BGP Prefix Origin Validation

draft-pmohapat-sidr-pfx-validate-02

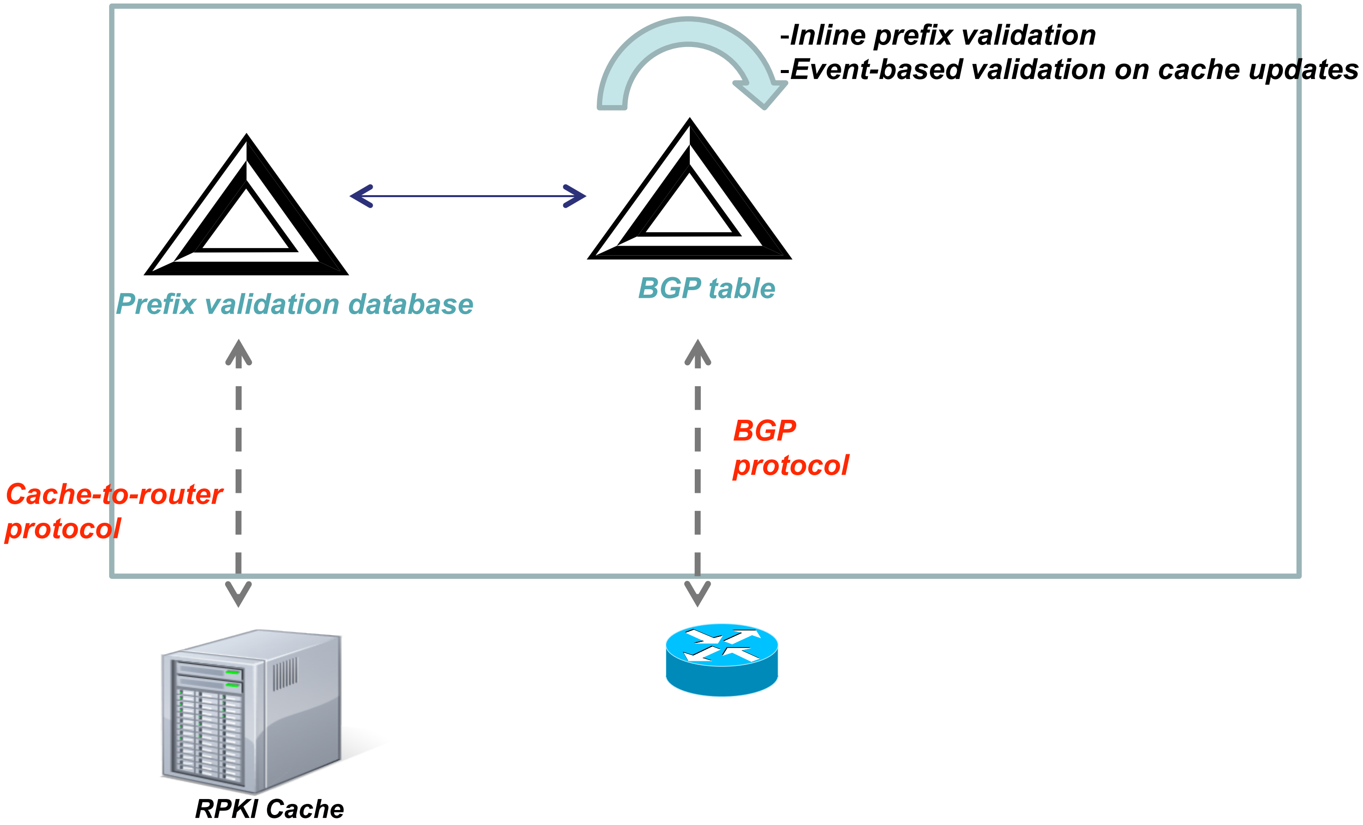
Pradosh Mohapatra <pmohapat@cisco.com> (Ed.),
John Scudder <jgs@juniper.net> (Ed.),
Geoff Huston gih@apnic.net (Ed.)

IETF 75, July 2009, Stockholm, Sweden

Other Authors/Contributors

Junaid Israr (jisra052@uottawa.ca),
Mouhcine Guennoun (mguennou@uottawa.ca),
Hussein Mouftah (mouftah@site.uottawa.ca),
Randy Bush (randy@psg.com),
Rob Austein (sra@isc.org),
Russ Housley (housley@vigilsec.com),
Rex Fernando (rex@juniper.net),
Miya Kohno (mkohno@juniper.net),
Shin Miyakawa (miyakawa@nttv6.jp),
Taka Mizuguchi (taka@nttv6.jp),
Tomoya Yoshida (yoshida@nttv6.jp),
David Ward (dward@cisco.com),

Approach



-Inline prefix validation
-Event-based validation on cache updates

Prefix validation Logic

```
query key = <BGP destination, masklen>, data = origin AS
result = BGP_PFXV_STATE_NOT_FOUND
walk prefix validation table to look for the query key
for each matched “entry” node in prefix validation table,
   prefix_exists = TRUE
   walk all records with different maxLength values
   for each “record” within range (query masklen <= maxLength)
       if query origin AS == record origin AS
           result = BGP_PFXV_STATE_VALID
           return (result)
       endif
   endfor
endfor
if prefix_exists == TRUE,
    result = BGP_PFXV_STATE_INVALID
endif
return (result)
```

Max_len fud for thought

Options when validating 10/24, AS A:

if ROA exists for 10/(min=8, max=16), AS A

- NOT_FOUND (draft currently says this)
- INVALID (will be covered in use cases presentation)

Bestpath selection

- Path’s validation states:

```
typedef enum {
    BGP_PFXV_STATE_VALID = 0,
    BGP_PFXV_STATE_NOT_FOUND = 1,
    BGP_PFXV_STATE_INVALID = 2,
} bgp_pfxv_state_e;
```

- Bestpath comparison

```
1. INPUT: received path, current bestpath
2. if either received path or current bestpath is IBGP learnt, skip this
     comparison and goto the next decision step.
3. if received path's validation state < current bestpath's validation state
4.      prefer received path
5. elseif received path's validation state > current bestpath's validation state
6.      prefer current bestpath
7. else /* they're equal */
8. proceed with rest of BGP decision process
```

Config and Policy overrides

1. Disable prefix validation globally
2. Disable prefix validation per EBGP peer
3. Disable prefix validation for a set of prefixes

When disabled, the "state" of such EBGP learnt routes will be set to “not-found”

1. Allow "invalid" routes for bestpath selection
2. Disallow "not-found" routes for bestpath selection
3. Set arbitrary communities based on "validity state" on neighbor outbound for debugging purposes

Implementations available

- Prototype in Cisco IOS-XR of RPKI data
  - cache-to-router protocol (draft-ymbk-rpki-rtr-protocol)
  - BGP prefix validation

- Testing in progress at multiple locations. Trying to get some real RPKI repository set up
- Early results:

  < 10usec of overhead for prefix validation per route

- Sample XR configuration:

```
router bgp <as#>
  bgp rpki cache <cache name> <port#> refresh-time <time>
  bgp bestpath prefix-validation {disable | allow-invalid | disallow-not-found}
```

Changes from -01

- Geoff Huston added as an editor for the document
- Changes in the decision process section to clarify that only EBGP paths are subject to prefix validation and their validation states are compared. There is no comparison of validation state between EBGP and IBGP paths
- Changes to the policy control section to indicate that the default validation state for routes for which prefix validation is disabled by policy is “not found”

Document Status

- Feedback please!
  - To authors or SIDR mailing list
- WG adoption