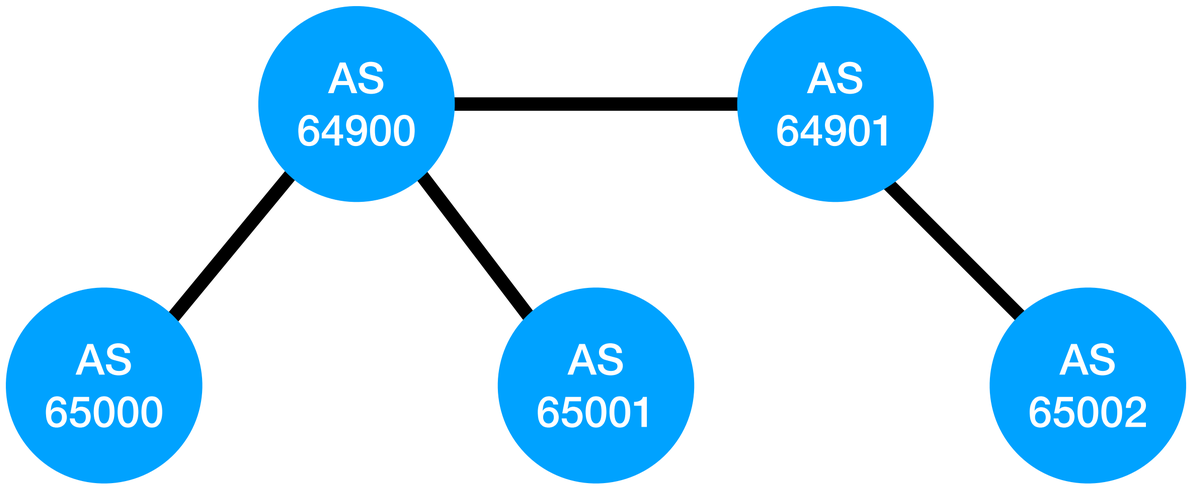
Securing the Internal Control Plane with Standards & OSS

