SAML-SASL integration

Klaas Wierenga
TF-EMC2, TF-Mobility Vienna
17-2-2010

© 2007 Cisco Systems, Inc. All rights reserved.

Problem

- Organizations deploy SAML IdP's
- But many non-Web apps (IMAP, Jabber/XMPP)

- Looking for a straightforward solution, i.e. no gazing at the sky ;-)
- Minimal effort for client application builders
- Many applications DO support SASL

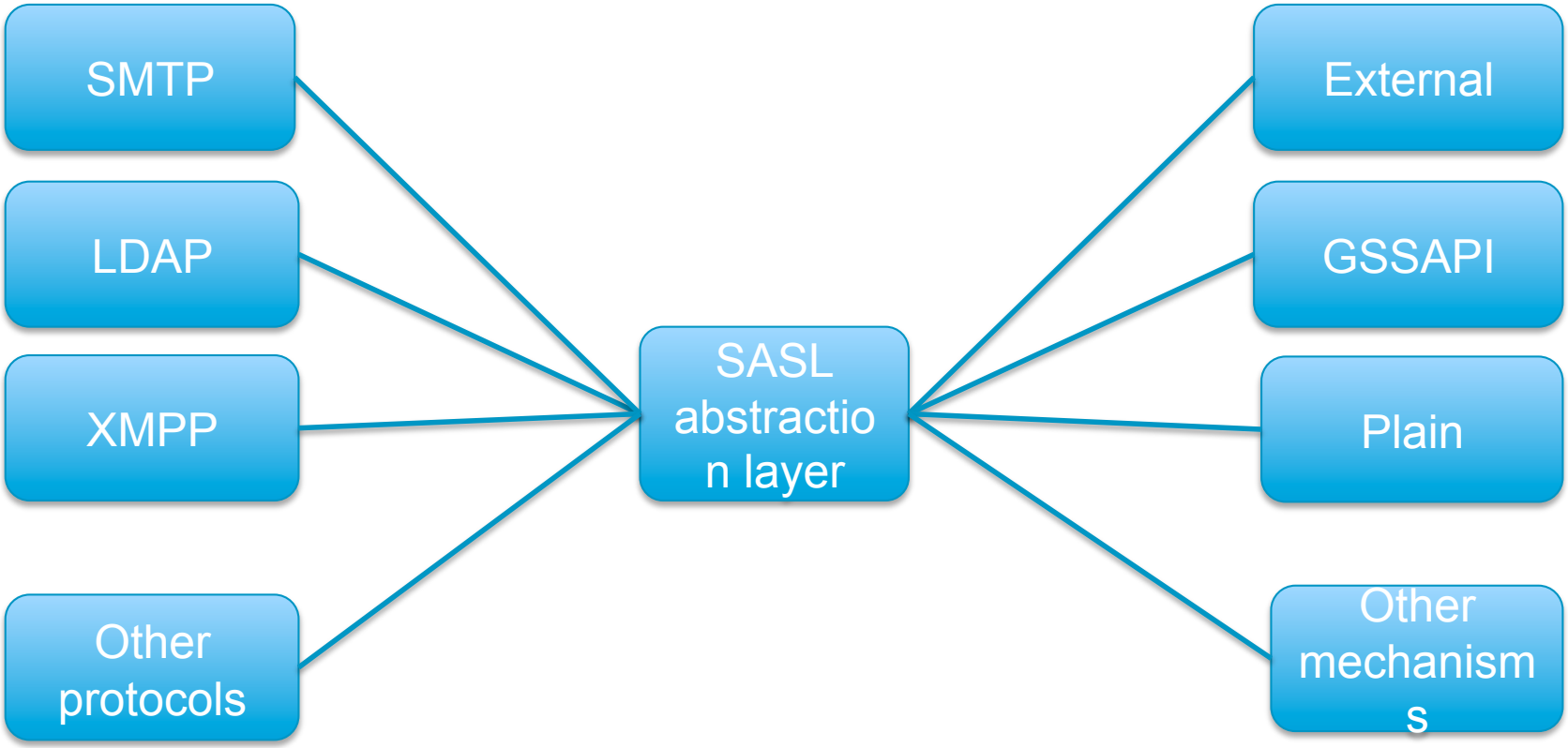
© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

SASL

- Simple Authentication and Security Layer
- RFC4422 (obsoletes RFC2222)
- Framework for providing authentication and data security services in connection-oriented protocols via replaceable mechanisms
  - –Old protocols can use new mechanisms
  - –New protocols can use old mechanisms
- Includes protocol for securing protocol exchanges within a data security layer

