ICMPv6 errors for discarding packets due to processing limits

draft-herbert-6man-icmp-limits-02

Tom Herbert <tom@quantonium.net>

Overview

- Hosts and middleboxes can drop packets because they have limits of processing headers
- Problem becomes apparent with more use of EH
- Proposal: new ICMP errors that nodes can send when such a limit is hit
- Goal: increase visibility into what’s happening and make use of extension headers more viable

Problem

- Real devices have practical limitations for processing protocol headers protocol headers
  - Hardware may have a limited size parsing buffer
  - Unbounded list of options is hard and potential DOS
  - Behavior may be nonconformant
- This is a host *and* intermediate node issue
  - Hosts may limit EH processing to avoid DOS attack (draft-ietf-6man-rfc6434-bis)
  - Intermediate nodes must process HBH (relaxed in RFC8200), or stateful FW needs to find L4 header

Effects of drops

- Source receives no indication why packets are dropped
- No visibility into what is happening
- Without feedback, source has no way to fix problem
- Middleboxes incur no obvious cost in dropping
- Net result: motivation to avoid using extension headers (essential de facto best practice)

Proposal

- New ICMP errors
- Can be sent by both hosts and intermediate nodes
- They *are* ICMP errors, so
  - All the caveats, limitations, & reasons to blocking
  - Security considerations similar to existing errors
- Source can at least log the error, possibly take corrective action

New ICMPv6 Parameter problems

- 1 - Unrecognized Next Header type
  - Not new, but allow intermediate nodes to send this it
- 4 - Extension header too big
  - Size in bytes
- 5 - Extension header chain too long
  - Either total size or number of headers
- 6 - Too many options in extension header
  - Applies to destination and hop-by-hop options

New ICMPv6 Dest. unreachable

- 8 - Headers too long
    - Usable for any header chain, not just IP

Reporting priority

1. Real ICMP error (existing codes)
2. Unrecognized Next Header Type (encountered by intermediate node)
3. Too many options in extension header
4. Extension header too big
5. Extension header chain too long (number of headers)
6. Extension header chain too long (size of chain)
7. Headers too long

ICMP error pointer

- Point to:
    - First byte beyond size limit, or...
    - First byte of EH or option that is beyond a count limit
- Methods
    - Pointer field in Parameter Problem
    - Multi-part ICMP for Destination Unreachable with a four byte pointer in multi-part data

Host reaction

- At least log error
- Report error to application
  - Application may be able to backoff what it’s sending
  - Probably needs some new API
- Path characteristics

Thank you!