PAM 기본구조

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

type	설명
auth	사용자 패스워드 유효성 검사와 같은 서비스 인증 절차에 사용 다른 인증모듈(Kerberos Ticket과 같은 연동도 가능)
account	서비스 사용자가 해당 서비스에 접근이 가능되는지 여부 검사 계정 활성화/비활성화 여부 확인 특정 시간대에 접속 시도 가능여부 확인
password	서비스 사용자가 패스워드 등의 인증 방법에 대한 변경을 시도할 때 사용 패스워드 변경 시 최소길이/복잡도 설정 등과 관련됨
session	사용자가 인증 받기 전/후에 해야 할 것들을 지정 디렉터리 마운트/언마운트, 메일함 할당 등

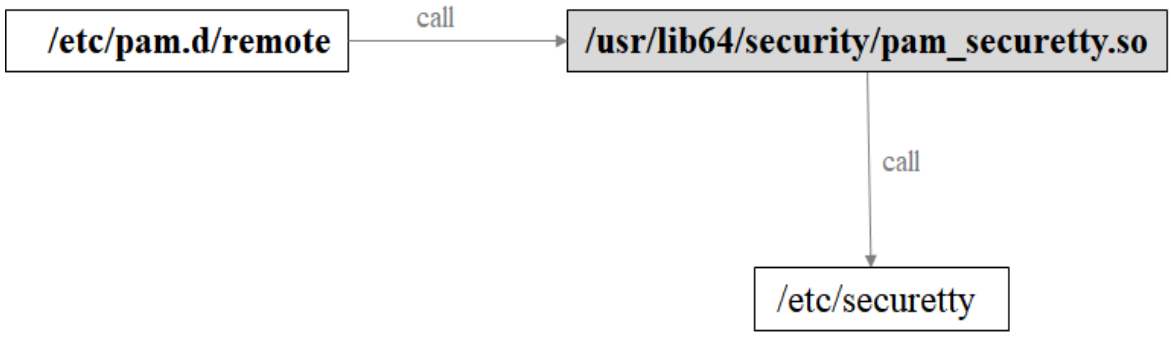
control	설명
sufficient	이전 인증이 실패하더라도 해당 모듈이 성공이면 PAM 인증을 승인
requisite	이 모듈의 인증이 실패할 경우 즉시 인증 거부
required	다른 인증들이 성공하더라도 해당 인증이 실패할 경우 인증 거부
include	인자(argument)에 지정된 또 다른 설정 파일의 내용이나 지침 포함 서로 다른 PAM 설정 파일의 내용을 연결하고 구성하는 방식으로 사용

라이브러리	설명
pam_permit.so pam_deny.so	항상 성공/실패를 리턴해서 접근을 허용, 거부.
pam_warn.so	호출한 사용자 및 호스트 정보를 syslog에 기록.
pam_access.so	/etc/security/access.conf 사용자 계정, 호스트/도메인을 통해서 시스템 접근을 허용
pam_pwdb.so	/etc/passwd 와 /etc/shadow
pam_env.so	/etc/security/pam_env.conf /etc/environment와 /etc/security/pam_env.conf 설정 파일이 비었을 경우 변수 설정 없이 성공 값을 반환한다.
pam_issue.so pam_motd.so	/etc/issue, /etc/motd 로그인 시 issue나 message를 출력하며, 성공적으로 로그인한 후 나타남
pam_tally2.so	로그인 시도 횟수를 세는 모듈로 일정 횟수 이상 실패 시 접근을 차단과 관리
pam_limits.so	/etc/security/limits.conf 시스템의 자원에 대한 사용자 제한을 설정할 때 주로 사용되는 모듈 시스템 로그인 전/후에 수행되는 session 인터페이스와 많이 사용
pam_nologin.so	/etc/nologin 파일이 존재하면 root만 로그인 가능하고, 일반 사용자 로그 인 불가능하며, root로 로그인 시 nologin 파일의 내용이 보여진다.
pam_rootok.so	UID가 0인 사용자를 인증하는 모듈로 보통 root가 암호 입력 없이 해당 서 비스에 대한 접근을 허용할 때 주로 사용됨.
pam_securetty.so	/etc/securetty 접속하는 계정이 root인 경우 /etc/securetty 파일에 기록된 터미널만 허용 나머지 사용자의 경우에는 항상 "성공"한 것으로 처리
pam_time.so	/etc/security/time.conf 시간, 사용자, 그룹, 터미널, 쉘 등으로 접근을 제어
pam_wheel.so	su와 관련된 /etc/pam.d/su에서 사용 특정 그룹에 속하지 않은 사용자는 root로의 접근을 거부, deny 옵션을 사용하여 특정 그룹만 접근을 거부할 수 있게 설정 가능
pam_cracklib.so	입력 받은 암호를 /usr/lib/cracklib_dict에 있는 디렉토리와 비교 새로운 암호를 /etc/security/opasswd에 저장되어 있는 이전 암호목록과 비교 이전 암호와 비슷한지 확인.
pam_succed_if.so	주어진 조건이 참일 경우 성공 값 반환, 인자로 quiet가 들어갈 경우 syslog에 알리지 않는다.

