LinuxCampus.net
trainings from the experts

Linux 2 - System Administration
Pluggable Authentication Modules

Peter Jahn, MSc
System Engineer
training@LinuxCampus.net

Kurs Agenda

Linux2 : Pluggable Authentication Modules

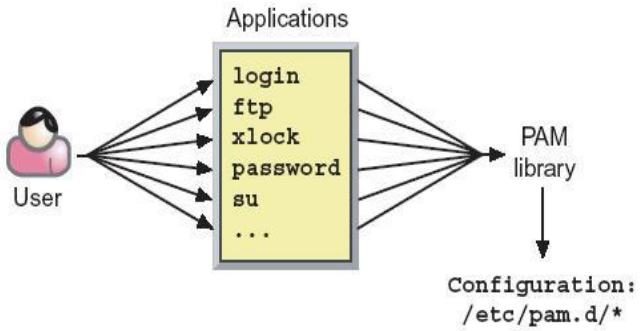
- Aufgabe von PAM
- PAM Module
- Konfigurationsbeispiele
- pam-config

PAM Grundlagen

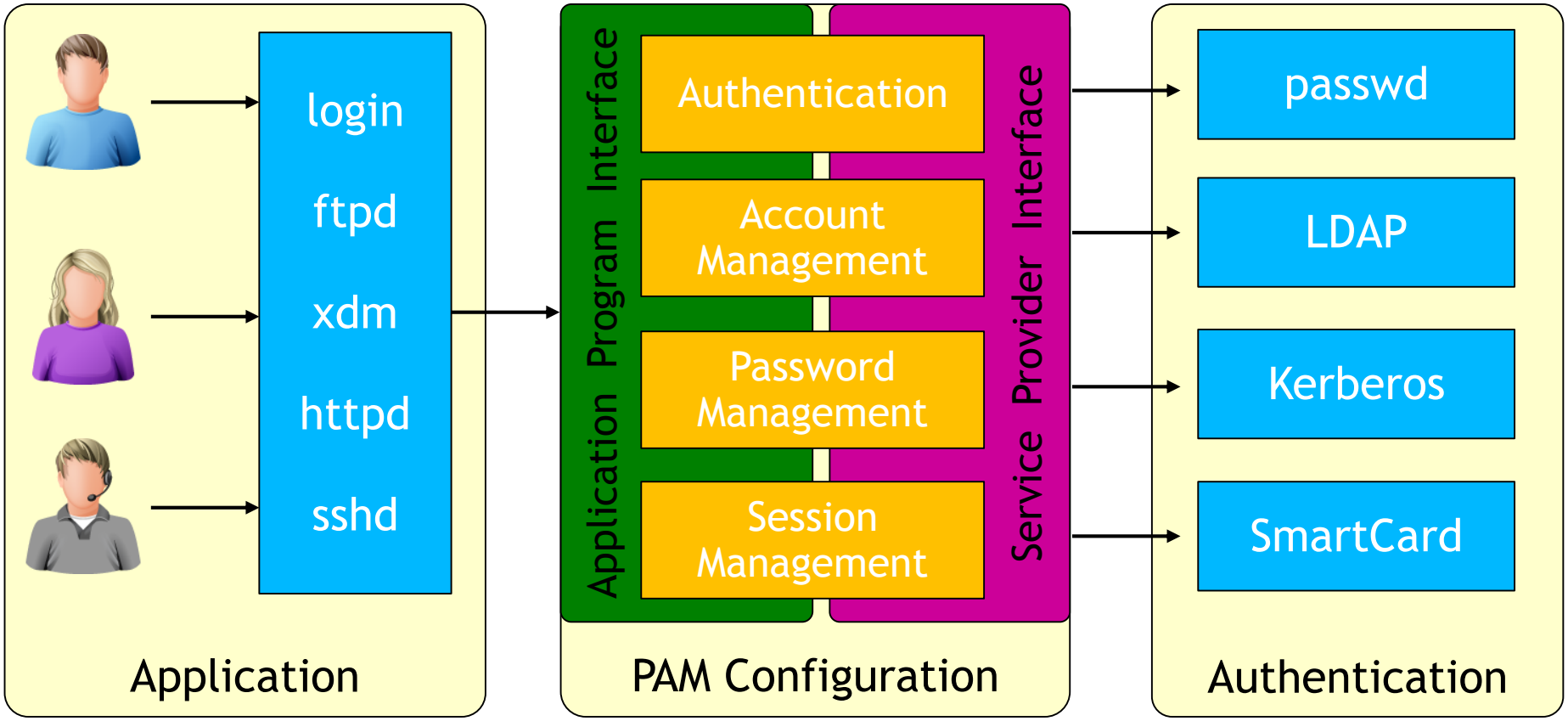
Pluggable Authentication Modules

- Was bedeutet PAM
  - Pluggable Authentication Modules
  - PAM wurde von SUN entwickelt

- Aufgabe von PAM
  - Strikte Trennung zwischen dem
    authentifizierenden Programm und
    dem eigentlichen Verfahren



