The Linux-PAM System Administrators' Guide

MorganAndrew G.

morgan (at) kernel.org

도운이

PAM team(안효성, 정경식, 허준영)

SPsoft (http://www.spsoft.co.kr)

pam (at) spsoft.co.kr

이 문서는 시스템관리자가 **Linux-PAM** 라이브러리에 대해서 알아야 할 것들을 다루고 있다. 여기서는 PAM 설정파일의 올바른 형식과 보안상으로 안전한 시스템을 유지하기 위한 전략을 이야기하고 있다.

DRAFT v0.75 2001/03/18

1. 소개(Introduction)

Linux-PAM(Linux용 착탈형 인증 모듈)은 시스템 관리자가 응용프로그램들이 사용자를 인증하는 방법을 선택할수 있도록 해주는 공유 라이브러리 묶음이다.

즉, PAM을 사용하는 응용프로그램을 재컴파일(재작성)하지 않고, 인증 방법을 변경할 수 있다는 것이다. 정말로, 누군가 응용프로그램 자체를 손대지 않고 인증시스템을 몽땅 업그레이드할 수 있을지도 모른다.

전통적으로 사용자 인증을 필요로 하는 응용프로그램은 특정 인증 방법을 사용하기 위해서는 컴파일을 해야만 했었다. 예를 들어, 전통적인 UN*X 시스템의 경우에, 사용자에 대한 신원확인을 올바른 패스워드를 입력하게 하여 확인하였다. 이 패스워드의 앞에 "salt"라는 두 글자를 붙여 암호화(crypt(3))된다. 사용자는 이 암호화된 패스워드가 시스템 패스워드 데이터베이스 (**/etc/passwd**)의 해당 사용자의 데이터의 두번째 필드와 일치할 경우 인증을 받게 된다. 이러한 시스템들에서, 만약 전부가 아니라면 대부분의 권한들은 이러한 하나의 인증 방식만으로 부여받게 된다. 권한은 사용자의 ID(uid)와 여러 그룹의 멤버쉽의 형태로 받게 된다. 서비스와 응용프로그램들은 사용자의 **UID**와 그룹 ID에 따라 사용여부가 결정된다. 전통적으로 그룹 멤버쉽은 **/etc/group**파일의 내용에 따라 부여받는다.

안타깝게도 컴퓨터의 속도가 향상되고 네트워크에 분산해서 연산을 하는 기법이 널리 사용되면서, 과거에 한때 안전하게 여겨졌던 이와 같은 인증 방법은 약점이 노출되었다. 이러한 현실에서 한가닥 희망으로 새로운 인증 방법이 꾸준히 계발되고 있다.

Linux-PAM 프로젝트의 목적은 권한을 부여하는 소프트웨어의 개발과 안전하고 적절한 인증안의 개발을 분리하려는데에 있다. 이는 응용프로그램이 사용자 인증을 처리하기 위해 사용될 함수의 라이브러리를 제공함으로써 가능하다. PAM라이브러리는 **/etc/pam.conf**(또는 **/etc/pam.d/**에 있는 여러 파일들)에서 각 시스템에 맞게 설정을 하여, 각 시스템에서 사용가능한 인증 모듈을 통해 사용자의 인증 요구를 처리한다. 모듈 자체는 일반적으로 **/usr/lib/security**에 위치해 있고, 동적으로 로드가능한 오브젝트 파일 (see **dlopen(3)**)의 형태를 갖는다.

2. 이 문서에 대한 약간의 주해(Some comments on the text)

이 문서 나머지를 계속 읽기 전에, 이 글에서 특정 파일들이 지정된 특정 디렉토리에 위치해 있다고 가정하고 있음을 주지해주기 바란다. 이 파일들의 위치에 대해 우리가 선택한 관례는 해당 RFC(RFC-86.0, see bibliography)에서 지정된 바를 따랐다. 만약 당신이 PAM을 지원하는 Linux배포판(또는 다른 OS)을 사용하지만 이 파일들을 다른 곳에 설치하도록 선택했다면(Red Hat이 이러한 배포판중 하나다.) 이 문서에서 예제를 바로 복사할 경우 주의하길 바란다.

앞의 예를 들면서 분명히 밝혀놓았듯이 이 글에서는 PAM 동적 오브젝트 파일 (모듈)들이 **/usr/lib/security** 디렉토리에 있다고 가정한다. 하지만 Red Hat Linux는 Linux파일 시스템 표준(FSSTND)과의 합의하에 이 파일들을 **/lib/security**에 두었다. 이 문서의 예제를 사용할 때, 주의해서 적절히 변용하기 바란다.

3. 개요

초심자를 위해, 예제를 먼저 살펴보겠다. 사용자에게 어떤 서비스를 제공해주는 응용프로그램을 생각해보자; login이 이런 프로그램중 하나이다. login은 두가지 일을 한다. 하나는 요청한 사용자가 자신이 주장하는 사용자가 맞는지 확인해서 원하는 사용자로 설정해주고 두번째로, 요구한 서비스를 제공해주는 것이다; **login**의 경우 서비스는 그 사용자의 UID로 실행되는 명령shell(**bash, tcsh, zsh,** 등등) 이다.

전통적으로, 앞의 단계는 **login** 프로그램이 사용자에게 패스워드를 물어 보고, 시스템에 있는 것과 일치하는지 확인함으로써 수행된다. 따라서 시스템에 관련된 한도 안에서 사용자가 그가 주장하는 사용자가 맞는지 확인한다. 이 작업을 **Linux-PAM**에게 위임하게 된다.

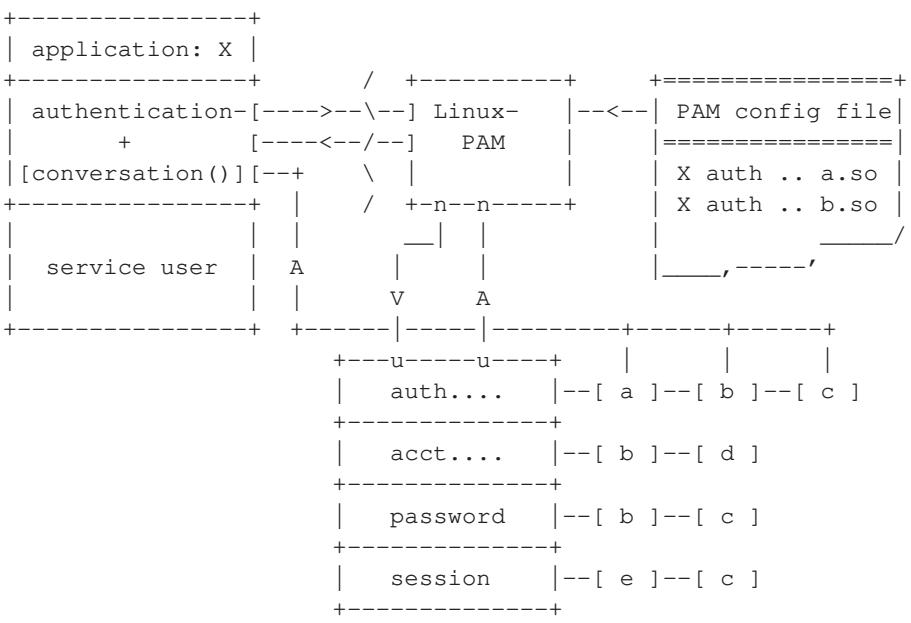
응용프로그램 개발자 입장에서 보면(이 경우에는 login프로그램을 작성하는 사람), **Linux-PAM**이 사용자의 신원을 확인하는 인증과정을 맡게 되는 것이다.