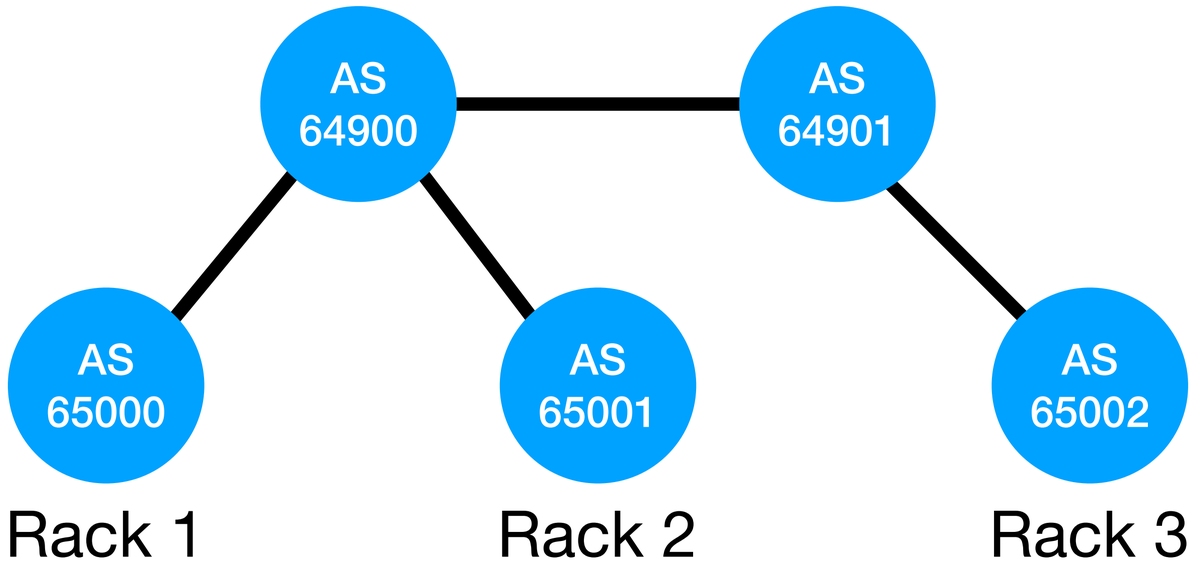
Antonios Chariton
<daknob@daknob.gov>

s/gov/net/

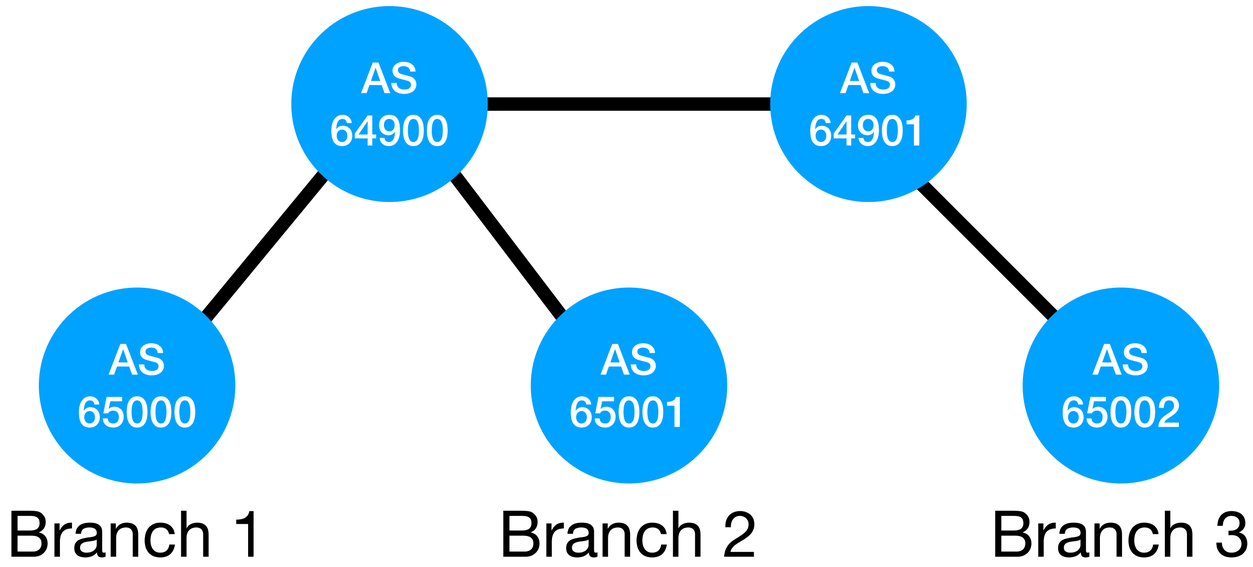
Enterprise Network



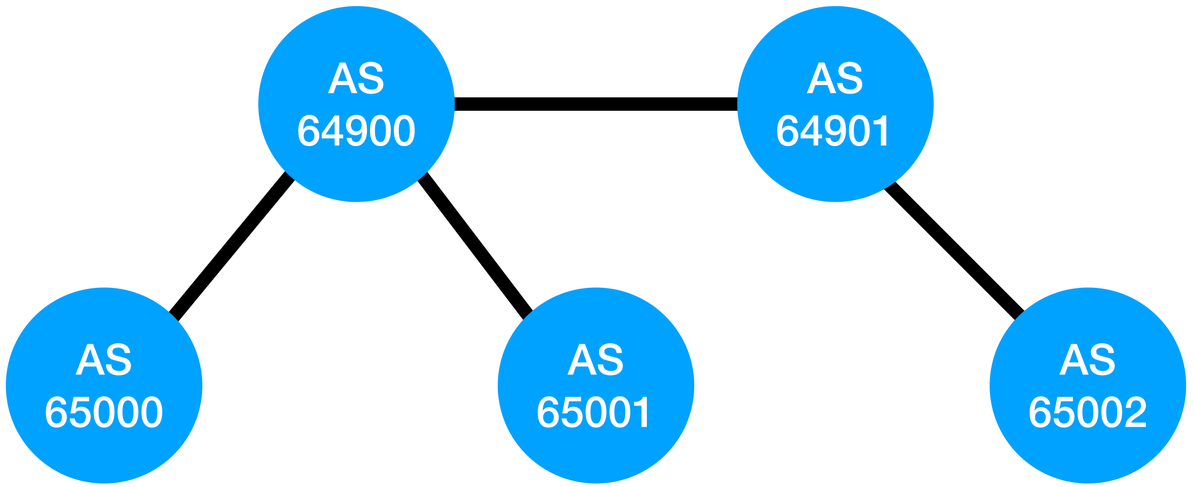
Enterprise Network



Enterprise Network



Enterprise Network



WiFi: 2001:db8::/64
VMs: 2001:db8:0:1::/64
Guest: 2001:db8:0:2::/64

...

