# Authentication, Authorization, & Identity Management

Mark Earnest
Lead Systems Programmer
Emerging Technologies - ITS

# Authentication & Authorization

## History

- Biometrics (Hi, I recognize you)
- Secret Handshakes
- Passwords
- Language and accents

# Authentication

- How do I know *you* are who *you* say *you* are?
- Comprised of multiple factors
  - Something *you* are (userID, biometrics)
  - Something *you* know (password)
  - Something *you* have (SecurID Token, digital certificate)
- Current thinking is that the question of authentication should be kept separate from authorization

# AuthN - Location (IP) Based

- Not even worth a slide to talk about
- IP does NOT equal identity
  - NAT, Wireless, proxy, spyware infected machines, etc
- Only mentioned because some still do this

# AuthN - UserID/Password

- UserID - Who I am
- Password - What I know
- Bad passwords are subject to brute force (dictionary attacks)
  - Bad passwords are dictionary words, short passwords, or easily guess-able with some knowledge of the user
  - Good passwords are long, comprised of numbers, letters, and symbols, and easy for the user to remember.

# AuthN - UserID/Password

- Pros
  - Common, understood, & accepted by users
  - Supported by virtually all applications
  - Requires no additional hardware (easy to roll out)
- Cons
  - Users generally do not pick good passwords
  - Sysadmins often make password policy too lenient or too strict
  - Users may not protect password - or worse, share it

# AuthN - Kerberos

- Developed by MIT, gold standard for enterprise authentication
  - works over insecure networks (password never transmitted)
- Single sign on
  - User gets a TGT which is then used to request service tickets
- Complicated, but provides a lot of security for authentication

# AuthN - Kerberos

- Pros
  - Single sign on
  - Secure
  - Users and applications can be authenticated
- Cons
  - Complicated to set up and administer
  - KDC is the basket with all the eggs, must be guarded carefully
  - Not historically well supported by end user apps

# AuthN - PKI

- Public key cryptography vs symmetric
- Certificate hierarchy
  - Offline validation
- Data Security (encryption)
- Data integrity (digital signature)

# AuthN - PKI

- Pros
  - Versatile - SSL, S/MIME, PDF signing, etc.
  - Recognized as strong, industry-wide and legally
- Cons
  - Only as secure as the protection of the private key
  - cumbersome, unintuitive, difficult to do right
  - Certificate Authority oligopoly
  - "PK" is easy, the "I" is difficult

# AuthN - Biometrics

- Oldest authentication known to man
- Fingerprint, retina & iris scan, voice, hand print, typing pattern, facial recognition
- While humans are suited to authenticate biometrics, computers are not.

# AuthN - Biometrics

- Pros
  - Cannot be easily shared or guessed
- Cons
  - Expensive (requires specialized hardware)
  - Impossible to erase or change "key"
  - Not 100% perfect (false positives & false negatives)
  - Unintended consequences - is what you are protecting worth more than your finger?

# AuthN - Passfaces

- Passfaces is a commercial product that takes an innovative approach to authentication
- The user's password is a series of faces, which they choose from a "line up"
- Easy to remember and hard to share
- Not used at PSU, just an example of something novel in the world of authentication

# Authorization

- Ok, you are who you say you are, what are you allowed to do?
- Can be centrally managed or managed at the edge
- Not as well understood as authN
- An area where we (PSU) are very much on the cutting edge.

# AuthZ - UserID Lists

- Most common AuthZ method
- Not easily scalable
- Easy to implement, difficult to manage
- Not centralized, must be maintained at each application.

# AuthZ - DCE & MS Kerberos

- Group based authorization
- Centrally managed groups
- Group membership sent with authN token
- History, this is how PSU handled authZ
- AuthN & AuthZ combined into a single token

# AuthZ - LDAP

- Group based and centrally managed
- Group membership is queried rather than send along with authN token.
- PSU's future direction
- DCE concept of user managed groups and groups within groups is being developed

# AuthZ - SAML

- Security Assertion Markup Language
- Open standard created by OASIS, used by many corporate products
- XML schema for asserting authN & authZ
- Builds on PKI by signing and/or encrypting assertions.
- Separates AuthN & AuthZ

# Tying it all Together at PSU

- Kerberos Authentication
- LDAP based groups for authorization purposes
- Need for both internal & external identity management
  - With external IdM, privacy becomes a big issue

# WebAccess (CoSign)

- Web single sign on project (web only)
- Open source developed at University of Michigan
- Builds on Kerberos and PKI
- Architecture is ideal for internal use, but only internal use
- Passes Kerberos credentials when needed

# WebAccess (CoSign)

- Two components, login server and web filter
  - Web filters available for Apache, IIS, & Java Servlets
- Session cookies are used, not shared across domain
- Sensitive information is passed over an encrypted connection directly between web server and login server

# WebAccess @ PSU

- Portal & Webmail
  - Both require Kerberos credentials (actually DCE) to access DFS
- downloads.its.psu.edu
- www.work.psu.edu
- Shibboleth

# Shibboleth

- Inter-institutional IdM
  - works with any internal AuthN & AuthZ setup
- Builds off of SAML & PKI
- Designed with privacy in mind, robust control of which attributes can be released where
- Becoming popular with federal government, library vendors, and course management apps.

# Shibboleth

- Comprised of Identity Providers & Service Providers
- All communication happens via signed SAML assertions
- PSU was one of the first Shibboleth trials

# Shibboleth @ PSU - Webassign

- Physics course management system
- First production shibboleth use
- Replaced ssn-based password
- Dynamic account creation
- 80% decrease in helpdesk calls

# Shibboleth & PSU - Napster

- Napster needs a system for account registration (one account per student)
- PSU needs to protect student's privacy
- Shibboleth TargetedID solved both needs
- PSU prototyped the now standard Napster approach to account creation using Shib.

# Final Thoughts

- We have fairly complicated security requirements
  - This is bad from a security perspective, is not likely to change
- When possible, we prefer to build upon the open standards and solutions we have.
- Identity management is still very immature, universities, government, and business are all currently grappling with it.

# Obligatory Final Slide

- Questions, comments, heckling?
- Thank you
- email: mxe20@psu.edu