Linux-PAM의 유연성은 시스템 관리자인 당신이 어느 인증안을 사용할 것인지 마음대로 요구할 수 있다는 것이다. 당신은 Linux시스템에서 PAM을 사용하는 어떤 또는 모든 응용프로그램에 대해서도 인증안을 정할 자유를 갖고 있다. 즉, 당신은 단순 신뢰(**pam_permit**, 모두 허용하는 것)같이 순진한 것부터 망막검사, 음성, 그리고 one-time 패스워드를 조합한 것처럼 대단히 의심많은 수준까지 사용해서 인증할 수 있다.

당신이 지금 보고 있는 **Linux-PAM**의 유연성을 보여주기 위해 다음의 상황을 생각해보자: 시스템 관리자(부모)는 그(녀)의 사용자들(아이들)의 수학 능력을 향상시키길 원한다. 그녀는 아이들이 좋아하는 "Shoot'em up game" (물론 PAM을 사용하는 프로그램이어야 한다)을 사용하도록 인증하는데 12보다 작은 임의의 몇몇 숫자들의 곱을 맞추도록 요구하도록 설정할 수 있다. 만약 그 게임이 좋은 점이 있다면 아이들이 곧 구구단을 배울 것이라는 사실은 확실하다. 아이들이 자람에 따라 인증방법도 (긴 숫자의)나눗셈을 포함하도록 업그레이드 할 수 있다.

Linux-PAM은 네가지 종류의 작업을 한다. 이것들은 인증**(authentication)**; 계정**(account)**; 세션**(session)**; 패스워드**(password)**관리작업이다. 선호하는 관리방안과 응용프로그램의 동작의 관련성은 해당하는 **Linux-PAM**설정 파일의 내용으로 정해진다. 관리작업은 설정 파일 내에 지정된 모듈에 의해 수행된다. 이 파일의 문법은 아래에서 다룬다.

다음 그림은 **Linux-PAM**의 전체 구성을 보여준다.



그림의 왼쪽은 X라는 응용프로그램을 나타낸다. **Linux-PAM** 라이브러리와 연결된 이런 응용프로그램은 설정된 인증 방법에 대해 전혀 모르는 상태이다. **Linux-PAM** 라이브러리(그림 가운데)는 PAM설정 파일의 내용을 보고 X라는 응용프로그램에에 맞는 모듈들을 로드한다. 이 모듈들은 네개의 관리 그룹(그림 아래의 가운데)으로 분류되고, 설정파일에 적혀있는 순서대로 처리된다. **Linux-PAM**에 의해 이 모듈들이 호출되면 응용프로그램을 위해 다양한 인증

을 수행한다. 사용자에게 제공되거나 사용자로부터 받아야할 텍스트 정보들은 응용프로그램이 제공하는 **conversation** 함수를 통해 교환된다.

3.1. 시작하기

[아래 글은 Seth Chaiklin이 제공한 것이다.]

여태까지 우리는 모든 응용프로그램들이 적절히 코딩된 이상적인 상황에서 PAM이 어떻게 동작하는지 설명했다. 하지만, 현재(1998년 10월)의 시점에서 이것은 실제 상황과 많이 다르다.
그래서, 여기에서는 시스템에 PAM을 사용하려고 시도할 때에 고려해야할 몇가지 사항을 적는다.
모든 문제에 PAM은 유용한 것인가? 만일 Linux를 1인 사용자 시스템으로 사용하거나 모든 사용자들을 신뢰할 수 있는 환경이라면, PAM을 사용하는 것은 실질적으로 이익이 없다.

Ed: 실제로 장점이 있기는 하다. Win95처럼 인증자체를 사실상 바보로 만들어 버릴 수 있다.

네트워크 환경에서, 사용자들이 어떻게 인증을 받는가 등에 대해 좀더 생각해볼 필요가 있다는 것은 확실하다.

Linux를 서버로 사용하고 있다면, 그리고 그 서버에서 몇몇 서로 다른 서비스(예를 들어 패스워드 컨트롤을 사용하여 일부 사용에 제약을 두는 WWW, PPP등)들을 제공하고 있다면, 실질적이고도 유익한 PAM의 가치가 있을 수 있다. 특히, 모듈 사용을 통해 PAM은 프로그램이 특정 패스워드 데이터베이스에 대한 코드를 가지고 있지 않더라도 여러 패스워드 데이터베이스을 검색할 수 있도록 해준다.
아래에 PAM이 가능하게 해주는 예가 있다.

* Apache는 PAM서비스를 제공하는 모듈을 가지고 있다. 이제 특정 디렉토리들을 사용하는 인증이 PAM에 의해 수행될 수 있다. 이것은 RADIUS, NIS, NCP 같이 PAM이 사용가능한 모듈 전부를 인증에 사용하다는 이야기가 된다.
(즉, 이것은 Novell 패스워드 데이터베이스가 사용가능하다는 것이다.)

* pppd는 PAM이 적용된 버전이 있다(RedHat에 있다.) 이제 ppp 사용자를 인증하기 위해 여러 데이터베이스를 사용하는 것이 가능하다. 일반 Linux기반의 패스워드 데이터베이스(/etc/passwd나 /etc/shadow같은 것들)뿐만 아니라, Novell 패스워드 데이터베이스나 NT기반의 패스워드 데이터베이스에 대해서도 인증을 하는 PAM모듈을 사용할 수 있다.

* 앞의 두 예의 조합이 가능하다. 당신의 사무실/부서에 있는 사람들이 이미 Novell이나 NT 기반의 LAN에 ID와 패스워드를 등록했다고 생각해보자. 이 데이터베이스를 당신의 Linux서버에서 사용하고 싶다면(PPP, Web access뿐아니라, 일반 shell access까지도), 양쪽 모두 별도의 데이터베이스를 유지하는 것보다 PAM을 사용해서 기존의 데이터베이스를 사용한 인증을 할 수 있다.

인증을 필요로하는 모든 프로그램에서 PAM을 사용할 수 있는가?
대답은 예/아니오 모두 가능하다.

소스코드에 접근 할 수 있고, 적절히 PAM기능을 추가할 수 있다면 대답은 '예'이다.
소스코드에 접근할 수 없고, 프로그램 자체가 PAM기능을
지원하지 않는다면 대답은 '아니오'이다.
즉, 어떤 프로그램이 PAM을 사용하려한다면 프로그램이 PAM을 지원하도록
만들어야 한다. 만약 그렇지 않다면, PAM을 사용하는 것은 불가능하다.

프로그램이 PAM지원 기능을 갖고 있는지 아닌지 어떻게 알 수 있는가?
간편한 방법(하지만 항상 믿을것은 못된다.)은 'ldd <programname>'이다.
만약 libpam과 libpam_misc가 프로그램이 사용하는 라이브러리들 중에 없다면,
그 프로그램은 PAM을 사용하지 않는 것이다.
그러나, 라이브러리를 가지고 있더라도 여전히 문제는 있는데 왜냐하면
PAM이 코딩된 프로그램이 제대로 동작하지 않을 수 있기
때문이다. 그래서, 믿을 만한 검증방법은 아래의 테스트를 하는 것이다.

