SOCKS Protocol Version 6

draft-olteanu-intarea-socks-6-06

Vladimir Olteanu
SOCKS Sessions

- (Elegantly) share state across multiple requests
  - Done since -01, but on a per-username basis

- Motivation
  - Tor
  - Credential sharing across clients (e.g. multiple browsers)
Tor

- One circuit per bar domain
- Current behavior: use SOCKS5 + username/password authentication
- Encode bar domain in username (i.e. username = “wikipedia.org”)
Idempotence + shared credentials (-05)

- Clients risk spending each other’s tokens
- At best: occasional wasted RTTs
Idempotence + shared credentials (-05)

- Clients risk spending each other’s tokens
- At best: occasional wasted RTTs
- At worst: livelock
Other motivators (-05)

- Unauthenticated clients:
  - Can’t use Idempotence options
  - Can’t use Listen Backlog options

- Shared credentials + BIND Backlog: X “listens” and Y can accept X’s connections
SOCKS Sessions

- Proxy holds shared state on a per-session basis
  - Was on a per-username basis
- (By default) authentication is waived once a session is established
- The proxy decides when to kill a session (e.g. using inactivity timers)
  - The client can also instruct it to do so
- All options are part of Requests and Authentication Replies
  - Only the client-proxy RTT is relevant
SOCKS Session options

<table>
<thead>
<tr>
<th>Kind</th>
<th>Length</th>
<th>Type</th>
<th>Session Option Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>Variable</td>
</tr>
</tbody>
</table>

- Type: Request, ID, Teardown, OK, Invalid, Untrusted
- Option data: only used by Session ID options
Establishing a session

- The session-initiating request also part of the session
  - Corollary 1: can also request a token window, etc.
  - Corollary 2: can’t initiate a session from within a session

- Session ID: opaque sequence of bytes (arbitrary length)
Further requests

- The client’s credentials are tied to the Session ID
- Authentication is no longer performed
Invalid Session ID

- Authentication automatically fails (even if not required by proxy policy)
Session teardown

- Free session state early, rather than after a timeout
- The session-killing Request is part of the session
Untrusted sessions

- The client must authenticate every time it makes a Request
  - With the same credentials
- Only protects against passive attackers
- Open question: Do we want this feature?
  - Or just leave security to TLS?
Other changes in -06

• Future-proofing: options have a 2-byte length field
  – Can fit X.509 certificates etc.

• Option count (max 255) replaced with options length (2 bytes, but capped at 16KB)

• Authentication methods: eliminated user data encryption
  – Backward compatibility: encryption can still be negotiated, but it is not honored
  – Can still run SOCKS over TLS.

• Nits and quality-of-life changes for implementers