WiFi: 2001:db8:2::/64
VMs: 2001:db8:2:1::/64
Guest: 2001:db8:2:2::/64

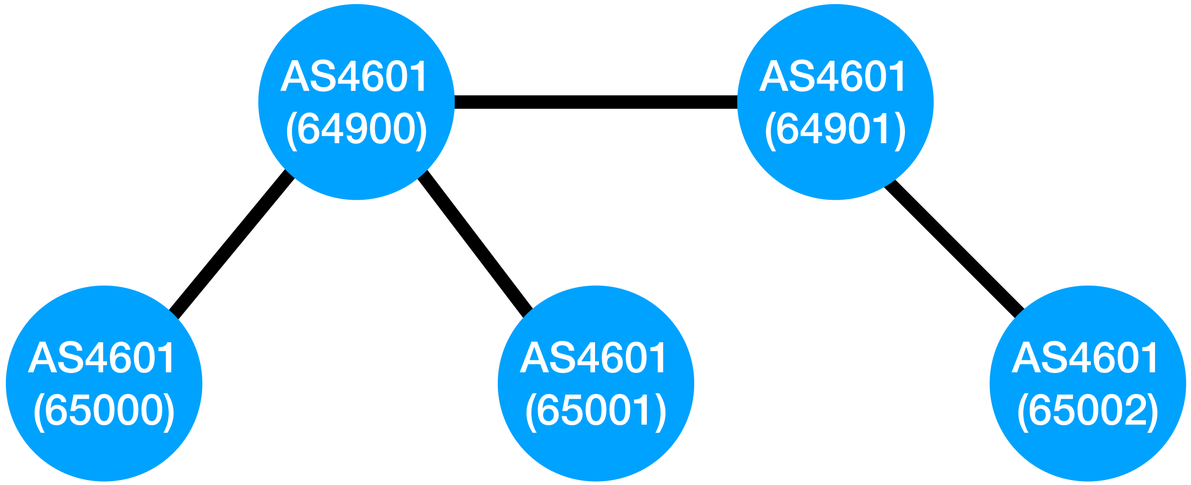
Provider Network


ASAG
ADDRESS
EXHAUSTION

Routing @ AS4601

- 100% eBGP for Internal Routing, no iBGP
- BGP Confederations, one per router
- Each router has its own RIB, makes independent decisions
- Collaboration via BGP Large Communities
- Multi-vendor, primarily Debian + bird2
- ~ 20 Core IP / Full BGP Routers

Provider Network



Towards Zero Trust

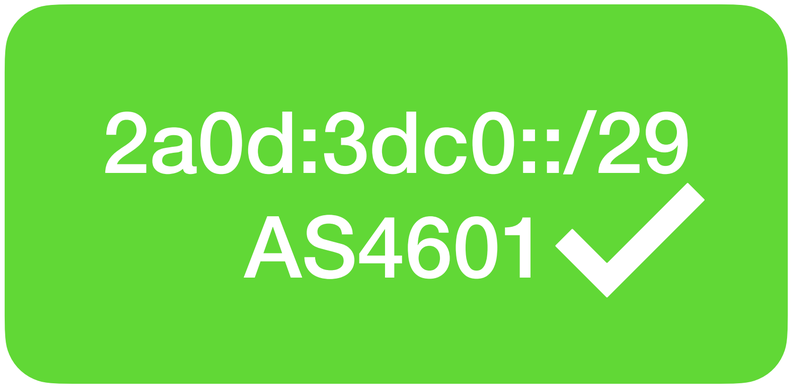
- Each router trusts only itself
- Treats information from others as “hints”
- Double-checks everything
- Kinda like each other Confederation AS being external / third-party
- Compromising a single router should in theory limit impact

Validating Public Routes

- Drop RPKI Invalid (and in some cases Unknown)
- IRR Filtering
- Bogon Lists
- Too large / too small
- […]

Validating Public Routes

- Drop RPKI Invalid (and in some cases Unknown)
- IRR Filtering
- Bogon Lists
- Too large / too small
- […]



Validating Internal Routes?

