UCTrust IAM Provisioning Workgroup - Requirements Outline

Objectives

1) Document the common need for the functionality (summary paragraph)
2) List the common specifications/requirements for a product that would meet the stated need
3) Compose a list of questions that might need to be asked of a vendor to determine viability (possibly used in an RFP)
4) Perform a preliminary review of the options in the market place (not intended to be an exhaustive or detailed product search/analysis)

Current working doc (addressing objectives 1 & 2 only)

Summary

Without accounts, people can't log in to anything so we need a way to provision them. Since there are many types of accounts used on each campus it is not efficient to have to go into each credential store and manually create accounts there (not to mention the fact that administrators need to have access in all of these places to do so). An account provisioning tool provides a repository for user identities, provisions and deprovisions accounts associated with those identities to external systems, and allows users to manage the passwords/credentials for those accounts centrally.

While a product (or set of products) would ideally be able to meet all of the following requirements, it's important that they have clear interfaces so processes that are developed outside of the system(s) can access and manipulate the data directly. Calls to custom external code, written in a language of choice with call backs to the provisioning system as necessary, may provide a better solution to meet some requirements.

Requirements

- Needs to have the capability to establish a new identity in the IdM database and look up or matching existing identities ("the merge"). Establishing identities would generally be based on incoming data (e.g., HR, student system), though some records may also be manually created in the system.
- Since the IdM databases at various campuses have different structures (eg schemas) and protocols (eg SQL, LDAP) and account provisioning system must have configurable and extensible plug-ins which enable these kinds of data access.
    - Should provide a framework for adding new connectors/adaptors
    - Should be able to use existing APIs to interact with provisioning targets - that is, customization should occur on the provisioning system side, and not require modifications to the targets.
- Since credential storage is varied (LDAP, AD, RACF, SQL, etc.) a provisioning system needs to support various plug-ins to manage accounts (e.g., passwords, certs, tickets) in all the possible credential stores.
- It needs to manage the mappings of identities to accounts (1:n), and should provide account reconciliation/discovery capability: what accounts "should be" and what "are" installed on a system.
    - Manage groups and roles directly within the provisioning system
- Sufficient user interface for support password management functions
- Audit and reporting of transactions/changes to accounts
- Provide general logic/rules/scripting to support data and account management:
    - Creation and transformation of data between source and target systems
    - Ability to add/remove individuals to groups, for groups and roles where (non-)membership directly implies (de)provisioning. The group object may live in an external group manager, but the provisioning system would ned to be able to provision people into/out of groups
    - Ability to create scheduled tasks (manually and by rule). E.g., Disable unused accounts, sunrising and sunsetting, force password resets.
    - User/supervisor/admin notification capability
- Data and schemas native to the provisioning system should be accessible, transparent and modifiable (for developers)
- Coding should be done in some standardized language (ideally language of choice)

Nice, but not required

- Mapping accounts to group identities or test identities for shared and functional accounts
- It would be nice to have the ability to manage groups and roles from the same UI in which accounts are created