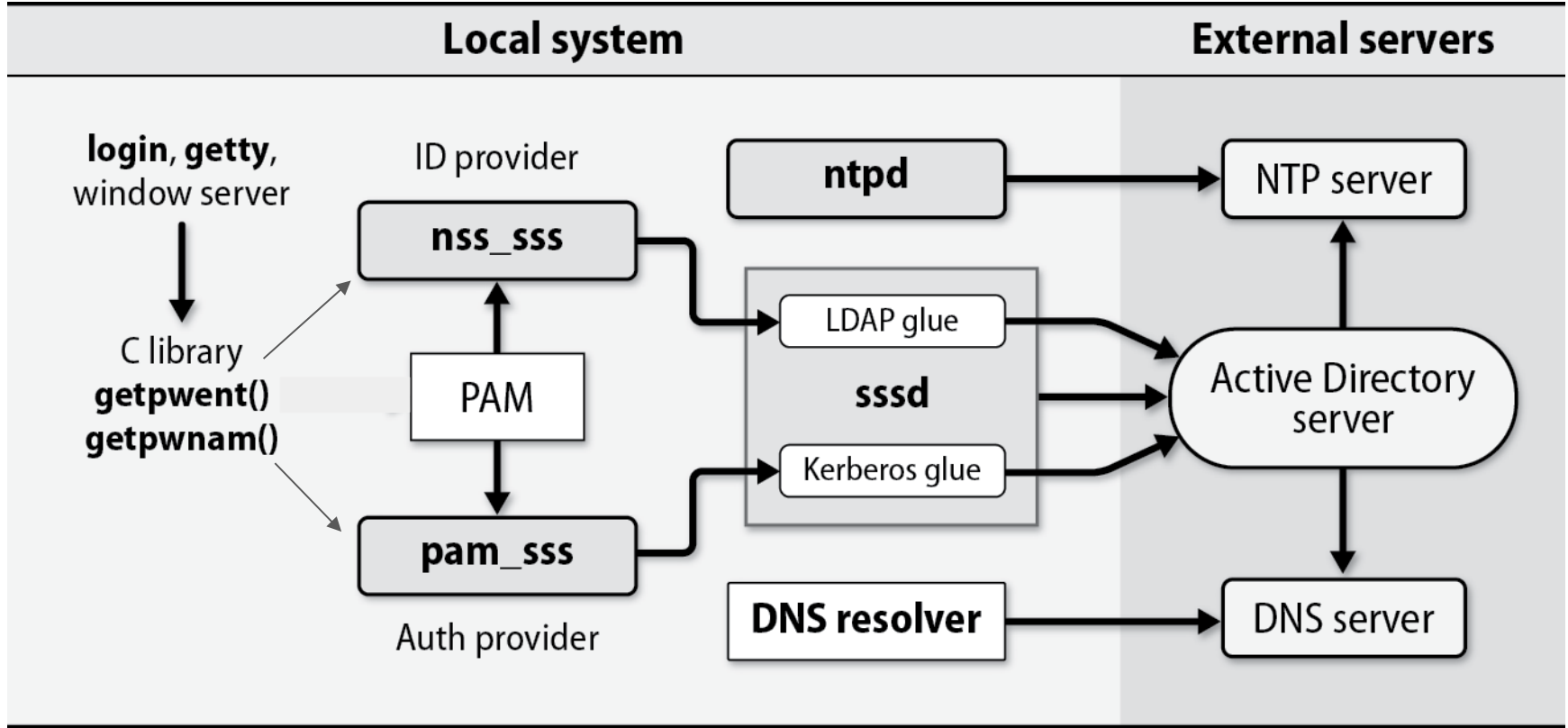# CS 447/647

Pluggable Authentication Modules

Goals

What is the purpose of PAM?

What are the four PAM control types?

How are PAM variables passed into libpam-script scripts?

What is the data structure used by PAM modules?

Local system

External servers

login, getty, window server

C library
getpwent()
getpwnam()

ID provider

nss_sss

PAM

pam_sss

Auth provider

ntpd

LDAP glue

sssd

Kerberos glue

DNS resolver

NTP server

Active Directory server

DNS server

Pluggable Authentication Modules (PAM)

- Provides an interface for authentication
  - login(1) utility calls the PAM libraries
  - Iterates over a stack composed of modules
- Configurations in /etc/pam.d/*
- "You can choose to have no security or absolute security (no access permitted)."
  - Errs toward the latter
- Configuration errors will lock you out.