arguments	설명
debug	시스템 로그에 디버깅 정보를 남김
no_warn	응용 프로그램에 경고 메시지를 제공하지 않음
use_first_pass	비밀번호를 두 번 확인하지 않음 대신 auth 모듈에서 입력한 비밀번호를 사용자 인증 과정 시에도 재사용 해야함 (이 옵션은 auth 및 password 모듈에 해당하는 옵션임)
try_first_pass	이 옵션은 use_first_pass 옵션과 비슷 사용자는 두 번 비밀번호를 입력할 필요가 없음 기존의 비밀번호를 다시 입력
use_mapped_pass	이전 모듈에서 입력된 텍스트 인증 토큰을 입력 받도록 함 이 값으로 암호화 또는 암호화가 해제된 키 값을 생성 모듈에 대한 인증 토큰 값을 안전하게 저장하거나 불러오기 위함
expose_account	이 값은 모듈로 하여금 계정 정보를 중요하다고 판단하지 않게 함 시스템 관리자에 의해 임의로 설정한 것이라 여겨짐
nullok	이 인자는 호출된 PAM 모듈이 null 값의 비밀번호를 입력하는 것을 허용

실습 1. Telnet Remote Access 제어

❶ telnet으로 root 로그인 금지



#touch /etc/securetty

#vi /etc/pam.d/remote

auth require pam_securetty.so

auth include password-auth

```
[root@localhost pam.d]# cat remote -nu
     1  #%PAM-1.0
     2  auth       required     pam_securetty.so
     3  auth       substack     password-auth
     4  auth       include      postlogin
     5  account    required     pam_nologin.so
     6  account    include      password-auth
```

Root 계정이 접근 할 수 있는 터미널이 /etc/securetty에 있는지 없는지 여부 확인

root 계정이 접근할 수 있는 터미널이 /etc/securtty에 있다면 성공값을 반환하여 3번 라인 실행

→ 3번 라인에 Password-auth을 실행하여 password가 root 계정에 일치하는 암호인지 확인

→ 암호가 일치하면 성공값을 반환되어 로그인 성공

root 계정이 접근할 수 있는 터미널이 /etc/securetty에 없다면 실패값을 반환

→ 실패값을 반환하지만 require이기 때문에 3번 라인을 수행하여 패스워드를 물어보는 작업 수행

→ 3번라인에서 암호를 비교하는데 이 값이 성공값/실패값 어느 것이라도 2번라인에서 실패 했기 때문에 로그인을 할 수 없음

❷ Root로 접속을 가능하게 하기 위해서는 pts/1, pts/2를 /etc/securetty에 추가 저장

(pts 는 원격 가상 터미널)

```
[root@localhost pam.d]# cat /etc/securetty
pts/1
pts/2
[root@localhost pam.d]# head -4 /etc/pam.d/remote
#%PAM-1.0
auth       required     pam_securetty.so
auth       substack     password-auth
auth       include      postlogin
```

required	ⓐ /etc/securetty에 설정된 터미널 접속이면 성공 값 반환 → 접근 성공 ⓑ /etc/securetty에 설정되지 않은 터미널이면 실패 값 반환 실패 값이 반환 시 아래 라인을 실행하지만 접근 실패
password-auth	/etc/pam.d/password-auth

