Peering Dragnet:
Anti-Social Behavior Amongst Peers, and What You Can Do About It

NANOG 38
October 09, 2006

Nathan Patrick, Sonic
Tom Scholl, AT&T Labs
Aman Shaikh, AT&T Labs
Richard Steenbergen, nLayer


SERGEANT
PEERING POLICE
179

What is this all about?

- Peering is the act of two networks coming together to exchange traffic freely, under certain mutually understood ground rules:
  - Consistent announcement across all locations
  - Only sending traffic to routes which are announced
  - Only announcing internal/customer routes
  - And many other requirements
- Peering is also by definition an act of trust.
  - “Trust, but verify”
- Every example included here has been caught in real life
  - The names have been changed to protect the guilty

How Can You Pull a Jack Move on a Peer?

- A Jack Move is one of two undesirable situations:
  - A network violating a peering agreement to steal service(s) from you, or;
  - A network violating a peering agreement by accident
- Example Jack Moves:
  - #1 Sending inconsistent announcements to obtain free back haul (or degrading routing quality if unintentional)
  - #2 Sending traffic to unadvertised routes via static route or altered next-hop, to obtain free outbound transit
  - #3 Sending more-specifics via a peer and less-specifics via a customer, to obtain inbound transit
  - #4 Sending packets between peering endpoints, to obtain free backbone transport

Jack Move #1: Inconsistent Announcements

- Consistent announcements ensure hot potato (closest exit) routing, which in MOST cases is:
  - Better for the traffic/routing quality
  - Fair and equitable for traffic which must be back-hauled
- Ways to send inconsistent announcements:
  - Send different or more specific routes at some locations
  - Pre-pend advertised AS-PATH with the peer's ASN to force loop-detection drops at certain locations
  - Alter BGP Origin code (IGP, EGP, Unknown)
- Most networks make only cursory attempts to verify
  - Common “accidents” are detected by prefix count
  - Intentional malicious behavior is harder to detect

Jack Move #1: Inconsistent Announcements (cont'd)

- Inconsistent announcements aren't always easy to detect
  - BGP installs and propagates only the best route
    - Route-servers or centralized monitors that collects data via BGP sessions will not show you all routes
- The only sure solution is to examine the BGP Adjacent-RIB-In or Received Routes for each peer
- Ways to do that are:
  - Sniff/tap your peering circuits
    - May not be feasible in production
  - Debug BGP updates
    - Gets complicated (e..g, **Borderguard** by Feamster *et al.*, IMC04)
    - In some cases may not even be possible
  - Execute CLI/Netconf commands on every router

Verifying Consistency
Task #1 – Getting the Data

- Netconf
  - A standardized XML-based transaction method to simplify script based interaction with routers
    - **Warning #1**: May not actually be all that standardized
    - **Warning #2**: Perl XML parsers may consume 4GB of RAM
- Telnet/SSH
  - Not as pretty, but more likely to work today
  - Can also use CLI to initiate commands, and “redirect” output to TFTP for further parsing
- RSH
  - Works fine but is a security nightmare

Verifying Consistency
Task #2a – Checklist for every peer

- Is peer sending same number of routes at all locations?
  - Probably one of the first things to look for
- Is peer sending RFC1918 addresses?
  - Or other addresses that might overlap with your infrastructure addressing?
- Is peer sending /25 to /32 mask lengths in prefixes?
  - Some networks permit this behavior, but only if you mark routes as “no-export”
    - Some don't
  - Perhaps a hopeful CDN wants you to receive it?
    - For fine-grained Traffic Engineering

Verifying Consistency
Task #2b – Checklist for (peer, prefix) pair

- Does peer announce the prefix at all locations?
  - Even if number of prefixes at all locations is same, the prefixes may not be same!
- Are routes same from BGP decision point of view?
  - Exclude hot-potato step and steps after that
  - Don't just look at the routes selected by the routers, but actually simulate the decision process on routes
- Do routes have consistent BGP Origin, AS-Path length and MED?
  - Are there loops in AS-Paths?
    - Due to AS pre-pending an AS might occur repeatedly
    - Look for same AS occurring **non-consecutively**
  - Is peer sending private ASNs in AS-Paths?
  - Do you observe your own ASN in the AS-Paths?