/etc/pam.d 디렉토리에서 실행시키려는 프로그램의 설정파일을 만든다. 정확한
파일의 이름은 프로그램내에 하드-코딩되어 있다. 보통은 프로그램 이름과
같은 이름이지만, 항상 그렇지는 않다. 설명의 편의를 위해 그 프로그램의
이름이 'pamprog'라고 하고, 설정파일은 /etc/pam.d/pamprog라고 하자.

/etc/pam.d/pamprog의 내용을 다음과 같이 하자:

```
auth required pam_permit.so
auth required pam_warn.so
```

이제 pamprog를 사용해보자. 설정파일의 첫번째 줄은 모든 사용자를 통과시키는
것이다. 두번째 줄은 syslog파일에 경고 메시지를 남기는 것이다. 이 테스트가
성공한다면, 당신은 이 프로그램이 pam을 제대로 지원하고 있다는 것을 알 수 있다.
그리고 /etc/pam.d/pamprog 파일내의 모듈 설정을 통해 어떻게 모듈을 쌓을 것인지를
결정하는 더 흥미있는 일을 할 수 있게된다.

4. Linux-PAM 설정 파일

Linux-PAM은 시스템 관리자에게 시스템에 있는 권한을 부여하는 응용프로그램들을 설정하는데 있어서 많은 유연성을 제공해주기 위해 디자인되었다. **Linux-PAM**에의해 조정되는 시스템 보안에 관련된 설정들은 **/etc/pam.conf**파일이나 **/etc/pam.d/** 디렉토리내의 파일들 중 하나에 들어있다. 이 장에서는 이 파일들이 참조하는 내용에 관한 올바른 문법과 옵션에 대해 설명할 것이다.

4.1. 설정 파일 문법(Configuration file syntax)

이 파일에서 **Linux-PAM**고유의 단어(토큰)들은 대소문자 구분이 없다는것을 유념하기 바란다. 그러나, 모듈 경로(path)는 파일 이름을 가르키는 것이고 전형적인 Linux 파일 시스템의 대소문자 구분에 따라 모듈 경로 역시 대소문자를 구분한다. 모듈에게 주어지는 인자(argument)들의 대소문자 구분은 모듈이 정한바에 따른다.

아래 설명된 것들외에, 시스템 관리자의 편의를 위해 두개의 특별한 문자가 제공된다: '#'으로 시작되는 문장은 주석이고 줄의 끝까지 유효하다. 모듈의 설정행에서 줄 끝에 '\'를 사용하면 다음줄도 앞과 같은 하나의 문장으로 처리된다.

/etc/pam.conf파일의 일반적인 설정 문장은 아래의 형태이다:

```
service-name   module-type   control-flag   module-path   arguments
```

아래에 이 각각의 단어들의 의미를 설명한다. **Linux-PAM**을 설정하는 두번째 방법(최근에 적용된 것이다.)은 **/etc/pam.d/**디렉토리내의 파일들을 사용하는 것이다. 위의 단어들의 의미를 설명한 다음에 이 방법에 대해 설명하겠다.

service-name

이 설정문장과 관련한 서비스의 이름이다. 주로 해당 응용프로그램의 일반적인 이름을 사용한다. 예를 들면, '**ftpd**', '**rlogind**', '**su**' 등이다.

디폴트 인증 방법을 정의하기 위해 정의된 특별한 서비스 이름이 있다. '**OTHER**'라는 것이 이것인데, 대소문자 구별없이 쓸 수 있다. 만약 어떤 서비스의 인증 모듈 설정이 이미 되어 있으면 '**OTHER**'항목은 무시된다.

module-type

네가지의(현재) 모듈 타입중 하나이다. 네가지는 다음과 같다:

- **auth**: 이 모듈타입은 사용자 인증의 두가지 면을 제공한다. 첫째는, 응용프로그램이 사용자에게 패스워드를 물어보거나 다른 확인 방법을 사용하도록 해서 그 사용자가 자신이 주장하는 사람이 맞는지 확인하는 것이다. 두번째는, 그룹 멤버쉽을 주거나 (앞서 이야기한 /etc/groups파일에 있는 내용을 통해서) credential(신임)속성을 통해 다른 권한을 주는 것이다.
- **account**: 이 모듈은 인증이 아닌 계정관리를 수행한다. 이것은 보통 시간/날짜나 현재의 시스템 자원 상황(최대 사용자수)이나 사용자의 위치('**root**'는 콘솔에서만 로그인이 가능하다)등에 따라 서비스에 대한 접근을 허가하거나 제한하는 것이다.
- **session**: 주로, 이 모듈은 서비스를 사용하기 직전/직후에 필요한 작업과 관련이 있다. 이러한 것들은 사용자와 교환할 어떤 데이터들의 opening/closing과 관련한 정보들의 기록, 디렉토리의 마운트 등이 해당된다.
- **password**: 이 마지막 모듈타입은 사용자와 연관된 인증토큰 (패스워드)들을 갱신할 때 필요하다. 보통, challenge/response 방식에 기반한 인증모듈 종류별로 각각 하나의 password모듈씩이 존재한다.

control-flag

control-flag은 관련되는 모듈의 성공/실패 결과에 따라 어떻게 PAM라이브러리가 반응할지를 지정한다. 모듈들이 쌓여(같은 타입의 모듈들이 순서대로 수행된다) 있기 때문에 control-flag는 각 모듈의 상대적인 중요도를 결정한다. 응용프로그램은 **/etc/pam.conf**에 열거된 모듈 각각의 성공/실패에 대해 알 수 없다. 대신에, **Linux-PAM**라이브러리로부터 요약된 성공/실패에 대한 결과를 받게된다. 모듈들의 수행 순서는 **/etc/pam.conf**파일에

열거된 순서대로 앞에 나온것이 뒤에것보다 먼저 수행된다. Linux-PAM v0.60부터, 이 **control-flag**은 두가지 문법중 하나로 정의할 수 있다.

보다 단순한(처음부터 있던) 문법은 특정 모듈별로 전체적인 성공/실패에 대해 주는 영향을 하나의 단어로 지시하는 것이다. 이것에는 네가지의 단어가 있다: **required, requisite, sufficient, optional**.