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

Abstraction layer between protocol and mechanism



- Protocols: AMQP, BEEP, IMAP, LDAP, IRCX, POP, SMTP, IMSP, ACAP, ManageSieve, XMPP
- Mechanisms: EXTERNAL, ANONYMOUS, PLAIN, OTP, SKEY, CRAM-MD5, DIGEST-MD5, NTLM, GSSAPI, GateKeeper

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

High level flow

- C = SASL client, S = SASL server

C: Request authentication exchange

S: Initial challenge

C: Initial response

<additional challenge/response messages>

S: Outcome of authentication exchange

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

SAML

- You know ;-)

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

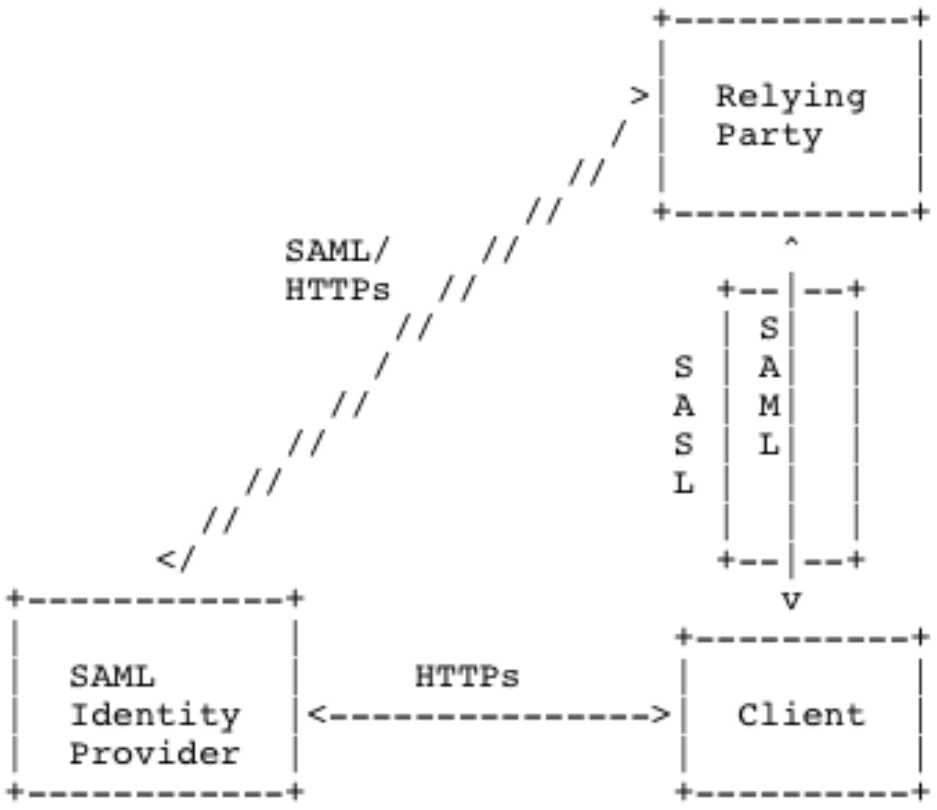
My proposal

- New SASL mechanism: SAML2
- http://tools.ietf.org/id/draft-wierenga-ietf-sasl-saml-00.txt

