EAP Extension to Allow Peer Configuration

Sandeep Tamrakar, Philip Ginzboorg, Pekka Laitinen
Huawei
Motivation

- We are working on IoT bootstrapping security for consumer IoT devices
- Bootstrap resource constrained devices such as temperature sensor to a resourceful device such as mobile phone (controller)
  - Use OOB channel to transfer information that is used to secure bootstrap process
- In the context of IoT, bootstrapping involves:
  - Pairing a resource-constrained IoT device with a controller device such as a smartphone
  - Taking ownership by exchanging identities and credentials for mutual authentication and securing communication
  - Configuring the device to be operational
- Bootstrapping with EAP
  - How to use EAP to bootstrap devices including long term credential provisioning
Related Work

• Credentials Provisioning and Management via EAP (EAP-CREDS)
  • A framework that has board goals

• EAP-TEAP
  • Allows peer device to provision client certificates
Goal

• Use EAP as a mechanism to enable Peer configuration from an EAP Authentication Server

• The configuration could be used for
  • Provision long-term credentials,
  • Set access control policies

• A simplest possible solution from implementation and specification point of view
Generic EAP Message Flow

**Peer**

- Common Messages
  - Start
  - Request Identity
  - Response Identity
  - Request 1
  - Response 1
  - Request n
  - Response n

- EAP method
  - Request 1
  - Response 1
  - Request n
  - Response n

- Common Messages
  - Success

**Authenticator**

- Request Identity
- Response Identity
- Request 1
- Response 1
- Request n
- Response n
- Success

**Authentication Server**

- Request 1
- Response 1
- Request n
- Response n
- Success
Possible approach for Peer configuration with EAP

- Configuration messages only to be sent after underlying EAP method has completed peer authentication
- Success message may or may not depend on the successful configuration however it must depend on EAP peer authentication

<table>
<thead>
<tr>
<th>Common Messages</th>
<th>Peer</th>
<th>Authenticator</th>
<th>Authentication Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Messages</td>
<td>Start</td>
<td>Request Identity</td>
<td>Response identity</td>
</tr>
<tr>
<td>EAP method</td>
<td>Request 1</td>
<td>Response 1</td>
<td>Response 1</td>
</tr>
<tr>
<td></td>
<td>Request n</td>
<td>Response n</td>
<td>Request n</td>
</tr>
<tr>
<td>Configuration Messages</td>
<td>Request 1</td>
<td>Response 1</td>
<td>Request 1</td>
</tr>
<tr>
<td></td>
<td>Request n</td>
<td>Response n</td>
<td>Request n</td>
</tr>
<tr>
<td></td>
<td>Success</td>
<td></td>
<td>Success</td>
</tr>
</tbody>
</table>
Issues to consider

• Message fragmentation
  • A simple way is to fit each payloads within a single EAP message

• Push vs pull model
  • Push: Support for server to push configuration messages
  • Pull: Let the peer device request configuration parameters

• Discovering mechanism
  • A simple and efficient way to find out if the other end-point supports configuration

• Security for the configuration
  • Based on EAP session key
  • Using MSK+EMSK for securing the configuration messages

• Avoiding unnecessary roundtrips
  • Probing for configuration needs to be very efficient

• Limits on the number of EAP messages caused by the AP and EAP server
Possible approaches

1. Define new EAP message type for configuration messages
   • EAP request and response type
   • EAP request can be sent in either direction
   • Reuse an existing Notification request and response

2. Define a new EAP method that uses existing EAP tunneling method for authentication
   • Similar to EAP-CRED

3. Define a mechanism that allows an end-point to indicate that another configuration protocol shall continue after the EAP session has ended
   • Mechanism allowing the configuration protocol to bind with the EAP session
     • E.g. a shared secret from the EAP session
Guidance from EAP group

Thank you.