

Project Moonshot

IETF 77, Anaheim

Sam Hartman, Painless Security LLC
Josh Howlett, JANET(UK)



Use-cases

- 1) Support federated authentication to out-sourcing providers
- 2) Federated authentication to applications beyond HTTP
 - IMAP, XMPP, enterprise-specific applications
- 3) High Performance Computing
 - Address HPC community requirements (Business Continuity & HPC-as-a-service)
 - Federated SSH, NFS, CIFS
- 4) Scalable SAML metadata trust

Learning from Web SSO

In creating federated authentication for new applications, avoid problems discovered with existing web federations:

1) Identity discovery

- User presented with hundreds of possible identity providers
- Solution: guide identity selection with what identities a client has

2) Multiple affiliation problem

- Difficult to choose the correct identity for a given service

Expected benefits I

- Users
 - Selection of an identity using a client-based “identity selector”.
- User-affiliated organisations
 - Apply federated identity to a range of new services, improving usability and reducing effort to support different authentication systems and credentials.
 - Increase ROI already made in federated identity.

Expected benefits II

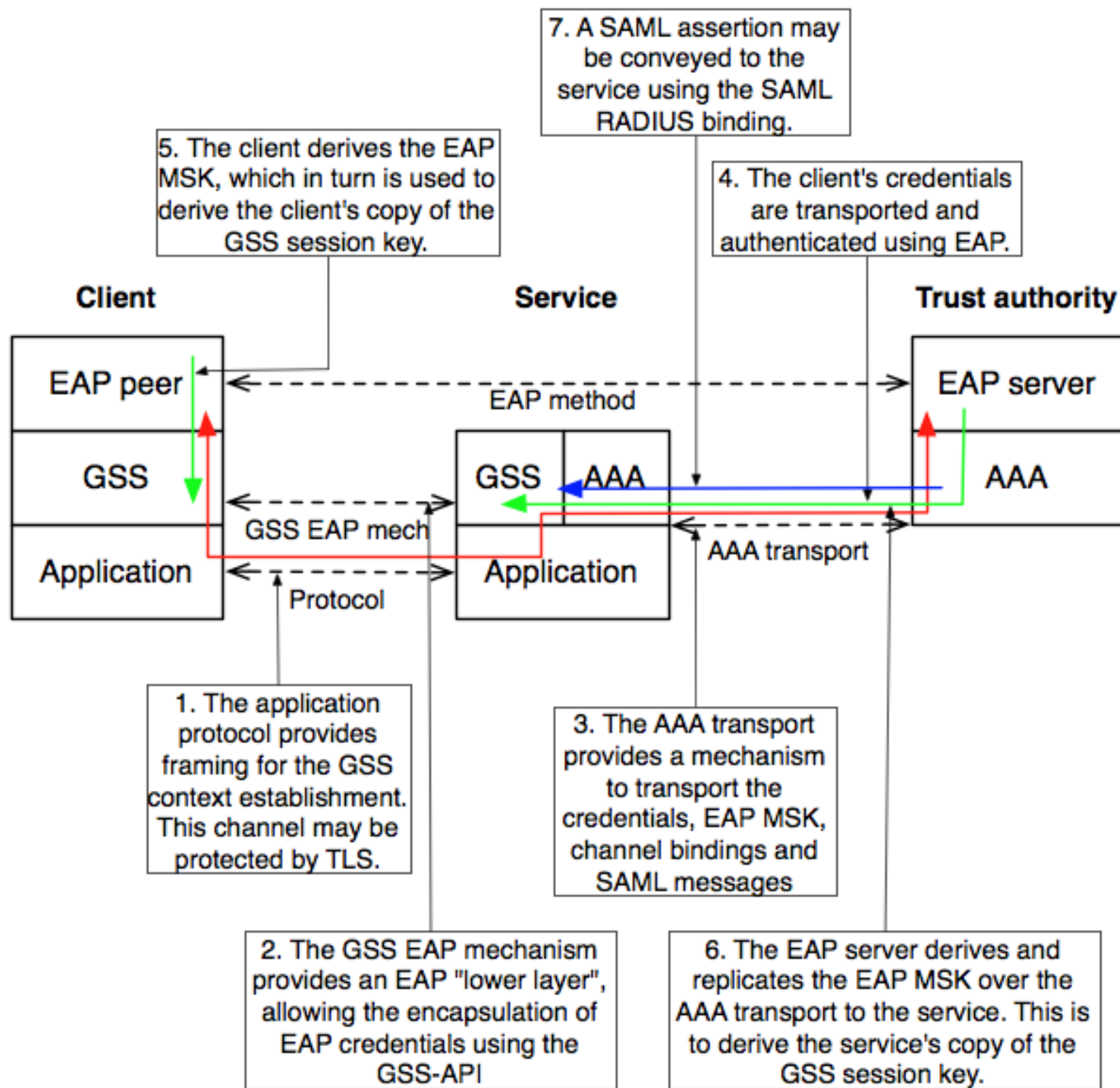
- Service providers
 - Introduces the benefits of SAML-based federated identity (anonymity, rich authorisation, etc) to new types of services.
- SAML implementations
 - Provides a SAML-based SSO profile enabling federated identity for arbitrary applications without requiring significant profiling (if any).
- Standards developers
 - Provides a SAML-based SSO profile to support federated identity without requiring significant profiling (if any).

