PAM – Pluggable Authentication Module

- Kurzakte -

<table>
<tr><th colspan="2">Installation</th></tr>
<tr><td colspan="2">• Installation der Pakete pam, pam-modules, yast2-pam</td></tr>
<tr><th colspan="2">Besondere Merkmale</th></tr>
<tr><td colspan="2">• Trennung von Applikation (z.B. FTP) und Benutzerauthentifizierung (z.B. LDAP)
• neue Authentifizierungsmechanismen sind ohne Veränderung der Anwendung integrierbar
• es sind mehrere Authentifizierungs-Verfahren alternativ oder hintereinander kombinierbar</td></tr>
<tr><th colspan="2">Konfiguration</th></tr>
<tr><td>/etc/pam.d/
(neue Syntax)</td><td>• Hauptverzeichnis der Konfigurationsdateien zur Authentifizierung
• für jedes Anwendungsprogramm, das eine Authentifizierung erfordert (Dienst), existiert eine gleichnamige Datei
z. B.: chage, chfn, chsh, cups, ftpd, login, other, passwd, ppp, rpasswd, samba, shadow, sshd, su, sudo, telnetd, useradd, xdm, dm-np, lock, xscreensaver
• definiert wird die Verbindung der eine Authentifizierung erfordernden Anwendungsprogramme (Dienste) mit den PAM-Modulen
• Achtung: Tippfehler führen zum kompletten Systemausschluss</td></tr>
<tr><td>/etc/pam.d/dienstname
(alte Syntax)</td><td>• Konfigurationsanweisungen speziell für diesen Dienst
<u>Syntax:</u> type control module-path module-arguments
die Module werden der Reihe nach abgearbeitet (Stack)
• die Dateinamen müssen klein geschrieben und mit dem Dienst identisch sein
• es kann eine Folge von PAM-Modulen verwendet werden = Stack</td></tr>
<tr><td>/etc/pam.d/other</td><td>• enthält Konfigurationsanweisungen für alle Dienste ohne eigene Konfigurationsdatei
• wird also abgearbeitet, wenn PAM keien eigene Datei mit gleichlautendem Dienstenamen unter /etc/pam.d/ findet
• als fallback-Variante muss sie also besonders sicher sein
z. B. alles verbieten, was nicht in einer gesonderten Datei unter pam.d steht:
auth required pam_deny.so
account required pam_deny.so
password required pam_deny.so
session required pam_deny.so</td></tr>
<tr><td>/etc/pam.conf</td><td>• kompakte Konfigurationsdatei alternativ zu /etc/pam.d/
<u>Syntax:</u> service type control module-path module-arguments
• zusätzlich gibt es also die Spalte service für beispielsweise login, su, passwd
• Datei wird bei bestehendem Verzeichnis /etc/pam.d/ ignoriert</td></tr>
<tr><td>/etc/security/</td><td>• Ablage von Konfigurationsdateien für komplexere Module</td></tr>
<tr><td>/lib/security/</td><td>• der Standard-Modulpfad
• wird bei relativer Pfadangabe zum Modul automatisch eingefügt
• hier liegen unter anderem die Module:
passwd/shadow pam_unix2.so
Terminalzugang pam_securetty.so
/etc/nologin-Test pam_nologin.so
Check auf neue Mail pam_mail.so
LDAP-Modul pam_ldap.so
Kerberos-Modul pam_krb5.so
SMB-Modul pam_winbind.so pam_smbpass.so
Radius-Modul pam_radius.so</td></tr>
<tr><th colspan="2">Kontrolle</th></tr>
<tr><td>/var/log/messages</td><td>• hier landen die Log-Meldungen
• in /etc/security/*.conf das Schlüsselwort debug einfügen, um komplexere Meldungen zu erhalten</td></tr>
<tr><th colspan="2">Dokumentation</th></tr>
<tr><td colspan="2">man: pam(8) pam.conf(8) pam.d(8) pam_localuser(8) pam_xauth(8)
/usr/share/doc/pam*</td></tr>
</table>

Aufbau der Konfigurationsdateien unterhalb von /etc/pam.d
vier Spalten: **PAM-Dienst** **Wichtigkeit** **Modul-Pfad** **Parameter** **PAM-Dienst(type):** beschreibt die Aufgabe des Moduls im Anmeldevorgang (Management-Aufgabe) • auth: checkt Benutzerauthentität (Passwort, Chipkarte) und kann zusätzliche Gruppenrechte vergeben • account: Zugangsbeschränkung nach Uhrzeit, Ressourcen, Anmeldeort • session: zusätzliche Jobs vor/nach Anmeldevorgang wie mounten, Logging, Variablen setzen • password: Update des Authentifizierungsmechanismus (Passwort ändern) **Wichtigkeit (control):** Bedeutung des Moduls für den Fortgang des Anmeldevorgangs • required: Modul ist für den Gesamterfolg nötig, Folgemodule werden bei Fehlschlag noch gestartet, erst dann erfolgt eine Fehlermeldung an den User • requisite: wie required, bei Fehlschlag wird die Anmeldung sofort abgebrochen (kein pw übers Netz bei pam_securetty.so) • sufficient: Erfolg des Moduls reicht für Anmeldung, wenn vorher kein required-Modul fehlgeschlagen ist, ein Fehlschlagen des sufficient-Moduls reicht nicht für den Fehlschlag des Stapels • optional: nur abgearbeitet, wenn ausschließlich optional-Module vorhanden sind, sonst ignoriert (nur bei pam_lastlogin.so) **Modul-Pfad(module-path):** Dateiname inklusive Pfad zum Modul **Parameter (module-arguments):** modulspezifische Argumente mit Leerzeichen getrennt Modul + Parameter: welche Module werden für welche Aufgabe in welcher Reihenfolge genutzt z. B.: `auth required /lib/security/pam_pwdb.so shadow nullok` → Anmeldung an der shadow-Datenbank ist notwendig, leere Passwörter sind zugelassen

PAM-Architektur: