Linux PAM 모듈을 이용한 보안장비 인증제어 체계의 보완 기술

박재필O, 홍세영*
*(주)시큐위즈 기술연구소,
O(주)시큐위즈 기술연구소
e-mail: foxyfeel@secuwiz.co.krO, hsykwb@naver.com*

AN Implement Enhanced Authentication system Using Linux PAM Module

Jae-Pil ParkO, se-young Hong*
*SECUWIZ CO. tech leb,
OSECUWIZ CO. tech leb

● 요 약 ●

본 논문에서는 네트워크 보안 장비의 OS로 사용되는 리눅스 시스템의 인증제어 체계인 PAM Module을 이용하여 강화된 보안장비들의 인증제도에 대응할 수 있는 기능들을 개발하며 이를 이용하여 최근 급격히 증가하고 있는 보안장비 해킹 공격에 효과적으로 대비할 수 있도록 한다.

키워드: 리눅스 시스템 보안, 네트워크 보안, linux authentication

I. Introduction

현재 국내외 적으로 네트워크 보안장비자체를 해킹하는 공격이 급격히 증가하고 있는 추세이다. 특히 내부 시스템에 접근할 수 있는 다리 역할을 하는 보안장비들(VPN, 망분리 장비, 망연계 장비, ...)에 대한 공격이 주를 이루고 있으며 이에 대응 하여 국내 보안장비의 인증제도역시 강화 되고 있는 추세이다.

하지만 국내 보안장비의 주 OS로 이용되고 있는 리눅스 시스템들에 탑제되어있는 인증 및 로깅(logging)기능들이 강화된 보안인증제도에 부합하지 못하는 것이 현실이다. 본 논문에서는 이처럼 강화된 보안 인증제도에 대응할 수 있는 기능들을 리눅스 PAM 모듈을 이용하여 구현해 보고자 한다.

II. Preliminaries

1. Related works

1.1 국내 동향

최근 기존에 활발하게 활동하는 해킹그룹들의 공격 추세를 살펴보면 전통적인 공격방법들인 해킹메일 기법이나 관련자의 계정 및 PC를 공격하는 방법에서 이러한 네트워크공격들을 방지하기 위해 도입되어 있는 네트워크 보안 장비 자체를 공격하는 방식으로 변화되고 있다.

특히 내부 시스템에 접근할 수 있는 다리 역할을 하는 VPN[1], 망연계, 망분리, VDI[2] 등의 장비들을 특정하고 이들 OS의 취약점을 이용하는 공격 방법들이 급증하고 있다.

이에 보안장비들의 인증을 관장하는 기관에서는 보안장비 자체 인증 및 접속에대한 인증제도의 강화와 이를 모니터링 할 수 있는 기능들을 보완, 강화하도록 인증제도 자체를 강화 하고 있으며 본 논문에서는 이를 위하여 리눅스 PAM모듈을 이용하여 이에 부합하는 기능을 구현해 보고자 한다.

III. The Proposed Scheme

가. 리눅스 PAM모듈

1) 개요

서버 취약점 진단을 받게되면 종종 "계정 잠금 임계값 설정","SU 명령어 사용 제한" 등의 항목에서 "PAM 모듈을 이용하여 취약점을 조치하세요."라는 안내를 받게된다. 그러나 막상 PAM 설정 파일을 열어 가이드에 명시되어 있는 대로 변경했다가는 낭패를 보기 십상이다. PAM 설정은 단순히 한 줄 수정한다고 해서 되는 것이 아니라 위/아래 줄에 선언된 설정에 의해 그 결과값이 달라지기 때문에 자칫하다간 SU 명령어를 사용할 수 없게 되거나, 로그인 잘 되던 계정이 갑자기 로그인할 수 없게 될 수 있어 설정에 주의가 필요하다.

2) PAM(Pluggable Authentication Module)이란?

PAM은 리눅스 시스템에서 사용하는 '인증 모듈(Pluggable Authentication Modules)'로써 응용 프로그램(서비스)에 대한 사용자의 사용 권한을 제어하는 모듈이다.

PAM을 사용하기 이전 리눅스 시스템에서는 사용자를 인증하기 위해 각 응용프로그램에서 자체적으로 로직을 구현하여 사용했다. 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 존재했다. 그 뿐만 아니라 응용프로그램마다 사용자 인증 방식이 상이하여 관리하기에 많은 어려움이 따랐다.

이를 해결하기 위해 PAM이 등장하게 되었으며, PAM의 동작 원리는 다음과 같다.

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.

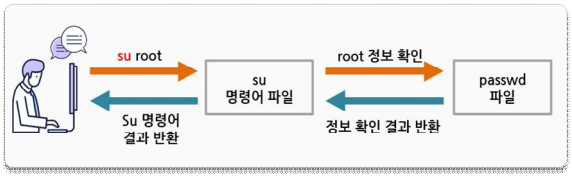


Fig. 1. 응용프로그램 자체적으로 사용자 인증하는 과정

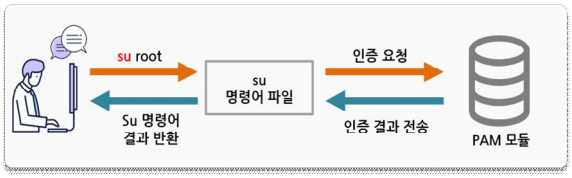


Fig. 2. PAM 모듈을 통한 사용자 인증 과정

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. 무엇보다도 가장 큰 장점은 시스템 운영자가 응용프로그램의 인증 동작을 제어할 수 있어 더욱 안전하게 시스템을 운영할 수 있다.

