AMT
draft-ietf-mboned-auto-multicast

Greg Bumgardner
Thomas Morin

AMT Specifications status

- History outline
  - First WG draft: 10 years ago
  - Last WG Last-Call in 2008
  - Tom Pusateri gave the ball to us to pursue the work
    - Original authors deserve all the praise !
    - And we will be the ones to blame for what breaks... ;-)
- What we propose
  - Identify the issues that need to be fixed before the doc can move forward to IESG
  - Identify the possible solutions to fix them, or the question that need to be answered
  - Identify stuff that could happen in an other document
- Let's go through the pending issues...

Sourcing multicast with AMT

- Initial idea:
  - be able to source SSM multicast traffic from behind a non multicast network
- Current issues
  - With current specs, a Gateway may have to send traffic to multiple relays
- Proposed resolution
  - **Move multicast sourcing out of the specs**
  - Anyone objecting to this ?
- Alternative would be to...
  - Document the approach outlined by Greg Shepherd
    - Just add a message letting the Relay provide a usable source address to the Gateway, but do not care about address allocation
  - Will only happen if someone wants to contribute some text
    - Secondary use case, people not that interested
  - (Not documenting sourcing in the base specs does not prevent the sourcing part to progress on its own in a separate document)

DoS on the relay resources

- Issue: **it's easy to create a denial-of-service condition on a Relay by making it instantiate a large number of AMT Tunnels**
  - (malicious intent, or even buggy code...)
- Relay could refuse to do more than one tunnel toward a said Gateway IP address, but this would break the legitimate use case where multiple Gateways are behind a NAT box
- Proposed resolution:
  - Recommend that Relay implementations limit the number of tunnels that can be setup toward a said Gateway IP address:
    - With a knob to tune the max to adopt to all use cases
    - With a default value big enough to allow a few devices behind a NAT box
  - Document that a Relay may withdraw is Anycast Relay prefix when it gets overloaded, to allow new clients to use another relay

Lifecycle

- It seems that current text is not explicit enough on the following:
  - What IGMP Queries are in AMT Queries : specific/general?
  - How shall a Gateway anticipate to anticipate for a loss of a Request message / when to retransmit these / how does the IGMP Query timer allows the gateway to determine when state would expire on the Relay ?
  - When to send discoveries ?
  - How can the Gateway determine how long a (nonce,MAC) tuple will be valid ?
- Proposed resolution
  - Determine when more text is needed to be fully explicit and write it

Feedback

- There are cases where the Gateway won't know that the Relay will not honor a Membership Update:
  - Relay is overloaded
  - (MAC,nonce) tuple isn't valid anymore ?
- Retransmission will solve the issue, but we might want to recover quicker
- Shall we allow some form of feedback to the Gateway ?
  - Flag in the AMT Query message ?
  - Revive IGMP Feedback proposal and send IGMP Feedback messages in an AMT message ?

Troubleshooting and metering when Gateway is behind a NAT

- Suggestion is to allow a gateway behind a NAT box to know about the (IP,port) seen by the Relay, to allow correlating Gateway and Relay logs for troubleshooting and metering
- Proposed solution
  - Extend the AMT Query message to include information on the Gateway (IP,port) of the Request message
  - Use part of the currently « reserved » bytes to indicate the presence of an additional field at the end of the Query message
    - Enough to allow smooth co-existence with existing pre-standard implementations ?
- Text essentially ready to be incorporated (AT&T contrib)

Teardown [1/2]

- Summary of the idea :
  allow a Gateway, after roaming, to indicate to the Relay that it can at once stop sending traffic to the old Gateway IP address
- Goal is to avoid the inefficiency of sending traffic uselessly until old state times out
- Lots of discussions during past meetings

Teardown [2/2]

- Obstacles to adopt this idea (our understanding)
  - Only a partial solution to the inefficiency problem
  - Need to extend messages sent by the Relay to let the Gateway know about its IP when its behind a NAT box
    - There may be other reasons to extend Query message (previous slide)
  - Does this solution introduce a security weakness ?
    - Currently, impersonating a Gateway requires spoofing its IP and guessing a 48 bit number
    - With the Teardown message, spoofing the IP source address is not needed anymore, but guessing a 48 bit number is still needed
    - Enumerating 2^48 values takes a long time (more than two years at 1Gb/s) - isn't it hard enough ?
- **Working group feedback wanted !**

UDP checksumming over IPv6

- Many discussions on this issue in the past
  - Blocking point was to have UDP/IPv6 specs relax the constraint on UDP checksumming
- 6man WG has now adopted draft-ietf-6man-udpchecksums
- We could revise text to say:
  - For IPv4, go back to what revision -09 was saying:
    - « The UDP checksum SHOULD be 0 in the AMT IP Multicast Data message »
  - For IPv6
    - Solution A:
      - « When carried over IPv6, the checksum MAY be set to zero [I-D.ietf-6man-udpchecksums].»
      - 'SHOULD' possibly too strong, because some receiver OS may not be able to follow [I-D.ietf-6man-udpchecksums] yet (?)
    - Solution B:
      - Extend the specs to let a Gateway indicate to the Relay, in the Update message, that it can receive UDP packets with a zero checksum

Security

- There are some undocumented security issues
  - Relay impersonation
    - illegitimate multicast packet injection
  - Issues due to sniffing, man-in-the-middle
- Proposition
  - Document them
  - When doable, recommend generic solutions, such as IPSec, or application-layer solutions
  - Do not necessarily seek to solve them in the document we will submit

Other possible improvements

- Roaming issues could be better solved if the Relay had a way to identify a gateway by something else than the (ip,port) tuple
- A mechanism to allow this could include some cryptographic mechanism to also improve robustness to sniffing/replay
- Authentication of receivers has already been talked about for plain IGMP/MLD ; AMT is a use case in which this would be even more useful
- The above is work in progress that could happen in a separate I-D

Conclusions

- We would like to be able to push these specs to IESG sooner rather than later
- Unless there are objections, we will make an update to the document with some the changes presented here
- Feedback welcome, especially on the less obvious questions