Linux-PAM 라이브러리는 이 단어들을 아래 설명과 같이 해석한다:

- **required**: 이것은 해당 **module_type**이 전체적으로 성공하기 위해서는 이 모듈의 성공이 필요하다는 것을 말한다. 나머지 모듈(같은 **module-type**의)이 모두 다 수행되기 전까지 사용자가 이 모듈이 실패했는지 확실히 알 수 없다.
- **requisite**: **required**와 비슷하나, 실패한 경우 제어가 바로 응용프로그램에게 넘어간다. 결과값은 처음의 실패한 **required**나 **requisite**모듈의 결과값이다. 이 플래그는 사용자가 안전하지 않은 매체(medium)에서 패스워드를 입력할 경우에 대비해 사용할 수 있다. 이러한 행동은 크래커에게 시스템의 유효한 계정을 알려줄 수 있다. 이 가능성은 민감한 패스워드를 적대적인 환경에 노출시키는 것에 대한 사소하지 않은 염려와 비교해서 검토해야 한다.
- **sufficient**: 이 모듈의 성공은 **Linux-PAM**라이브러리에게 이 module-type이 성공했다고 판단할만큼 충분하게 만족스럽다는 것이다. 이전의 **required**모듈이 실패하지 않았다면 더이상 이 type의 나머지 모듈들은 수행되지 않는다. (이경우에 뒤따르는 **required**모듈이 수행되지 않는것에 유의하라) 이 모듈의 실패는 이 **module-type**의 성공여부 판단에 치명적이지는 않은것으로 간주된다.
- **optional**: 이름이 의미하듯이, 이 **control-flag**는 모듈이 서비스에 대한 응용프로그램의 성공/실패에 중요하지는 않다는 것이다. 보통, **Linux-PAM**은 모듈들의 성공/실패 판단시에 이런 모듈은 무시한다. 그러나, 이전/이후의 모듈들이 명확한 성공/실패가 없다면 이 모듈이 응용프로그램에게 주는 결과를 결정짓는다. 후자의 경우의 예로, 다른 모듈들이 **PAM_IGNORE**같은 것을 리턴할 경우이다.

좀더 정교한(새로운) 문법은 더욱 명확하고 시스템 관리자가 사용자를 어떻게 인증할지에 있어 더 많은 제어방법을 제공한다. 이 control-flag는 대괄호([])안에 **value=action**들이 나열되어 있는 형태이다:

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

valueI는 다음의 리턴값들 중 하나이다: **success; open_err; symbol_err; service_err; system_err; buf_err; perm_denied; auth_err; cred_insufficient; authinfo_unavail; user_unknown; maxtries; new_authtok_reqd; acct_expired; session_err; cred_unavail; cred_expired; cred_err; no_module_data; conv_err; authtok_err; authtok_recover_err; authtok_lock_busy; authtok_disable_aging; try_again; ignore; abort; authtok_expired; module_unknown; bad_item; default**. 마지막(**default**)것은 명확히 정의되지않은 리턴값들에 대한 action을 정하기 위해 사용된다.

actionI는 양의 정수나 다음의 토큰들 중 하나이다: **ignore; ok; done; bad; die; reset**. 양의 정수 **J**가 action으로 정해지면 같은 type의 모듈들중 다음의 **J**개 모듈들을 건너뛰게하는 것이다. 이 방법으로 관리자는 여러개의 서로 다른 실행과정을 갖는 정교한 모듈목록을 적절히 만들 수 있게 된다. 어느 경로가 될지는 각 모듈의 반응에 의해 결정된다.

- **ignore** - 모듈의 결과값이 응용프로그램이 갖게될 결과값에 영향을 주지 않는다.
- **bad** - 이것은 결과값이 모듈 실패의 표시로 생각해야 한다. 이모듈이 실패한 첫번째 모듈이면, 이 상태값은 전체 모듈의 결과로 사용될 것이다.
- **die** - bad와 같지만 모듈 수행이 중지되고, PAM이 응용프로그램으로 바로 리턴한다는 것이 다르다.
- **ok** - 이것은 PAM에게 관리자가 이 결과값이 모듈 전체의 결과 상태로 바로 사용되길 바란다고 알려주는 것이다. 즉, 모듈목록의 수행과정에서 앞까지의 상태가 PAM_SUCCESS이면 이 모듈의 결과값은 앞까지의 상태값을 덮어쓴다. 반면 앞까지의 상태가 실패라면 이 'ok'값은 상태를 덮어쓰지 않을 것이다.
- **done** - ok와 같지만 모듈 수행이 중지되고 PAM이 응용프로그램에게 바로 리턴한다는 것이 다르다.
- **reset** - 모듈들의 결과상태 기록을 지우고, 다음 모듈부터 새로 시작한다.

새로운 문법의 능력을 체감하기 위해, 이것으로 무엇을 할 수 있는지 보여주겠다. **Linux-PAM-0.63**부터 client plug-in agents라는 것이 소개되었다. 이것은 PAM이 client/server프로그램들이 고유하게 가지고 있는 전송 규약을 사용해서 서로다른 기계간의 인증을 지원하는 것을 가능하게 해 준다. "**[... value=action ...]**" 제어 문법으로 응용프로그램이 유순한 클라이언트 프로그램에게는 바이너리 프롬프트를 지원하지만 동시에 우아하게도 오래된 구식의 응용프로그램에게는 다른 인증모드를 선택하도록 설정할 수 있다. 유연하지 않은가?

module-path

동적으로 로드될 오브젝트 파일(모듈 자체)의 경로명이다. module path의 첫 문자가 '**/**'라면, 전체경로를 의미한다. 그렇지 않은경우에는 default module path(**/usr/lib/security**)를 붙여서 해석된다. (그렇지만 2장 내용을 참고하라).

args

args는 모듈이 불려질때 넘겨주는 토큰 리스트이다. 전형적인 Linux shell명령에 주는 인자들과 비슷하다. 보통, 유효한 인자는 옵션이고 특정 모듈에만 해당된다. 무효한 인자들은 모듈에 의해 무시되지만, 무효한 인자들을 보게되면 모듈은 **syslog(3)**에 에러를 보내도록 요구받는다. 일반적 **generic**옵션의 목록은 다음절(4.3)에서 볼 수 있다.

설정 파일의 어떤 줄이라도 제대로 작성되어있지 않을 경우 일반적으로 인증 절차가 실패하게 된다. 해당 에러는 **syslog(3)**를 통해 시스템 log파일에 기록된다.

4.2. 디렉토리에 설정파일을 기록할 때(Directory based configuration)

하나의 설정 파일보다 더 유연성을 갖기 위해 0.56버전 부터 **/etc/pam.d/** 디렉토리의 파일들로 **libpam**을 설정하는것이 가능해졌다. 이경우에 그 디렉토리의 파일은 파일이름과 같은 이름(소문자)을 갖는 서비스에 대한 설정을 갖게 된다: 서비스 각각에 대한 설정 파일인 것이다.

Linux-PAM은 두가지 모드로 컴파일 할 수 있다. 추천하는 쪽은 **/etc/pam.d/**나 **/etc/pam.conf**둘 중 하나만 사용하는 것이다. 하지만 둘다 같이 할 수는 없다. 즉, **/etc/pam.d/**디렉토리가 있다면, libpam은 그 디렉토리에 있는 파일들만을 사용한다. 그러나 **/etc/pam.d/** 디렉토리가 없는 경우에는 **/etc/pam.conf**를 사용한다. 다른 쪽은(현재 RedHat 4.2이상에서 지원되고 있다) **/etc/pam.d/**와 **/etc/pam.conf**를 순서대로 모두 사용하는 것이다. 이 때는 **/etc/pam.conf**의 내용보다 **/etc/pam.d/**의 내용이 우선적으로 사용된다.

/etc/pam.d/의 파일의 문법은 **/etc/pam.conf**의 문법과 비슷하다. 각 문장은 다음과 같은 형태이다:

```
module-type   control-flag   module-path   arguments
```

서비스 이름**(service-name)**이 없다는것 하나만 다르다. 서비스 이름(service-name)은 물론 설정파일의 이름이다. 예를 들어, **/etc/pam.d/login**은 **login**서비스에 대한 설정을 가지고 있다.

