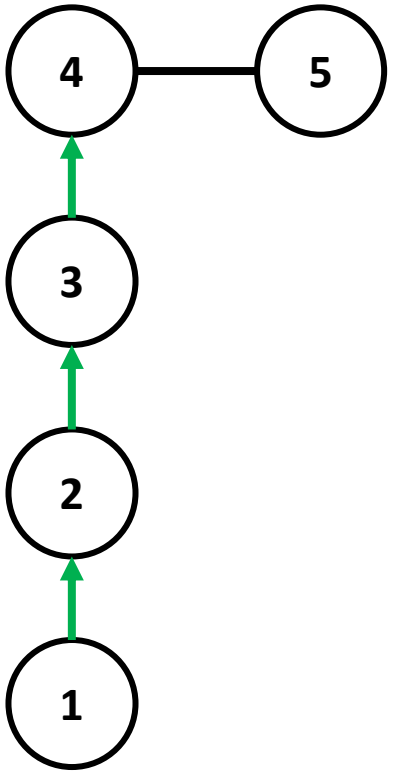


(1, 2), (2,3) are **Valid**
The path is **Valid**



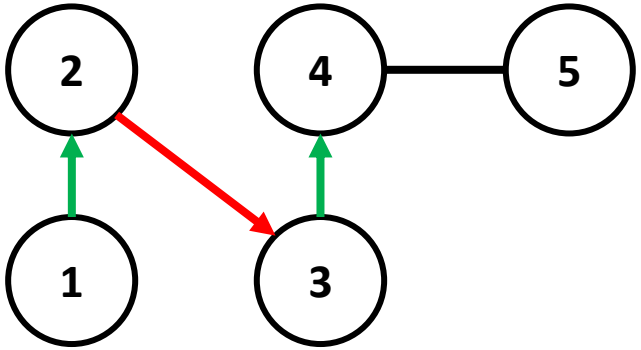
(1, 2) is Valid, (2, 3) is **Invalid**
The path is **Invalid**