Pluggable Authentication Modules



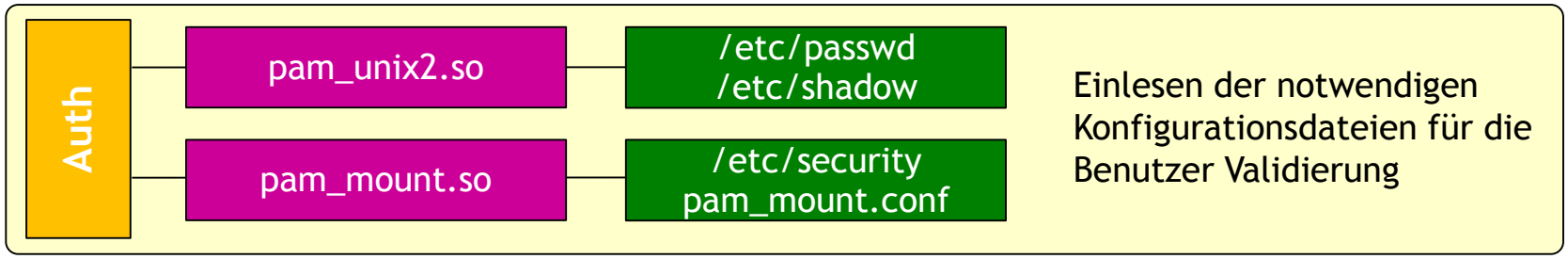
einfache Benutzer Validierung

Benutzer Login

Zeit

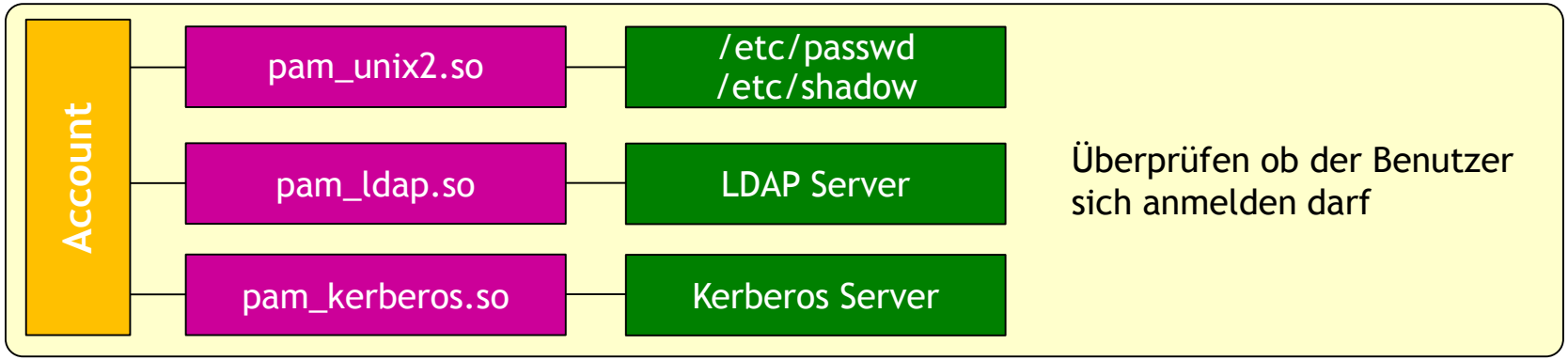
Auth	pam_unix2.so	/etc/passwd /etc/shadow	Einlesen der notwendigen Konfigurationsdateien für die Benutzer Validierung
	pam_mount.so	/etc/security pam_mount.conf	
Account	pam_unix2.so	/etc/passwd /etc/shadow	Überprüfen ob der Benutzer sich anmelden darf
	pam_krb5.so	Kerberos Server	
Session	pam_unix2.so	schreibt zu syslog	Protokollieren der Anmeldung und Erstellen der Arbeits-umgebung
	pam_mount.so	mount Verzeichnis	

Authentication Group



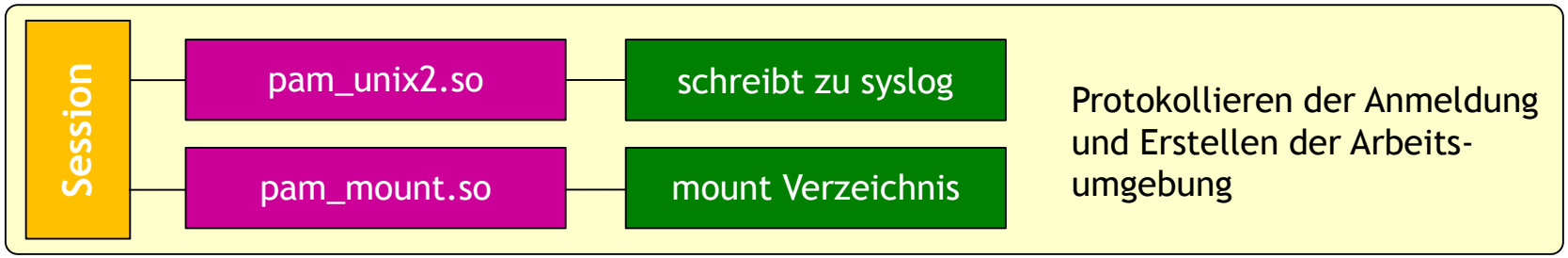
- Authentication Group Aufgabe
  - unabhängig vom verwendeten Back-End (passwd, LDAP,...) werden folgende Schritte durchgeführt
    - Einlesen der Konfigurationsdateien
    - Überpüfen des Benutzers aufgrund seiner Credentials
      - Benutzername, Passwort, Smartcard,...
    - Definieren der Gruppenmitgliedschaften

Account Group



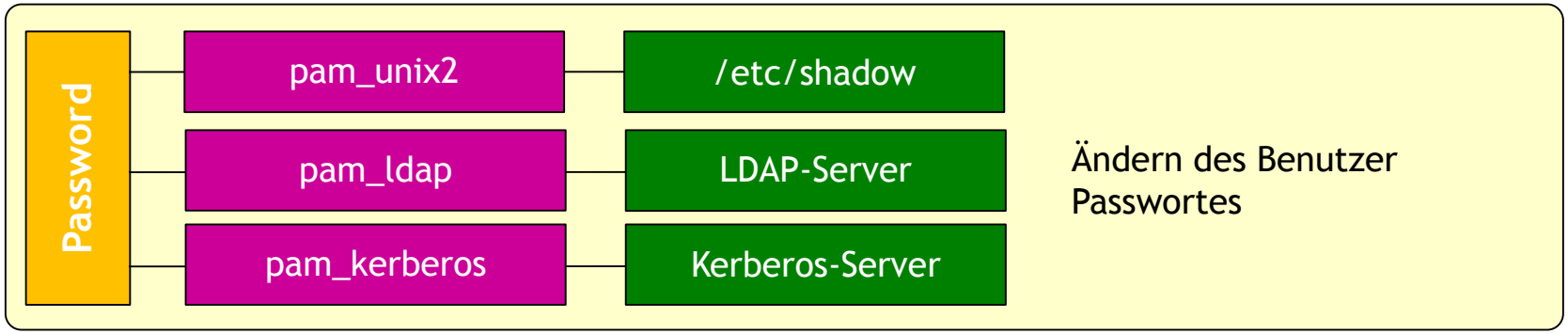
- Account Group Aufgabe
  - Ist der Benutzer Account abgelaufen?
    - /etc/shadow, LDAP, Kerberos,...
  - Darf sich der Benutzer um diese Zeit anmelden?
  - Soll auch ein Kerberos Ticket geholt werden?

