outbound-04 open issues

Rohan Mahy

rohan@ekabal.com

STUN UDP DoS attack

- Currently when STUN response indicates change in mapped address, UA assumes NAT rebooted and reregisters immediately. There is no authentication of these STUN keepalives
- On unprotected 802.11 networks, an attacker who can see the STUN request, can easily send a response with a different mapped address and will usually beat the real response. This causes UA and registrar (and intervening proxies) to do work. Can melt the registrar.
- Options:
  - add timer delaying re-registration if binding appears to change more than seems reasonable in some period?
  - do nothing (use TLS if you care)
- Proposal: add timer

Discovering STUN support

- Current text says that keepalive STUN support is “configured” (using a very broad definition of configure).
- Incompatible with current DHCP proxy discovery (but so is sigcomp) due to limitation of DHCP option
- Do we need another way of discovering/probing for STUN support defined **in this document?**
- Options
  - progress the document and fix later with new mechanism
    - fix DHCP option (write a new DHCP option that returns a URI)
    - create a new DNS-based mechanism
    - use the configuration framework
    - use something like service-route
  - delay this document yet again waiting for one of these
  - add a probe mechanism: UA can send OPTIONS to first-hop, can “shift-up” to keepalive=stun.
- Proposal: go forward as is

Validating STUN support

- Currently presence of keepalive=stun used as indicator its OK to send STUN requests
- Some expressed concern about misconfiguration causing proxies that don’t do STUN to get confused. More of a problem for TCP.
- Options:
  - treat this as a non-problem
  - revisit STUN keepalives over TCP problem
  - don’t send STUN until UA validates its OK, by including a parameter in reflected Path, but Path not always present...
  - have UA try the request with a Proxy-Require: sip-stun
  - probe with OPTIONS. only way to validate keepalive=stun
- Proposal: treat as non problem

STUN keepalive definition

- Many requests that STUN keepalive description is pulled into a separate section that can be implemented independently

- Does anyone object to doing this?
  - move keepalive=stun to new section (and discovery/validation if we added that)
  - leave avalanche restart timers in main sections of outbound

How many flows (minimum)?

- Current draft says (from Section 4.2):
  - For each outbound proxy URI in the set, the UA MUST send a REGISTER in the normal way using this URI as the default outbound proxy.
- Options:
  - leave as is (MUST)
  - change to SHOULD. allows UA to setup fewer flows at its discretion (does no harm)
- Proposal: change to SHOULD

Additional discovery and semantics of outbound-proxy-set

- Various proposals to discover or manufacture outbound-proxy-set or apply different semantics
- Can be addressed in future extensions without any changes to outbound
- We don’t even have all the requirements written down for these proposals.

- Proposal:
  - address in future specs. can add substantial complexity and delay.

Why does registrar send to EP over same connection?

- Only needed if EP is in a foreign domain—for authorization

- Draft defines EP obliquely as in a foreign domain, and other proxies as a disaggregation of a proxy/registrar
- Incredibly confusing. Needs to be more explicit

How to verify edge proxy support

- Currently draft says that edge proxy adds a flow token to a Path header and forwards. No discussion about how the registrar decides that edge proxy actually did this.
- Proposal:
  - edge proxy adds new parameter to Path URI to indicate it added flow token appropriately
  - ex: Path: <sip:*token*@ep.example.com;lr**;ob**>
  - If no token present, registrar ignores reg-id parameter, doesn’t return Supported: outbound

Presence of Supported: outbound

- What does this mean?
  - Today it means that the registrar supports outbound.
  - Client can use this to understand that if it needs to cleanup old Contacts using RFC 3261 style matching
- Proposals:
  - clarify this in the text
  - only include Supported: outbound if any edge proxy supports outbound as well

Re-registering with same reg-id

- Original intent of reg-id is so that UA can indicate if it wants to add a new flow or to ***replace*** an existing flow
- Dave Oran asked for change to language (ex: SHOULD replace) to allow for a private use case
- WG was later uncomfortable with idea of deleting/replacing flows. In Dallas seemed to have consensus to use most recent flow. Since then, many complaints on the mailing list about this, and no one motivating/defending previous choice.
- Problems with this:
  - proxy can send traffic to the wrong place (for UDP flows)
  - no way for registrar to delete state until registration expires
  - no way for UA to say it really wants to replace flow
  - harder to implement on registrar
  - not motivated by requirements
- Options:
  - leave draft as is
  - change back to SHOULD replace flows with same reg-id
  - say registrar MUST replace flows with same reg-id
- Proposal
  - SHOULD replace flows

Refresh registration on same flow

- Erkki pointed out that registration refresh should happen over the same flow it refreshes.
- Duh! Thanks Erkki.
- Will clarify in next rev

Usage of 410 response Code

- Adam Roach pointed out that semantics of 410 response code is inconsistent with our usage.
- Options:
  - 480 Temporarily Unavailable
  - 481 Call Leg Does Not Exist (for mid-dialog)
  - New 4xx response
- Proposal:
  - Use 480