Route Received from Peer



(1, 2), (2,3), (3,4) are **Valid**

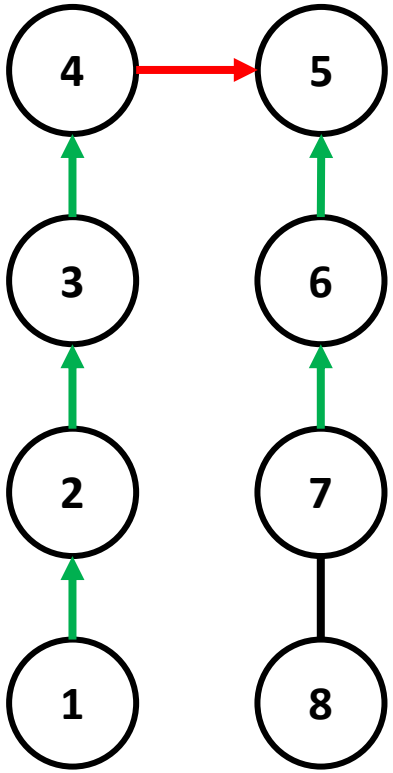
The path is **Valid**



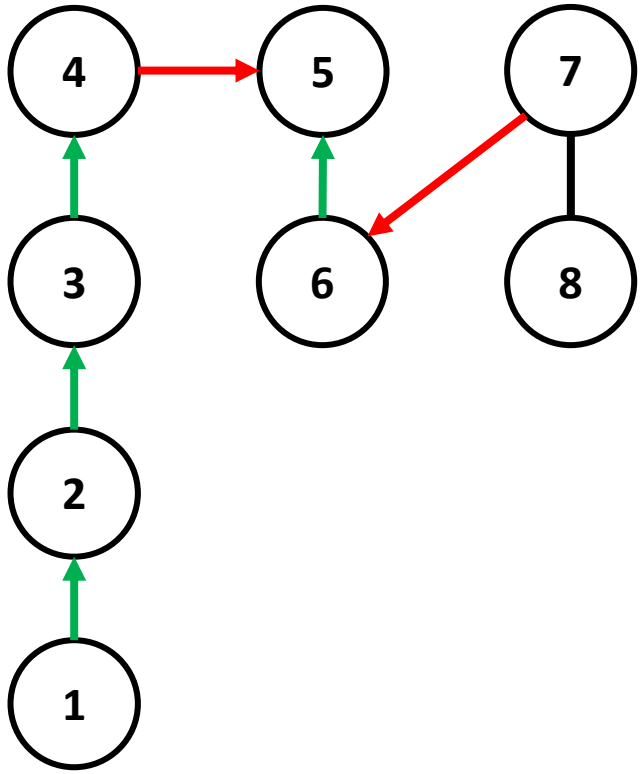
(1, 2) is Valid, (2, 3) is **Invalid**

The path is **Invalid**

Route Received from Provider



(1, 2), (2,3), (3,4) are **Valid**
(4,5) is Invalid, but it's **OK!**
(6,5), (7,6) are **Valid**
The path is **Valid**



(1, 2), (2,3), (3,4) are **Valid**
(4,5) is Invalid, but it's **OK!**
(6,5) is **Valid**, (7,6) is **Invalid**
The path is **Invalid**

ASPA

ASPA Verification Can be Used to:

- filter **mistake** route leaks from customers, peers and providers;

ASPA Verification + ROA Validation Can be Used to :

- filter **mistake** and **malicious** hijacks;
- filter **mistake** and **malicious** route leaks;

In reality:

- **It works!**

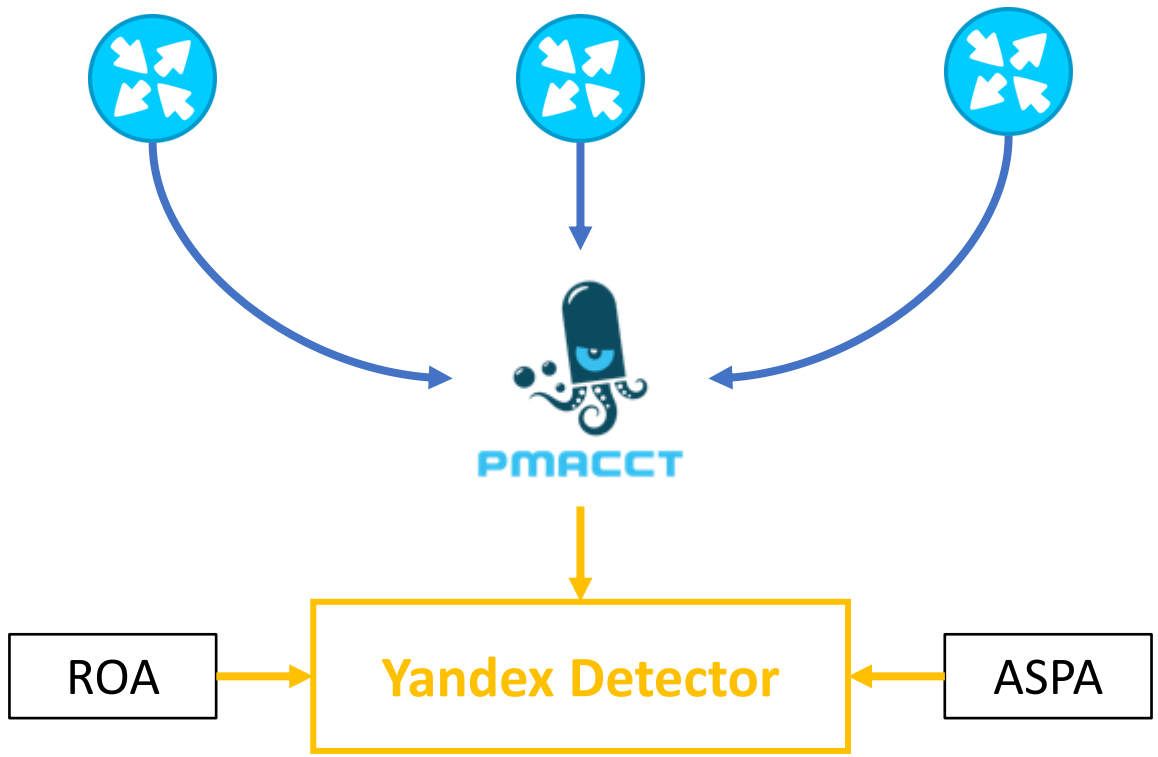
How It Works: NTT Peering Lock

- Uses AS Path regular expression;
- Uses known default free networks;
- Uses known customer-provider pairs;
- Detects leaks from customers and peers.

