Contents

- Background
- Use-cases
- Brief overview of architecture
- Progress to date, and future plans
- Kerberos & Moonshot integration proposal
Research & Education Networks

- Provide advanced network services for R&E.
- Traditionally focused on connectivity services.
- Rapid growth in trust and identity services.
  - X.509 PKI
  - RADIUS federation
  - SAML federation
RADIUS federation for network authentication
SAML federation for Web single sign-on
Motivations

1. Provide customers with a single ‘federation backhaul’.

2. Address our customers’ emerging use-cases.

3. Fix some known issues with SAML and RADIUS federation today.
Use-case 1: Out-sourcing

- Our customers increasingly want to:
  - Reduce costs by out-sourcing commodity services to third party service providers.
  - Use their own managed identities to provide SSO and enable conformance to data protection legislation.
- SAML provides this for Web-based services...
- ...but not other types of services (IMAP, POP3, SMTP, CalDAV, etc).
- Identity Provisioning APIs exist, but they’re typically not appropriate.
Use-case 2: High Performance Computing

- HPC facilities are increasingly critical to our customers.

- Requirements:
  - Improve Business Continuity by federating access to HPC facilities.
  - Offer HPC-as-a-service to external customers.
  - Reduce costs incurred in operating HPC-specific authentication service.
  - Provide a better user experience.
Learning from SAML federation

- In federating new applications, avoid problems already discovered with SAML federation today (and fix them).

- As a federation grows in size:
  - Users are presented with an ever-growing list of identity providers (“IdP discovery problem”).

- As a federation grows in scope:
  - Users may acquire more than one identity provider (“multiple affiliations problem”).
Technology choices

- SAML provides authorisation and attributes.
- GSS-API mechanism for application integration.
- EAP authentication encapsulated in GSS-API to gain existing credential support.
- RADIUS transport provides federation.
Background: EAP for network access

- EAP peer (supplicant)
  - EAP lower Layer (e.g., 802.11i)
  - EAP MSK

- Authenticator
  - AAA
  - EAP lower Layer (e.g., 802.11i)
  - EAP method

- EAP server
  - AAA
  - EAP server
  - EAP MSK
Client

- Supplicant
  - GSS-API
  - Client application

Server

- AAA
  - GSS-API
  - Server application

EAP server

- EAP server
  - AAA

- draft-ietf-abstract-aaa-saml
  - sstc-saml-binding-aaa-draft

- draft-ietf-abstract-gss-eap
  - draft-ietf-abstract-gss-eap-draft
Goals

- To deliver
  - A standardised architecture.
  - A production-quality open-source implementation.
  - Packaged and shipped with Debian Linux.
  - A test-bed for interoperability testing.
  - High quality documentation.
  - An active community of users and developers.

- To enable
  - Third-party implementations by vendors and other communities.
  - Available for all computing platforms.
Software development

- GSS EAP library supporting MIT Kerberos & Heimdal
- SASL support through Cyrus GS2 plugin.
- Apache: implement a new mod_auth_gss.
- Firefox: update the Negotiate implementation.
- Shibboleth SP: extend to permit use for SAML processing in the non-Web case.
- FreeRADIUS: extend to support EAP channel bindings.
- libradsec: library for RadSec clients (i.e. the GSS EAP acceptor) and servers.
- Extend Open1x and wpa_supplicant to support application authentication ("identity selector") and EAP channel bindings.
What have we achieved so far?

- Phases 1-3 (January 2010 → April 2010)
  - Feasibility Analysis & draft specifications.
  - Bar BOF @ IETF 77.

- Phase 4 (April 2010 → June 2010)
  - Use-case development
  - Started development of draft project plan.
  - Started development of IETF Working Group charter.

- Phase 5 (June 2010 → August 2010)
  - IETF 78 “FedAuth” BoF: consensus to form a working group (ABFAB).
  - Project plan completed
    - See http://www.project-moonshot.org/plan
Current & planned activities

- Phase 6 (August 2010 → January 2011)
  - First project meeting (September, Copenhagen)
  - Advance specifications through IETF and OASIS.
  - Implement the core technologies
  - Proof of concept demonstration.

- Phase 7 (February 2011 → July 2011)
  - Second project meeting (East coast US, Jan/Feb)
  - Develop remaining technologies.
  - Implement test-bed.
Current limitations

• EAP takes lots of round trips

• No support for n-tier applications

• No resource domain concept
Borrowing from Kerberos

• Kerberos with a ticket is one round-trip

• Kerberos provides authorisation mapping within a domain.

• Kerberos has good n-tier support.
Extending Moonshot with Kerberos

• Optionally return ticket from acceptor to initiator.

• Future round-trips use ticket as optimisation.

• Service ticket or TGT.

• Operation with or without a KDC.
Moonshot initial Kerberos

Client
- EAP peer (Identity selector)
- GSS-API
- Client application

Server
- AAA
- GSS-API
- Server application

EAP server
- EAP server
- AAA
- IdP

Kerberos ticket
Moonshot same server

Client

- Kerberos client
- Moonshot
- Client application

Server

- Kerberos service
- Moonshot
- Server application

EAP server

**Kerberos AP-REQ**

**Kerberos AP-REP**
Moonshot with a KDC

- KDC sits between server and RADIUS within the resource domain.
- EAP over Kerberos FAST, then over RADIUS.
- KDC issues service ticket to service and TGT to client.
- Key hierarchy protects TGT from service.
Get involved!

• Your opinions and ideas.

• Use-cases, use-cases, use-cases.

• Join the Project Moonshot mailing list.

• Join the IETF ABFAB mailing list.

• Participate in the test-bed.
http://www.project-moonshot.org

**Project partners**

JANET(UK) (http://www.ja.net)

GÉANT (http://www.geant.net)