SOCKS Protocol Version 6
draft-olteanu-intarea-socks-6-00

Vladimir Olteanu, Dragoș Niculescu
University Politehnica of Bucharest
Motivation

- SOCKSv5 makes liberal use of round trips
  - Authentication method negotiation
  - Authentication
  - Remote connection establishment

- 0-RTT authentication possible after pre-negotiation

- Hot use case: “Bond” 3G/4G/LTE and WiFi using MPTCP
  - Little to no MPTCP support on the server side
  - Use proxy to convert to regular TCP
  - Mobile networks have high latency
Improvements over v5

- Client sends as much information as possible upfront
  - Optimistic, doesn’t wait for authentication to conclude
  - Method advertisement, server address, some application data
- Client can specify if it wants TFO on the proxy-server leg
- Extensible: TCP-like options
- 0-RTT authentication support via options
SOCKSv5 vs. SOCKSv6 [1/2]
• Can include authentication data in the request on subsequent connections
### SOCKSv6 Request

<table>
<thead>
<tr>
<th>Version</th>
<th>Number of Methods</th>
<th>Methods</th>
<th>Command</th>
<th>TFO</th>
<th>Address</th>
<th>Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major</td>
<td>Minor</td>
<td>Methods</td>
<td>Code</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Variable</td>
<td>1</td>
<td>1</td>
<td>Variable</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Options</th>
<th>Initial Data</th>
<th>Initial Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>Size</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Variable</td>
<td>2</td>
</tr>
<tr>
<td>Variable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Includes auth. method advertisement
- Includes initial data
- Options in TLV format
  - May include authentication data
SOCKSv6 Authentication Reply

<table>
<thead>
<tr>
<th>Version</th>
<th>Type</th>
<th>Method</th>
<th>Number of</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major</td>
<td>Minor</td>
<td></td>
<td>Options</td>
<td>Variable</td>
</tr>
</tbody>
</table>

- Informs client whether more authentication is needed or not
  - If 0-RTT authentication failed: selects which authentication method to use
  - If 0-RTT authentication succeeded: informs client which method was used
SOCKSv6 Operation Reply

<p>| Version | Reply | Address | Bind | Bind |</p>
<table>
<thead>
<tr>
<th>Major</th>
<th>Minor</th>
<th>Code</th>
<th>Type</th>
<th>Address</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Variable</td>
<td>2</td>
</tr>
</tbody>
</table>

<p>| Number of | Options | Initial Data |</p>
<table>
<thead>
<tr>
<th>Options</th>
<th></th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Variable</td>
<td>2</td>
</tr>
</tbody>
</table>

- Reply code indicates whether the connection was successful or not (and why: RST, timeout, etc.)
- Initial data offset lets the proxy avoid buffering data while the client authenticates
SOCKSv6 in action: no TFO anywhere

- Data reply in 2 RTTs
  - No worse than vanilla TCP
SOCKSv6 in action: TFO on proxy-client leg

- Data reply in 1 end-to-end RTT + 1 proxy-to-server RTT
  - **Negative overhead**: We save 1 client-to-proxy RTT, assuming the proxy is on path
  - Highly advantageous for mobile networks, where layer 2 has high delay
SOCKSv6 in action: TFO everywhere

- Data reply in 1 RTT
  - Same as when contacting the server directly
Implementation

- Early prototype (some differences from draft)
  - Message library: https://github.com/45G/socks105
  - Proxifier + proxy: https://github.com/45G/shadowsocks-libev