I. Pluggable Authentication Modules (PAM)

1. Що е то - обща идея

   - Трябва ли всички пароли да се държат в system password file?
   - Един файл, описващ модулите, през които всеки service минава за автентикация, а после и за други неща.

2. Начин на работа (сървър - библиотеки – conversation)

   - Клиентът се представя с username;
   - Сървърът се представя на PAM libs с името на услугата си;
   - Сървърът започва PAM сесия с username на клиента;
   - PAM libs минават през модулите
   - Сървърът дава credentials, като при нужда използва conversation функциите,за да поиска още информация от клиента;
   - PAM libs решават дали да спрат по средата (sufficient или failure);
   - PAM libs попълват struct passwd за използване от сървъра.

3. Реализации

   - Sun/Solaris;
   - Linux-PAM;
   - OpenPAM.

4. Предимства и недостатъци от гледна точка на usability

   - con: още едно нещо, което сървърите трябва да поддържат;
   - con: още не е съвсем стандартизирано като API;
   - con: не много добра поддръжка на template user accounts;
   - pro: само едно нещо, което сървърите трябва да поддържат;
   - pro: централизирана конфигурация с една стъпка;
   - pro: конфигурация на auth на много машини с централна user account db;
   - pro: лесно добавяне на нови auth methods без прекомпилиране или дори прелинкване на приложенията - компилираш новия PAM модул, слагаш го в конфигурационния файл et voila;
   - pro: всякакви странни възможности (pam_checkmail, pam_alreadyloggedin);
   - pro: връзване на authentication с account mgmt.

5. Предимства и недостатъци от гледна точка на security

   - con: паролите все пак се предават в clear text;
   - pro: всичко, което улеснява работата на sysadmin-а, е добро;
   - pro: pam_cracklib и подобни;
   - pro: централизиран logging.