- SASL server sends SAML AuthnRequest as challenge
- Out-of-band SAML AuthN

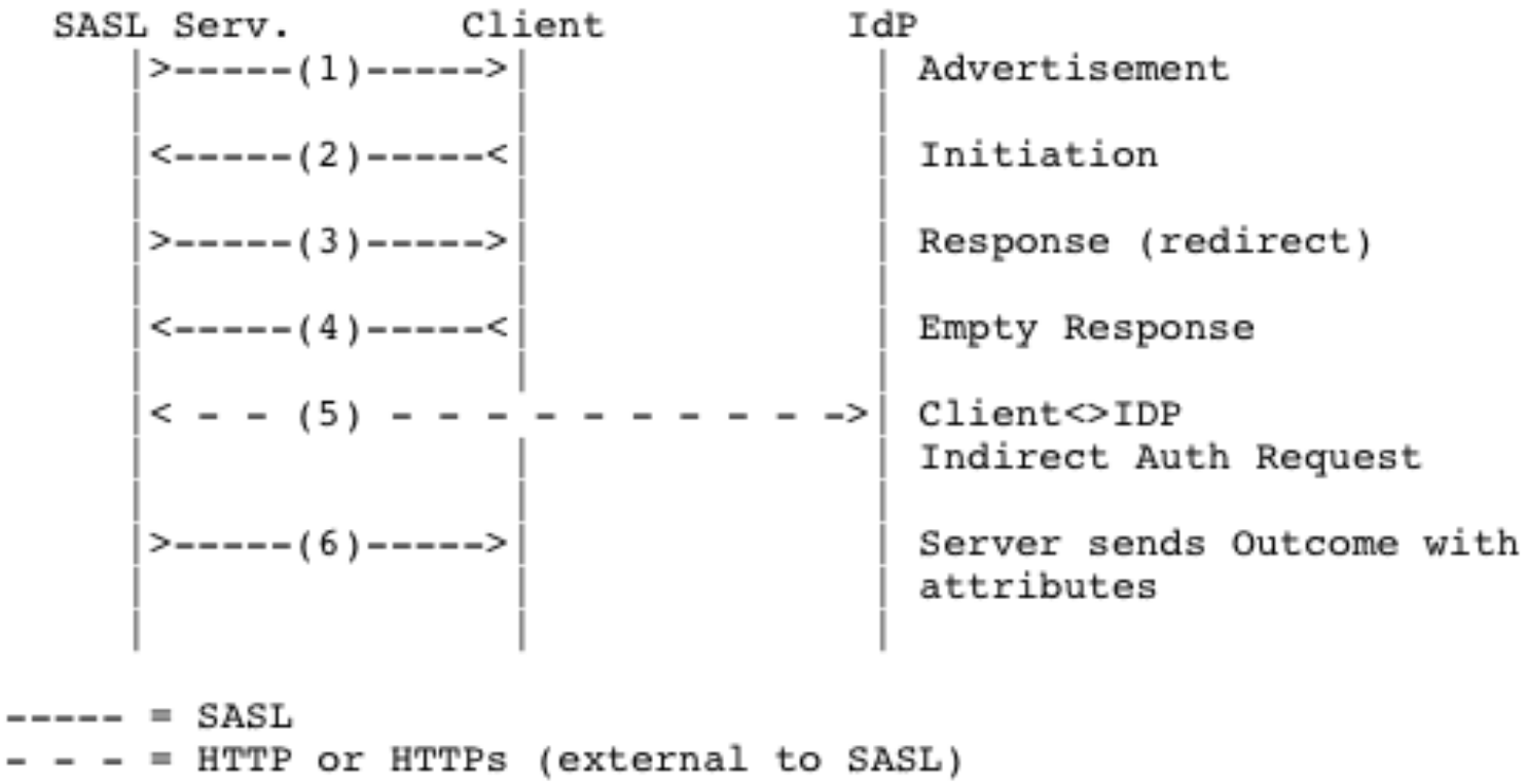
© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

Interworking architecture



© 2007 Cisco Systems, Inc. All rights reserved.

Authentication flow



© 2007 Cisco Systems, Inc. All rights reserved.

Example: Jabber (XMPP) AuthN

- J=Jabber, S=SASL
- J: Step 1: Client initiates stream to server
- J: Step 2: Server responds with a stream tag sent to client
- S: Step 3: Server informs client of available authentication mechanisms (PLAIN, SAML20 etc.)
- S: Step 4: Client selects an authentication mechanism (SAML20)
- S: Step 5: Server sends a BASE64 [RFC4648] encoded challenge to client in the form of an HTTP Redirect to the SAML assertion consumer service with the SAML Authentication Request as specified in the redirection url)
- S: Step 6: Client sends a BASE64 encoded empty (!) response to the challenge
- […… SAML magic……..]
- S: Step 7: Server informs client of successful authentication
- J: Step 8: Client initiates a new stream to server
- J: Step 9: Server responds by sending a stream header to client along with any additional features (or an empty features element)
- J: Step 10: Client binds a resource
- J: Step 11: Server informs client of successful resource binding

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

Self assessment

- Good:
  - –Leverages SASL
  - –Client builders don’t have to understand SAML
  - –Browser ubiquitous
  - –Clients can be made more intelligent
- Bad:
  - –Ugly kludge with browser
  - –Only works for SASL enabled apps

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

Alternative approach (in-band)

- Dixit Scott Cantor
  - –SASL server challenges SASL client with an SAML AuthnRequest
  - –SASL client relays AuthnRequest to its IdP over SOAP, HTTP, TLS, ….SAML
  - –IdP sends SAML AuthnStatement back to client
  - –Client sends SAML AuthnStatement to SASL server
- Good: Much cleaner
- Bad: chicken and egg between client and server builders, need to define SASL binding in SAML, yet another client you trust with your credentials

© 2007 Cisco Systems, Inc. All rights reserved. Cisco Public

QUESTIONS?

© 2007 Cisco Systems, Inc. All rights reserved.

CISCO

© 2007 Cisco Systems, Inc. All rights reserved.