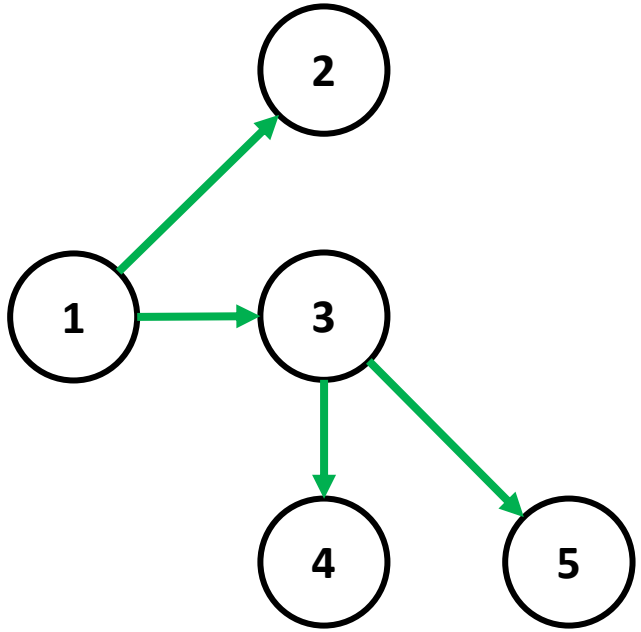
```
$bignetwork ASN anywhere in the AS_PATH. H

ip as-path access-list 99 permit \

_(174|209|286|701|1239|1299 \

|2828|2914|3257|3320|3356 \

|3549|5511|6453|6461|6762 \

|7018|12956)_

route-map ebgp-customer-in deny 1

match as-path 99
```

How It Works: Yandex Detector

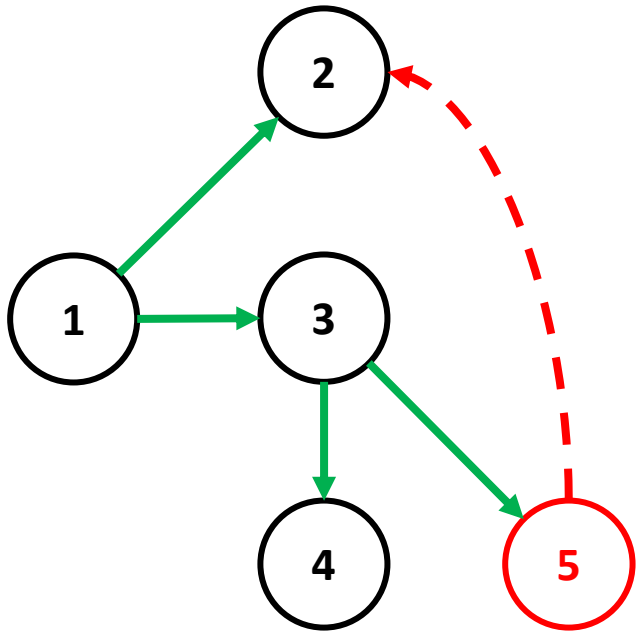
- Uses BMP as a source (pmacct);
- Uses known default free networks;
- Uses known customer-provider pairs;
- Full support of ASPA algos: capable to detect leaks from all directions;
- Can detect anomalies for Yandex itself!



