Pluggable Authentication Modules

Michał Grzejszczak

20 stycznia 2003

Spis treści

1 Wprowadzenie

PAM (Pluggable Authentication Modules) został utworzony przez firmę SUN na potrzeby systemu Solaris. Później został zaadoptowany przez Open Software Foundation i jest obecnie wykorzystywany w wielu open-source'owych systemach. PAM został utworzony aby pokonać trudności związane z wprowadzaniem nowych mechanizmów autentykacji. Każdy taki mechanizm powodował konieczność przepisywania na nowo kodu odpowiedzialnego za autentykację (login, passwd, telnet itp).

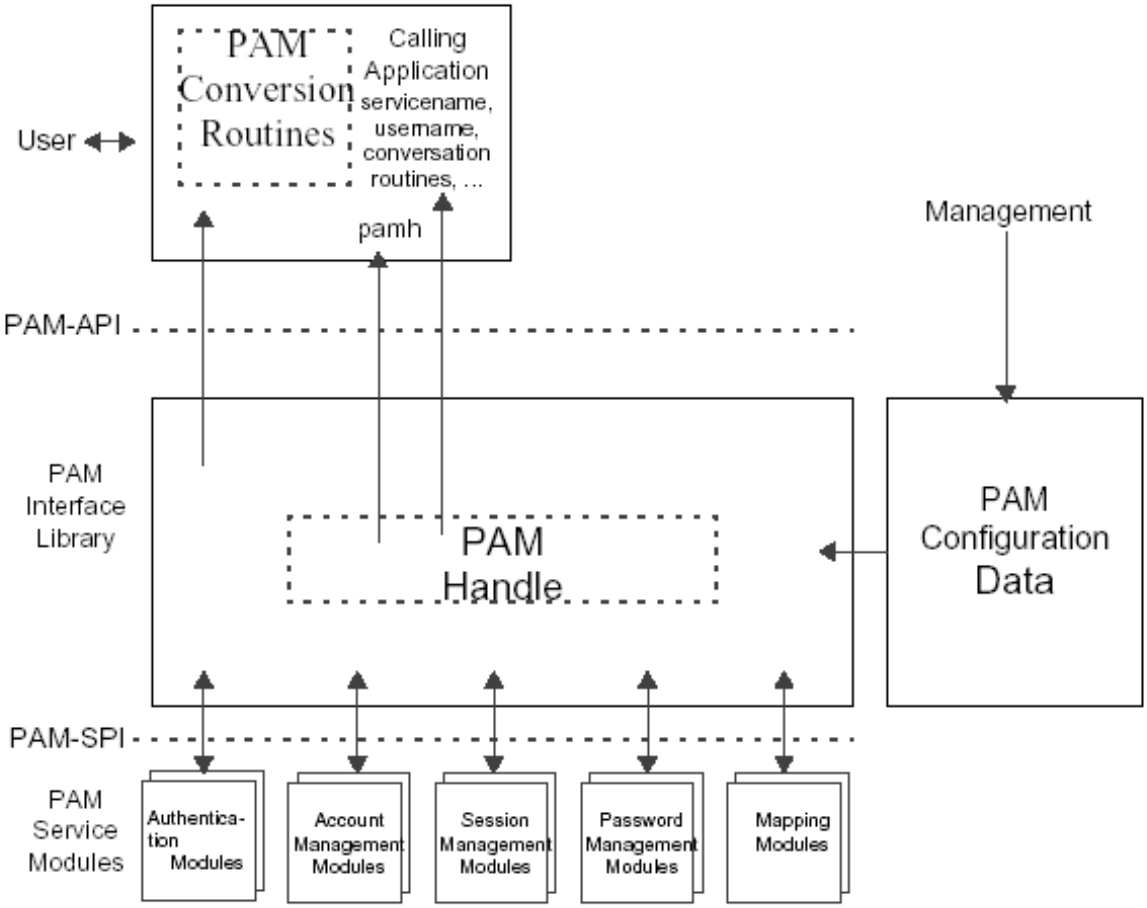
Mechanizmy dostępne za pomocą PAM są implementowane jako moduły. Są przezroczyste dla aplikacji. Co więcej pozwalają na konfigurację uwierzytelniania osobno dla każdej aplikacji, tzn. np. lepsze zabezpieczenia dotyczą dostępu zdalnego niż lokalnego. PAM udostępnia korzystanie z mechanizmów w stosie, czyli aplikacja może korzystać z uwierzytelniania przez kilka mechanizmów.

