MOBIKE issues

Pasi Eronen
IETF63 MOBIKE WG
August 3, 2005

Issue tracker

- http://www.vpnc.org/ietf-mobike/issues.html

Issues handled in –01

- Mostly editorial
  - 21: Editorial comments from Lakshminath
  - 25: Editorial comments from Mohan
  - 26: Window size and latest update counter
  - 29: Editorial comments from Tero
- Editorial fixes and clarifications in –01
- Word “path” now used only in senses that include the route

Issues handled in –01 (cont.)

- 23: Payload type of addresses
  - – Separate payloads for IPv4/IPv6 in –01
- 30: Protocol ID in notifications
  - – Protocol ID 0 used in –01

Some easy issues

24: NAT prevention details

- This feature needs a better name
- My proposal:
  - Discuss better names on mailing list
  - May depend on issue 22

Issue 35: Version number

- Should Mobike_Supported payload also contain a version number?
- Arguments against
  - Future extensions or MOBIKEv42 can use same negotiation mechanism as MOBIKE (add Notification payloads), a separate one not needed
- Arguments for
  - Some future extensions might save some bytes
    - But only if future extensions are “linear” MOBIKE v2,v3,v4,v5,…, not if they’re orthogonal features
- My proposal: Don’t add

Issue X(37)

- Move Mobike_Supported notification from IKE_SA_INIT to IKE_AUTH
  - Allows per-user policy about whether MOBIKE is allowed

Issue 36: Unacceptable_Addresses

- If Update_SA_Addresses message was retransmitted with different src/dst IP, we don’t know which path was unacceptable
  - Because can’t add/change anything to the messages after they have been sent once
- My proposal:
  - The initiator knows if it has retransmitted the message with several addresses
  - If it has, just try again (send a new Informational request with Update_SA_Addresses)

27: Security and path testing

- Security considerations section needs text about path testing
- Details depend on issue 34
  - But currently it looks like no big differences between path test outside IKE_SA (message not encrypted/MACd) vs. using Informational exchange
- My proposal: Wait until issue 34 is closed, then write text.

Other open issues

- 22: Is disabling NAT traversal a possibility?
- 28: Comments about security considerations
- 31: Responder address changes
- 32: Omitting COOKIE2 for non-RR messages
- 33: Changing ports 500/4500 and RR

Issue 34: Path testing, changes in NAT mappings

34: Background

- NAT mappings (~“port seen by peer outside NAT”) can change if NAT is rebooted or keepalive interval is too long, etc.
- If peer outside NAT continues using the old port, its packets won’t reach the other peer

34: Background

- IKEv2 NAT Traversal recovers from this by automatically updating the information from authenticated packets (ESP or IKEv2)
- Specified as a “SHOULD”
  - Non-trivial to implement
  - Some people thought that handling this situation is not important (i.e., breaking the connection is OK)
  - At least one implementation known to update only port number (and accept updates only with same IP address)

34: Approaches

- 1) mobike-protocol-01 keeps the IKEv2 NAT Traversal approach as “SHOULD”
  - 1b) Upgrade “SHOULD” to “MUST”
- 2) Change to “MUST NOT” (break if NAT mappings change)
- 3) Change to “MUST NOT”, but specify some other way to recover

34: Approach 3

- Current host behavior without NATs:
  - If outgoing traffic but no incoming traffic within X (e.g. 15-300 sec), send empty Informational request
  - Also send it if no incoming traffic within Y (e.g. 30 min)

34: “Approach 3a”

- If outgoing traffic but no incoming traffic within X (e.g. 15-300 seconds), send Informational request with NAT detection payloads
  - If reply contains different NAT_D payloads than last time, mappings have changed → send Update_SA_Addresses
- Also send it if no incoming traffic within Y (e.g. 30 min)
- If no outgoing traffic within Z (e.g. 30 sec), send NAT-T keepalive

34: “Approach 3b”

- Same as 3a, but also include Update_SA_Addresses in the first Informational request just in case

34: Comparison of 1 vs 3

- In approach 1, ESP packets cause updates
  - Faster recovery
  - But may complicate implementation
- In approach 3
  - Recovery is slower
    - Possible pressure to decrease X considerably (e.g. 100→10 seconds) → increased load on gateway?
    - Increased number of IKEv2 exchanges has impact on high availability scenarios (client-transparent failover between gateways)
  - Possibly cleaner implementation

34: Comparison of 1 vs 3

- Approach 3 allows client to detect when NAT mappings change
  - In Approach 1, this can be done using Path_Test messages
- Choose more appropriate keep-alive interval → less load on gateway?

34: Relationship to Path_Test

- Without a separate Path_Test exchange, there are time periods when the initiator can’t do path testing without possibility of disrupting things (if window size 1)
- With approach 1
  - There are more of these time periods → more need for separate exchange
  - If you want to do NAT detection in path testing → need separate exchange
- With approach 3
  - Fewer of these time periods

34: Summary

- Both approaches 1 and 3 for handling changes in NAT mappings work (?)
- Both have some pros and cons
- Which we choose has an impact on how path testing is done
  - Although “at any time without disrupting” needs a separate exchange even with 3
  - Separate exchange means additional code, but possibly much simpler
    - In MOBIKE work, bottleneck does not seem to be lines of code, but complexity and intellectual effort needed…

Next steps

- Get rough consensus on most important technical issues
  - How to handle changes in NAT mappings
  - Path testing
  - “NAT prevention” details
- Handle editorial comments received
  - Fix security considerations
  - Clarify role of multihoming
- Then WGLC?