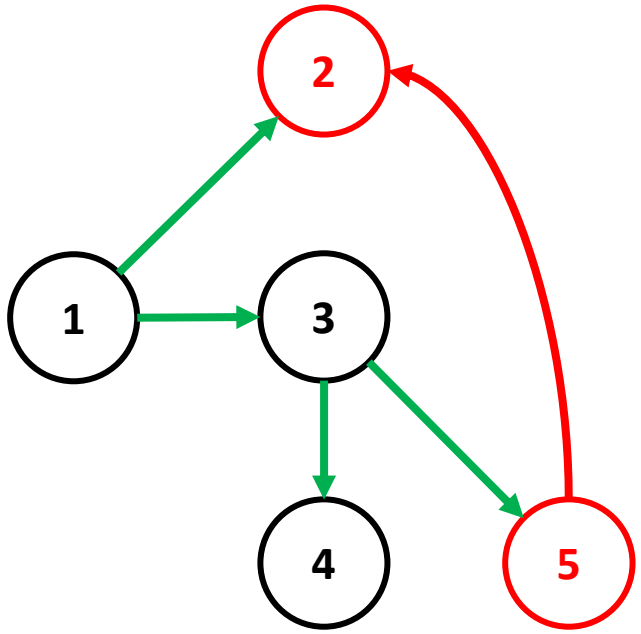
No Leaks – Good Leaks



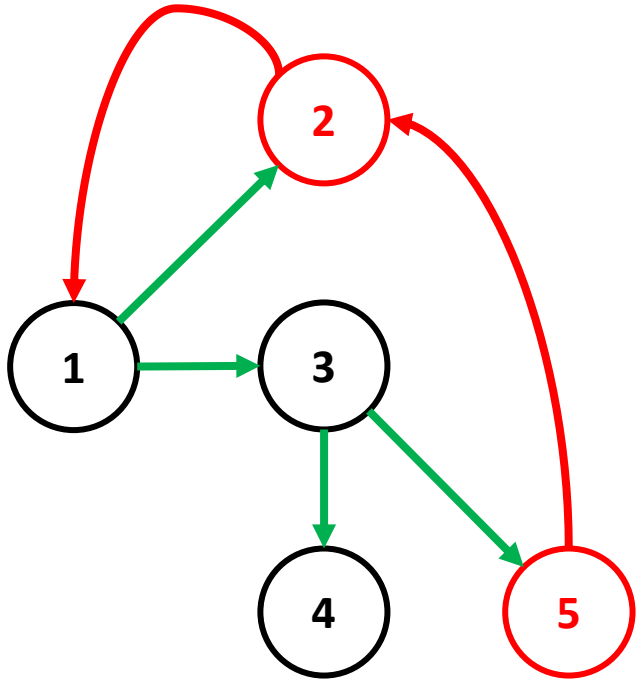
Not Propagated Leaks – Good Leaks

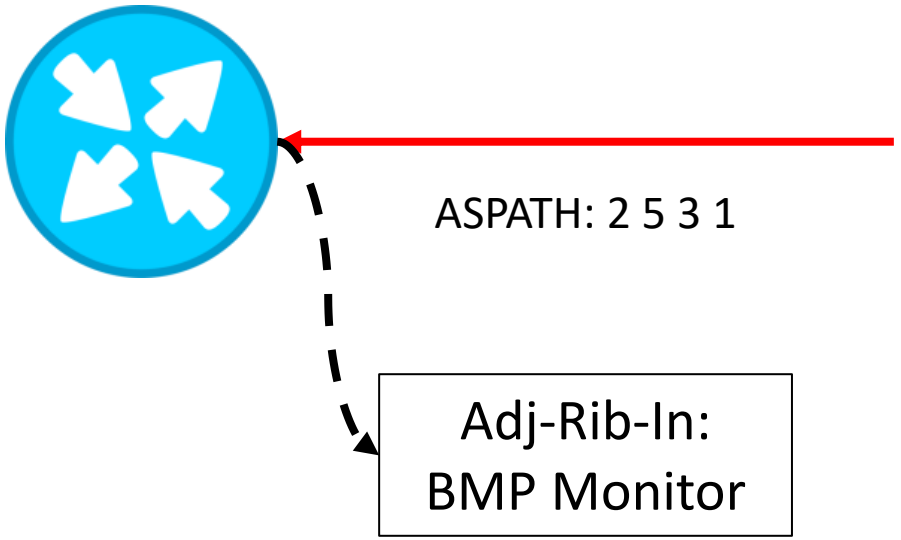


Propagating Leaks – Detection is Needed



Yandex Detector: Key Idea





If your neighbor accepts leaked/hijacked prefix, it will send it to you.
It will send your own address space too!

Proof of Concept

	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 38.122.63.37, aspath: 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 149.11.124.73, aspath: 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 185.70.202.152, aspath: 6762 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 213.242.69.249, aspath: 3356 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 213.248.90.186, aspath: 1299 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 4.14.97.241, aspath: 3356 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 62.115.54.165, aspath: 1299 174 31133 13238
14h	CRIT	bmp_monitor_4_Leaks prefix: 213.180.202.0/24, peer_ip: 87.245.248.8, aspath: 9002 3356 174 31133 13238

How Many Records Do You Need?

18

ASPA: It Works!

To make it work in your network:

- Contribute to IETF documents;
- Ask your favorite vendor for support plans;
- Ask your favorite RIR for support plans;

Alexander Azimov: a.e.azimov@gmail.com