Session Group



- Session Group Aufgabe
  - Erstellen der User-spezifischen Arbeitsumgebung beim Login
    - Protokollieren des Logins
    - Erstellen und Aktivieren des (verschlüsselten) Heimatverzeichnisses
  - Beim Logout des Benutzers
    - Deaktivieren des Heimatverzeichnisses
    - Protokollieren des Logouts

Password Group



- Password Group Aufgabe
  - Diese Gruppe wird nur ausgeführt wenn der Benutzer sein Passwort ändern möchte
    - Definition von Passwortregeln
    - Ändern der Passwörter in allen Bereichen
      - lokales Linux Passwort
      - LDAP Passwort
      - Kerberos Passwort

PAM und Verzeichnisstruktur

- Globale Konfigurationsdateien
  - /etc/security/
- PAM-Modul Konfigurationsdateien
  - /etc/pam.d/*programm-name*
- Standard PAM-Module
  - /lib/security/pam_*.so

PAM-Modul Konfigurationsdateien

- PAM-Modul Konfigurationsdateien
  - neben den Standard Konfigurationsdateien hat jedes PAM-aktivierte Authentifizierungsprogramm auch eine eigene Konfigurationsdatei unter /etc/pam.d/

/etc/pam.d	
xdm, gdm	GUI-Login
su, sudo	User Wechsel
other	wenn nichts anderes zutrifft
samba	Samba Server
cups	Drucker

/etc/pam.d	
login	allgemeiner Login
sshd	SSH Server
useradd	Benutzer Erstellung
passwd	Passwort Änderung
shadow	Passwort Speicher

PAM Anwendungscheck

- Prüfen ob eine Anwendung PAM unterstützt
  - wenn PAM unterstützt wird zeigt ldd libpam Einträge an

```
#PAM check mit ldd
peter@tux# ldd /usr/bin/passwd
        linux-gate.so.1 =>  (0xffffe000)
        libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7f15000)
        libpam.so.0 => /lib/libpam.so.0 (0xb7f09000)
        libldap-2.4.so.2 => /usr/lib/libldap-2.4.so.2 (0xb7eca000)
        liblber-2.4.so.2 => /usr/lib/liblber-2.4.so.2 (0xb7ebb000)
        libnsl.so.1 => /lib/libnsl.so.1 (0xb7ea4000)
        libnscd.so.1 => /lib/libnscd.so.1 (0xb7ea1000)
        libc.so.6 => /lib/libc.so.6 (0xb7d5d000)
        ...
```

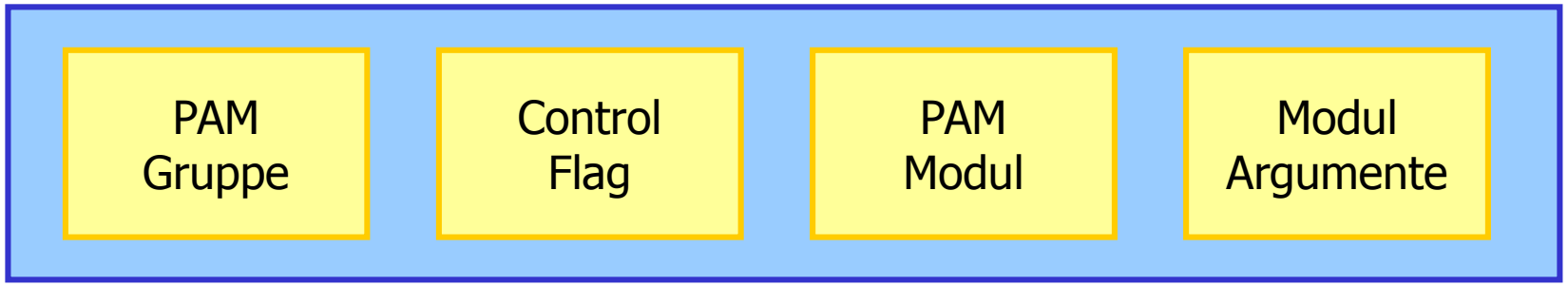
Konfigurationsdateien

Beispiel PAM Konfigurationsdatei

- /etc/pam.d/login

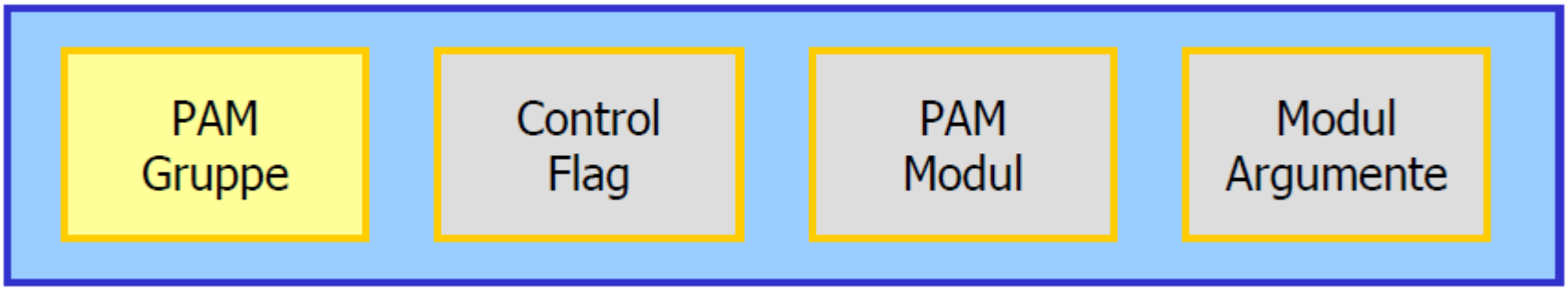
```
#%PAM-1.0
auth       required     pam_securetty.so
auth       include      common-auth
auth       required     pam_nologin.so
account    include      common-account
password   include      common-password
session    include      common-session
session    required     pam_lastlog.so nowtmp
session    required     pam_resmgr.so
session    optional     pam_mail.so standard
```