#cat /etc/pam.d/password-auth

```
[root@localhost pam.d]# cat password-auth
# Generated by authselect on Tue Nov 23 21:20:58 2021
# Do not modify this file manually.

auth   required                                  pam_env.so
auth   required                                  pam_faildelay.so delay=2000000
auth   [default=1 ignore=ignore success=ok]  pam_succeed_if.so uid >= 1000 quiet
auth   [default=1 ignore=ignore success=ok]  pam_localuser.so
auth   sufficient                                pam_unix.so nullok try_first_pass
auth   requisite                                 pam_succeed_if.so uid >= 1000 quiet_success
auth   sufficient                                pam_sss.so forward_pass
auth   required                                  pam_deny.so
```

실습 2. SSH 접속 시 root 계정 로그인 제한

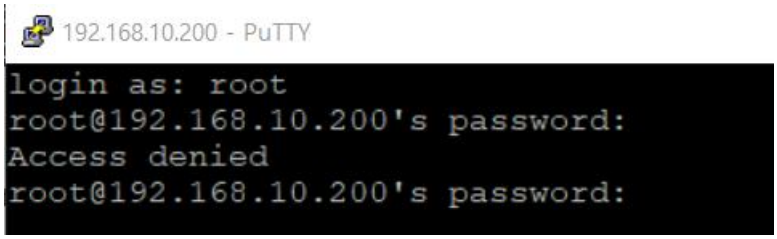
- ssh 설치 시 기본적으로 root 계정으로 로그인 허용
- root 로그인 제한은 /etc/ssh/sshd_config 파일에서 설정

#vi /etc/ssh/sshd_conf

PermitRootLogin yes ➔ PermitRootLogin no 로 변경

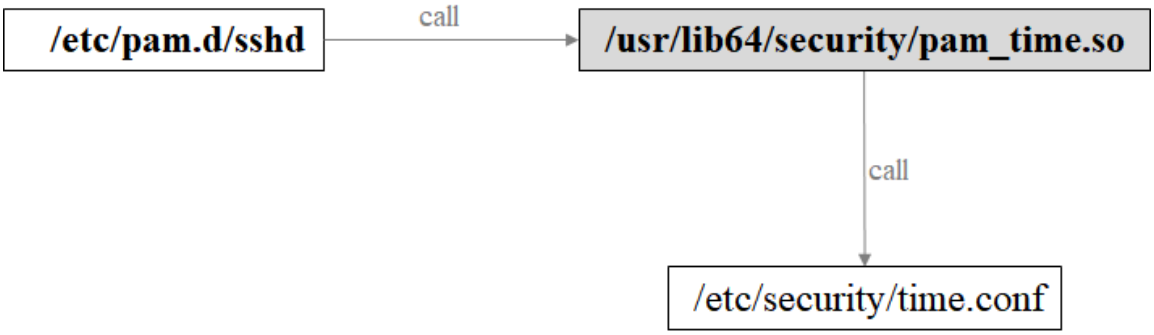
#systemclt restart sshd

```
[root@localhost pam.d]# vi /etc/ssh/sshd_config
[root@localhost pam.d]# cat /etc/ssh/sshd_config -nu | grep PermitRootLogin
    46  PermitRootLogin no
[root@localhost pam.d]# systemctl  restart sshd
```



실습 3. 특정 사용자의 ssh 로그인 시간 제한

평일 : AM 9시 30분 ~ PM 18시 30분 로그인 가능(주말 : 로그인 불가)