이 설정 방법은 단일 파일로 하는 것에 비해 여러 장점을 갖는다. 어느 방법을 적용할지 선택하는데 도움을 주기 위해 여기에 그 장점을 나열한다:

- 잘못 설정할 가능성을 줄여준다. 설정파일을 손으로 수정할 때, 필드가 하나 적기 때문에 그만큼 실수도 적다.
- 유지가 쉽다. 한 응용프로그램 설정을 다시할 때, 시스템의 다른 응용프로그램을 방해할 위험이 없다.
- 여러 서로 다른 서비스의 설정 파일을 하나의 파일에 심볼릭(symbolic) 링크하는것이 가능하다. 이것은 다른 응용프로그램간의 보안정책을 동일하게 유지하기 쉽게 해준다. (공간을 절약하기 위해, 하드**(hard)**링크 하는 것도 가능하다. 하지만, 하드 링크된 파일을 수정하는 것은 링크를 끊을 수 있기 때문에 조심해야한다.)
- 설정 파일 분석(parsing)이 빠를 수 있다. 서비스가 해당 모듈을 필요로 할 때, 관련된 파일만 읽어서 분석하면 된다.
- 각각의 **Linux-PAM**설정 파일들을 파일시스템의 파일 보호방법을 사용해서 읽기 권한을 제한할 수 있다.
- 패키지 관리가 쉬워진다. 새로운 응용프로그램이 설치될 때마다, **/etc/pam.d/xxxxxx** 파일을 같이 설치하기만 하면 된다.

4.3. 일반적인 옵션 인자들**(generic optional arguments)**

다음은 어떠한 모듈이든 인식할 수 있는 가능성이 큰 옵션 인자들이다. 인자들(이것들도 포함해서)은 대게 선택적**(optional)**이다.

debug

syslog(3) 호출을 사용해서 시스템 로그 파일에 디버그 정보를 남기도록 한다.

no_warn

모듈이 응용프로그램에게 경고(warning) 메시지를 보내지 않도록 한다.

use_first_pass

모듈이 사용자에게 패스워드를 요구하지 않도록 한다. 대신에, 이전에 입력받았던 패스워드(먼저 수행된 **auth**모듈 로부터)를 받아서 사용해야 한다. 패스워드가 없다면, 사용자는 인증을 받지 못하게 될 것이다.(이 옵션은 **auth**와 **password** 모듈들만 적용된다.)

try_first_pass

모듈은 이전에 입력받은 패스워드(먼저 수행된 **auth** 모듈로부터)로 인증을 시도한다. 만일 패스워드가 없다면, 사용자에게 패스워드를 물어본다.(이 옵션은 **auth**모듈만 적용된다.)

use_mapped_pass

이 인자는 현재 배포되고 있는 **Linux-PAM**에 있는 어떤 모듈도 지원하지 않는다. 이는 미국 암호화 수출 제한과 관련하여 생길 수 있는 문제 때문이다. 미국내에서는 개발자들이 물론 자유롭게 이 기능을 구현할 수 있다. (다른 나라에서도 마찬가지이다.). 호환성의 이유로 **DCE-RFC 86.0** (bibliography 장을 보라) 에서 제안된데로 이것의 사용을 기술한다.

use_mapped_pass인자는 암호화되지 않은 인증토큰을 요구하는 앞의 모듈로부터 인증토큰을 받은후 다시 이것을 사용해 암호/복호 키를 생성하여 이 모듈에게 필요한 인증토큰을 안전하게 저장하고 읽을 수 있게 한다. 이런 방법으로, 사용자는 한번의 인증 토큰 입력만으로 여러개의 모듈에 의한 인증 과정을 조용히 통과할 수 있다. 안전하게 하기 위해 믿을만한 튼튼한 암호화를 요구하지만 편한 기능임은 분명하다. 이 인자는 **auth**와 **password**모듈에만 적용된다.

expose_account

보통 사용자 계정에 관한 정보를 누설하는 것은 모듈이 채택할만한 안전한 보안정책이라 할 수 없다. 때때로, 사용자 이름이나 홈 디렉토리나 사용할 쉘에 대한 정보들은 사용자 계정을 공격하는데 사용될 수 있다. 그러나 어떤 상황에선 이런 종류의 정보가 위험요소가 될 것 같지는 않다: 보안이 유지된 환경에서 패스워드를 물어볼 때, 사용자의 실제 이름을 표시하는 것은 친근한 느낌을 준다고 볼 수도 있을 것이다. **expose_account**인자는 관리자에 의해 적절한 것 같은 계정 정보에 대해 덜 엄격하게 구분하도록 하는 표준 인자이다.

4.4. 설정파일 내용의 예(Example configuration file entries)

이 장에서는 **Linux-PAM**설정 파일에서 볼 수 있는 내용에 대한 예제 몇개를 설명한다. 시스템을 처음으로 설정할 때, 이것들을 구현하는 것보다 더 심각한 짓을 할지도 모른다.

4.4.1. 디폴트 정책(Default policy)

시스템이 안전해야한다고 생각되면, '**OTHER**' 항목에 대해서 안전한 설정을 해야 좋을 것이다. 다음은 심하게 불신하도록 설정되어 있는 것이다(처음 시작하기에 나쁜 것은 아니다!):

```
#
# default; deny access
#
OTHER auth  required /usr/lib/security/pam_deny.so
OTHER account  required /usr/lib/security/pam_deny.so
OTHER password required /usr/lib/security/pam_deny.so
OTHER session  required /usr/lib/security/pam_deny.so
```

근본적으로 안전한 반면, 잘못 설정된 시스템에게는 대단히 좋다고 말할 수 없다. 예를들어, 나머지 파일들이 잘못 씌여지면 모든 사용자들을 막아버릴 가능성이 있게된다.

pam_deny모듈(나중에 설명된다)은 설정하기에 좋지 않다. 예를 들어, 수행될때 로그를 전혀 남기지 않기때문에, 사용자가 서비스 프로그램을 사용하는데 실패를 했을때 관리자에게 알리지 않는다면 관리자는 시스템의 설정 실수에 대한 사실을 모른채로 오랫동안 지낼것 이다.

위의 예제와 같은 설정앞에 다음을 추가하면 관리자에게 적절한 경고 메시지를 줄 것이다.

```
#
# default; wake up! This application is not configured
#
OTHER   auth     required       /usr/lib/security/pam_warn.so
OTHER   password required       /usr/lib/security/pam_warn.so
```

두줄의 'OTHER auth'가 같은 type의 여러모듈 적층사용의 예이다.

/etc/pam.d/설정을 사용하는 시스템에서는 아래와 같은 파일이 위에서 얘기한 디폴트 설정이 된다:

```
#
# default configuration: /etc/pam.d/other
#
auth  required /usr/lib/security/pam_warn.so
auth  required /usr/lib/security/pam_deny.so
account  required /usr/lib/security/pam_deny.so
password required /usr/lib/security/pam_warn.so
password required /usr/lib/security/pam_deny.so
session  required /usr/lib/security/pam_deny.so
```

이게 **/etc/pam.d/**형태의 설정 방법의 예로 보여주는 유일한 예이다. 나머지 예들에서 이 설정 방법으로 변환하는 방법은 명확할 것이다.

