Stronger / Multi-factor Authentication
For Enterprise Applications
(Identity Assurance using PKI, Smart cards and Biometrics)

Presented to OWASP Seminar, Hartford (Feb 10, 2009)

Ramesh Nagappan
Sun Microsystems, Burlington, MA
http://www.coresecuritypatterns.com/blogs
Agenda

- The Identity Dilemma
- Identity Assurance vs. Stronger Security
- Multi-factor Authentication Strategies
  - OTPs, Smartcards, PKI and Biometrics
  - Choosing the credential: Pros and Cons
- Understanding Real-world Implementation
  - Tools of the Trade
  - Role of JAAS
- Role of Sun OpenSSO Enterprise
- Architecture and Deployment
- Demonstration
  - Multi-factor SSO with PKI, Smart cards and Biometrics.
- Q & A
Who am I?

• A technical guy from Sun Microsystems, Burlington, MA.
  > Focused on Security and Identity Management technologies

• Co-Author of 5 technology books and numerous articles on Java EE, XML Web Services and Security.

• Holds CISSP and CISA.

• Contributes to Java, XML security, Biometrics, Smart cards standards and open-source initiatives.

• Contributes to Graduate curriculum of Information Security programs at multiple universities.

• Ph.D drop-out.

• Read my blogs at http://www.coresecuritypatterns.com/blogs

• Write to me at nramesh@post.harvard.edu
The Identity Dilemma: Who Are You?

- Internet is a faceless channel of interaction.
  - No mechanisms for physically verify a person – who is accessing your resources.

- Identifying the legitimate user has become crucial.
  - With higher strength of authentication and security.
  - Mandates mechanisms driven by human recognition characteristics.
  - Growing trends on on-demand SaaS, Cloud computing infrastructures.
  - Everyone is concerned about their private information and privacy.
How do I know..it's you?

• **Identity thefts** and on-line frauds: The fastest growing crime in the world.
  > Someone wrongfully obtains or abuses another person's identity – for economic or personal gain.
  > Impersonation, Counterfeits, stolen or forged credentials (PINs, Passwords, ID cards), Phishing are widely becoming common.
  > Most frauds happens through trusted insiders.
  > Fake credentials are everywhere: Few detected and many undetected!

• **Identity thefts results huge losses to organizations.**
  > Loss of consumer confidence and leading to incur huge government penalties.
  > Growing needs for stringent “Personal Identity Verification and Assurance” (*i.e HSPD-12, ICAO 9303)*.
  > Growing mandates for protecting Identity information and compliance (*i.e. Massachusetts 201 CMR 17.00*)
Growing need for Identity Assurance

- High degree of authentication and assurance is the most critical requirement for physical and logical access control.
- Acquire *Identity credentials* that tightly binds an event to a person's proof of possessions, physiological characteristics and behavioural traits.
  - Identification and authentication as equivalent to Face-to-Face verification of a person.
  - Credentials must provide at-least some long-term stability.
  - Credentials should be non-intrusive but still qualitatively and quantitatively measured.
  - Integrate/Interoperable with physical and logical infrastructures for assured identity verification.
  - Support for pervasive use (*On-demand SaaS and Cloud-computing based application infrastructures*) for authenticating a person with irrefutable proof.
  - Lesser impact on privacy and social values.
Human Factors of Identity Assurance

Human attributes as Identity Assurance Credentials

- **Proof-of-Knowledge**
  - Something I know?
  - Passwords, PIN, Mom's Maiden Name, Phone #, etc.

- **Proof-of-Possession**
  - Something I have?
  - Smartcards, Tokens, Driver's license, PKI certificates

- **Proof-of-Characteristics**
  - Something I physiologically or behaviorally own?
  - Fingerprints, Hand geometry, Facial image, Iris, Retina, DNA, voice, signature patterns

- **Proof-of-Physical Presence**

![Venn Diagram](image)
Strong Authentication Strategies

• Authentication Questions
• HTTP/s Request/Response attributes
• Hardware/Software Token based One-time Passwords (OTP)
• Hardware/Software Token based Challenge/Response OTP
• Phone call based OTP
• SMS based OTP
• PKI Certificate
• USB Tokens/Smart cards (PIN and PKI Certificates)
• Biometrics (Fingerprints)
• USB Token/Smart cards (PKI and Match-on-card Biometrics).
One-Time Passwords

Hardware/Software Tokens

• Generate one-time passwords
  > Mathematical problem or Crypto function or Random number generation
  > Challenge/Response Dynamic password, Asynchronous Password
  > Time synchronization between client & server.

• Deliver Passwords
  > Proprietary devices, USB, Key fobs
  > SMS Messages, Email, Phone

• Known issues
  > Vulnerable to MITM, Phishing attacks where Time-synchronization not effective.
  > DES key and Lost token issues

• Standards: OAUTH (Open Authentication Initiative)

One-Time Passwords : Alternative to static passwords
Smart cards w. PKI

**Smart cards**
- A credit card sized computing device acts as a Cryptographic token.
  - Contact / Contactless cards
- Allows performing security functions
  - Key generation
  - Public/Private key operations
  - PIN/Biometric authentication
  - Challenge/response authentication
- Supports the use of Public-key infrastructure to verify the Identity claim.
  - PKI credential issuance.
  - Credential validation/verification via OCSP, CRLs
- Defends against tampering and hacking.
  - PKI/Private key protection
- Issues: Lost cards, Key compromise recovery is difficult.

**Standards**
- ISO-7816
- Java Card, Multos
- Global Platform
- PC/SC
- FIPS-201/PIV, CAC
- PKCS#11, PKCS#15
- GSM/PCS
- EMV (Europay/Mastercard/Visa)

Smart cards as a Cryptographic Token
Biometric Assurance

Biometric Identity

- Use of Physiological or Behavioral characteristics to identify a person.
  > High degree of assurance with proof of presence.
  > Fingerprints, Facial image/geometry, Iris, Retina, Voice, Hand geometry, Keystroke, Signature
- Biometric templates can be stored on Smart card for personal identification.
  > Fingerprint template is ~200 bytes
  > Iris template is 500 bytes
- Biometric credential must be exchanged in a secure network channel (Trusted path)
- Issues:
  > Biometrics is not a secret
  > False Acceptance (FAR) & False Reject (FRR) rates
  > Vulnerable to Message replay/MITM attacks, if not exchanged in secure channel.

Standards

- INCITS 378 / CBEFF (Fingerprints)
- INCITS 379 (Iris)
- OASIS BIAS
- BioAPI
- JavaCard BioAPI
- FIPS-201 / PIV

Biometric Assurance: Who I claim to be
Real-world Scenario: Authentication

Identity Assurance Credentials
- User name, password
- One-time Passwords
- Smart Card (PKI Certificates)
- Biometrics

Ideally, credentials are:
- something you know +
- something you have +
- something you are

Validated

Authentication Server / Directory

Enterprise Applications
Real World Scenario: SSO / Federation

IDP/SSO Server + Directory

Multi-factor Identity Credentials (e.g. username/password, Smartcard PKI, Biometrics)

Acquire Credentials

1. 2. 3.

SSO Token
- SAML
- Artifact
- Proprietary token
- Kerberos ticket
- etc

Application A

Application B
Tools of the Trade: What do you need?
Multi-factor Authentication for Enterprise Applications

- Web Authentication
  - JAAS Login Module
- Desktop Authentication
  - PAM Module (Solaris, Linux)
  - GINA (Windows XP, 2003)
- Identity Provider Infrastructure (IDP)
  - Single Sign-on (SSO)
  - Multi-factor authentication
- Directory Server
  - Repository for user accounts
- Your target applications
Tools of the Trade: From Authentication Providers
Multi-factor Authentication for Enterprise Applications

• Browser Plugin
  > PKCS#11 Client for Smartcard
  > ActiveX/Java Plugin for USB Biometric Scanners

• Enrollment Middleware
  > Biometric Enrollment, Smartcard/Token Credential Issuance/Management
  > One-time Password (Token) registration/issuance