❶ #vi /etc/pam.d/sshd

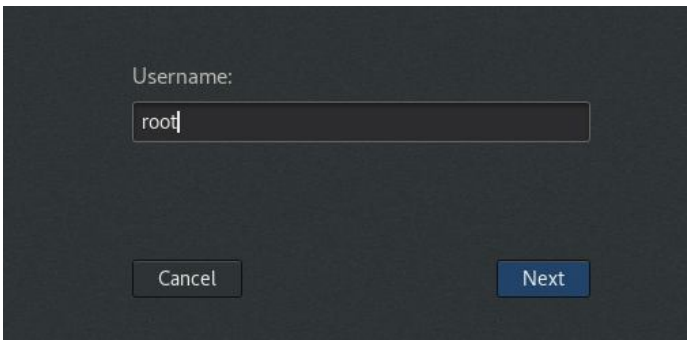
account required pam_time.so

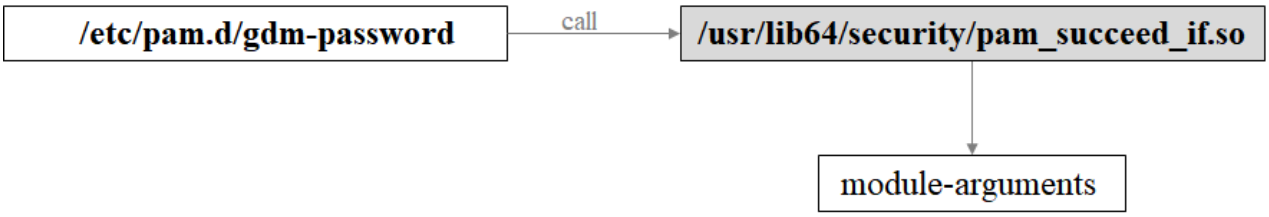
```
[root@localhost pam.d]# cat /etc/pam.d/sshd
#%PAM-1.0
account    required     pam_time.so
auth       substack     password-auth
auth       include      postlogin
```

❷ #vi /etc/security/time.conf

sshd;*;root;!WK0900-17:00

실습 4. Gnome 루트 사용자 로그인 금지





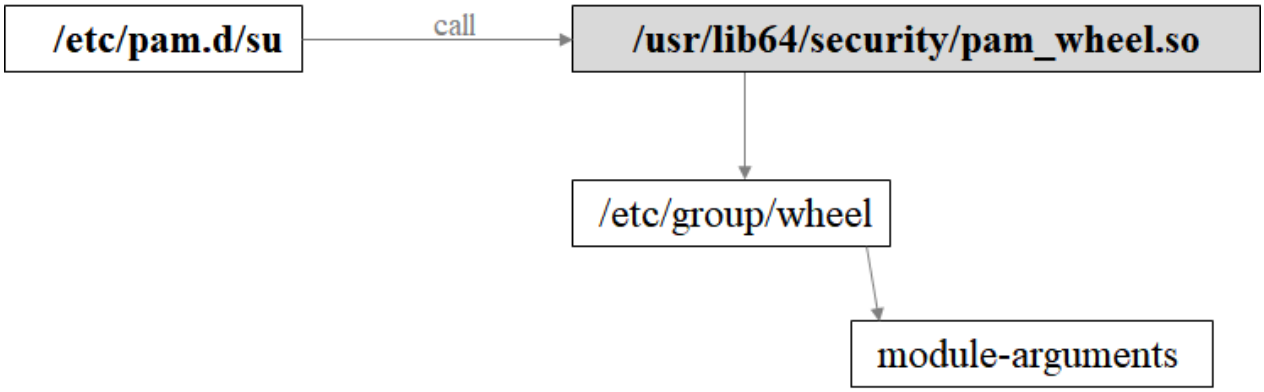
- UID가 1000 이상인 사용자만 로그인 가능

  #vi /etc/pam.d/gdm-password

  auth required pam_succeed_if.so ***uid >=1000***

  module-arguments

실습 5. su 명령어 사용자 제한



❶ wheel 그룹에 포함되어 있는 사용자들이 su 명령어 사용 시 패스워드 없이 로그인 가능

#cat /etc/group | grep wheel

#vi /etc/pam.d/su

auth **sufficient** pam_wheel.so trust use_uid

```
#%PAM-1.0
#auth            required        pam_env.so
auth            sufficient      pam_wheel.so trust use_uid
auth            sufficient      pam_rootok.so
```

