접근제어, PAM

1. /etc/securetty

- Telnet 접속 시 root 접근 제한 설정 파일
- 루트가 로그인 가능한 터미널(tty) 장치들을 나열해 둔것
- 누군가 로그인 시도 하면 /bin/login이라는 프로세스가 /etc/securetty파일을 참조하여 /etc/securetty파일에 터미널 목록으로 로그인을 시켜줌
- shadow패스워드시스템을 설정하여 사용하고 있는 경우엔 /etc/login.defs에서 root로의 접속가능한 터미널 설정 -> 대부분이 경우라고 함
- 파일 내 pts/x 관련 설정이 존재하는 경우 PAM 모듈 설정과 관계없이 root 계정 접속을 허용하므로 pts/x관련 설정 제거 필요
    - tty(terminal-teletype) : 서버와 연결된 모니터, 키보드 등을 통해 사용자가 콘솔로 직접 로그인함
    - pts(pseudo-terminal, 가상터미널) : Telnet, SSH, 터니멀 등을 이용하여 접속함

2. 접근제어

https://www.igloo.co.kr/security-information/리눅스-pam-모듈의-이해/

- 사용자 로그인관련 명령어(su, who, last)
- PAM 설정 파일을 잘 못 변경하면 자칫 su 명령어를 사용할 수 없게 된다거나 계정이 갑자기 로그인 할 수 없게 될 수 있어 설정에 주의가 필요

2.1 PAM 모듈

- Pluggable Authentication Modules
- https://www.youtube.com/watchv=iNGRQaJDMWM&list=PL0d8NnikouEXVn9FfoX2XVlGgEArLDiLZ&index=8&t=7s (7:00부터 참고)
- 리눅스 시스템에서 사용하는 인증모듈로써 응용 프로그램(서비스)에 대한 사용자의 사용 권한을 제어하는 모듈
- PAM을 사용하기 이전 리눅스 시스템에서는 사용자를 인증하기 위해 각 응용프로그램에서 자체적으로 로직을 구현하여 사용, 특히 시스템에 저장된 사용자 정보를 통해 인증할 경우, 응용프로그램이 사용자 정보가 담긴 주요 시스템 파일(etc/passwd)에 대한 접근 권한을 가지고 있어야 하므로 침해사고의 위험이 존재
- PAM 모듈은 소프트웨어의 개발과 인증 및 안전한 권한 부여 체계를 분리하고자 하는 목적으로 만들어졌기 때문에 이를 통한 인증을 수행할 경우, 응용프로그램에서 직접 인증 로직을 구현하지 않아 개발이 간소화될 뿐만 아니라 passwd 파일 등 시스템 파일을 열람하지 않아도 되는 장점이 있다.

2.2 PAM 동작원리

1. 인증이 필요한 응용프로그램은 더 이상 passwd 파일을 열람하지 않고 'PAM' 모듈에 사용자 인증을 요청한다.
2. PAM은 인증을 요청한 사용자의 정보를 가지고 결과를 도출하여 응용프로그램에 전달한다.

2.3 PAM 구성

- /etc/pam.d 디렉토리 : pam을 적용한 프로그램이나 서비스의 설정 내용을 저장해둔 파일들이 모여 있는 디렉토리

auth	sufficient	pam_rootok.so	
module_type	config_flag	module_path	module_argument

Module_type	PAM이 어떤 타입의 인증을 사용할 것인지를 지정 auth : 사용자 인증에 사용하며 올바른 패스워드인지 아닌지 확인 account : 계정 관리를 수행, 사용자의 위치, 시간, 권한 등을 지정하여 접근을 결정 ex.특정 사용자는 콘솔로만 아침 9시부터 6시까지만 root로 접근 가능 password : 사용자가 패스워드를 변경할 수 있는 모듈 지정 session : 사용자가 인증 받기 전후에 수행되어야 할 작업을 지정 ex. 로그 기록
Config_flag	PAM에서 사용되는 모듈들이 결과에 따라 어떤 동작을 취할 지 결정 required : 이 모듈이 성공값을 반환해야 최종 인증에 성공(실패 시 밑의 다른 설정도 전부 다시 확인)(덜 엄격한 설정) requisite : 이 모듈이 성공값을 반환해야 최종 인증에 성공(실패 시 바로 최종 인증 실패 시)(엄격한 설정) sufficient : 이 모듈이 성공값을 반환하면 바로 인증 성공 optional : 선택사항, 일반적으로 무시, 다른 플래그에 의한 최종 결과가 불분명할 시 optional 값이 적용 include : 다른 설정 파일을 불러옴
Module_path	/usr/lib64/security 디렉토리 내의 어떤 모듈을 사용할 지 지정 모듈 이름을 지정(→알아서 모듈이름 찾음)
Module_argument	모듈에 전달되는 매개변수 값을 나타냄 debug : 시스템 로그 파일에 디버그 정보를 남기게 함 no_warn : 모듈이 경고 메시지를 보내지 않게 함