Verifying Consistency
Task #3 – Long term analysis

- Track inconsistencies over time
  - Do peers play the tricks based upon time of day?
    - To shift traffic around as demands change through the day
  - Are peers dumping traffic on you to get around their failures?
- Other ways of using the data
  - Determine how to set max-prefix counters and when to increase max-prefix counters
  - Trend routing table size sent by peers for capacity planning, trouble-shooting, auditing ...
  - Track what customers are behind what peers and how the business relationships are changing

The challenges

- Polling from all routers takes significant time (on the order of hours)
  - BGP updates occur frequently
    - Routes may not be consistent if the snapshots from different routers are taken at different times (even minutes apart)
  - Need a better method to retrieve data from all routers
    - pipe dream!
- How often to poll?
  - Poll too frequently => impact production routers
  - Poll too infrequently => miss fine time-scale traffic shifts peers might be doing

Our Solution: The Peering Dragnet Tool

- Route-Retriever
  - Retrieves data from the routers
  - Places output in files
    - One file for every (peer, peering location) pair
    - File name of the form: “ASN.BGP-NBR-IP-ADDRESS.IN”
- Route-Analyzer
  - Analyzes the routes
  - Generates reports of inconsistencies and anomalies

What data did we look at?

- Routes received in SBCIS (AS7132)
  - Looked at inbound and outbound routes
    - Pre-policy and post-policy inbound routes
      - We will only present results for post-policy inbound routes
    - And of course, we won't talk about outbound routes here ;-)
- Ongoing collection of route snapshots at peering routers since August 2006
  - Present results for snapshots from August 09 through September 30, 2006
  - One snapshot per day
  - Total snapshots ~ 50

Overall Results

- How often was cold-potato routing seen?
  - Different total routes at different locations:
    - 64.40% **(mean)**, 2.67% **(std_deviation)**
  - Missing routes at some locations: 2.3%, 0.8%
  - Different origins at different locations: 0.003%, 0.001%
  - Different AS-Path length at different locations: 1.6%, 0.2%
- How many routes are inconsistent?
  - /25 - /32 prefixes: 0.16%, 0.01%
  - Prefixes in RFC1918 address space: 0.0%, 0.0%
    - Only one snapshot had few routes in this space
  - Loops in AS-Paths: 0.087%, 0.0099%
  - Un-allocated ASNs in AS-Paths: 0.0%, 0.0% (surprise!)
  - Private ASNs in AS-Paths: 0.005%, 0.004%

Report Output: BGP Origin Inconsistencies

- AsiaPac peers de-preferring specific west coast peering locations
  - Example #1 – Australian Peer
    - (1) /19 advertised at: Palo Alto (IGP), Seattle (INCOMPLETE)
  - Example #2 – AsiaPac Peer
    - (4) /24's advertised at: LA (INCOMPLETE), Palo Alto (IGP)
  - Example #3 – Oceania Peer
    - (4) /24's (1) /21 advertised in: Palo Alto (INCOMPLETE), LA (IGP)
  - Example #4 – AsiaPac Peer
    - (9) /24's advertised at: LA (INCOMPLETE), San Jose (IGP)
- An AsiaPac peer load balancing?
  - (14) /24's advertised at two locations:
    - Palo Alto– 5 prefixes advertised as IGP, rest as INCOMPLETE
    - San Jose- 9 prefixes advertised as IGP, rest as INCOMPLETE

Report Output: BGP Origin Inconsistencies (cont'd.)

- Canadian Peer
  - (1) /23 and (1) /16 advertised at:
    - Chicago (INCOMPLETE), NYC (INCOMPLETE), Seattle (IGP)
    - /16 is broadband users out of Ontario
      - Why an incomplete in Chicago and NTC? Are their capacity issues?
    - /23 dies at the first hop in the network
- Domestic US Peer
  - (1) /29 advertised at:
    - San Jose (INCOMPLETE), Ashburn (INCOMPLETE), Atlanta (IGP)
    - Customer network based in DC area; Misconfiguration?
- Domestic US Peer
  - (2) /24's advertised at:
    - LA (De-Pref"d) – Customer network is based in Irvine, CA
    - Chicago (De-Pref'd) – Customer network based in Chicago

Report Output: RFC1918 Address Space Leak

- Leak from a large cable MSO (1 day only)
  - 8.0.0.0/5
  - 128.0.0.0/1
  - 10.0.0.0/8
  - 192.168.0.0/16
  - 172.16.0.0/12

Report Output: ASN Loops

- Typically AS-Paths with loops have this form:

  AAA BBB XXX BBB BBB

- Our conjecture: ASN BBB is doing the pre-pend so that they can make ASN XXX to drop the route
  - Often XXX are large networks
- But sometimes there just seem to be typos ...