[Test 결과]

```
[root@localhost pam.d]# su sora
[sora@localhost pam.d]$ su root
[root@localhost pam.d]# su test
[test@localhost pam.d]$ su test
Password:
su: Authentication failure
[test@localhost pam.d]$ su test
Password:
su: Authentication failure
[test@localhost pam.d]$ exit
exit
[root@localhost pam.d]#
```

* wheel 그룹에 속한 사용자는 신뢰할 수 있는 사용자로 지정

Wheel 그룹 사용들이 su 명령어 사용시 패스워드를 물어보지 않게 설정

* sufficient : 해당 조건이 만족하면 아래 내용들은 확인 하는 과정을 걸치지 않음

❷ wheel에 포함되어 있는 일반 사용자라도 su 명령어를 시 패스워드 입력 요구

#vi /etc/pam.d/su

auth **required** pam_wheel.so use_uid

```
#%PAM-1.0
#auth            required        pam_env.so
auth             required        pam_wheel.so  use_uid
auth             sufficient      pam_rootok.so
```

[Test 결과]

```
[root@localhost pam.d]# su sora
[sora@localhost pam.d]$ su root
Password:
[root@localhost pam.d]# su test
[test@localhost pam.d]$ su root
Password:
su: Permission denied
[test@localhost pam.d]$
```

trust 가 있으면 패스워드를 물어보지 않음/ trust 가 없으면 패스워드를 물어봄

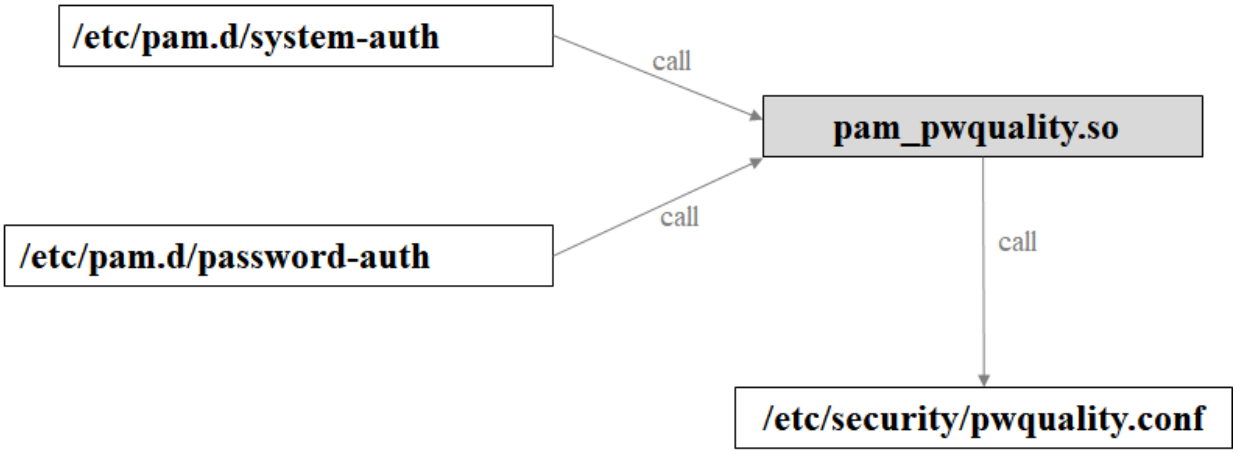
* wheel 그룹에 속한 사용자만 성공값을 반환하게 되어 su 명령어를 사용하여 로그인이 가능 (지정된 사용자만 su 명령어를 사용)

* 특정 사용자를 wheel 그룹에 포함

usermod -G 10 [사용자명]

실습 6. 패스워드 복잡도 설정

- 리눅스 6 이상버전부터는 system-auth, password-auth를 분리하여 관리
- 리눅스 6 이상버전부터는 pam_tally.so, pam_tally2.so 둘다 적용가능
- CentOS8/RHEL8에서는 'pam pwquality' 모듈을 이용하여 패스워드 정책 적용
  (pam_cracklib 대신에 pam_pwquality 모듈이 적용)
- Pam pwquality 모듈은 CentOS에 기본적으로 설치되어 있음 (#dnf install libpwquality)



