IKE over TCP

draft-nir-ipsecme-ike-tcp

Y. Nir

Why?

- A few months ago we started getting reports from some of our customers
- IKE had stopped working
- Turns out that the larger IKE packets were not making it to the other side
  - Packets #5 and #6 in IKEv1 Main Mode
  - Both packets of IKE_AUTH in IKEv2
- The culprit turned out to be the ISP. They were dropping all fragments

Why?

- The reports seem to be coming from SE Asia and from Australia, where IPv4 is becoming scarce, and CGNs are being deployed.
- This is prohibited by RFCs, including the CGN draft, but it's happening anyway.
- Regardless of the reason, it looks like we can no longer assume that the Internet will carry any size UDP datagram that we throw at it.
- We need a way to avoid fragmentation

This has happened before

- We have run into this issue before
- In the late '90s and early '00s, when remote access was just starting to get deployed, we have already had the issue of middleboxes dropping fragments
- At the time it was home routers that were broken. This time it's the ISPs.

This has happened before

- At the time, we added a new transport for IKE: IKE over TCP
- This has worked fine for remote access.
- We tried to bring it to the IETF before
- We got the “your customers should buy better routers” response
    - Even got the “TCP? No, SCTP is better” response

What's Changed?

- It's no longer bargain basement home routers
- With more CGNs, there may be more of this at the ISPs.
- While you might be able to get yourself whitelisted with your own ISP, it gets much harder when it's some other ISP down the line.

Other Solutions

- One of our competitors has added a fragmentation layer within IKE, so large IKE messages get sent in several packets and re-assembled on the other side.
- IMO this is exceedingly complicated, and if we do that then IKE has:
  - A segmentation layer
  - exponential back-off of retransmissions
  - A transmission window (in IKEv2) initialized to 1
- Those who would not use TCP are doomed to re-create it.

Other Solutions

- We could try to name & shame ISPs into not dropping fragments
  - or making exceptions for UDP ports 500 and 4500.
- Make packets smaller:
  - Hash & URL
  - Move to PSK
  - Configuration to avoid certificate chains
    - But everybody's going to 2048-bit certs
  - Don't send CRLs.

IKE over TCP

- This solution is rather lightweight. IKE messages already have a length field, so it's easy to get them in a stream protocol like TCP.
- We can use port 500, as it's already allocated to “isakmp”.
- It can move arbitrarily large messages
- We have over a decade of operational experience running it with thousands of peers.

IKE over TCP

- Possible policies:
  - Send only IKE_AUTH over TCP, everything else over UDP
  - Send the whole Initial+IKE_AUTH over TCP, everything else over UDP
  - Send every IKE request over TCP

Open Issues

- Do we want to also specify IKEv1?
- Should retransmissions in TCP be forbidden, or just discouraged
- Liveness checks, should they be sent over TCP?
- Is it OK to send the Initial exchange over TCP?
  - That means discovery through a 3-way handshake
  - Alternatively we could have discovery through a Notify payload in the Initial request and response
    - Which are sent over UDP