IAM for Workday: How to Embrace an 800 Pound Gorilla

Michael Brogan & Jonathan Pass
UW-IT, Identity & Access Management

Agenda

- Background
- IAM Integrations
- Parting Thoughts
- Questions

Background

- The UW has been using HEPPS, a 35 year old payroll system running on the mainframe
- HEPPS doesn't meet UW business needs
- UW went through a multi-year business process redesign, RFP, and contracting process
- Workday and IBM (implementation partner) got the contract
- Go-live scheduled for June 2016

IAM Integrations

- Identity Registration
- Credentialing
- Directory Services
- Authentication
- Multi-factor Authentication

Identity Registration

- Goal: Real-time integration between Workday and UW Identity Registry for hires/rehires
- Requirements
  - IdReg must: store legacy worker records, be the source for EIDs used by WD, provide worker NetIDs back to WD
  - Registration, matching, and EID creation must work even when WD lacks core matching data

Identity Registration - Solution

- Loaded legacy workers into IdReg
- Ported EID algorithm over from HEPPS
- WD business process triggers an integration with Identity Registry Web Service (IRWS)
- WD searches for matching identities in IdReg
- WD adds/updates employee data for existing identity or creates new one with worker affiliation
- IRWS generates EID and returns in payload
- Second IRWS call gets the worker's NetID, if exists

Identity Registration - Issues

- Took some time to get Workday up to speed on using our API (IRWS)
- We use certificates for client authentication to IRWS
- Workday uses a self-signed cert for outbound API calls and will only accept server certs from CAs in their trust store (no UW CA or InCommon)
- No WD business process to manage merges of employee records

Credentialing - Goals

- Project Goals / Requirements
  - Users can establish UW NetID via process similar to experience today
  - Record UW NetID in Workday in 1-2 days
  - Keep complexity of Workday integration to a minimum
- IAM Goals
  - Loose coupling and keep real-time credentialing
  - Don’t lower assurance with weaker handling of secret

Credentialing - Solution

- Sequence diagramming: Utilize AWS with notification events from IdReg
- Utilize hire's "home" email address as address of record for delivery of secret
- Supervisor receives copy (Cc) of email with the secret
- Utilize Workday SOAP UI for syncing UW NetID as "Workday Account" in near real time

Credentialing - Issues

- While having just one “employee ID” users may end up with three WD records (employee, contingent, affiliate)
  - Prefixing one or two of the WD accounts to allow UW NetID linking to active role
  - Mid version API release allowed for affiliate UW NetID integration

Directory Services

- Goal
  - Maintain current service design and service levels
- Requirements
  - Real time not a priority
  - Support display of all job information instead of only two from mainframe schema
  - Continue to support publishing preference options

Directory Services – Solution

- Person Directory no major differences
  - Primary appointment info instead of self selected
- White pages
  - Use official working title and include all jobs
  - Publishing preferences moving to IdReg
  - All or nothing

Directory Services - Issues

- Faculty and Staff will not be able to freely edit their working title and department name
- Still nightly delay but there may be opportunities for real-time
- Original design was all or nothing privacy controls
- IAM driving privacy control enhancements, hiring a developer to help

Authentication

- Goal
  - Workday integrates with UW web authentication services to provide UW NetID authentication for users
- Requirements
  - Use SAML2 and UW Shibboleth IdP

Authentication - Solution

- Configured Workday SAML SP via WD UI
- Workday configured UW IdP metadata
- Workday SP registered metadata with UW IdP via SP Registry app
- IdP provides UW NetID in nameid format
- SP does not consume attributes
- Workday uses forced reauth (kills SSO)

Authentication - Issues

- Workday is not an InCommon member
- Workday doesn't consume federation metadata
- IdP metadata typed into WD config UI
- IdP-intiated flow initially, SP-initiated later
- UW needed to educate IBM implementation team on SAML integration
  - WD originally wanted to use an entityID of "WorkdaytoUW"

Multi-factor Authentication

- Goal
  - Enforce MFA for Workday users that have access to more privileged data and processes
- Requirements
  - MFA should be handled as part of the SAML authentication flow
  - Use existing Entrust hard tokens

Multi-factor Authentication

- Challenges
  - Workday doesn't support authnContextClass
  - Workday doesn't support security metadata that could be used to drive step-up to MFA
  - How to construct WD roles and security groups to identity users that require MFA?
  - How to make that information available to IdP?
  - How to enforce application business rules within web authentication system?

MFA Solution

- Created WD roles, identified which needed MFA, assigned NetIDs to roles
- Export MFA NetIDs as a file and sync to an “activator” group in UW Group service
- IdP makes pubcookie request and attaches a special AppID for WD
- Pubcookie server authenticates user normally then checks if they are in the activator group
- If a user is a member, pubcookie requests Entrust token value

Parting Thoughts

- Waterfall vs. agile project methodology
- Keep HR functional team involved in design phase
- Nature of design specs authored by WD made it difficult for functional team to approve
- Be prepared to teach and guide implementers
  - SAML, certificate config issues
- Limitations on WD API capabilities
- Better MFA may be coming from WD
- WD community web site has helpful information

Questions?

Contact:
Michael Brogan – mbrogan@uw.edu
Jonathan Pass – pass@uw.edu