2.3.1 PAM의 기본 구조

- 기본구조

Module Type	Control Flag	Module Name	Module Arguments

- 기본 구조의 예 : /etc/pam.d/su

Module Type	Control Flag	Module Name 및 Module Arguments
auth	sufficient	pam_rootok.so
auth	required	pam_wheel.so debug group-wheel
auth	required	pam_wheel.so use_uid
auth	include	system-auth
account	sufficient	pam_succeed_if.so uid=0 use_id
account	include	system-auth

- Module Type
  - 모듈 타입 필드는 PAM에 어떤 종류의 인증을 사용할 것인가를 지정하는 필드로 아래와 같이 4종류의 타입을 설정할 수 있다.

auth	사용자에게 비밀번호를 요청하고 입력 받은 정보가 맞는지 검사하는 모듈
account	명시된 계정이 현재 조건에서 유효한 인증 목표 인지 검사하는 것으로 계정에 대한 접근 통제 및 계정 정책 관리하는 모듈
password	사용자가 인증 정보(password)를 변경할 수 있도록 비밀번호 갱신을 관장하는 모듈

session	사용자가 인증을 받기 전 /후에 수행해야 할 일을 정의하는 모듈

- Control Flag
  - PAM에서 사용되는 모듈들이 결과에 따라 어떤 동작을 해야하는지 결정하는 필드이다.
  - 5개의 Control Flag 중 “Required”의 경우, 해당 모듈의 결과와 상관없이 다음 모듈을 실행시킨다. 심지어 다음에 실행된 모듈의 결과보다 더 높은 우선순위를 가지므로 “Required”에서 실패가 되면 최종 인증 결과는 실패가 된다는 사실에 주의해야 한다.

requisite	인증 결과와 실패일 경우, 인증 종료 - 인증 결과가 **성공**일 경우, 다음 인증 모듈 실행**(최종 인증 결과에 미반영)** - 인증 결과가 **실패**일 경우, **즉시 인증 실패를 반환**
required	인증 결과와 관계없이 다음 인증 실행 - 인증 결과가 **성공**일 경우, **최종 인증 결과는 무조건 성공** - 인증 결과가 **실패**일 경우, **최종 인증 결과는 무조건 실패**
sufficient	인증 결과와 성공일 경우, 인증 종료 - 인증 결과가 **성공**일 경우, **즉시 인증 성공을 반환** - 인증 결과가 **실패**일 경우, 다음 인증 모듈 실행**(최종 인증 결과에 미반영)**
optional	일반적으로 최종 인증 결과에 반영되지 않음 단, 다른 인증 모듈의 명확한 성공/실패가 없다면 이 모듈의 결과를 반환
include	다른 PAM 설정 파일 호출

- Module Name
  - 이 필드는 사용하고자 하는 모듈의 경로와 이름을 지정하는 필드이며 PAM 모듈은 대부분 /lib/security 또는 /etc/pam.d 디렉터리에 위치한다. 다음은 주요 모듈에 대한 설명이다.