/etc/pam.d/system-auth	로컬 로그인 (콘솔 로그인) 일반 계정에서 root 계정 전환 시(su)
/etc/pam.d/password-auth	원격 로그인 (SSHD)

 # vi /etc/pam.d/system-auth

password requisite pam_pwquality.so try_first_pass local_users_only

```
password    requisite     pam_pwquality.so try_first_pass local_users_only
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password    sufficient    pam_sss.so use_authtok
password    required      pam_deny.so
```

 # vi /etc/pam.d/password-auth

password requisite pam_pwquality.so try_first_pass local_users_only

```
password    requisite     pam_pwquality.so try_first_pass local_users_only
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password    sufficient    pam_sss.so use_authtok
password    required      pam_deny.so
```

vi /etc/security/pwquality.conf

minlen = 8	패스워드 최소 길이 설정
minclass=2	패스워드 설정 시 대문자, 소문자, 숫자, 특수문자 조합 수 설정
maxrepeat = 2	동일 문자 사용 최대 허용 문자 수 설정 (예 aaa,111 형태를 사용 못함)
maxclassrepeat = 4	동일한 타입(대소문자, 숫자, 특수문자)의 최대 허용 연속 문자 수 설정 (예 abc,123 형태를 사용 못함)
lcredit = -1	소문자 최소 사용 수 설정
ucredit = -1	대문자 최소 사용 수 설정
dcredit = -1	숫자 최소 사용 수 설정
ocredit = -1	특수문자 최소 사용 수 설정
usercheck = 1	패스워드에 유저 ID가 포함되어 있는지 점검 [1 권장/0 체크 안 함] (예 ID:gildong PASS:gildong123 형태를 사용 못함)
maxsequence = 2	단조로운 문자열 허용 최대 길이 설정
difok = 5	이전에 사용하던 패스워드에서 사용한 문자 허용 금지 수 설정
badwords	패스워드에 사용하지 않을 단어 리스트 작성 (예시 : test, toor, admin, 회사이름 등)
dicpath	지정된 디렉터리에 있는 파일의 내용이 패스워드로 사용하지 못함

[참고]

password requisite pam_cracklib.so try_first_pass retry=3 type= minlen=8 lcredit=-1 dcredit=-1 ocredit=-1

pam_cracklib.so : Dictionary에 등록된 단어를 이용한 password 설정 등을 방지하기 위한 비교 및 검사를 수행

retry=3 : 새로운 password를 설정 시 횟수 지정, 변경하고자 하는 password가 기준에 미달(복잡도, 최소길이 등) 할 경우 몇 번의 입력을 추가로 허용할지를 설정

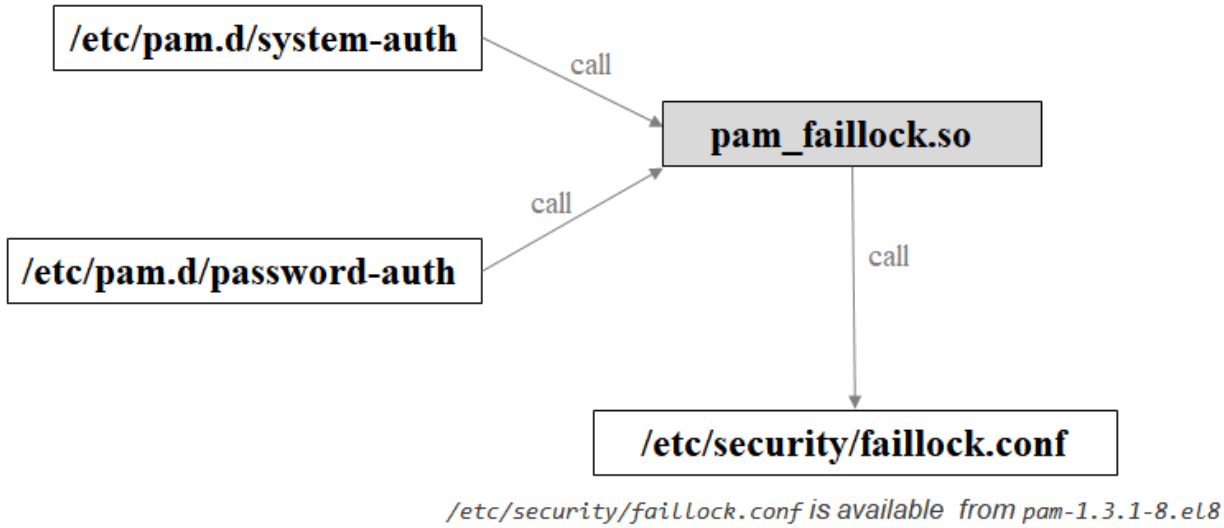
type= : 새로운 password를 입력할 때 화면에 표기되는 문구를 지정, 없으면 기본값.

