RIPE NCC
RIPE NETWORK COORDINATION CENTRE

BGP Collector Communities

Randy Bush, IIJ
Emile Aben, RIPE NCC

2015-11-19 | RIPE71

BGP communities - It's complex

- Extracting customer cones from RIS data
  - What prefixes are originated and propagated by a given ASN?
- Why not use existing BGP communities?
  - TL;DR: It's complex!
  - https://labs.ripe.net/Members/emileaben/a-tale-of-bgp-collectors-and-customer-cones

BGP communities - Example AT&T

7018:1000 - large aggregates (e.g. 12.0.0.0/8 and 2001:1890::/29)
7018:2000 - routes from customers, announced to other customers and to peers
7018:2500 - routes from customers who request AT&T to announce only to other AT&T customers and not to AT&T peers
7018:5000 - peer routes

Each BGP route will have one and exactly one of these four communities. In addition some routes will have a second community in the range 7018:[30000-39999], but these communities have nothing to do with determining AT&T's 'customer cone.'

The set of routes received by AT&T's customers who want to see all of AT&T's customer routes is the union of the sets of routes tagged with communities 7018:1000, 7018:2000, and 7018:2500.

The set of routes received by AT&T's peers is union of the sets of routes tagged with communities 7018:1000 and 7018:2000

- Customer Cone: 48,576 (best guess from RIS)

BGP communities - Example Level3

- AS3356 uses “3356:123" to tag customer routes
- AS3356 originates 2,555 prefixes
  - 1,052 tagged “3356:123” vs. 1,533 not
- Customer cone: 48,576 (best guess from routeviews)

These are Not Even Commensurate

- Fix:
- draft-ymbk-grow-bgp-collector-communities

Customer Cone ASN:64994

External Route ASN:64995

Internal Route ASN:64996

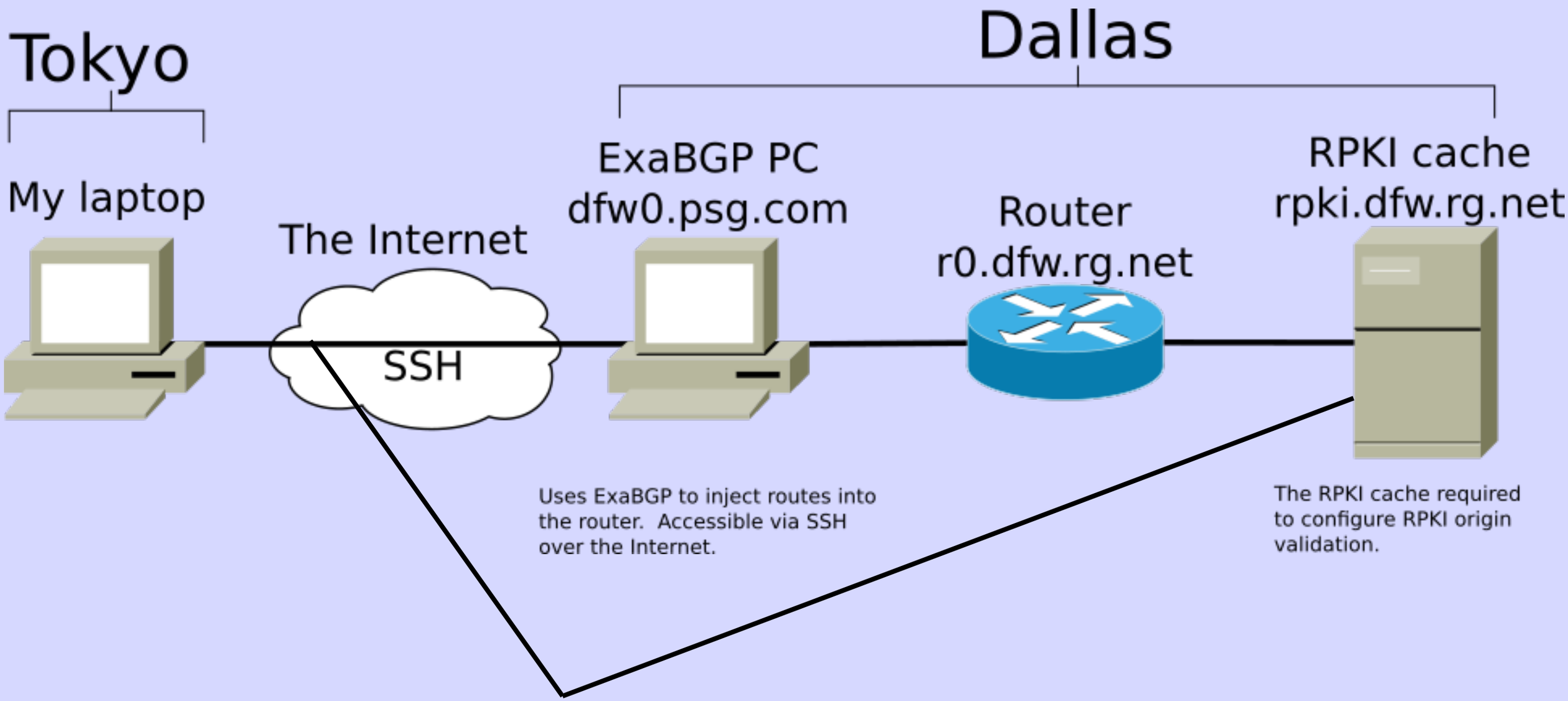
RPKI Origin Validation vs Route Filters

Lachlan Kang <lachlan.kang@student.adelaide.edu.au>
Cristel Pelsser <pelsser@unistra.fr>
Randy Bush <randy@psg.com>

Aim of the Project

Measure the difference in performance and configuration time between RPKI origin validation and route-policy prefix filtering.

Configuration



Experiment 1

- Route policy / prefix-sets created from Hurricane Electric customer closure.
- Routes extracted from RIB from the Route-Views Equinix router and filtered so that only routes in the HE customer closure are present (101,900 routes).
- Custom RPKI data used. One ROA per announced route.

Experiment 1 Results

Configuration loading time
RPKI: 9.1 seconds (includes time it takes for the RPKI cache to fill the router)
Prefix-Filter: 11.4 minutes!!

Configuration memory usage
RPKI: 9.4MB
Prefix-Filter: 39.9MB

Route processing time
RPKI: 3.678 seconds
Prefix-Filter: 3.703 seconds

Experiment 3

- Multiple BGP sessions, each announcing a different set of routes.
- 5 sessions totalling 715,009 routes.
- Announced routes extracted from route-views RIBs.
- Tier-1 customer closures extracted by examining BGP communities of routes.
- Route-policy / prefix-sets created from announced prefixes. One prefix-set entry per announced route for each peer.
- Custom RPKI data used. One ROA per announced route.

Experiment 3 Results

Configuration loading time
RPKI: 13.4 seconds (includes time it takes for the RPKI cache to fill the router)
Prefix-Filter: 72.5 minutes!!

Configuration memory usage
RPKI: 39.4MB
Prefix-Filter: 290.8MB

Route processing time
RPKI: 25.4 seconds
Prefix-Filter: 31.7 seconds

And a Taxonomy

leak - i receive P and send it on to folk to whom i should not send it for business reasons (transit, peer, ...)

mis-origination - i originate P when i do not own it

hijack - an intentional mis-origination

laundered - i receive P (or some sub/superset), process it in some way (likely through my igp), and re-originate it, or part(s) of it, as my own

Questions