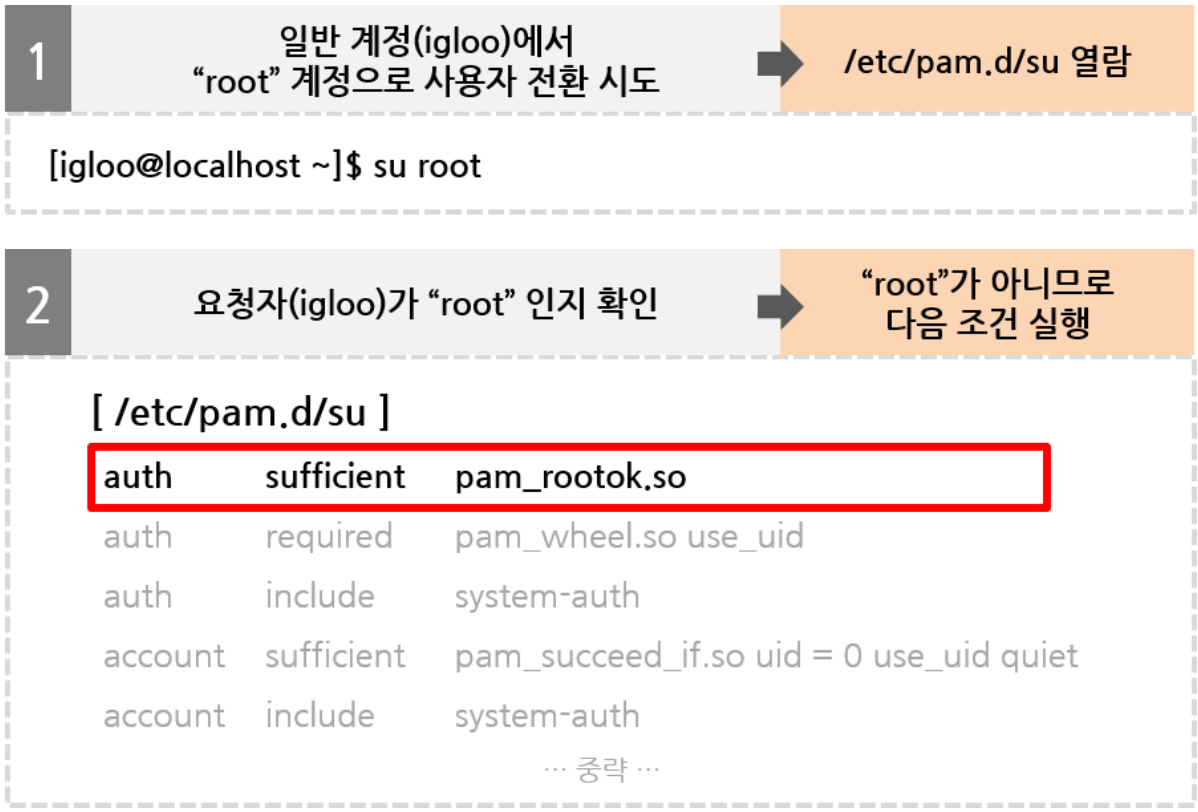
pam_rootok	root 계정인 경우, 추가 인증 없이 무조건 허용하는 모듈
pam_wheel.so	SU명령어 사용 인증에 사용되며 특정 그룹(wheel)에 대한 인증 제어하는 모듈
pam_ succeed_if.so	인수로 주어진 조건에 따라 인증을 제어하는 모듈
pam_securetty. so	root 계정인 경우에만 적용되는 모듈로써 /etc/securetty 파일을 참고하여 해당 파일에 root가 있으면 특정 서비스에 대한 root 접근을 허용하는 모듈. (root 이외의 계정일 경우, 항상 인증 성공값을 반환)