Aufbau der Konfigurationsdatei




```
#%PAM-1.0
auth       required     pam_securetty.so
auth       include      common-auth
auth       required     pam_nologin.so
account    include      common-account
password   include      common-password
session    include      common-session
session    required     pam_lastlog.so nowtmp
session    required     pam_resmgr.so
session    optional     pam_mail.so standard
```

PAM Gruppe



- Die PAM Modulgruppe
  - gibt an im welchem Bereich das PAM Modul eingesetzt wird.
  - eine Gruppe kann auch mehrere Module beinhalten die dann nacheinander abgearbeitet werden
  - Es gibt 4 Gruppen: auth, account, password, session

auth

- auth (wie muss sich der Benutzer ausweisen)
  - Dieses Modul bestimmt wie sich ein User Authentifizieren muss
    - Passwort, Fingerabdruck, Spracherkennung

Argumente	Beschreibung
debug	Aktiviert Logging zu Syslog
use_first_pass	Benutzt das Passwort vom vorherigen Modul
try_first_pass	wie use_first_pass jedoch wenn es fehlschlägt wird der Benutzer nach einem Passwort gefragt
nullok	Erlaubt Accounts ohne Kennwort (Null = OK)
ndelay	Bei einem Fehler sofort melden. Nicht verwenden da es das Hacken erleichtern würde (No Delay)

Authentifizierungstyp

- account (Vorraussetzung für eine Anmeldung)
  - Normalerweise wird es benutzt, um den Zugang zu Diensten aufgrund der Tageszeit, der momentan verfügbaren Systemressourcen (maximale gleichzeitige Benutzer) oder vielleicht der Lage des "root"-Logins auf der Konsole zu erlauben oder verweigern
    - Benutzername, Zeiteinschränkung, IP-Adresse,...

Authentifizierungstyp

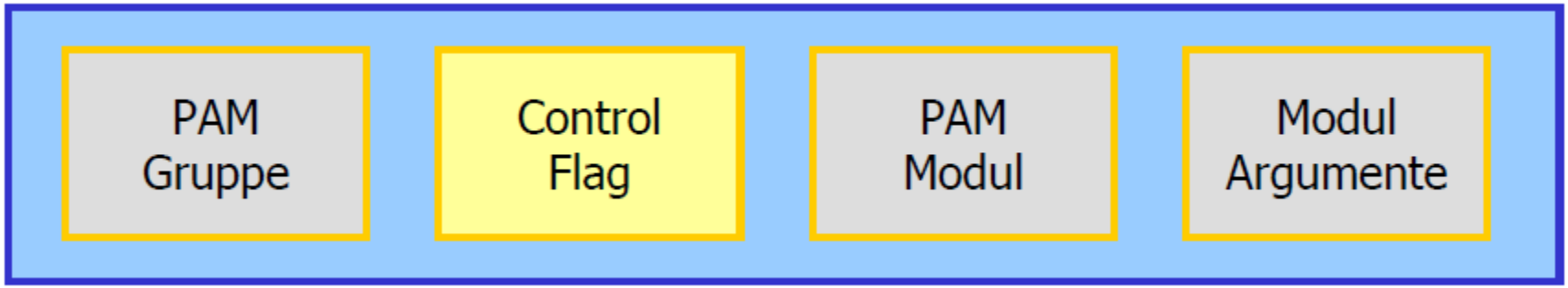
- session (vor/nach der Session ausführen)
  - Dieses Modul wird vor allem verwendet, um festzulegen, was ausgeführt werden sollte, bevor ein Benutzer einen Dienst benutzt oder beendet
  - Dies könnte zum Beispiel das Aufzeichnen von Informationen bezüglich des Datenaustauschs mit dem Benutzer das Mounten von Ordnern ein Eintrag in einem LOG, oder ein Begrüssungstext sein

Authentifizierungstyp

- password (Passwort Einstellungen)
  - Dieser letzte Modultyp wird gebraucht, um das mit dem Benutzer verbundene Authentifizierungstoken zu aktualisieren
    - z.b Passwort Änderung
  - Normalerweise gibt es ein Modul für jeden "challenge/response" basierten Authentifizierungstyp

Argumente	Beschreibung
retry=N	Prompt User N mal vor Abbruch
use_authtok	Erzwingt die Benutzung des vorherigen Passwortes
minclass=N	Unterschiedliche Klassen die ein Passwort enthalten muss (Digits, Upper, Lower,...)

Control Flag



- Das Control Flag
  - bestimmt wie PAM auf einen positiven oder negativen Rückgabewert eines Modules reagieren soll
  - Mögliche Flags sind: required, requisite, sufficient, optional

Control Flag

- Requisite
  - Ist das stärkste Flag und wenn es nicht erfüllt ist, wird sofort der Fehler der Applikation gemeldet ohne weitere Tests durchzuführen

- Required
  - Im Falle eines Fehlers wird das Problem vermerkt, jedoch wird der Stack nicht abgebrochen, sondern es werden die nächsten Tests durchgeführt
    - Die Überprüfung der weiteren Tests erfolgt nur damit ein Hacker nicht genau weiß in welchem Modul der Fehler war

Control Flag

- Sufficient
  - Wenn dieses Modul erfüllt ist und vorher kein anderer Fehler war dann werden keine weiteren Tests mehr durchgeführt
- Optional
  - Wenn dieses Modul fehlschlägt macht das nichts und es werden die nächsten Tests durchgeführt

Beispiel Control Flags und SSH Login

```
# /etc/pam.d/sshd
auth        required        pam_nologin.so
auth        sufficient      pam_winbind.so
auth        required        pam_unix2.so   use_first_pass
```

- required pam_nologin.so
  - Wurde der Login für Standard Benutzer deaktiviert?
- sufficient pam_winbind.so
  - Gibt es den Benutzer in der Windows/Samba Domain?
- required pam_unix.so use_first_pass
  - Auch wenn der vorherige Test fehlschlägt dürfen sich trotzdem lokale Benutzer anmelden

Die Reihenfolge ist wichtig!