Design Factors

- Leverage existing EAP infrastructure for authentication
- Leverage existing SAML infrastructure for attribute release and authorization
- Standards-based solution for application integration
- No standardization required for each application

Architecture Overview

- GSS-API mechanism for application integration
- EAP authentication encapsulated in GSS-API to gain existing credential support
- RADIUS transport provides federation
- SAML provides authorization and attributes
-



Background Slides

Architecture

- EAP GSS mechanism
- RADIUS SAML attributes
- SAML RADIUS binding
- SAML EAP profile

Architecture

- EAP GSS mechanism
 - EAP lower-layer between an EAP peer (GSS initiator) and Authenticator (GSS acceptor).
 - Uses RFC 4121 tokens
 - Base key derived from the EAP key.
 - If acceptor is acting as a pass-through authenticator
 - EAP key and channel bindings are replicated from EAP server over AAA channel
 - Can be used with any AAA transport
 - Work needed to define necessary EAP channel bindings
 - No support for delegation presently

Architecture

- RADIUS SAML attributes
 - Defines RADIUS attributes for transporting SAML messages
 - All values are compressed and expressed using the RADIUS “string” binary format.

Architecture

- SAML RADIUS Binding
 - Specification that will be taken to OASIS SSTC
 - Defines how to bind SAML request/response protocol messages to AAA transport using the RADIUS SAML attributes

Architecture

- SAML EAP Profile
 - Specification that will be taken to OASIS SSTC
 - Defines how to compose the EAP GSS mechanism and the SAML RADIUS binding to effect SSO for application protocols.

Notes for architecture diagram

- The following slide has a protocol stack diagram showing how the three layers (application, GSS/AAA, EAP) interact for the three parties (EAP peer, authenticator, EAP server)
- The top of the stack has the EAP layer, with an arrow directly between the peer and server representing the EAP method transport.
- The middle of the stack has the GSS and AAA layer, with an arrow between the peer and the authenticator representing the EAP over GSS transport, and an arrow between the authenticator and the EAP server representing the EAP over AAA transport.
- The bottom of the stack has the application protocol, with an arrow between the peer and the authenticator. All of these transport arrows are black.
- There are four coloured arrows superimposed on this that illustrate the flow of the security information:
 - A green arrow representing the export of EAP key and bindings from the peer's EAP layer to the GSS layer.
 - Another green arrow representing the equivalent export of EAP key and bindings from the EAP server replicated over AAA to the authenticators GSS layer.
 - A blue arrow showing the push of SAML assertion from the server to the authenticator.
 - A red arrow showing the passage of the EAP messages through the various boxes and along the arrows representing transport.