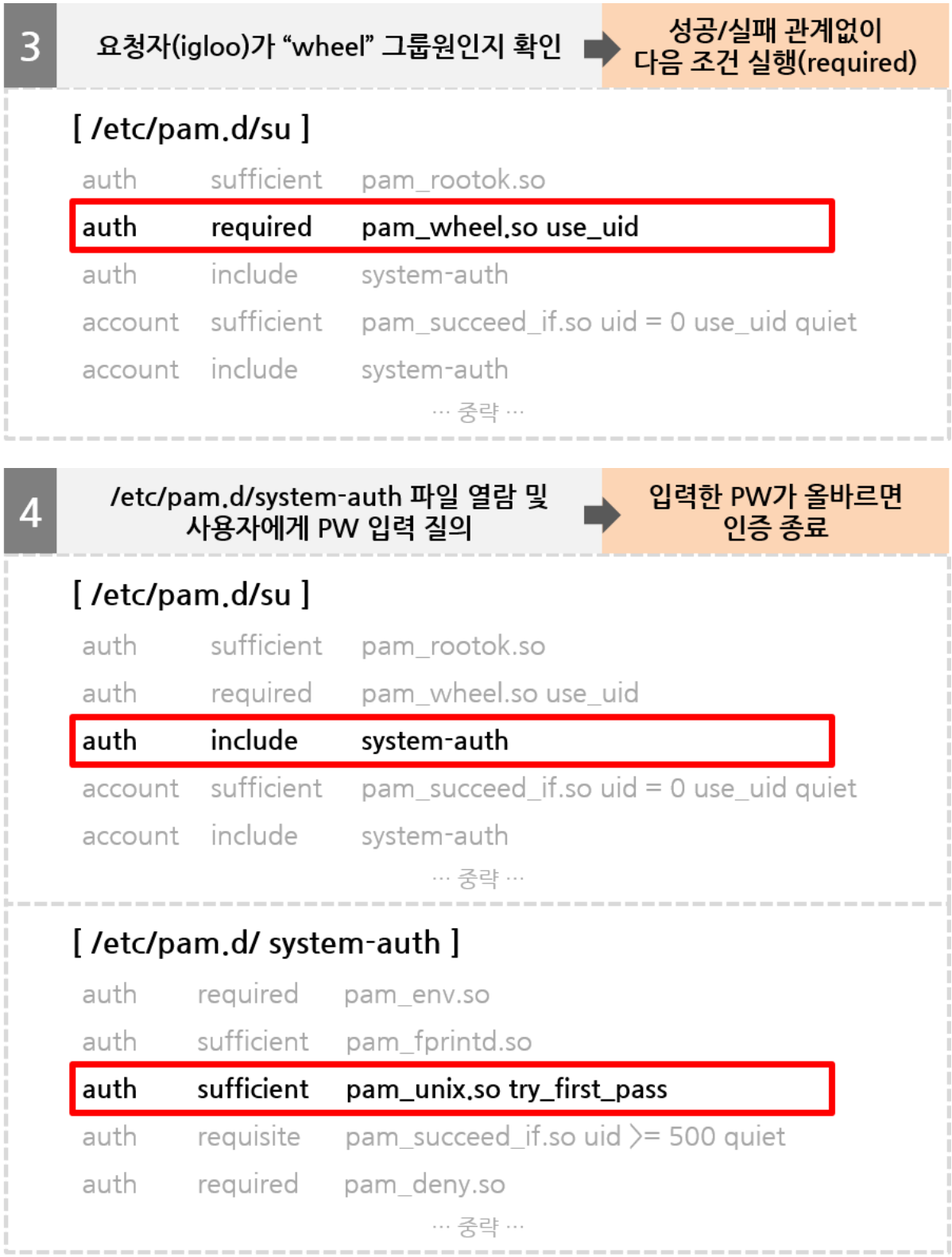
- Module Arguments

- 모듈-인수는 모듈에게 전달되는 인수를 나타낸다. 각각의 모듈은 각각의 인수를 가지고 있다. 모듈마다 인수가 필요할 수도 필요 없을 수도 있다.

Debug	시스템 로그 파일에 디버그 정보를 남기도록 지정
No_warn	모듈이 경고 메시지를 보내지 않도록 지정
Use_first_pass	사용자에게 password 입력을 요구하지 않도록 지정하는 인수로 이전 모듈에서 입력 받은 password가 존재하지 않을 경우, 인증 실패 반환
Try_first_pass	이전 모듈에서 입력 받은 password로 인증 시도하며, 이전에 입력받은 password가 존재하지 않을 경우, 사용자에게 입력 요구

2.3.2 PAM을 통한 인증과정 상세





3.1 root 비밀번호 초기화?

```
# root 로그인 시도 횟수확인
pam_tally2 --user root -r

# 비번 틀린 히스토리 초기화
# 1. /sbin/pam_tally2 --user 계정명 --reset
# 2./etc/pam.d/system-auth 파일의 deny, unlocking_time 설정
```

3.1.2 보안 취약점 - 계정 잠금 임계값 설정

- 시스템 정책에 사용자 로그인 실패 임계값이 설정되어 있는지 점검합니다. 무작위 대입 공격 등으로 시스템에 로그인 시도에 대한 차단을 위하여 임계값 설정을 하는 것을 권고
- deny 값이 5 이하 (금융감독원) 3 이하
- auth required pam_tally2.so onerr=fail deny=5 unlock_time=120 reset

```
#Step 1) /etc/pam.d/system-auth 파일 수정
# 설정 추가 또는 수정
 auth required /lib/security/pam_tally.so deny=5 unlock_time=120 no_magic_root
 account required /lib/security/pam_tally.so no_magic_root reset

# 참고항목
# deny=5 : 5회 입력 실패 시 패스워드 잠금
# no_magic_root : root에게는 패스워드 잠금 설정을 적용하지 않음
# unlock_time : 계정 잠김 후 마지막 계정 실패 시간부터 설정된 시간이 지나면 자동 계정 잠김 해제 (단위: 초)
# reset : 접속 시도 성공 시 실패한 횟수 초기화
```