Juniper NETWORKS™

BGP Origin Authentication

The Problem

- Any AS can inject any prefix
  - Mistake (most commonly)
  - Malicious
- Effective DoS attack
- No automated way of excluding bogons
- Need mechanism to differentiate between bogons and legit prefixes

Different Problems

- Anyone can masquerade as another AS
- Anyone can tamper with advertisements
- Valid problems
- Need practical solutions
- Not this talk

Impractical problems

- Compromise of a BGP speaker
- Global Byzantine computations are intractable
- Not this talk

Authenticate the AS path?

- AS path performs two functions
  - Prevents routing & forwarding loops
  - Differentiates between paths
- Attacks
  - DoS
    - Add AS number
    - Delete AS number (causing a loop)
  - Shift traffic
    - Move traffic towards or away

Authenticate the AS path?

- Threat environment
  - Must be a transit ISP
  - Global advertisement provides auditing
  - Transit ISPs can attack the data stream too
  - Hard to hide from traceroute
- Is this a problem worth solving?
- At what cost?

Our Approach

- Encode prefixes in DNS
- Use DNSSEC to provide authentication
- Have BGP look up each prefix in DNS
- Paths to prefixes fall into three classes
  - Authenticated
  - Unauthenticated
  - Authentication failures (bogon)

The Easy Part: The AS RR

- **Syntax:**

  <name> AS <AS number> <prefix length>

- **Semantics:**
  - **The prefix represented in <name> can be advertised with origin <AS number> with the given <prefix length> or longer**

An Example

- An AS RR:

```
125.128.bgp.in-addr.arpa.  AS   47      16
```

- Prefix 128.125/16 is allocated to AS 47
- Longer prefixes also match!

On the BGP side

- BGP does a lookup for each prefix
- Compare results against each path
- Performance issues:
  - BGP speakers can cache relevant RR’s
  - Entire allocation tree fits on secondary storage
  - Cache can persist across reboots

Fun with BGP

- If there’s a matching AS RR
  - And the origin doesn’t authenticate
    - BOGON!!!
    - Log prefix, origin
    - Select a different path
    - Withdraw it, if it has been advertised
    - Generate SNMP trap, ring bells, send pages, wake the dead, etc.

More fun with BGP

- If there's a matching AS RR
  - And the origin authenticates
    - Authenticated paths may be preferred over unauthenticated paths
    - Authentication has a lifetime
      - min TTL of all RRs
    - Authentication should be rechecked before lifetime expires

Even more fun with BGP

- If there's no authentication information
  - Paths are unauthenticated
    - Paths are useable
    - Same as today -- eases migration
  - Exception: authenticated less-specific prefixes are preferred over unauthenticated more-specific prefixes

Circular DNS dependency

- If there is an authenticated path, it is preferred to an unauthenticated path
- Only the authenticated path is announced
- Transitivity holds: the authenticated path always wins and propagates
- Only holds if domains authenticate the origin

Migration

- Inaction results in the status quo
- Action results in increased protection
- Database configured by address assignors
- Transit providers must deploy new code
- No (intractable) flag days
- Security improves with additional deployment

Aggregation

- How do we deal with aggregates?
- Include aggregates in bgp.in-addr.arpa
- Looks just like any other prefix, where the owner is the aggregator

The Hard Part: DNS

- How do we encode prefixes and prefix allocation?
- Awkward on non-octet boundaries
- Use the classless in-addr hack
- Root is bgp.in-addr.arpa. (or ipv4.nlri.ietf.org., or … ????)
- Root is administered by ???

Prefix encoding rules

- A name is

  <label>.<label>• • •<label>.bgp.in-addr.arpa

- Rule 1: Add a label and NS RR for every assignment
- Rule 2: For non-octet assignments:
  - The label is <octet>/<length>
  - Add CNAME records for each octet value in the assignment

Advantages

- Solves 95% of the real problems now
- Tractable amount of computation
- Leverages existing technologies
- Readily implementable
- Scales linearly with the number of paths in the global routing table
- Straightforward migration path

Forward progress?

- We need one global solution
- Debate has not selected an alternative
- Need a practical solution
- Prevent the next incident
- Debate must come to a close soon
- Otherwise:
  - The market will decide
  - After the horse has left the barn

Acknowledgments

- My co-authors
  - Yakov Rekhter
  - Tony Bates
  - Randy Bush
- The Classless in-addr gang
  - Havard Eidnes
  - Geert Jan de Groot
  - Paul Vixie
- The DNSSEC folks
- Jerry Scharf