Server Reply Sequence Number

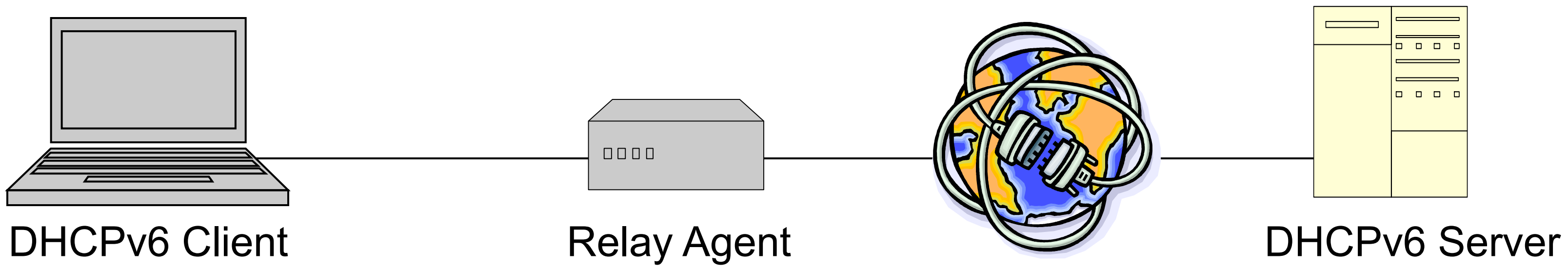
Ralph Droms

Slides prepared by Bernie Volz

Problem

- draft-ietf-dhc-dhcpv6-agentopt-delegate (RAAN) option may be received out of order by relay
- While likely not common, cloud between relay agent and DHCPv6 server can delay packets and reorder packets
- Client is protected because of its state machine and transaction ID
- Relay has no corresponding capability

Example



1. Client sends REQUEST(1), relayed, received by Server.
2. Server sends REPLY(1) with RAAN, delayed in network.
3. Client retransmits REQUEST(2), relayed, received by Server.
4. Server sends REPLY(2) with RAAN, relayed, received by Client.
5. Client sends RELEASE or DECLINE(3), relayed, received by Server.
6. Server sends REPLY (3) with RAAN, relayed, received by Client.
7. Relay receives delayed REPLY (1). Relay processes RAAN options and adds client's addresses/prefixes back. (Client ignores.)

Proposed Solution

- SRSN option is sent by server when sending RAAN option in Reply
- Contains a 64-bit monotonically increasing server reply sequence number
- Relay uses SRSN and Server’s DUID to only process “new” packets for a client on a specific link
  - New determined by comparing against previously received client’s data
- Note - doesn’t protect against multiple servers
- Requires RAAN to include all of client’s data for link
  - Text missing from that I-D

Relay Impact

- A relay MUST
  - Store Server’s DUID and SRSN with client binding data
    - A relay can not retain just a single SRSN for all bindings as packets for different clients may not be in order or may not have been sent in order by the server
  - Retain above information for deleted bindings for a short period in case of delayed packets (similar to TCP’s maximum segment lifetime)
- A relay MUST NOT discard “old” packets instead of relaying them to the client
- A rebooting relay isn’t protected since it has no data with which to compare

Server Impact

- Send SRSN and Server DUID options when sending RAAN option to a relay
- One “simple” technique
  - Split 64-bits into two 32-bit numbers
    - Most significant bits are incremented by 1 each time server starts
      - Requires read and write to storage at start up
      - If no value in storage, initialize from current time
    - Least significant bits (LSBs) can start at 0
    - Only need another write to disk if LSBs roll over (every 2^32 messages)

Security Issues

- New Denial-of-Service attack:
  - A rogue device can inject SRSN with all bits set or some sufficiently “large” value but it must do that once for each client and server

Questions

- This draft needs to progress if RAAN progresses