덜 민감한 컴퓨터, 관리자가 **Linux-PAM**의 많은 기능들을 무시한채로 두고 싶은 경우는 다음의 줄들의 예 (**/etc/pam.conf**에서)가 전통적으로 친숙한 linux 설정을 모방한 설정이 될 수 있다.

```
#
# default; standard UNIX access
#
```

```
OTHER   auth     required       /usr/lib/security/pam_unix_auth.so
OTHER   account  required       /usr/lib/security/pam_unix_acct.so
OTHER   password required       /usr/lib/security/pam_unix_passwd.so
OTHER   session  required       /usr/lib/security/pam_unix_session.so
```

[역주: 지금 버젼에서는 pam_unix_auth, pam_unix_acct, pam_unix_passwd, pam_unix_session이 pam_unix.so로 통합되었고, 좀더 발전한 형태로 pam_pwdb.so가 존재한다.] 이 내용이 일반적으로 대부분의 응용프로그램에 대해 초기설정으로 사용될 수 있다. 불행하게도, 대부분이 모두라는 것은 아니다. **ftpd**같은 경우는 **anonymous-ftp**를 가능하게 하기 위해 추가적인 설정이 필요하다.

anonymous-ftp를 가능하게 하기 위해 다음의 줄이 디폴트(**OTHER**) 설정을 대체해서 사용할 수 있다. (*경고* 1996/12/28이후, 어떠한 ftpd와도 이 설정이 제대로 동작하지 않았다. 결과적으로 이 설명은 바뀌거나 응용프로그램이 고쳐질 것이다.)

```
#
# ftpd; add ftp-specifics. These lines enable anonymous ftp over
#       standard UNIX access (the listfile entry blocks access to
# users listed in /etc/ftpusers)
#
ftpd auth sufficient  /usr/lib/security/pam_ftp.so
ftpd auth required    /usr/lib/security/pam_unix_auth.so use_first_pass
ftpd auth required    /usr/lib/security/pam_listfile.so \
onerr=succeed item=user sense=deny file=/etc/ftpusers
```

/etc/pam.conf에 해당 서비스(여기서는 **ftpd**)의 설정파일이 있으면 디폴트 설정이 무시되기 때문에 두번째 줄이 필요하다. 다시 한번 이 예제는 authentication모듈에 있어서 같은 type의 여러 모듈을 적층해서 같이 사용될 수 있음을 보여준다. **sufficient** control-flag을 사용했음을 주목하라. 이것은 이 모듈이 사용자를 인증하면 이후의 **auth**모듈은 무시하라는 것이다. 또한 '**use_first_pass**' module-argument를 사용했음을 주목하라. 이것은 UNIX 인증 모듈이 사용자에게 패스워드를 물어보지 않고 앞서 실행된 ftp모듈에 의해 얻은 패스워드만을 사용하도록 설정한다.

5. Linux-PAM의 보안 이슈(Security issues of Linux-PAM)

이 장에서는 Linux-PAM을 보안감각을 갖고 사용하는 좋은 사례에 대해 논의할 것이다. 내용이 부족한데**...** 제안은 환영한다**!**

5.1. 뭔가가 잘못 되고 있다면(If something goes wrong)

Linux-PAM은 시스템의 보안을 변경시키는 심각한 잠재력을 갖고 있다. 당신은 아무런 보안도 하지 않게 할 수도 있고, 완벽보안 (아무도 접근할 수 없게)을 할 수도 있다. 보통은 **Linux-PAM**이 후자의 경우 에러를 낸다. 설정에러는 시스템을 부분적으로 또는 완전히 접근을 막아버릴 수 있다.

가장 **Linux-PAM**을 설정할 때 부닥치게 되는 가장 기가 막힌 상황은 설정파일(**/etc/pam.d/*** 나 **/etc/pam.conf**)을 지우는 것이다. 이렇게 되면 당신의 시스템은 당신자신도 쓸 수 없게 완전히 잠겨버린다!

이를 복구하기 위한 최선의 방법은 시스템을 재부팅한 후 싱글 유저 모드로 들어가서 문제를 고치는 것이다. 다음은 이 문제에 대해 David Wood가 답한 email에서 귀중한 내용을 인용한 것이다.

```
> 내가 도대체 지금 무엇을 한겁니까?

알았습니다. 진정하세요. 당신이 우선 PAM을 사용하는 사용자의 절반 가량에게
이 일이 생긴다는 것을 아셔야 할 필요가 있습니다.
여기에서도 한번도아니고 두번도 아니고 세번이나 일어났습니다.
결국, 매번 해결책은 항상 같았습니다.
첫째로, 당신이 LILO를 딜레이를 갖도록 설정했기를 바랍니다. 가능하다면, 리부팅을
하고 shift나 tab을 친 후 아래와 같이 치십시오:
LILO boot: linux single
('linux'를 '당신이 쓰는 linux 이미지 이름'으로 바꾸세요).
이것은 당신이 로그인 없이 시스템에 들어가게 해줄 것입니다. 콘솔에서
linux머신에 침입하는 것이 얼마나 쉬운지 궁금해하지 않았습니까?
이제 당신도 압니다.

이것을 할 수 없다면, bootkernel 플로피와 slackware의 rescue.gz같은
root디스크를 구하십시오. (RedHat의 설치 디스크를 이 모드로도 사용가능합니다.)

두 경우 모두, root 프롬프트를 다시 볼 수 있을겁니다.

두번째로, 당신이 pam 설치를 완전히 망가뜨리지 않았다고 가정하겠습니다. 즉,
설정파일만 없앴것으로 알겠습니다.
여기에 어떻게 당신의 설정파일을 다시 멋있게 만드는지 보여드리겠습니다:
  cd /etc
  mv pam.conf pam.conf.orig
  mv pam.d pam.d.orig
  mkdir pam.d
  cd pam.d
그리고 vi를 사용해서 "other"라는 파일을 이 디렉토리에 만드십시오. 이 파일에
아래의 네줄을 포함하도록 하십시오:
  auth required pam_unix_auth.so
  account required pam_unix_acct.so
  password required pam_unix_passwd.so
  session required pam_unix_session.so

이제 당신은 당신이 사용하던대로 동작하는 가장 단순한 PAM설정 파일을 갖게
되었습니다. 모든게 마법처럼 다시 동작할 것입니다. ALT-F2를 누르고, 다른 버츄얼
```

```
콘솔에서 로그인 해보십시오. 만일 제대로 되지 않는다면, 당신은 더 큰 문제를
가지고 있거나, 설정파일에 오자가 있을 겁니다. 이 시스템의 이상한일이
설정파일에 오타가 있는 것이라면 콘솔에서는 에러 메시지를 볼수 없고,
log파일에서 볼 수 있습니다. 한번 보세요.
('tail /var/log/messages'라고 해보세요)

이제부터 예전처럼 돌아가서 실제 설정파일을 가져옵시다. 망가뜨려도 상관없는
시스템에서 시험을 거친후에 가져오는 것이 좋겠지요. :/

약간의 지시사항(RedHat에서는 모두 잘 될겁니다.):
    최신 pam, pamconfig, pwdb를 redhat의 current 디렉토리에서 인스톨하세요.
rpm -Uvh [아마 --force도] pam-* pamconfig-* pwdb-*

    그리고, 최신의 libc, util-linux, wuftp, NetKit을 설치/재설치했는지
확인하세요. xlock같은 X 응용프로그램에 영향을 주는 새 버젼의 설치를 시도하다가
당하는 퇴짜는 아직까지는 없었습니다.
```

5.2. 약한 'other' 설정하지 않기(Avoid having a weak 'other' configuration)

약하게 디폴트(**OTHER**)설정을 두는 것은 좋지 않다. PAM을 지원하는 응용프로그램들에 대해 이 디폴트 설정이 적용되고 만약 이 설정이 약하다면, 시스템이 공격받기 쉽다.

