draft-ietf-tram-stunbis-08

Marc Petit-Huguenin – Gonzalo Salgueiro
2016-07-18

STUNBis is now feature complete

All changes since RFC 5389:

- Added support for DTLS-over-UDP (RFC 6347).
- Made clear that the RTO is considered stale if there is no transactions with the server.
- Aligned the RTO calculation with RFC 6298.
- Updated the cipher suites for TLS.
- Added support for STUN URI (RFC 7064).
- Added support for SHA256 message integrity.
- Updated the PRECIS support to RFC 7613.
- Added protocol and registry to choose the password encryption algorithm.
- Added support for anonymous username.
- Added protocol and registry for preventing biddown attacks.
- Sharing a NONCE is no longer permitted.
- Added the possibility of using a domain name in the alternate server mechanism.
- Added more C snippets.
- Added Test Vector.

Since stunbis-04:

- Userhash support.
- Generic bid down attack protection.
- MESSAGE-INTEGRITY-SHA256 truncation.
- Removed Salted SHA256, added MD5 in password algorithm registry.
- Added test vector with USERHASH, bid down protection and SHA256.
- Updated the change list.

Userhash support

- Inspired by a similar feature in RFC 7616, a revision of the HTTP Authentication Digest mechanism.
- Permits to transmit a hash of the username instead of the username itself, which is useful when DTLS is not in use.
- SHA256 is always used and the resulting 32 bytes are transmitted without encoding.

Generic bid down attack protection

- As for the password algorithm, the userhash feature requires bid down attack protection, but adding more special prefixes to the nonce would have made things more complex.
- Instead we added a 24 bit bitmap encoded in base64 between the prefix and the remaining of the nonce.
- Each bit signals the protection of one security feature, and we allocated two bit in a newly created registry, one for the password algorithm and one for the userhash.

M-I-256 truncation

- As proposed, the draft now permits to truncate the HMAC stored in the MESSAGE-INTEGRITY-256 attribute.
- Each STUN Usage should explicitly explain if this feature is used and what is the minimum truncation size.
- STUNbis need a review to set the minimum truncation size that STUN Usages can use.

Test vector

- In the same spirit than RFC 5769, we provide a STUN packet protected with a MESSAGE-INTEGRITY-SHA256 attribute.
- We also used a USERHASH and a nonce cookie to serve as example.
- Note that the example will be correct only after receiving a codepoint from IANA for USERHASH and M-I-256.

Ship it, it’s done

- Questions?