3) PAM 구성

3.1) PAM 기본구조

*기본구조

Module type	Control Flag	Module Name	Module Arguments

*기본구조의 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_uid
account	include	system-auth

3.2) 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

3.3) Control Flag

PAM에서 사용되는 모듈들이 결과에 따라 어떤 동작을 해야하는지 결정하는 필드이다.

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

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.

3.4) 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)

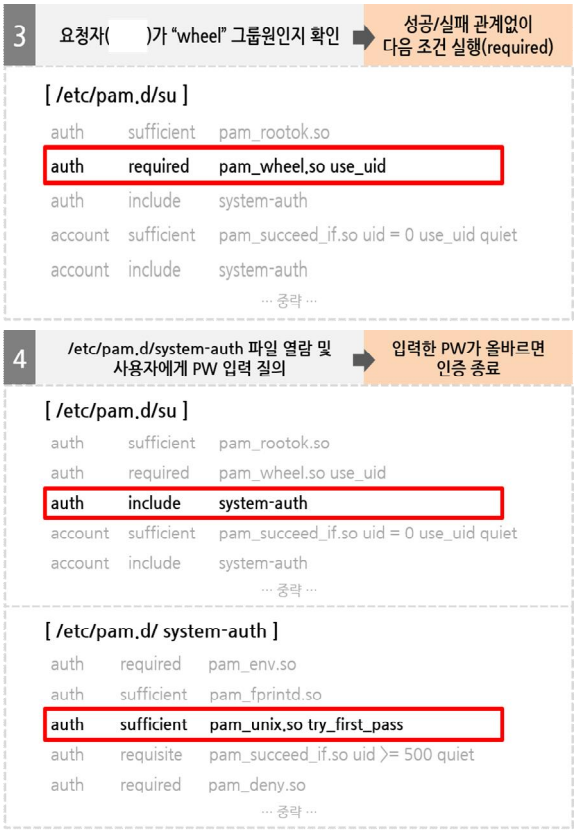
3.5) Module Arguments

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

4) PAM을 통한 인증과정 상세

그럼 이제 PAM을 통한 인증 과정을 상세하게 분석 해보자. 여기서는 “su” 명령어 응용프로그램을 기준으로 분석해보고자 한다.



※ *try_first_pass*

이전 모듈에서 입력받은 패스워드가 있을 경우, 해당 패스워드로 먼저 인증 시도한다. 이전에 입력받은 패스워드가 없을 경우, 패스워드 입력을 요청한다.

위의 과정 중 가장 주목해야 하는 단계는 바로 "3" 단계이다. "3" 단계에서는 "required" flag를 사용하여 모듈을 실행했기 때문에 인증 요청자가 wheel 그룹에 속하지 않아도 다음 인증 모듈이 실행된다. 따라서 wheel 그룹원뿐만 아니라 "모든" 인증 요청자가 "5" 단계까지 인증 과정을 거치게 되며 모두 동일하게 password 입력 요청까지 받게 되는 것이다.

여기서 앞서 말한 ″required″ flag의 진면목이 나타난다. 인증 요청자는 wheel 그룹에 속하지 않아서 인증 실패가 났음에도 불구하고 password 입력 요청을 받았기 때문에 단순히 password 입력 오류라고 생각할 수 있다. 이는 에러 발생 원인이 직접적으로 표출되지 않기 때문에 보안상 매우 좋다는 장점을 가진다.

5) PAM모듈과 보안장비 기능 연동

이제 PAM 모듈과 보안장비 기능을 연동할 수 있도록 연동 모듈을 준비한다. 연동 모듈은 보안장비의 설정을 저장 하고 있는 DB연동 모듈과 설정값을 암호화 및 복호화, hash, encoding등을 처리할 수 있는 보안 모듈로 분리하여 구현 할 수 있도록 한다. 특히 보안장비 인증을 위해서는 암호화시 키관리를 처리할 수 있는 표준들(예:pbkdf2)을 사용해야 하므로 이를 연동 하는 모듈로서 구현토록 한다. PAM 모듈은 자체적으로 기능별 모듈로 나누어져 구성되어있으

며 이들은 각각 shared library형태로 빌드되어 동작 되게 된다.

이중 보안장비인증에 직결 되는 모듈로는 pam_access, pam_tally2, pam_unix, pam_xauth, pam_limit가 있다.

5.1) pam_tally2

ssh 및 보안장비 admin의 연속 인증횟수 초과 알림, 문자메시지, 이메일통보 기능을 연동한다. 인증횟수, 인증실패에 대한 감사기록을 생성한다.

5.2) pam_limits

동일 계정의 중복접속 차단 및 그 로깅을 연동한다. 중복 접속시 감사기록을 생성한다.

5.3) pam_unix

로그인/로그아웃 감사기록을 생성한다.

5.4) pam_access

IP인증에 대한 감사기록을 생성한다.

IV. Conclusions

본연구의 목표는 국내 네트워크 보안 인증제도의 강화에 따른 보안장비의 사용자 인증기능 강화와 감사기록의 생성, 보관에 대해 리눅스를 OS로 사용하는 보안장비들에 효율적으로 적용할 수 있는 기능들을 개발하는것을 목적으로한다. 이는 또한 최근 급증하고 있는 네트워크보안장비 자체 해킹공격의 효과적인 대응 방법이 될것이라고 판단된다.

REFERENCES

[1] Virtual Private Network
[2] Virtual Desktop Infrastructure