  1273 11269 11139 11139 11139 11139 11139 1139 11139 11139

  13237 1327 13237 13237 33891 33891

  6453 9729 9279 9729

  3491 9116 9116 9116 9116 9116 116 9116

  12883 13228 24995 24994 24995 24995 24995 35781

  3491 4637 4637 467 4637 9942

Report Output: ASN Loops (cont'd.)

- ASN's that are injected in the path (targets):
  - 5 ASNs from Europe
  - 7 ASNs from North America
  - 3 ASNs from AsiaPac
  - 1 ASN from Latin America
    - 1 prefix inserted with Latin America ASN
- Who is originating the prepends?
  - 6 ASNs from Europe
  - 7 ASNs from North America
  - 3 ASNs from AsiaPac
  - 1 ASNs from Latin America (Suspected typos removed)
- ASN pre-pending appears to occur with other ASNs in the same geographical area

Report Output: Unallocated / Private ASNs

- Surprisingly, no unallocated ASNs seen (yet)
- Private ASNs
  - 20 unique peers advertising paths with private ASNs
    - 7 North American peers
    - 9 European peers
    - 4 Asia Pacific peer
  - 73 unique prefixes received across peers with private ASNs
    - All /19 to /24's
    - 1 /16 advertised with multiple private ASNs:
      - XXXX 14201 **65001** 2009 2005 **65500**

Jack Move #2: Altering/Static Next-hop

- You can advertise authorized routes via BGP, but you can't stop another network from sending you unauthorized traffic to other arbitrary destinations
  - At a public IX, you may have no relationship with them
    - Easier to detect traffic from people who you don't peer with
    - But harder to make them stop, since you have no agreements
    - The network could claim it was an accident
  - With a relationship, it can be even harder to detect
    - You have to distinguish between real and unauthorized traffic
- Only two ways to combat forced routing
  - Accounting (detecting the behavior after the fact)
  - Filtering (blocking packets to unauthorized destinations)

Jack Move #2: How to Detect it?

- Accounting: Check for traffic on “wrong” incoming links
  - NetFlow
    - Check if source address in data packets make sense
    - **Problem:**
      - No data for the peer that sent you the inbound traffic
      - Thus may only be useful for private peering, not public peering
  - MAC accounting
    - Check if we have eBGP session with the remote router
      - Use source MAC address to determine the remote router
    - **Problem:** May not be possible depending on your linecards
      - Or may require Quad Price PICs
  - ACL counters

Jack Move #2: How to Prevent it?