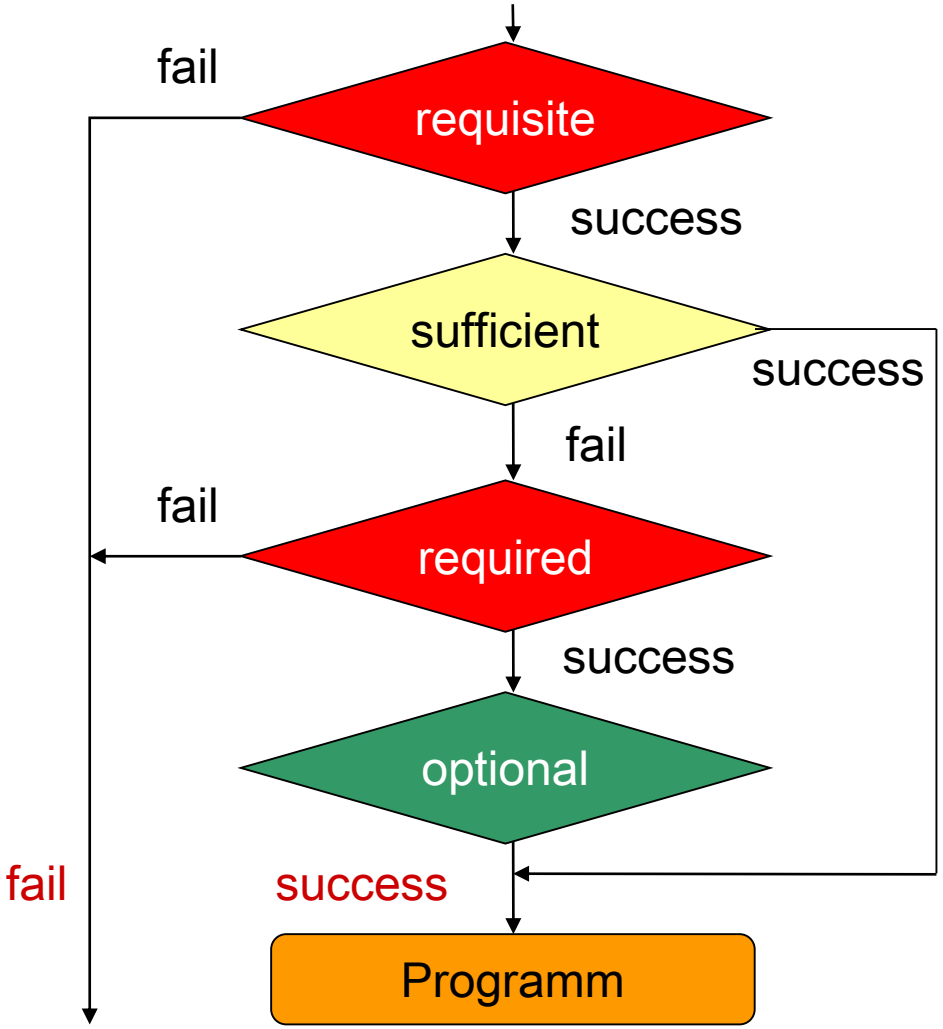
- richtige Reihenfolge
  - pam_deny trifft nur zu wenn pam_unix nicht erfüllt wurde

```
# /etc/pam.d/login
auth        required      pam_unix2.so
auth        optional      pam_deny.so       #verhindert login
```

- falsche Reihenfolge
  - pam_deny trifft zu bevor pam_unix überprüft werden kann

```
# /etc/pam.d/login
auth        optional      pam_deny.so       #verhindert login
auth        required      pam_unix2.so
```

Verkettung der Module



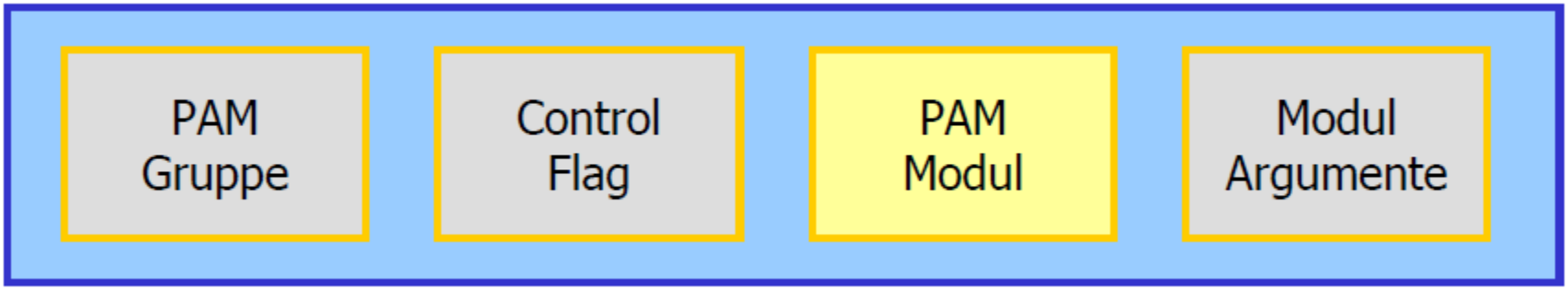
....unbedingt notwendig

...ausreichend, keine weiteren Überprüfungen mehr notwendig

...unbedingt notwendig

...egal ob ja oder nein

PAM Modul



- Das PAM Modul
  - beinhaltet den Namen des eigentlichen Moduls
  - wenn das Modul unter /lib/security liegt reicht der Dateiname ansonsten muss der ganze Pfad angegeben werden
    - z.b: pam_nologin.so