Pluggable Authentication Modules (PAM)

- Four separate types of (management) module-type
  - Authentication
  - Account
  - Session
  - Password
- man pam.conf

Authentication

- Instructs application to prompt for username and password
  - Verifies access credentials
  - pam_unix.so checks /etc/passwd and /etc/shadow

Account

- Non-authentication based account management
- Restrict/permit access to a service based on
  - Time of day
  - Resources available
  - Location of a user

Session

- "Does things" before/after a user can be given service
  - ○ Logging
  - ○ Mounting filesystems

Password

- Used for changing and manipulating passwords.

Example

module-type control-flag module-path [arguments]

Add:

auth   optional   pam_warn.so

To:

/etc/pam.d/common-auth

Exit and log back-in

tail -n20 /var/log/auth.log

Absolute security

```
#
# default; deny access
#

other    auth     required      pam_deny.so
other    account  required      pam_deny.so
other    password required      pam_deny.so
other    session  required      pam_deny.so
```

```
¯\_( ツ )_/¯

#
# default; any access
#

other    auth     required      pam_permit.so
other    account  required      pam_permit.so
other    password required      pam_permit.so
other    session  required      pam_permit.so

#This module is very dangerous. It should be used with
extreme caution. man 8 pam_permit
```

Modules

pam_access - Provides access management

pam_unix - Authenticate against /etc files

pam_env - Control environmental variables

pam_systemd - Registers sessions in systemd hierarchy

pam_ldap - LDAP authentication

pam_sss - SSS authentication

pam_permit - Always allows access

pam_listfile - Allows you to authenticate based on lines in a file

module-type control-flag module-path [arguments]

module-type

auth   - Identify user and grant permissions
account - enforces restrictions
session - tasks before login
password - changing a password

control-flag

include   - Includes another file
optional   - Only important if the only module
required   - Failure eventually causes stack to fail
requisite - Same as required but stack fails immediately
sufficient - Exits upon success but does not override

module-path (/lib/x86_64-linux-gnu/security)

pam_unix.so - Unix file authentication   *Local*
pam_ldap.so - LDAP authentication   *Network*
pam_krb5.so - Kerberos Authentication   *Network*
pam_sss.so - SSSD authentication   *Network*

```
auth        optional    pam_faildelay.so   delay=3000000
auth [success=ok new_authtok_reqd=ok ignore=ignore user_unknown=bad default=die] pam_securetty.so
auth        requisite   pam_nologin.so
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
session     required        pam_loginuid.so
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
session         required    pam_env.so readenv=1
session         required    pam_env.so readenv=1 envfile=/etc/default/locale

@include common-auth
auth        optional    pam_group.so
session     required    pam_limits.so
session     optional    pam_lastlog.so
session     optional    pam_motd.so motd=/run/motd.dynamic
session     optional    pam_motd.so noupdate
session     optional    pam_mail.so standard
session     optional    pam_keyinit.so force revoke

@include common-account
@include common-session
@include common-password
```

Syntax [value1=action1 value2=action2 ...]

Values   Actions
success   ignore - will not contribute to return
open_err   bad   - module failed
symbol_err   die   - same as bad but immediately exits
service_err   ok   - PAM_SUCCESS
system_err   done   - Terminate the stack and return
buf_err   N   - Same as OK but skips N modules
perm_denied   reset - clear all memory and start with
abort   next
default   module
...

Handling authentication

```
apt install libpam-script libpam-mkhomedir

pam-auth-update #Friendly

vim|emacs|nano /etc/pam.d/common-auth
```

libpam-script

- Invoke scripts within the PAM stack
  - Authentication
  - Passwd changes
  - Session opening
  - Session closing
- Scripts stored in /usr/share/libpam-script

libpam-script scripts

- **pam_script_auth - Authentication**

- pam_script_acct - Account

- pam_script_passwd - Password changes

- **pam_script_ses_open - Session open**

- pam_script_ses_close - Session close

pam_script_auth

/usr/share/libpam-script/pam_script_auth

```python
#!/usr/bin/env python3
import os
import sys

print("PAM Script Start")
for k, v in os.environ.items():
  print("{0}, {1}".format(k, v))

print("PAM Script End")

sys.exit(0)
```

PAM

/run/pwchk

uid: greater than 0

pam_script_auth

user:password

uid: 0

pwchk.py

/etc/pam.d/nginx_pwchk

```
auth    optional                        pam_warn.so
auth    [success=1 default=ignore]      pam_script.so
dir=/usr/local/pam
auth    requisite                       pam_deny.so
auth    required                        pam_permit.so
```

/etc/pam.d/login #or

```
auth    optional                        pam_warn.so
auth    [success=1 default=ignore]      pam_unix.so
auth    requisite                       pam_deny.so
auth    required                        pam_permit.so
```

True or False

socket
0666 rw-rw-rw