draft-perez-abfab-kerberos-preauth-options

IETF 83, Paris
Kitten & Kerberos WGs

Alejandro Perez & Josh Howlett
Objective

• Improve inter-organisational use of Kerberos
  – RFC5868 describes some issues of contemporary Kerberos cross-realm operation in large-scale systems

• Some interest in using a AAA-based cross-realm architecture
  – AAA is highly effective at federated authentication for network access (i.e., mobile telephony & data, IEEE 802.1X, LTE, etc)

• ABFAB provides AAA-based federation architecture for application-level protocols
  – EAP for authentication & SAML for authorisation
  – RADIUS / Diameter provides federation (and authorisation)
  – GSS-API enables integration with applications
Options

- Two different models to integrate ABFAB and Kerberos have been discussed:
  1. The Kerberos client is the ABFAB initiator
  2. The Kerberos client is the ABFAB acceptor
- The models are similar but address different use cases
- Two different approaches for binding EAP to Kerberos have also been discussed:
  1. Bind EAP directly to Kerberos
  2. Bind via a GSS pre-authentication mechanism, using ABFAB’s GSS-EAP mechanism
Model 1: Kerberos client is the ABFAB initiator

- **ABFAB Initiator/Kerberos Client** obtains a TGT from the KDC using an EAP-based pre-authentication mechanism
  - Straightforward integration of Kerberos and AAA infrastructures
  - User is authenticated by the home domain, but obtains a ticket from the KDC (acting as an EAP pass-through authenticator)
  - After Kerberos pre-authentication, the client uses standard Kerberos to obtain STs for the services within the visited domain
Model 2: Kerberos client is the ABFAB acceptor

- **ABFAB Acceptor/Kerberos Client** obtains a ST from the KDC using an EAP-based pre-authentication mechanism
  - Kerberos client uses EAP tokens from ABFAB initiator to authenticate against KDC
  - User is still authenticated by home domain
  - Abfab RP and KDC act as ‘split EAP authenticator’

```
User         ABFAB          Kerberos         EAP
Agent <-------> RP <----------> KDC <------> Server
            (GSS-EAP)    (EAP over   (EAP over AAA)
                        Kerberos)       AAA)
```
Approaches for implementing EAP-based Kerberos pre-authentication

• Initial description of EAP over Kerberos and GSS-EAP over Kerberos in:

• More detailed description:
Approaches for implementing EAP-based Kerberos pre-authentication

1. EAP pre-authentication
   
   – Use Kerberos as EAP lower layer
   
   – Architecture more straightforward
   
   – Need to define the whole interface with EAP stack and framing
Approaches for implementing EAP-based Kerberos pre-authentication

2. GSS pre-authentication
   - Introduces an additional layer
   - GSS becomes the lower layer
   - GSS-EAP is already defined and implemented by ABFAB WG
     • Makes GSS-preauth simpler to define than EAP-preauth
   - Flexibility: allows the use of other non-EAP GSS mechanisms
     • Extensible to other forms of federation
## Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>GSS-preauth</th>
<th>EAP preauth</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MODEL 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UA</td>
<td>EAP peer GSS initiator Kerberos client</td>
<td>EAP peer Kerberos client</td>
</tr>
<tr>
<td>RP</td>
<td>Application server</td>
<td>Application server</td>
</tr>
<tr>
<td>KDC</td>
<td>EAP authenticator GSS acceptor</td>
<td>EAP authenticator</td>
</tr>
<tr>
<td><strong>MODEL 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UA</td>
<td>EAP peer Abfab GSS initiator</td>
<td>EAP peer Abfab GSS initiator</td>
</tr>
<tr>
<td>RP</td>
<td>EAP authenticator (split) Abfab GSS acceptor GSS pre-auth initiator Kerberos client</td>
<td>EAP authenticator (split) Abfab GSS acceptor Kerberos client</td>
</tr>
<tr>
<td>KDC</td>
<td>EAP authentication (split) GSS pre-auth acceptor</td>
<td>EAP authentication (split)</td>
</tr>
</tbody>
</table>
Summary

• The models address similar but different use cases
• Model 1
  – Preference for GSS pre-auth
• Model 2
  – Preference for EAP pre-auth
• Can we do both? If not, which?