Interoperable Authentication for the Virtual Observatory: an Update

Ray Plante, Bill Baker, Mike Freemon
NCSA

Ramon Williamson, Patrick Duda
NCSA

Matthew Graham
Caltech

Andrew Cooke, Chris Miller
NOAO
Reminder: What we’d like to see from interoperable security

• Trustworthy access to restricted resources
  - Remote storage: VOStore
  - Proprietary data: new data from observatory archives
  - CPU cycles: services that require significant computing power
    Resource providers need full control of access rights and auditing

• Single sign-on
  - User enters a username/password once per session
  - Can access many restricted resources in an operation from different providers
    - User can use the NOAO portal to access proprietary Hubble data in MAST
  - Interoperates with public (non-restricted) data and services seamlessly
  - Interoperability with Grid security

• A framework that can be leveraged by observatories
  - A common way to get at proprietary data
  - A common way to support security in portals
    - Simple toolkit for portal developers
Portal–Managed Certificates

A short-lived *proxy* certificate is returned to the portal.
Portal–Managed Certificates

The proxy certificate is used to access restricted services—e.g., to retrieve proprietary data.
Portal–Managed Certificates

During the same session, the portal can even retrieve proprietary data from other VO–compatible archives.
Framework Principles

• X.509 Certificates for authenticating to restricted services
  - Globus convention for certificate chaining, proxies

• Weak and strong certificates
  - Weak allows immediate access to some services
  - Strong certificates are backed by identity verification

• Emphasis on portal-managed certificates
  - Users experience similar to current portals
  - Users unaware of use of certificates
  - Power users still have access to certs for use with specialized clients.
  - Implemented with PURSE, MyProxy, pubcookie

• Identity Verification Services
  - Leverage community structures to verify user’s identity

• Authorization is mainly a local issue
  - Service providers know who is allowed to do what
  - Access rights are not centrally managed for community
  - VO projects can assist with attribute management
Framework Design

• Each major regional VO project runs a User Authentication Server (UAS)
  - User registers with UAS’s CA to create a VO identity
  - Each regional UAS may specialize the user registration process to their needs
  - Each may employ their own user identity verification mechanisms
  - Total number of UASs worldwide should be \( \leq 5 \)
    • Hold down the number of CAs that services need to support

• Portals use UAS standard interfaces to register, login users
e.g. observatory archive portals, service portals, etc.
  - Users talk directly with UAS; portals never see passwords
  - UAS returns a short-lived certificate to portal.
  - Authorization is handled by the portal provider
    • May be based on user attributes contained in certificate
  - Users generally must register with portal on first visit
    • Allows portal to track and manage own user information
Demo

http://sky1.ncsa.uiuc.edu/nsahome/nsahome.html
Framework Implementation

Components

- **User Authentication Server**
  - **Globus PURSE**
    - Support portal-managed credentials
    - Contains user database
  - **Pubcookie**
    - Provides login service
    - Enables cross-portal single sign-on
  - **MyProxy**
    - For retrieval of credentials
    - Added support for Pubcookie authentication
      - Cookie provided by login service to user’s browser contains secure authenticating token
      - Browser delivers cookie to portal
      - Portal passes pubcookie token to MyProxy in lieu of password

- **Portal Server**
  - **Apache modules: mod_pubcookie, mod_myproxy**
    - Helps Portal manage pubcookie and myproxy interactions transparently
Framework Implementation

- Two phase implementation
  - Phase 1: based on current PURSE implementation
    - PURSE includes a CA
    - Long-term cert stored in MyProxy repository
    - MyProxy delivers proxy credentials
  - Phase 2: move CA to MyProxy
    - PURSE manages user data
    - MyProxy produces short-lived certificate based on latest user data
Pubcookie-based authentication

When astronomer visits portal, she is directed to the UAS Login Service to log in.
Pubcookie-based authentication

The astronomer provides the Login Service with a username and password; if valid, the Login Service sets 2 secure tokens validating the user.
Pubcookie-based authentication

The portal retrieves the cookies from the browser, extracting and validating the secure token. The token is passed to MyProxy In lieu of password to Retrieve credentials.
Pubcookie-based authentication

When astronomer visits 2nd portal, she is directed to the login service.
Pubcookie-based authentication

The Login Service detects the logon cookie from 1st portal’s session. It skips query for username/password and redirects user to portal with session cookies.
Pubcookie-based authentication

Portal

Archive #1
e.g. NOAO Science Archive

Cookies

username/
Pubcookie token

Portal

Archive #2
e.g. NRAO

VO Project
UAS e.g.,

Login
Service
(pubcookie)

CA
(Purse)

Certificate
Repository
(MyProxy)

credentials

2nd portal transparently
retrieves credentials.
Becoming a compliant portal

- Obtain portal support kit
- Register with UAS
  - Obtain a server certificate to allow portal to connect to myproxy service
  - Obtain a symmetric key for pubcookie decryption

Supporting user–managed certs in client apps
- Power users enable direct download of credentials in user profile
  - To discourage phishing on naïve users
- Use standard myproxy tools/libraries to retrieve creds
Identity Verification

- **Weak Identities**
  - Confirm only that registrants email is correct
  - Many services don’t need strong identity assurance
    - won’t need to know that users are who they say they are
    - Just need to know that an identity is the same user across sessions
  - Provides immediate access to some restricted resources

- **Strong Identity Verification**
  - Engage asynchronously one or more Identity Verification Services (IVS)
    - Example IVS hosts: observatories, home academic depts.
    - UAS sends registration info, IVS confirms information
  - UAS records IVS confirmations into user DB as they come in
  - When certificate is issued, identifiers for confirming IVSs are encoded into certificate
  - Services decides which IVSs it trusts/requires to allow access
    - Service would apply local authorization tests on top
Identity Verification Policies

- What a “yes” response means depends on IVS policy
  - Home department:
    - The named person is a member of the dept. with the given email address
    - The named person has confirmed having registered with the NVO with the given login name
  - Observatory:
    - The named person with the supplied email address has been award time on our telescope
  - NVO would verify persons via traditional means to help get network of trusted identities going.

- Each registered IVS publishes policy with VO project
  - Project would provide toolkit/assistance to local authority for installing IVS.

- Mechanics and Procedures still under development
  - Globus project is developing tool support for encoding IVS IDs into certificates
  - What is a trustworthy but workable system from community’s perspective?
    - Engage existing trust procedures
Authorization

• VO Project UAS may manage regionally related user attributes
  - Ex: UK astronomer, EU astronomer
  - Attributes encoded in certificates
    • As SAML embedded assertions
  - Leverage Shibboleth infrastructure
    • GridShib
  - Setting of attributes may be incorporated into IVS mechanism

• Services would use attributes to manage authorization policies
Conclusions

• We’ve built a working prototype
  – Built on existing grid tools
  – Through close collaboration between the NVO and grid specialists
  – Ready to begin working with portal developers

• Based on a user–friendly but scalable model
  – Single-Sign On that operates across administrative domains
    • Regional CAs
    • Pubcookie: interoperability across portals in a secure way
  – Focus on portal–managed certificates
    • But also allow access to services via specialized clients
  – Weak & Strong certificates
    • Weak lowers the users entry barrier
    • Strong engages framework for verifying identities
  – Locally–managed authorization policies aided by regionally–managed attributes