Digest Based Authentication

James Undery - Ubiquity

Sanjoy Sen - Nortel Networks

Vesa Torvinen - Ericsson

Digest Authentication

- SIP & RFC 2617
  - It’s deployed today
  - It’s simple
  - UAC to UAS
  - UAC to proxy
- draft-undery-sip-auth-00
  - proxy to UAS
  - Bid-down protection
  - Integrity protection
  - Mutual authentication
  - Suggested replay protection implementation

What’s New

- proxy to UAS authentication
  - UAS-Authenticate
  - UAS-Authorization
  - UAS-Authentication-Info
  - 492 response

- Bid-down protection
  - Prefixes added to nonces
  - Protects scheme and quality of protection
  - Doesn’t protect algorithms (See open issues)

What’s New

- □ Mutual Authentication
  - ○ *-Authentication-Info non digest specific
- □ Integrity
  - ○ Complete one hop message integrity
  - ○ Negotiated header integrity
    - ⋄ With bid down protection

Open issues (1/4)

- □ No algorithm protection
  - ○ If a hashing algorithm is broken (i.e. one of the following hold,) we need algorithm revocation
    - ◇ given H(m) you can extract m
    - ◇ given m and H(s+m) you can find n, st H(s+m) = H(s+n)
  - ○ Proposal - make limitation explicit, algorithm revocation is out of scope
- □ No negotiation of body integrity protection
  - ○ Proxies can’t alter message bodies
  - ○ Proposal - leave unchanged

Open issues (2/4)

- □ No protection against weak passwords
  - ○ If a scheme uses a weak password / session key protection can be compromised
  - ○ Care should always be taken to match the strength of protection against the tim

    you wish a secret to remain secret
  - ○ Proposal - make limitation explicit, the solution is out of scope
- □ 492 Mechanism requires UAC support
  - ○ UAS has already applied policy that could result in failure of this dialog
  - ○ Proposal - leave unchanged, explicitly note deficiency

Open issues (3/4)

- Client side can’t initiate authentication
  - Could include *-Authorization header or new header / Require option
    - Require option only applies to target of challenge and generally a bad idea
  - Can’t respond to responses in general
    - Can immediately respond to tear-down dialog created by requests
  - Only one server responding in a non 4xx/5xx/6xx manner will cause problems
  - Proposal - make limitation explicit

Open issues (4/4)

- Forking and response collation issues
  - Can’t guarantee upstream entities see challenges
  - Response collation oriented towards success
  - Proposal - make limitation explicit