dcredit=-1 : 숫자가 갖는 기본 크레딧 값을 지정, 기본값은 1이며 -1로 설정된 경우 password에 반드시 숫자가 하나 이상 있어야 함

*크레딧(예, ucredit) 값이 -1이 아닌 2로 지정되었다면 password 최소길이가 8글자로 설정되었을 때 대문자 하나는 2의 값을 갖게 됨. (예를 들면 Abcdefg라는 password는 얼핏 보기에 총 길이가 7문자이지만 대문자 A는 2크리딧 값을 갖게 되어, 이 문자열은 8크리딧이 되고 이에 password 최소길이 8글자라는 조건을 충족하게 됨)

실습 7. 로그인 실패 잠금 설정

- CentOS 8/RHEL8부터는 pam_tally2를 기본으로 사용되지 않으며 faillock 이용
- pam_faillock.so 모듈을 이용하며 일정 간격 동안 사용자별로 실패한 인증 시도 목록을 유지 관리
- 인증 실패가 연속적으로 거부될 경우 계정을 잠금



- /etc/security/faillock.conf파일이 없는 경우 system-auth 또는 password-auth부분에 아래 형식과 같이 잠금 설정을 직접 지정

auth ... pam_faillock.so {preauth|authfail|authsucc}
[dir=*/path/to/tally-directory*]
[even_deny_root] [deny=*n*]
[fail_interval=*n*]
[unlock_time=*n*]
[root_unlock_time=*n*]
[audit]
[silent]
[no_log_info]

#vi /etc/pam.d/system-auth

auth required pam_faillock.so preauth silent audit deny=5 unlock_time=600

```
auth        required      pam_env.so
auth        required      pam_faildelay.so delay=2000000
auth        required      pam_faillock.so preauth silent deny=4 unlock_time=1200
```

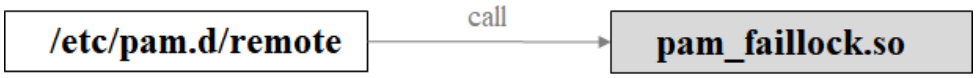
vi /etc/pam.d/password-auth //일반계정자의 접속 금지 설정

auth required pam_faillock.so preauth silent audit deny=5 unlock_time=600
auth [default=die] pam_faillock.so authfail audit deny=5 unlock_time=600

*even_deny_root // ROOT 계정을 잠금 설정 시 추가

auth required pam_faillock.so preauth silent audit eny=5 **even_deny_root** unlock_time=600

실습 8. Telnet 접속 시 계정 잠금 설정



#vi remote

auth required pam_faillock.so deny=3 unlock_time=60

인증을 처리하는데 pam_faillock.so 모듈을 사용하여 실패 3회(deny=3)이면 30초동안 계정을 잠근다 (unlock).

```
[root@localhost pam.d]# cat /etc/securetty
[root@localhost pam.d]# head -4 remote
#%PAM-1.0
auth        required      pam_securetty.so
auth        required      pam_faillock.so deny=3 unlock time=60
auth        substack      password-auth
```

계정 잠금 관련 명령어

- authconfig/authselect(pam_tally/pam_faillock)

현재 인증 설정 확인	authselect current
Faillock 활성화	authselect enable-feature with-faillock
	grep –n faillock /etc/pam.d/system-auth grep –n faillock /etc/pam.d/password-auth
사용자의 로그인 실패 횟수 표시	faillock --user gildong
잠긴 계정을 수동으로 잠금 해제	faillock --user gildong --reset