Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

eToken unter Mac OS X

Seminar Betriebsadministration

Stefan Wehrmeyer

Agenda

- Motivation und Security Standards
- Keychain und Sicherheit in Leopard
- Smartcard Integration und Open Directory
- eToken-Login: Methoden

Motivation

- 2-Faktor-Sicherheit für sensible Daten
- OS X mit normierten Sicherheitsstandards nutzen
- Common Access Card des DoD (US-Verteidigungsministerium)
  - US-Militär will auch Mails vom Mac aus abrufen können

Security Standards

- Mac OS X Leopard unterstützt Smartcards standardmäßig.

> “Now you can use a smart card to unlock FileVault volumes and your keychain, and configure your Mac to lock the screen when a smart card is removed. Leopard supports the PIV standard for Federal employees and contractors.”
>
> 300+ Feature Ankündigung zu Leopard

- FIPS 201 (Federal Information Processing Standards)
  - Kryptographie-Standard für US-Bundesbehörden
  - Unterstützt in Firefox und Thunderbird

OS X Keychain

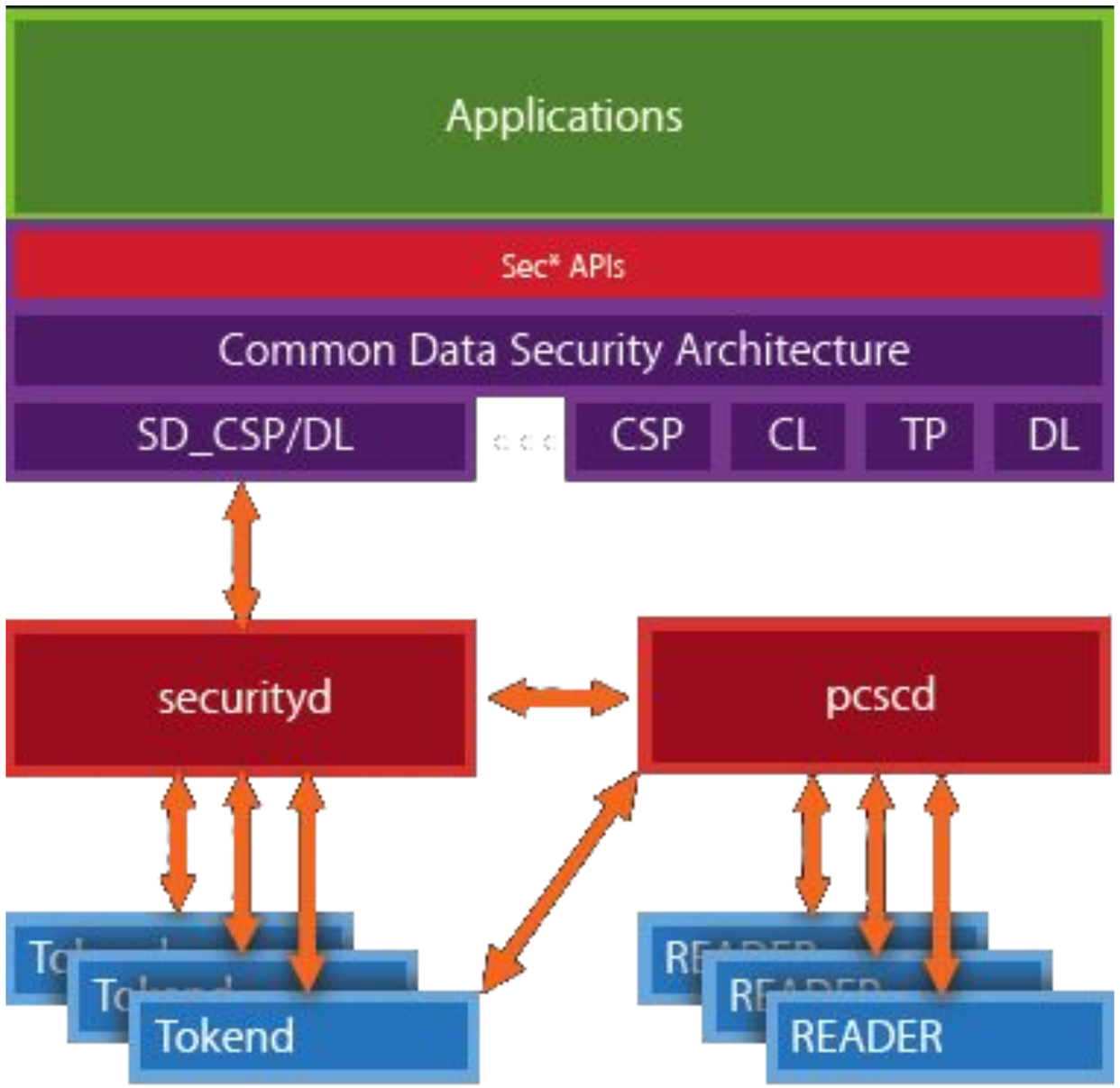
- Zugriff über Keychain Access
- Speichert verschlüsselt:
  - Passwörter (Anwendungen, Internet, WLAN...)
  - Schlüssel, Zertifikate, ...
- Zugriff durch Anwendungen granular und bequem einstellbar