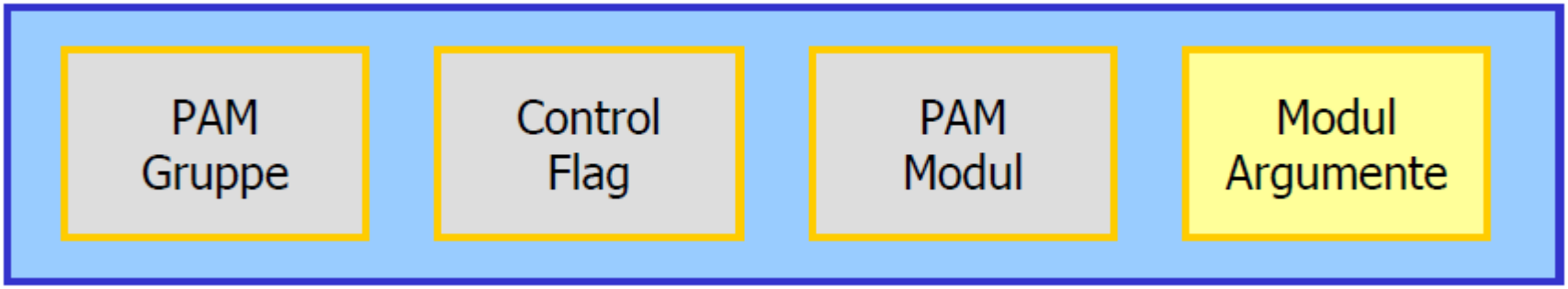
PAM Modul

- Welche PAM Module existieren auf dem System

```
#Anzeigen der verfügbaren PAM Module
peter@tux# ls /lib/security
pam_access.so          pam_homecheck.so   pam_permit.so
pam_deny.so            pam_limits.so      pam_rootok.so
pam_echo.so            pam_listfile.so    pam_rpasswd.so
pam_env.so             pam_localuser.so   pam_securetty.so
pam_exec.so            pam_loginuid.so    pam_shells.so
pam_faildelay.so       pam_mail.so        pam_smbpass.so
pam_gnome_keyring.so   pam_namespace.so   pam_time.so
pam_group.so           pam_nologin.so     pam_tty_audit.so
...
```

man pam_tally ...zeigt Dokumentation zum dem PAM Modul

Modul Parameter



- Die Spalte Modul Parameter
  - beinhaltet (meistens) optionale Parameter
  - Mögliche Parameter sind vom jeweiligen Modul abhängig
    - minimum_uid=1000
    - use_first_pass

Modul Parameter

- debug
  - Schickt Diagnose Informationen zum Syslog Daemon
    - Dieser Parameter kann in allen 4 Management Groups verwendet werden
- use_first_pass
  - Übergibt das Passwort das der Benutzer bei einem vorherigen Schritt eingegeben hat zusätzlich an dieses Modul
    - Wird hauptsächlich in der Auth Management Group verwendet

Modul Parameter

- try_first_pass
  - Im Gegensatz zu use_first_pass fragt dieses Modul noch einmal nach einem weiteren Passwort falls das zuerst übergebene Passwort nicht gültig war
    - Wird hauptsächlich in der Auth Management Group verwendet

- expose_account
  - Dieser Parameter erlaubt PAM das die sensiblere Informationen "Username" ausgegeben werden darf

PAM Module

pam_mkhomedir

- pam_mkhomedir.so
  - Erstellt automatisch das Heimatverzeichnis eines Benutzers wenn es nicht vorhanden sein sollte
    - Wird oft bei Logins via LDAP eingesetzt

Argumente	Beschreibung
skel=/path	Alternativer Pfad für das Skeleton Verzeichnis
umask=xxxx	Spezielle Rechte am Heimatverzeichnis

```
# /etc/pam.d/common-session
session optional      pam_mkhomedir.so  skel=/etc/skeladmin umask=0022
...
```

pam_mount

- pam_mount.so
  - Dieses Modul kann Heimatverzeichnisse einbinden
    - verschlüsselte Heimatverzeichnisse
    - Windows & NetWare Shares
  - Es gibt viele Parameter für dieses Modul
  - http://pam-mount.sourceforge.net/pam_mount.conf.5.html

```
# /etc/pam.d/common-auth
auth   optional      pam_mount.so   use_first_pass
```

```
# /etc/pam.d/common-session
session optional      pam_mount.so
```

pam_succeed_if

- pam_succeed_if.so
  - Wird verwendet um Filter für Benutzer und Gruppen zu definieren die sich anmelden dürfen
    - Ideal für LDAP Integration
  - Es gibt viele Filtermöglichkeiten
    - http://linux.die.net/man/8/pam_succeed_if

Argumente	Beschreibung
gid=X,Y	Erlaubt nur Gruppen mit einer GUID X oder Y
uid >=1000	Erlaubt nur Benutzer mit einer UID grösser 1000

pam_nologin

- pam_nologin.so
  - Verhindert das sich nicht Root-Benutzer am System anmelden können, sobald eine Datei /etc/nologin existiert
  - Der Inhalt der Datei wird als Fehlermeldung bei einem Loginversuch ausgegeben
    - echo "Derzeit kein Zugriff möglich" > /etc/nologin

```
# /etc/pam.d/login
auth        required        pam_nologin.so
```

```
# /etc/pam.d/sshd
auth        required        pam_nologin.so
```

pam_deny

- pam_deny
  - Ist ein Modul das immer einen Login verhindert
  - Kann am Ende einer Konfiguration eingebunden werden um einen erfolgreichen Login aufgrund einer Fehlkonfiguration zu verhindern
  - Kann auch am Anfang der Passwort Konfiguration gesetzt werden um einen Passwortwechsel zu verhindern

```
# /etc/pam.d/login
auth        required        pam_nologin.so
auth        required        pam_unix2.so
auth        optional        pam_deny.so      #verhindert login
```