• Authentication Middleware
  > Biometric Authentication, One-time password authentication
  > PKI Credential validation via OCSP, CRL, Directory, Certificate Authority
Java Authentication Authorization Service (JAAS)

- JAAS plays a vital role delivering Multi-factor authentication.
  - All Java EE compliant Application server provide support for JAAS.
- JAAS allows to enable Multi-factor authentication in Java EE Enterprise environment.
  - Facilitates pluggable authentication providers as “Login Modules”.
  - Ensure Java EE remain independent of authentication providers.
- Implementing a Login module is not cumbersome.
  - Callback handler – Prompt the user for acquiring credentials
  - Login (), commit (), Logout ()
- Choose your own JAAS based Identity Provider Infrastructure?
  - Get introduced to Sun OpenSSO
Sun OpenSSO Enterprise

- **Identity Services Infrastructure** facilitates Single Sign-On (SSO) for Web applications residing within an enterprise or across networks.
- Based on **Sun's Open-source initiative**.
- **Open standards based framework** supports centralized authentication, authorization and auditing.
  - JAAS based authentication services
  - Agent-based and XACML based policy enforcement
  - Identity-enabled XML Web services for AuthN, AuthX, Audit and Provisioning
  - Identity Federation Protocols support include SAMLv2, ID-*, WS-Federation, WS-Policy
  - Multi-factor authentication via chaining
  - Centralized configuration, logging and auditing services
  - Supports multiple Java EE application servers and Web containers
  - Fedlets
- **Deployed as a Web application (single WAR file)**
Multi-factor Authentication and Session Upgrade

OpenSSO Authentication Chain and Session upgrade thru’ AuthN

• OpenSSO facilitates **stronger/multi-factor authentication** through authentication chain including multiple authentication providers.
  > Enables an authentication process where an user must pass credentials to one or more authentication modules before session validation.
  > Session validation is determined based on the JAAS control flag (Required, Requisite, Sufficient, Optional) configured to the authentication module instance chain.
  > The overall authentication success or failure is determined based on the control flag assigned to each module in the authentication stack.
  > OpenSSO is tested and verified to provide multi-factor authentication chain that include BiObex Login, Smartcard/PKI and other OpenSSO supported authentication providers.

• **Session Upgrade** allows upgrading a valid session based on a successful “second-factor authentication” performed by the same user.
  > Allows user authenticate to access second resource under the same or different realm
  > If authentication is successful - OpenSSO updates the session based on the second-level authentication. If authentication fails, the current session will be maintained.
**OpenSSO Policy Agents**

**Authorization and Policy Enforcement**

- **Policies** are managed by Policy Configuration Service in OpenSSO.
  - Policy service authorizes a use based on the policies stored in OpenSSO.
  - Policy consists of Rules, Subjects, Conditions and Response providers.

- **OpenSSO Policy Agents** enforce policy and Policy decisions on protected resources.
  - Intercepts the requests from user clients and applications and redirects them to OpenSSO server for authentication – If no SSO token exists.
  - Once authenticated, the policy agent communicates with OpenSSO Policy service to grant/deny access to the user based on policy evaluation.
Multi-factor Authentication w. Biometrics
Multi-factor Authentication
Smartcard/PIN/PKI and Biometrics

Input your PIN, press login and then input the appropriate fingerprint.

PIN
Fingerprint

Finger stored on your PIV card.

Log In
Deployment Architecture

- Users with Smartcard/PKI credentials or Biometrics
- Web Server
- Agent
- OpenSSO Enterprise Sun Directory Server
- Biometric Provider (BioBex)
- FBCA PKI SSP / OCSP Responder / CRL DP
- Federated Service providers using SAML 2.0 with X.509 Tokens
- Web/Application Servers
Participate in OpenSSO Community!

- Join 700 project members at opensso.org

Join:
- Sign up at opensso.org

Download:
- OpenSSO Enterprise 8

Subscribe:
- OpenSSO Mailing Lists
dev, users, announce

Chat:
- #opensso on freenode.net
Demonstration...

Thank You

Ramesh Nagappan
Sun Microsystems, Burlington, MA
http://www.coresecuritypatterns.com/blogs