- Smart Card Plugin bindet eToken als Keychain ein
  - Smart Card Keychains werden immer zuerst abgefragt

Sicherheit in Leopard (1)

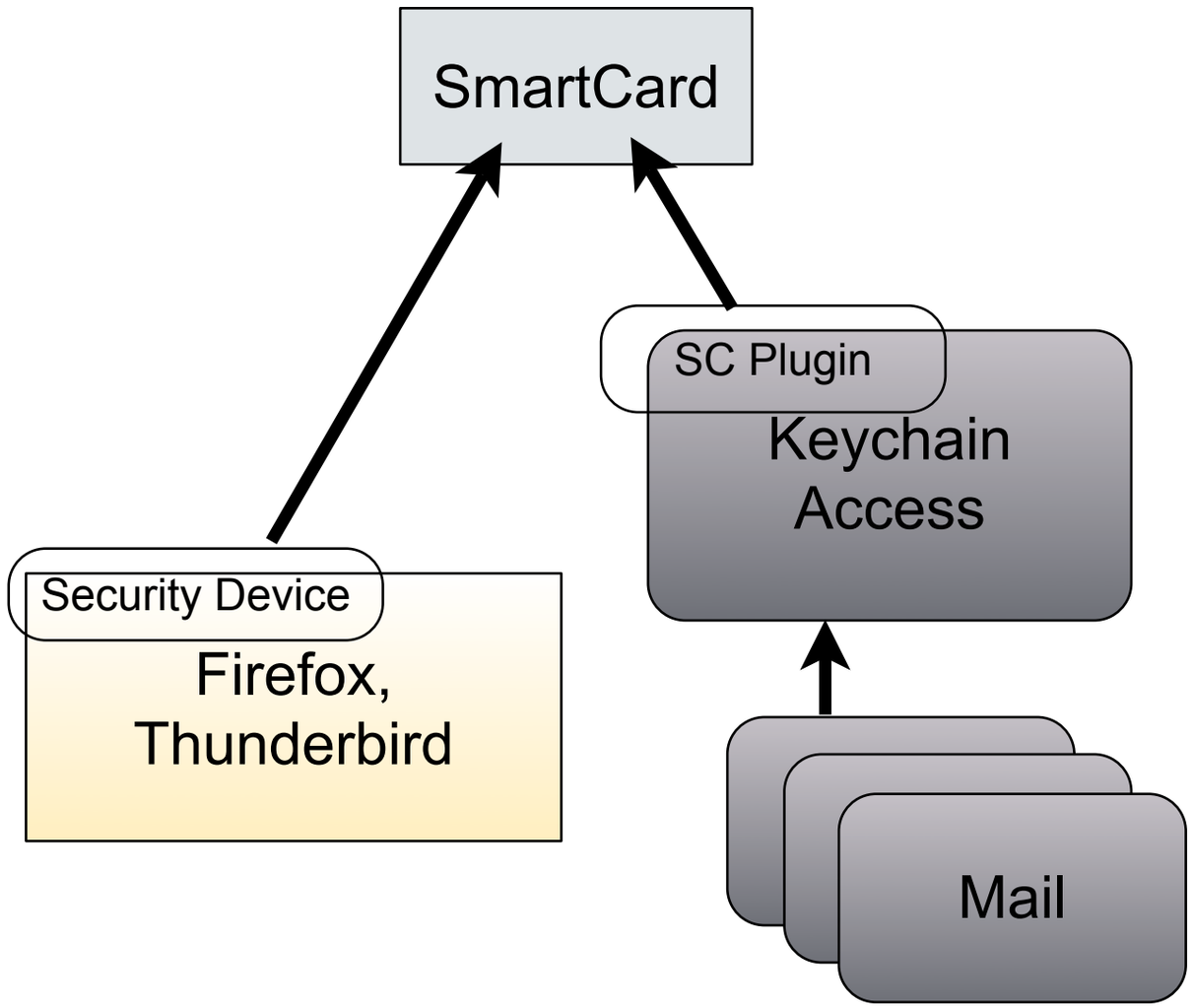
- Common Data Security Architecture (CDSA)
  - Framework-Standard für Kryptographie, Zertifikatverwaltung,
  - Schicht zwischen OS X APIs und Kern
- Apples benutzt MUSCLE-Driver-Framework für Smartcard-Zugriff