여기에 'other' 설정파일에 대한 예가 있다. **pam_deny** 모듈은 항상 접근을 불허하고 **pam_warn** 모듈은 syslog 메시지를 **auth.notice**에게 보낼 것이다:

```
#
# The PAM configuration file for the 'other' service
#
auth      required   pam_deny.so
auth      required   pam_warn.so
account   required   pam_deny.so
account   required   pam_warn.so
password  required   pam_deny.so
password  required   pam_warn.so
session   required   pam_deny.so
session   required   pam_warn.so
```

6. 사용가능한 모듈들에 대한 자세한 설명(A reference guide for available modules)

이장에서는 **Linux-PAM**에서 사용가능한 여러 모듈들에 대한 설명을 하고 있다. 대부분 이 모듈들은 무료로 사용할 수 있다. 그렇지 않은 경우는 표시를 해놓을 것이다.

예제들을 복사할 때에는 2장에서 언급한 내용을 꼭 읽고 주의하기 바란다.

6.1. The access module

6.1.1. 개요

모듈 이름:

pam_access

저자:

Alexei Nogin <alexei@nogin.dnttm.ru>

유지보수:

저자

제공되는 관리 그룹:

account

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

설정 파일을 요구한다. 디폴트는 /etc/security/access.conf이지만, 바꿀 수 있음.

네트워크 인식:

PAM_TTY가 지정되어 있다면, PAM_TTY를 통해서 얻고, 그렇지 않다면 ttyname()으로 stdin파일 설명자(descriptor)의 tty 이름을 얻는 시도를한다. 표준 gethostname(), yp_get_default_domain(), gethostbyname()호출. netgroup지원을 위한 NIS가 사용.

6.1.2. 모듈의 전반적인 설명

logdaemon형태의 login 접근 제어를 제공한다.

6.1.3. 계정관리 구성요소

인식하는 인자들:

accessfile=/path/to/file.conf

설명:

이 모듈은 login name, 호스트(또는 도메인) 이름, 인터넷 주소(또는 네트워크 번호), 또는 네트워크가 아닌 login일경우 터미널 이름을 바탕으로 logdaemon형태의 login접근 제어를 제공한다. 진단은 syslog(3)를 통해 보고된다. Wietse Venema가 제작한 logdaemon-5.6의 login_access.c를 A. Nogin이 몇가지 수정해 사용하였다.

이 모듈의 동작은 다음의 인자로 제어할 수 있다:

- * accessfile=/path/to/file.conf - 디폴트 대신 사용할 access설정 파일을 가르킨다. 이것은 서로 다른 서비스들이 각자의 access파일을 사용할 때 유용하다.

예제/추천하는 사용법:

몇몇 계정은 동작하지만 전부 login 하게 할 필요가 없는 NIS서버나 메일 서버같은 관리용 기계에 이 모듈을 사용할 것을 권장한다.

/etc/pam.d 형태의 설정의 경우에, 모듈이 /lib/security에 있다면, /etc/pam.d/login, /etc/pam.d/rlogin, /etc/pam.d/rsh 그리고 /etc/pam.d/ftp에 다음의 줄을 우선 추가한다:

```
account required /lib/security/pam_access.so
```

시스템이 .rhosts파일을 무시하지 않는다면 이 모듈의 사용이 효과가 없다는 것을 주목하라. pam_rhosts_auth문서를 보라.

예제 access.conf설정 파일은 배포판에 포함되어 있다.

6.2. Chroot

6.2.1. 개요

모듈 이름:

pam_chroot

저자:

Bruce Campbell <brucec@humbug.org.au>

유지보수:

저자; proposed on 20/11/96 - email for status

제공되는 관리 그룹:

account; session; authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

기록되지 않았음

시스템 의존성:

네트워크 인식:

localhost로 가정함

6.2.2. 모듈의 전반적인 설명

이 모듈은 일반 사용자들에게 파일시스템의 루트를 속이기 위한 목적으로 만들어졌다.(예를 들어, 사용자의 '/'는 실제로는 /some/where/else이다.)

여러 분류의 사용자들이 시스템을 사용하고, 당신이 보안에 관해 약간 신경을 많이 쓴다면 유용할 것이다. 시스템에서 사용자들이 볼수 있는 것을 제한하고 실행시킬 수 있는 프로그램을 제한하는데 사용할 수 있다.

6.2.3. 계정관리 구성요소:

내용이 아직 없음.

6.2.4. 인증관리 구성요소:

내용이 아직 없음.

6.2.5. 세션관리 구성요소:

내용이 아직 없음.

인식하는 인자들:

PAM 버젼에 맞는 인자들과 로그 수준은 작업중이다.

설명:

예제/추천하는 사용법:

적당한 프로그램들의 리스트를 제공하라 - 'cat', 'ls', 'rm', 'cp', 'ed' 같은 것들.

너무 심하게 하지마라.(예를들어 각 사용자마다 별도의 환경을 구성할 수 있지만 만약 그렇게 한다면 디스크 낭비가 심할것이다.)

6.3. Cracklib pluggable password strength-checker

6.3.1. 개요

모듈 이름:

pam_cracklib

저자:

Cristian Gafton <gafton@redhat.com>

유지보수:

저자

제공되는 관리 그룹:

password

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

libcrack 시스템 라이브러리와 시스템 사전:/usr/lib/cracklib_dict.

네트워크 인식:

6.3.2. 모듈의 전반적인 설명

이 모듈은 password의 튼튼한 정도를 검사하도록 하기 위해 응용크로그램의 패스워드 스택에 등록해서 사용된다.

이 모듈은 다음과 같이 동작한다: 우선 패스워드의 튼튼함을 검사하기 위해 Cracklib 루틴을 호출한다. 패스워드가 crack되지 않으면 다음의 추가적인 검사를 한다:

- 회문 - 새 패스워드가 예전 패스워드의 회문(palindrome) - 역주: 앞으로 읽으나 뒤로 읽으나 같은 글자 - 이 아닌가?
- 대소문자 변경만 한 것인가? - 새 패스워드가 예전 패스워드에서 대소문자 변경만 된게 아닌가?
- 유사성 - 새 패스워드가 예전 패스워드와 얼마나 많이 비슷한가? 이것은 'difok' 인자에 의해 주로 조절된다. 'difok'는 새 패스워드가 예전 패스워드와 얼마나 많은 다른 문자들을 가지고 있어야 하는가를 나타낸다. 이것의 디폴트값은 10이거나 새 패스워드의 크기의 1/2중 작은 값이다. 길고 복잡한 패스워드로 변경해야만 하는 것을 피하기 위해 'difignore'가 있다. 이것은 새 패스워드 길이의 최소값을 정해서, 길이가 그 이상이면 'difok'를 무시한다. difignore의 디폴트값은 23이다.
- 단순함 - 새 패스워드가 너무 짧지 않은가? 이것은 다음 5개의 인자로 조절된다. 'minlen', 'dcredit', 'ucredit', 'lcredit', 'ocredit'이다. 이 인자들에 대한 기본값과 동작에 대한 자세한 설명은 인자에 대한 설명이 있는 란을 보라.
- 돌려썼는가? - 새 패스워드가 예전 패스워드를 돌려쓰기만 한 것이 아닌가?
- 이미 사용된 패스워드 - 새 패스워드가 과거에 사용되었는가? 예전 패스워드는 /etc/security/opasswd에 저장된다.