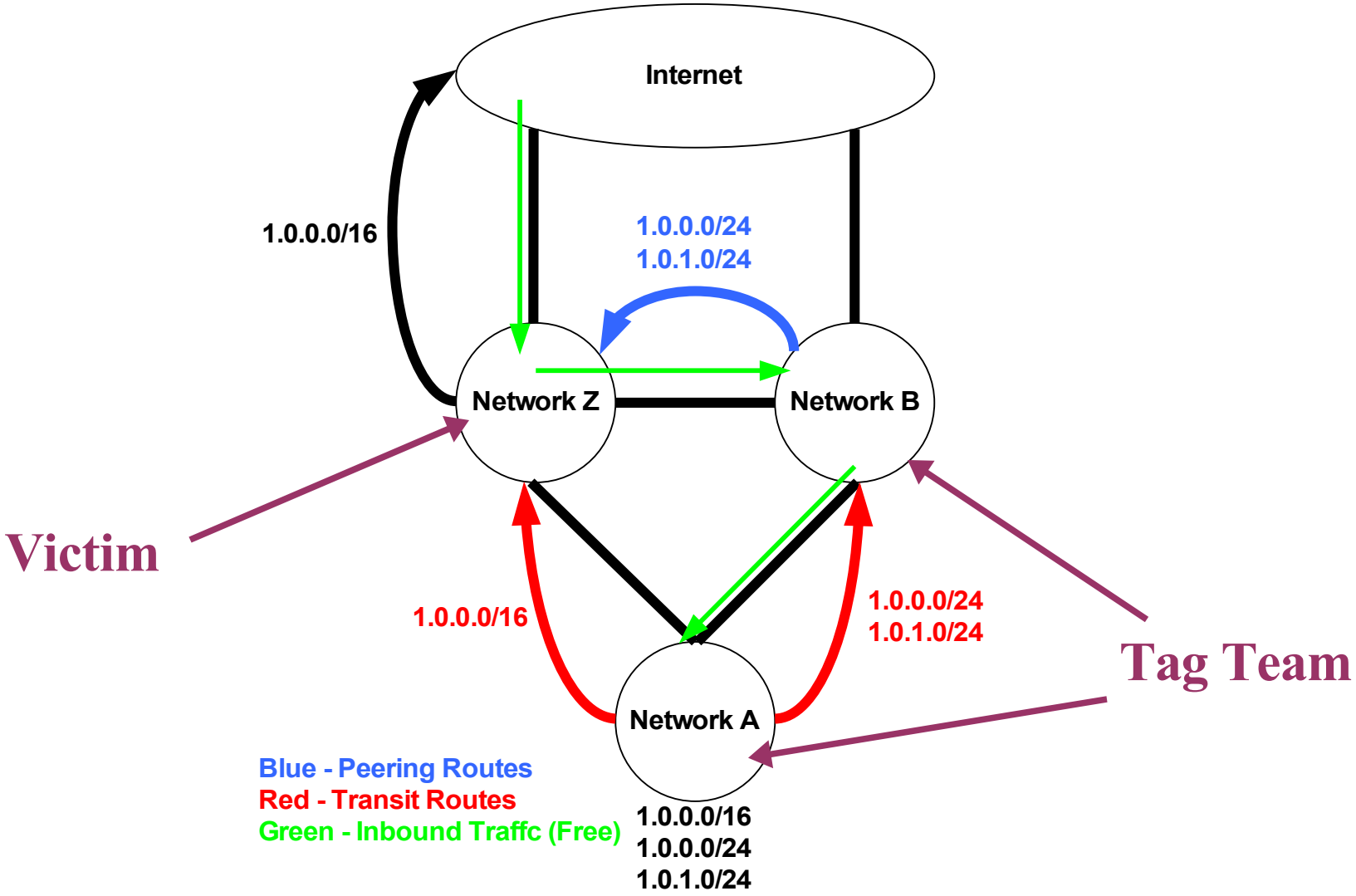
- Anti-Default Provisioning/Configuration
  - Solution in “Dot-com era”: separate peering routers which carried only customer and internal routes, but no default
    - Problems in “Dot-bomb era”: extra routers means extra cash!
  - Better solution: do it virtually instead
    - Routing-instances, BGP policy accounting
    - Cisco can potentially do this with QPPB on some platforms
    - Juniper can on some platforms
- MAC-Filtering
  - Not as powerful, but prevents individuals you don't peer with from routing traffic to you
  - Some IXPs provide you with participant MAC address <–> IP address mapping consistent with switch configurations
    - **Example:** TorIX

Jack Move #3: Peering + Transit Tag-Team

- This one is a little more complicated and unusual
  - And obscure enough that almost nobody checks for it
- How this is done?
  - Network A is a customer of network Z; B is a peer of Z
  - Network A "becomes" transit customer of network B
  - Network A announces aggregates to network Z, and more-specifics to network B which sends them to peer Z
  - Network Z only announces the aggregates to the world
- The result
  - Traffic from Internet to Network A goes through Network Z and then through network B
  - Network A does not have to pay its provider Network Z

Thus, Networks A and B pull the Jack Move on network Z

Jack Move #3: Peering + Transit Tag-Team (cont'd)