Problems

- Internal Route Leaks
- Internal Hijacks
- Internal `$anything`
- No IRR, no RPKI, looser “valid” meaning
  - “Up to /24, except these 3 /26’s we have from an M&A in Atlanta”
  - “No, I think A.B.C.D/24 is in NYC, not in Seattle”

Your entire playbook is
useless :(

What do people do?

- Nothing ¯_(ツ)_/¯
- O(`n!`) Prefix Lists or O(`m!`) Route Maps / Filters / …
  - Usually hand-written, unmaintained
  - No visibility: sub-optimal routing, asymmetries, packet loss, …

Let’s reuse the playbook!

The Public Route Playbook

- Drop RPKI Invalid (and in some cases Unknown)
- IRR Filtering
- Bogon Lists
- Too large / too small
- […]

The Public Route Playbook

- Drop RPKI Invalid (and ~~in some cases~~ Unknown)
- ~~IRR Filtering~~
- ~~Bogon Lists~~
- ~~Too large / too small~~
- ~~[…]~~

The IGP Playbook

- Run a private RPKI internally
- Drop Invalids, Drop Unknowns for your prefixes
- Single source of truth for IP -> AS mapping

AS4601 RPKI Dashboard

Prefix	Origin
193.5.16.0/31-32	AS64900
193.5.16.80/29-29	AS65001
2a0d:3dc0:100::/48-64	AS65001
2a0d:3dc0:16::/64-128	- AS65050 - AS65051 - AS65052

Filters are simple now!

```
function internal_backbone_import_v6 {

   if net ~ IGP_PFX_V6 {

      return roa_check(irpki6, net, bgp_path.last) = ROA_VALID;

   }

   return false; # or handle Public BGP Routes

}
```

Awesome, how do I do it?

Getting a Private RPKI

- NLNet Labs has Krill (Free / MPL 2.0 Software)
  - It’s for Delegated RPKI, but it works for TALs, too
  - You can create your own TAL, also HSM-Backed
  - You can issue ROAs for your prefixes and private ASNs

Populating Data

- Depends on the current source of truth (or lack thereof)
  - You can easily create an XLSX / CSV / YAML to Krill Importer
  - If you use Netbox, it’s easy to do the same via its API
  - Perhaps your favorite IPAM solution has something?
- You want to create a map of Prefix -> AS
  - There’s flexibility in the map properties

Keeping Data in Sync

- If you use an outdated format, consider using Krill as source of truth
  - No data to sync, after initial import this is your new SOT, congratulations!
- If you have existing automation, periodically or on-change create / delete ROAs

`curl(1)` `cron(8)`

Getting the data to the routers

- You can deploy Routinator from the same vendor
  - Point it to your Krill instance / distribution points
  - Add it directly or via RTRTR to your routers
- Identical to Public RPKI

ROUTINATOR

Done!

Right?

Nope :(

Caveats

- Not all BGP implementations support independent RPKI sources
  - Ideally you need the Private RPKI in a separate table, not merged with RIRs' data
- BGP Confederation handling is a wildcard
  - bird2 has no way of separating confederation AS 65000 from eBGP AS 65000 in the filters / paths (but can in the UI)
  - At least that's only relevant if you use these. If you use eBGP then it's fine :)

Caveats

- RPKI does not protect from all types of attacks today
  - You'd need to deploy ASPA as soon as it's more readily available
  - Which will hopefully be more complete, as you know all the possible legitimate links in your network
    - It will probably be a bit trickier to generate the objects from XLSX / Visio / XML than just Prefix->AS…
  - A list of internal ASes expected behind every BGP session is a stop-gap

Caveats

- If your RPKI Validator / Cache / […] goes down, your network will drop all routes as “Not Valid” (while on the Internet it’d be Unknown and still accepted)
  - Maybe you have an OOB directly connected network, which can now also host RPKI and it’s 100% independent *(wishful thinking)*
  - I host RPKI (Routinator / Krill / …) in a special prefix that:
    - Is okay to be Unknown (but not Invalid)
    - Serves data only on application-layer authenticated protocols (mTLS, SSH)
    - Is close to all routers (sometimes chassis-local)

The Long Term Goal

- Automate the entire IGP Routing Security
- Enforce at every router, independently
- Enhanced visibility: SOT & RIB Dumps exist
- Unmatched alerting: many existing eBGP tools are now usable internally!
  - AS Path Mismatch? Hijack? All detectable via RPKI violation monitoring
    - More issues will be detected as more RPKI features are added (ASPA)