2 Budowa PAM

PAM składa się z następujących komponentów (na przykładzie Solaris 9):

- „Framework" - umożliwia modularne zarządzanie mechanizmami autentykacji. Składa się z czterech komponentów:
  - PAM API
  - PAM Framework - implementacja API
  - PAM SPI (Service Provider Interface) - implementuje funkcjonalność „back-end" dla API
  - /etc/pam.conf - plik z konfiguracją (które SPI obsługują które aplikacje)
- typy modułów PAM - obecnie są zdefiniowane cztery typy modułów:
  - Autentykacja - obsługuje autentykację użytkownika i udostępnia metody obsługi listów uwierzytelniających.
  - Zarządzanie kontem - kontroluje wiek haseł, i inne ograniczenia w dostępie do konta (decyduje o przyznaniu dostępu).
  - Obsługa sesji - obsługuje otwieranie i zamykanie sesji a także może rejestrować aktywność użytkownika.
  - Obsługa haseł - zmiana haseł.
- aktualizacja pliku z konfiguracją PAM
- aktualizacja modułów
- rozszerzenia zarządzania hasłami

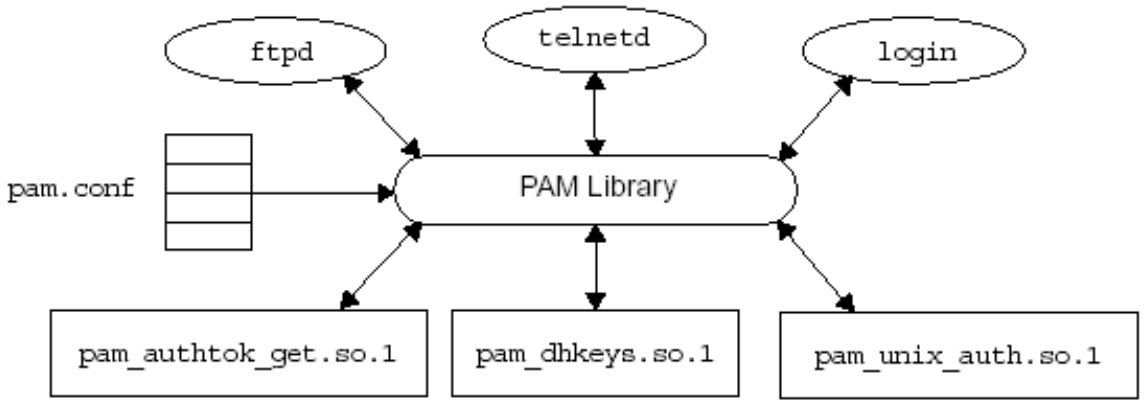
Rysunek 1: Struktura PAM



3 Jak działa PAM?

(Solaris 9)
Aplikacja korzystające z uwierzytelniania używają biblioteki /usr/lib/libpam.so, która jest łącznikiem pomiędzy aplikacją a modułami. Każdy moduł musi posiadać przynajmniej jeden typ (może więcej).

Rysunek 2: Działanie PAM



Moduły z obrazka (Solaris 9):

- `pam_authtok_get`

  : typ: autentykacja i obsługa haseł
  Odpowiada za pobieranie haseł (tokenów) od użytkownika.

- `pam_dhkeys`

  : typ: autentykacja i obsługa haseł
  Odpowiada za tworzenie i zmianę kluczy typu Diffie-Hellman używanych przy Secure RPC (NIS+ i Secure NFS)

- `pam_auth_unix`

  : typ: autentykacja
  Sprawdza zgodność haseł od użytkownika z hasłami w repozytorium przy użyciu szyfrowania crypt(3c)

4 Literatura

Wyczerpujących informacji na temat PAM i jego konfiguracji, a także użycia i pisania własnych modułów można znaleźć na stronie SUNa: http://www.sun.com/software/solaris/pam/.
Informacje o implementacji linuksowej można znaleźć na http://www.kernel.org/pub/linux/libs/pam.