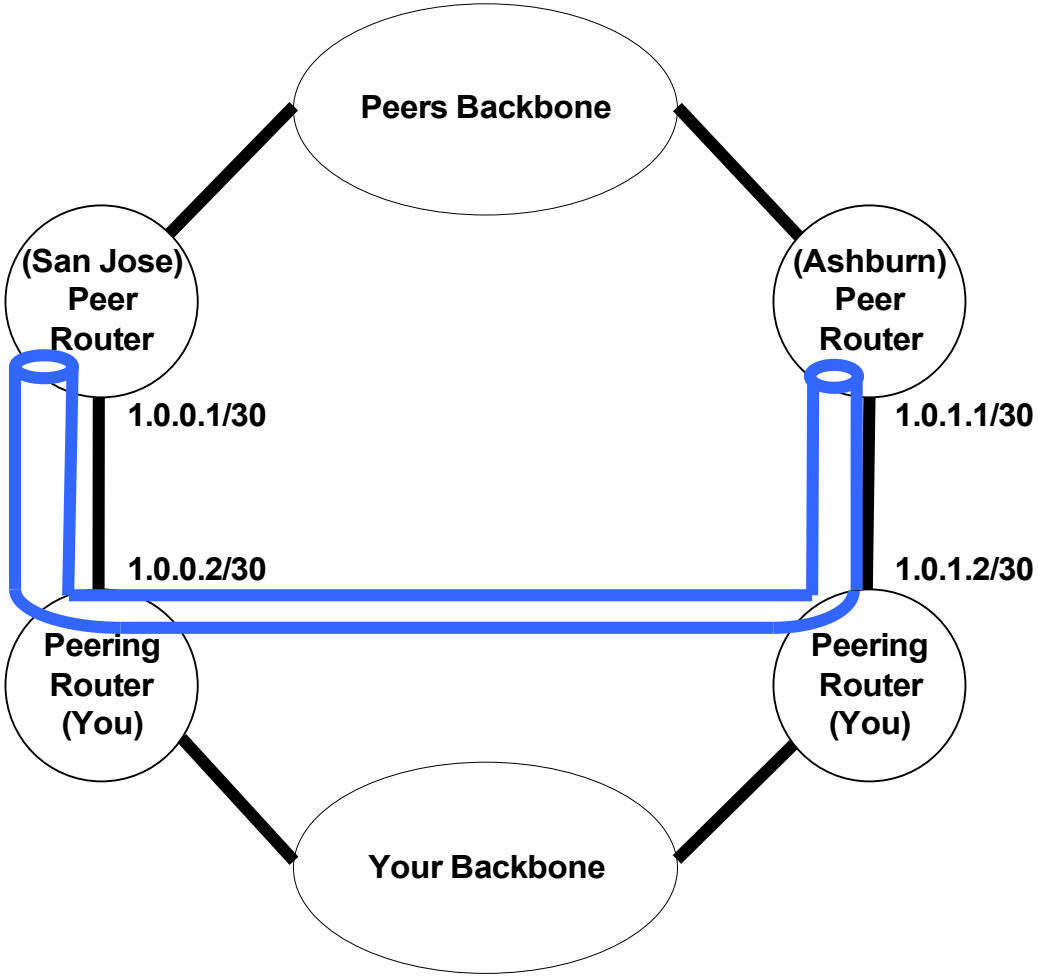
Jack Move #3: Peering + Transit Tag-Team (cont'd)

- Why this works
  - Most networks local-pref customers over peers
  - But prefix-length overrides the local-pref during forwarding
- How to prevent it
  - The only way is to watch for peers advertising more-specifics of routes for routes learned from customers

Jack Move #4: No officer I didn't see that tunnel

- How this is done
  - Network A peers with Network Z in multiple locations
  - Network Z assigns /30s for the interconnections
    - These probably come out of larger aggregate announced by Z
  - Network A sets up a tunnel between these endpoints
  - Network A steals free transport from network Z
- Why it may not be noticed
  - Since the /30 assignments came from network Z's announced space, technically network A isn't violating any peering rules
  - Network A can say they reset next-hop to loopbacks in their iBGP, and didn't redistribute the /30 into their IGP
- How to protect against it
  - Watch for unusual traffic to interconnection /30s, or
  - Rate-limit unusual traffic to interconnection /30s

Jack Move #4: No officer I didn't see that tunnel (cont'd)



Summary

- Peering Dragnet is about detecting peers that may violate peering agreements
  - Our tool checks for violations in BGP control plane
  - Results demonstrates a variety of behaviors:
    - For traffic engineering, load balancing, ...
    - Some mis-configurations as well
  - Several violations require careful analysis after detection
- More sophisticated and subtle tactics
  - In data plane
  - Colluding networks playing a tag team
  - Through tunnels
- Bottom-line: pay close attention to control and data traffic coming from your peers into your network

Thank You

Name	Email Addresses
Nathan Patrick	np@sonic.net
Tom Scholl	tom.scholl@att.com
Aman Shaikh	ashaikh@research.att.com
Richard Steenbergen	ras@nlayer.net