Sicherheit in Leopard (2)



aus “Integrating Smart Card Solutions” von Shawn Geddis

Smartcard Integration in Leopard



OS X Open Directory Architecture

- Über Open Directory werden Verzeichnisdienste verwaltet
  - NetInfo und `lookupd` existieren in 10.5 nicht mehr: `ds*`-Tools
- Abstrahiert den Zugriff auf beliebige Verzeichnisserver
  - Unterstützt Active Directory, LDAPv3, NIS
- Steuerbar mittels `dscl DOMAIN COMMAND [PATH [KEY [VALUE]]]`
  - `dscl . read /Users/steve`

```
[...]
AuthenticationAuthority: ;ShadowHash; ;Kerberosv5;;steve@LKDC:SHA1.368C
5CCD6DEF5593594AECA0EAEA561BBAEB5FAB;LKDC:SHA1.368C5CCC6DEF5593694AECA0
EAEA561BBAEB5FAB;
AuthenticationHint:
GeneratedUID: EFF6F82F-DE17-4FE1-9FF1-BCCAA40E04FC
NFSHomeDirectory: /Users/steve
Password: ********
Picture:
 /Library/User Pictures/Flowers/Sunflower.tif
PrimaryGroupID: 20
RealName:
 Stefan Wehrmeyer
RecordName: steve
RecordType: dsRecTypeStandard:Users
UniqueID: 502
UserShell: /bin/bash
```

Smartcard/eToken Login

- Aktiv in 10.5
  - 10.4: Änderungen nötig an /etc/authorization
- eToken zu Nutzer zuordnen: Directory Service
  - mittels Public Key Hash
  - oder mittels Zertifikat-Attributen (/etc/caclogin config.plist)

```
<dict>
    <key>fields</key>
    <array>
        <string>Common Name</string>
    </array>
    <key>formatString</key>
        <string>$1</string>
    <key>dsAttributeString</key>
        <string>dsAttrTypeNative:_writers_realname</string>
</dict>
```

Mittels Public Key Hash

- Bequemste Methode
  - Aber nicht nachhaltig (Zertifikat läuft ab)
- Liste aller verfügbaren Public Key Hashes: `sc_auth hash`
  - `F986DC8E29565F89925EF0ED49CA9A7D7CC2078A Stefan Wehrmeyer's Hasso-Plattner-Institut ID`
- Eintragen in Directory Service
  - `sudo sc_auth accept -u Steve -h F986DC8E29565F89925EF0ED49CA9A7D7CC2078A`

Demo

Fazit

- Smartcard-Unterstützung ist gut
  - Noch nicht Mac-like
- Keychain verbindet Smartcard und OS X
  - Optimale Lösung

Quellen

- “Mac OS X 10.4: Enabling Smart Card Login” http://docs.info.apple.com/article.html?artnum=304035
- “Integrating Smart Card Solutions” WWDC07-Folien von Shawn Geddis
- “About PIV of federal employees and contractors” http://csrc.nist.gov/groups/SNS/piv/index.html
- Apple Developer Connection Documentation http://developer.apple.com/documentation/