TCP Encapsulation of IKE and IPsec Packets Update

draft-smyslov-ipsecme-rfc8229bis

Valery Smyslov
svan@elvis.ru

Tommy Pauly
tpauly@apple.com

IETF 108

TCP Encapsulation in IKEv2

- Defined in RFC 8229
- Modifies IKEv2 behavior in various situations:
  - original Initiator is responsible for restoring TCP connection if it is broken
  - with MOBIKE if IP address is changed then first try UDP and then switch to TCP
  - NAT keepalives are redundant
  - IKE Fragmentation is redundant
  - etc.
- However, some nuances in using TCP are missing. Most of them affect performance, however few are essential for reliability and interoperability
- This draft is intended to replace RFC 8229 adding missing clarifications

Retransmissions

- RFC 7296 requires exchange initiator to retransmit request periodically until either response is received or the SA is deemed to have failed
- TCP is reliable protocol, there is generally no need to retransmit
- Moreover, in congested networks retransmitting requests can increase congestion making things worse
- However, if TCP connection is lost and then restored, then IKE implementation must retransmit all outstanding requests

Using COOKIE and PUZZLE

- Using COOKIE allows responder to make sure the initiator's IP address is real
- In general COOKIE is not useful with TCP:
  - TCP itself verifies that initiator's IP address is real
  - TCP creates state on responder before first packet ever reaches IKE, that violates stateless nature of COOKIE
- Using PUZZLE still makes sense
- If COOKIE (or PUZZLE) request is sent by responder:
  - TCP connection should be immediately closed by responder (to keep responder stateless)
  - COOKIE calculation must not include initiator's port number (since it will most probably be different)

Error Handling in IKE_SA_INIT

- RFC 7296 advises initiator not to act immediately if error notification is received in IKE_SA_INIT because it can be forged; instead wait for more responses
- With TCP this makes little sense:
  - if this is genuine message from responder, then other responses won't be sent
  - if TCP is hijacked and this is message is forged by attacker, then genuine response won't be received or will be corrupted (because TCP sequence numbers will already be consumed by attacker's message)

Interaction with MOBIKE

- RFC 4555 defines MOBIKE protocol
- RFC 8229 recommends, that if IP is changed, then initiator first tries to send UPDATE_IP_ADDRESSES notify using UDP and then switches to TCP if no response is received
- Clarifications on the NAT_DETECTION_*_IP content and Message ID are still missing
- When switching to TCP:
  - the content of the NAT_DETECTION_*_IP notifications must be recalculated if source/destination ports differ from UDP's
  - Message ID for TCP-based exchange must remain the same as for (failed) corresponding UDP-based one

Interaction with High Availability Clusters

- RFC 6311 defines IKE Message ID & ESP SN synchronization mechanism between IKE peer and HA cluster:
  - when cluster failover takes place the new active node initiates INFORMATIONAL exchange containing new Message IDs & SN gap
- In case of cluster failover the existing TCP connection is most likely broken and the new active node cannot initiate the exchange until the client restores it (by sending fresh IKE or ESP packet):
  - client is unaware of the fact that the connection is broken, so if it has nothing to send, the connection won't be restored for a long time, and the cluster would eventually tear down the IKE SA
- Clients should periodically send Liveness Check messages if the partner is HA cluster and there is no outgoing ESP traffic

Thank you!

- Comments? Questions?
- More details in the draft
- WG Adoption?