pam_securetty

- pam_securetty.so
  - erlaubt den Root-Login nur über die in /etc/securetty definierten Terminals
  - kann auch für XDM verwendet werden um zu bestimmen ob sich Root an der GUI (tty7) anmelden darf

```
# /etc/pam.d/login
auth        required        pam_securetty.so
auth        required        pam_unix2.so
```

```
# /etc/securetty
tty1
tty2
```

pam_tally

- pam_tally.so
  - Zählt unerfolgreiche Loginversuche und sperrt den Account
  - Kann in der Modulgruppe auth und account verwendet werden
  - http://linux.die.net/man/8/pam_tally
  - http://www.administrator.de/Kurzanleitung_%28HowTo%29_zur_Konfiguration_des_Linux_PAM_Moduls_pam_tally_am_Beispiel_des_SSH_Daemons.html

```
# /etc/pam.d/sshd
auth required pam_tally.so  onerr=fail  deny=5  no_reset
auth required pam_unix2.so
```

weitere PAM Module

PAM-Modul	Aufgabe
pam_time.so	Zu welcher Zeit darf man sich anmelden Konfigurationsdatei: /etc/security/time.conf
pam_limits.so	Darf (nicht) folgende Resourcen verwenden Konfigurationsdatei: /etc/security/limits.conf
pam_access.so	Wer darf (nicht) sich wo anmelden Konfigurationsdatei: /etc/security/access.conf
pam_group.so	Ist Mitglied von Gruppe Konfigurationsdatei: /etc/security/group.conf
pam_env.so	Export von Umgebungsvariablen Konfigurationsdatei: /etc/security/pam_env.conf

PAM Module für Backends

Backend: pam_unix2

- pam_unix2.so
  - Ist das am meisten verwendete Modul um Benutzer in /etc/passwd und /etc/shadow zu überpüfen
  - Abhängig von der verwendeten Modulgruppe stehen eine Vielzahl von Argumenten zur Verfügung
    - debug, nullok, use_first_pass, ...

```
# /etc/pam.d/login
auth        required      pam_nologin.so
auth        required      pam_unix2.so
auth        optional      pam_deny.so      #verhindert login
```

Backend: pam_winbind

- pam_winbind.so
  - Dieses Modul wird durch durch Samba geliefert um Benutzer aus Active Directory in Samba einzubinden
  - Mögliche Argumente
    - http://www.samba.org/samba/docs/man/manpages-3/pam_winbind.8.html

```
# /etc/pam.d/login
auth     required       pam_nologin.so
auth     sufficient     pam_winbind.so    #Domain Accounts
auth     required       pam_unix2.so       #Lokale Accounts
```

Backend: pam_ldap

- pam_ldap
  - Dieses Modul wird verwendet um Benutzer aus einem LDAP Verzeichnis einzubinden
    - eDirectory, Active Directory, OpenLDAP,...
  - Mögliche Argumente
    - http://linux.die.net/man/5/pam_ldap

```
# /etc/pam.d/common-auth
auth    required        pam_env.so
auth    required        pam_unix2.so
auth    required        pam_ldap.so   use_first_pass
```

Backend: pam_mysql

- pam_mysql.so
  - Dieses Modul wird verwendet um Benutzer aus einer MySQL Datenbank einzubinden
  - http://pam-mysql.sourceforge.net/

PAM und OpenLDAP Login

Authentifizierung

```
# /etc/pam.d/common-auth
auth    required        pam_env.so
auth    required        pam_unix2.so
auth    required        pam_ldap.so  use_first_pass
```

- pam_env.so
  - Liest Umbebungsvariablen aus /etc/security/pam_env.conf
- pam_unix2.so
  - Überprüft Einstellungen in /etc/nsswitch.conf
- pam.ldap.so use_first_pass
  - Auch der LDAP Tree soll nach Usernamen durchsucht werden
  - das bereits übergebene Passwort für LDAP verwenden

Autorisierung

```
# /etc/pam.d/common-account
account required        pam_unix2.so
account sufficient      pam_localuser.so
account required        pam_ldap.so  use_first_pass
```

- pam_unix2.so
  - Stellt sicher das ein Account nicht abgelaufen ist
- pam_localuser.so
  - Ist der User in einer Passwort Datei enthalten
- pam_ldap.so use_first_pass
  - Prüft im DIT nach User Einstellungen

Passwort Änderung

```
# /etc/pam.d/common-password
password        requisite       pam_pwcheck.so   nullok cracklib
password        required        pam_unix2.so     use_authtok nullok
password        required        pam_ldap.so  try_first_pass use_authtok
# Diese Module werden nur aktiv wenn ein Passwort verändert wird
```

- pam_pwcheck.so nullok cracklib
  - liest die Passworteinstellungen aus /etc/login.defs
- pam_unix2.so use_authtok nullok
  - Verlangt nach altem Kennwort bei PW Änderung
- pam_ldap.so try_first_pass use_authtok
  - zusätzlich zu /etc/shadow wir auch das Passwort im DIT gesetzt

Session

```
# /etc/pam.d/common-session
session optional        pam_mkhomedir.so
session required        pam_limits.so
session required        pam_unix2.so
session optional        pam_ldap.so
session optional        pam_umask.so
```

- pam_ldap.so
  - Überpüft auf Regeln im DIT
- pam_mkhomedir.so
  - Versucht für den User ein Heimatverzeichnis zu erstellen wenn es noch keines gibt

/etc/ldap.conf

```
# /etc/ldap.conf
# OpenLDAP SSL mechanism
# start_tls mechanism uses the normal LDAP port, LDAPS typically 636
#ssl start_tls
ssl   no
ldap_version    3
pam_filter                  objectClass=posixAccount
tls_checkpeer   no
```

- ssl no
  - wir haben noch kein SSL und daher deaktivieren
- pam_filter
  - hier können komplexe Filter stehen die definieren wer sich am System anmelden darf

