DAMe
Deploying Authorization Mechanisms for Federated Services in the eduroam Architecture

Sascha Neinert
Marseille, 06.02.2008
Overview

- Project Goals
- Partners
- Network Authorization
- Unified Single Sign On
Project Goals

1. Network Authorization
   - Further development of eduroam, the Europe-wide NREN roaming federation
   - Fine-grained network access control based on attributes
   - For properties of the network

2. Unified Single Sign On
   - Using eduGAIN, the European AAI confederation architecture
   - Interoperability with existing AAIIs based on Shibboleth, PAPI, …
   - Token-based authentication for web services
   - Unified Single Sign On for network, web- and Grid services
**Goal 1: Network Authorization**

Diagram illustrating the process of network authorization and authentication in the eduroam system. The diagram shows the interaction between a user's device (supplicant) and the network access point, followed by communication with the network access server (RADIUS) and policy decision point. The user's details are then sent to the Authentication Authority (RADIUS) and Attribute Authority for authorization and attribute release policies. The diagram also highlights network authentication and authorization protocols (RADIUS/EAP/SAML, HTTPS/SOAP/SAML).
Network AuthZ Components

- XSupplicant
  - Recovery and storage of the eduToken

- FreeRadius
  - Request of the eduToken from the HomeBE
  - Delivery of the eduToken using a TLV in the tunneled success message
  - New RADIUS attribute in the response with the user's handle

- LDAP_RemoteBE
  - Receives the user's handle via LDAP
  - Requests the user's attributes using eduGAIN
  - Consults the PDP to get the user's network properties

- PDP
  - Implemented as a servlet and using the XACML library
  - Using the XACML policies, decides the network properties based on the user's attributes

- ...
Network AuthZ Workflow

1. The user requests access to the network.
2. The request is forwarded to the home Radius server.
3. The request is validated using a key shared with the Radius server.
4. The supplicant is notified about the success.
5. The user is authenticated based only in its identity.
6. The request is validated using the DN of the requesting BE.
7. An authn assertion is sent back.
8. The eduToken is built based on the assertion.
9. The supplicant stores the token.
10. Based on the ARP and using the handle to identify the user, the attributes are recovered from the LDAP and sent back.
11. The Access-Accept message is sent including the network properties.
12. The properties are enforced and the access is granted.
13. The handle is included as an attribute in the Radius response.
14. The eduToken is sent to the user through the PEAP tunnel.
15. The handle is used to request the network properties to the LDAP server.
16. Acting as BE, this element requests the user's attributes.
17. The request is forwarded to Shibboleth.
18. Shibboleth consults the PDP using the attributes to get the network properties.
19. Shibboleth builds the Authn Assertion and handle.
20. The request is validated.
21. The eduToken is sent to the user through the PEAP tunnel.
22. The handle is included as an attribute in the Radius response.
23. The eduToken is sent to the user through the PEAP tunnel.
24. The handle is used to request the network properties to the LDAP server.
25. Acting as BE, this element requests the user's attributes.
26. The request is forwarded to Shibboleth.
27. Shibboleth consults the PDP using the attributes to get the network properties.
28. Shibboleth builds the Authn Assertion and handle.
29. The request is validated.
30. The eduToken is sent to the user through the PEAP tunnel.
31. The handle is included as an attribute in the Radius response.
32. The eduToken is sent to the user through the PEAP tunnel.
33. The handle is used to request the network properties to the LDAP server.
34. Acting as BE, this element requests the user's attributes.
35. The request is forwarded to Shibboleth.
36. Shibboleth consults the PDP using the attributes to get the network properties.
37. Shibboleth builds the Authn Assertion and handle.
38. The request is validated.
39. The eduToken is sent to the user through the PEAP tunnel.
40. The handle is included as an attribute in the Radius response.
41. The eduToken is sent to the user through the PEAP tunnel.
42. The handle is used to request the network properties to the LDAP server.
43. Acting as BE, this element requests the user's attributes.
44. The request is forwarded to Shibboleth.
45. Shibboleth consults the PDP using the attributes to get the network properties.
46. Shibboleth builds the Authn Assertion and handle.
47. The request is validated.
48. The eduToken is sent to the user through the PEAP tunnel.
49. The handle is included as an attribute in the Radius response.
50. The eduToken is sent to the user through the PEAP tunnel.
Goal 2: unified SSO

- User's Device (Supplicant + Token Client)
- Network Access Server (RADIUS)
- Access Point (802.1X)
- Visited Domain
- eduroam confederation
- Home Domain
- Authentication Authority (RADIUS)
- Attribute Authority (Shibboleth, PAPI, ...)
- Service Domain
- Service Provider (Shibboleth, PAPI, ...)
- eduGAIN confederation
- eduGAIN

Network Authentication (RADIUS/EAP/SAML)
Web Authentication and Authorization (HTTPS/SOAP/SAML)
uSSO Components

- **DameTokenManager**
  - Java *Client* Application (eduGAIN + openSAML libraries)
  - Receives eduToken from supplicant
  - Provides eduToken to DameTokenFetcher

- **DameTokenFetcher**
  - Signed Java *Applet*
  - Fetches eduToken from DameTokenManager
  - Sends eduToken to DameTokenServlet

- **DameTokenServlet**
  - Java *HttpServlet* (eduGAIN + openSAML libraries)
  - Receives eduToken from DameTokenFetcher
  - Create Shibboleth assertions and send to Service Provider
  - Using fromSAML and toShibbolethSAML of Shibboleth remote Bridging Element
uSSO Workflow

802.1X Access Point

RADIUS

eduroam

Username
Password

DameTokenManager
uSSO Workflow

802.1X Access Point

RADIUS
eduroam

DameTokenManager

Access-Accept + eduToken

DAMe Project, Sascha Neinert, 06.02.2008
uSSO Workflow

1. User enters URL
2. DameTokenManager
3. DameTokenServlet (eduGAIN r-BE)
4. Web/Grid Service + Shibboleth SP
5. Authentication needed
6. RADIUS
7. 802.1X Access Point
8. eduroam
uSSO Workflow

- 802.1X Access Point
- RADIUS
- eduroam
- DameTokenManager
- DameTokenFetcher
- DameTokenServlet (eduGAIN r-BE)
- Web/Grid Service + Shibboleth SP

- enter URL

DAME Project, Sascha Neinert, 06.02.2008
uSSO Workflow

1. **Enter URL**: User enters the URL for the service.
2. **DameTokenManager**: Validates the token and creates an assertion.
3. **Web/Grid Service + Shibboleth SP**: The assertion is validated by the Shibboleth SP.
4. **DameTokenServlet (eduGAIN r-BE)**: The token is sent to the DameTokenServlet for further processing.
5. **RADIUS**: The token is sent to RADIUS for authentication.
6. **802.1X Access Point**: The user is authenticated through the access point using 802.1X protocol.
7. **eduroam**: The user is granted access through the eduroam network.
**uSSO Workflow**

1. **User** enters URL
2. **DameTokenManager** requests token
3. **EDUAGAIN r-BE** issues token
4. **DameTokenServlet** authenticates user
5. **Web/Grid Service + Shibboleth SP** grants access
6. **RADIUS** and **802.1X Access Point** validate credentials
7. **eduroam** provides network access
Questions?

Any questions or comments?

Visit the DAMe website:  http://dame.inf.um.es/

... see you @ DAMe-2 😊