KSAMS - Executive Overview

Problem

While many campus applications use WebAuth/Shibboleth as a centralized solution for authentication (verify identity - "who are you?") they don't use a centralized solution for authorization (verify allowed access - "are you allowed to do X?"). Instead they usually fall into one of two camps: 1) they use only WebAuth authentication, which more than likely isn't fine-grained enough for proper access control, or 2) they use locally defined role memberships or hard-coded user rights for access control which makes it impossible to: a) audit with complete visibility all of the access a user has on campus, b) perform security forensics / incident response of which high risk resources a user had access to at a certain time if their credentials were breached, c) quickly and comprehensively revoke all access a user has on campus when they leave (relying on UCnetID activation is not secure enough), and d) provide users with a consistent interface and process for requesting and approving access to applications on campus.

Solution

KSAMS (new Security Access Management System) is a web-based centralized authorization (role-based access control) system that provides a central point for defining roles and managing user membership for those roles. When a user tries to access a campus application it will query KSAMS (either through a SOAP Web Service or LDAP API) to ask if the user is a current member of the required role(s). Roles defined centrally are mapped to permissions defined locally within the application, for example Role X is allowed to view a particular page, data field, or submit/approve a specific type of transaction. User access is not directly defined by one person, access is instead requested and approved by the appropriate people via workflow with a complete audit trail. This offers a consistent interface and process for users requesting and approving access to resources, as well as a single location to give security visibility to audit what people have access to and comprehensively terminate all access when needed.

How?

- Talk to KSAMS project team about how to best define the centralized roles, who should approve access to those roles, and how they will map to local application permissions
  - KSAMS New Role Request Procedure
- Modify campus application to query role membership from KSAMS to determine appropriate access controls
  - KSAMS Developer Guides - Web Services SOAP API (role membership with qualifiers) and LDAP API (role membership) currently available
- Communicate to users new way to request changes to access
  - Refer them to KSAMS User Guide but let them know which roles are important for that specific application