PAM und Kerberos

Kerberos und pam auf DEBIAN

```
# /etc/pam.d/common-auth
auth    [success=1 default=ignore]      pam_winbind.so krb5_auth
    krb5_ccache_type=FILE cached_login try_first_pass
```

```
# /etc/pam.d/common-account
account required        pam_krb5.so  minimum_uid=1000
```

```
# /etc/pam.d/common-password
password    requisite     pam_krb5.so  minimum_uid=1000
```

```
# /etc/pam.d/common-session
session optional    pam_krb5.so  minimum_uid=1000
```

- minimum_uid=1000
  - stellt sicher das Service Account wie auch Root sich anmelden können auch wenn kein KDC erreichbar ist

Kerberos und pam auf SUSE

```
# /etc/pam.d/common-auth
auth    sufficient krb5_auth  use_first_pass
```

```
# /etc/pam.d/common-account
account required  pam_krb5.so  use_first_pass   ignore_unknown_principals
```

```
# /etc/pam.d/common-password
password    requisite     pam_krb5.so
```

```
# /etc/pam.d/common-session
session optional    pam_krb5.so
```

- ignore_unknown_principals
  - Wenn ein Principal nicht existiert wird ein PAM_IGNORE anstatt einem PAM_USER_UNKNOWN zurück geliefert

Kerberos und pam auf RHEL

```
# /etc/pam.d/system-auth
# This file is auto-generated. User changes will be destroyed the next time
    authconfig is run.
auth  sufficient  pam_krb5.so  use_first_pass
...
account [default=bad success=ok user_unknown=ignore]  pam_krb5.so
...
password sufficient  pam_krb5.so  use_authtok
...
session optional  pam_krb5.so
```

- user_unknown=ignore
  - Wenn ein Principal nicht existiert wird kein Fehlercode zurück geliefert

Dokumentation

- Gute PAM Dokumentationen
  - http://www.techrepublic.com/article/controlling-passwords-with-pam/1055267
  - http://www.linux.ie/articles/pam.php
  - http://gertranssmb3.berlios.de/output/pam.html
  - http://pig.made-it.com/pam.html

PAM und pam-config

Standardkonfiguration

- /etc/pam.d/common-[account,auth,password,session]
  - Zur Vereinfachung der Erstellung und Verwaltung von PAM-Modulen stehen Dateien mit gängigen Standard Konfigurationen für die Module auth, account, password und session bereit. Diese werden den PAM-Konfigurationen der einzelnen Anwendungen entnommen.
  - Aktualisierungen der globalen PAM-Konfigurationsmodule in common-* werden daher auf alle PAM-Konfigurationsdateien übertragen. Die manuelle Aktualisierung jeder einzelnen PAM-Konfigurationsdatei durch den Administrator entfällt somit.

Import in SSHD-Konfiguration

```
# /etc/pam.d/sshd
auth        requisite     pam_nologin.so
auth        include       common-auth
account     requisite     pam_nologin.so
account     include       common-account
password    include       common-password
session     required      pam_loginuid.so
session     include       common-session
```

- include
  - Einzelne Dienste können die Standardkonfiguration des Systems importieren

Standardkonfiguration anpassen

- pam-config
  - Die globalen, allgemeinen PAM-Konfigurationsdateien werden mit dem Tool pam-config verwaltet. Dieses Tool fügt der Konfiguration automatisch neue Module hinzu, ändert die Konfiguration vorhandener Module oder löscht einzelne Module oder Optionen aus den Konfigurationen.

Option	Erklärung
-a --add	Optionen/PAM-Module hinzufügen
-c --create	Neue Konfiguration erstellen
-d --delete	Optionen/PAM-Module entfernen
-q --query	Abfrage auf installierte Module und Optionen
--list-modules	Alle unterstützten Module anzeigen

Konfigurieren von PAM mit pam-config

- Automatische Generierung einer neuen PAM-Konfiguration
  - Das Kommando pam-config --create erstellt eine einfache UNIX-Authentifizierungskonfiguration, die Sie später erweitern können
  - Bereits vorhandene, nicht von pam-config verwaltete Konfigurationsdateien werden überschrieben.
    - Allerdings werden von diesen Dateien Sicherungskopien mit dem Namen *.pam-config-backup erstellt.

Konfigurieren von PAM mit pam-config

- Hinzufügen einer neuen Authentifizierungsmethode.
  - Zum Hinzufügen einer neuen Authentifizierungsmethode zB. LDAP zu Ihrem PAM-Modulstapel benötigen Sie lediglich das Kommando
    - pam-config --add --ldap
  - Die LDAP-Authentifizierungsmethode wird allen common-*-pc-PAM-Konfigurationsdateien hinzugefügt, auf die diese Methode anwendbar ist.

Konfigurieren von PAM mit pam-config

- Aktivieren der Debug-Funktion für Testzwecke.
  - Um sicherzustellen, dass die neue Authentifizierungs-methode wie geplant funktioniert, aktivieren Sie die Debug-Funktion für alle PAM-Vorgänge
  - Im Falle von LDAP verwenden Sie dazu das Kommando
    - pam-config --add --ldap-debug
      - Die Debug-Ausgabe finden Sie in /var/log/messages
  - Entfernen der Debug-Optionen. Im Falle von LDAP verwenden Sie dazu das Kommando
    - pam-config --delete --ldap-debug

Red Hat

- Red Hat und PAM
  - Auf RHEL6 gibt es kein pamconfig aber dafür gibt es das Kommandozeilenprogramm authconfig

Passwort Einstellungen

Passwort Richtlinien

- pam_cracklib
  - ermöglicht das Setzen von schärferen Passwort Richtlinien
    - gelten nicht für den Root Benutzer
  - pam-config --cracklib-minlen=8

pam_cracklib-so	Auswirkung
minlen=8	Min Passwortlänge ist 8
lcredit=-1	Min Nummer von Kleinbuchstaben ist 1
ucredit=-1	Min Nummer von Großbuchstaben 1
dcredit=-1	Min Nummer von Zahlen ist 1
ocredit=-1	Min Nummer von anderen Zeichen ist 1


LinuxCampus.net
trainings from the experts