이 모듈은 인자 없이 사용되면 표준 유닉스 패스워드 암호화 방식을 사용한다. md5암호화를 사용하면, 패스워드는 8자 이상이어도 되고, 이 모듈의 디폴트 설정은 사용자가 만족스러운 새 패스워드를 선택하는데 어렵게 만들 수 있다. 명백히, 새 패스워드가 예전 패스워드의 1/2이상의 문자들을 포함하지 않도록 하는 것은 사소하지 않은 제약이 되었다. 예를 들어, 예전 패스워드가 'the quick brown fox jumped over the lazy dogs'이면 바꾸기가 어렵다. 게다가 디폴트설정이 5자의 짧은 길이의 패스워드를 허용한다. md5시스템에 대해서, 새 패스워드의 최소 길이를 늘리는 것은 좋은 생각이다. 이렇게 하면 서로다른 종류의 문자들을 사용해서 패스워드를 만들게하는 것을 장려하고, 동시에 예전 패스워드에 있는 문자들 대부분을 그대로 새 패스워드에서도 쓸 수 있게 하는 것을 허용할 수 있다.

6.3.3. 패스워드관리 구성요소

인식하는 인자들:

```
debug; type=XXX; retry=N; difok=N; minlen=N; dcredit=N; ucredit=N; lcredit=N; ocredit=N;
use_authtok;
```

설명:

이 모듈의 동작은 나쁜 패스워드를 알아내기 위해 사용자에게 패스워드를 묻고, 시스템 사전과 몇가지 규칙으로 패스워드의 튼튼함을 검사한다.

기본 동작은 하나의 패스워드를 묻고, 튼튼함을 검사하고, 튼튼하다고 생각되면 다시 패스워드를 묻는다(첫번째것이 올바로 입력된 것인지 확인하기 위해서). 모두 통과하면 패스워드는 설치된 다음 모듈로 새 인증 토큰으로서 전달된다.

기본 동작은 인자들을 사용해서 여러가지로 고칠 수 있다.

- debug - 이 옵션은 모듈이 동작을 보여주기 위해 syslog(3)에 정보를 남기도록 한다.(이 옵션이 log파일에 패스워드 정보를 남기지는 않는다.)
- type=XXX - 모듈의 디폴트 동작은 패스워드를 물어볼 때 "New UNIX password: ", "Retype UNIX password:"라고 묻는데, 이 옵션을 사용해서 'UNIX'라는 말 대신 'XXX'로 바꿀 수 있다.
- retry=N - 새 패스워드를 물어보는 횟수로 디폴트는 1이다. 이 옵션을 사용해서 N만큼 횟수를 늘릴 수 있다.
- difok=N - 새 패스워드에서 예전 패스워드에 있지 않은 문자들을 몇자나 사용해야 하는지 나타내는 수로 디폴트는 10이다. 그리고, 새 패스워드에서 1/2이상의 글자가 이전과 다르다면 새 패스워드는 받아들여진다.
- minlen=N - 새 패스워드의 최소 크기에 1을 더한 크기이다. 새 패스워드엔 사용된 문자열의 길이 외에 각 문자종류 (숫자, 대문자, 소문자, 기타문자)를 사용한 것에 대해서 각각 크레딧을 준다. 이 인자의 초기값은 9인데, 같은 문자 종류만을 사용하는 예전 UNIX패스워드에 적당하지만 md5시스템의 추가된 보안능력을 충분히 쓰기에는 적당하지 못하다. (역주: 문자 종류에 대한 크레딧들이 1이라고 하면, 모두 같은 종류로만 된 패스워드는 8의 길이를 가져야 하고, 두가지 종류를 갖는 경우에는 7의 길이를 가져야 하는 것이다.)

  Cracklib자체에 두가지 길이 제한이 있다는 것을 염두해 두라. Cracklib에는 4자 이상으로 하도록 하드코딩되어 있고 minlen값을 참고하지 않고 검사하도록 정의된 값(6)이 있는데, 만일 더 짧은 패스워드를 허용하고 싶으면 이 모듈을 사용하지 말거나 Cracklib과 이 모듈을 다시 컴파일해라.

- dcredit=N - 숫자 문자가 가질 수 있는 크레딧의 최대값이다. 만약 패스워드에 N개 이하의 숫자를 사용했다면 사용된 숫자의 갯수가 현재 정의된 minlen값을 계산하는데 크레딧값으로 적용된다. 이것의 기본값은 1인데, 1은 minlen이 10보다 작을때 권장값이다.
- ucredit=N - 대문자가 가질 수 있는 크레딧의 최대값이다. 만약 패스워드에 N개 이하의 대문자를 사용했다면 사용된 대문자의 갯수가 현재 정의된 minlen값을 계산하는데 크레딧값으로 적용된다. 이것의 기본값은 1인데, 1은 minlen이 10보다 작을때 권장값이다.
- lcredit=N - 소문자가 가질 수 있는 크레딧의 최대값이다. 만약 패스워드에 N개 이하의 소문자를 사용했다면 사용된 소문자의 갯수가 현재 정의된 minlen값을 계산하는데 크

레딧값으로 적용된다. 이것의 기본값은 1인데, 1은 minlen이 10보다 작을때 권장값이다.

- ocredit=N - 앞서 정의된 문자 이외의 기타 문자가 가질 수 있는 크레딧의 최대값이다. (역주: 즉, 기타 문자는 숫자, 대문자, 소문자를 제외한 나머지 문자를 가르킨다.) 만약 패스워드에 N개 이하의 기타 문자를 사용했다면 사용된 기타문자의 갯수가 현재 정의된 minlen값을 계산하는데 크레딧값으로 적용된다. 이것의 기본값은 1인데, 1은 minlen이 10보다 작을때 권장값이다.
- use_authok - 이 인자는 사용자에게 새 패스워드를 묻지말고, 앞서 실행된 password모듈에서 받은 것을 사용하도록 모듈에게 강제한다.

예제/추천하는 사용법:

이 모듈 사용의 예를위해, pam_pwdb의 password component와 적층구조로 설정하는 것을 보여준다.

```
#
# 이 줄들은 2개의 password 타입의 모듈을 갖고 있다. 이 예에서 사용자는 3번의
# 튼튼한 패스워드를 입력할 기회를 갖는다. 'use_authok'인자는 pam_pwdb모듈이
# 패스워드를 묻지 않도록하고, 대신에 pam_cracklib이 제공하는 것을 사용하도록
# 한다.
#
passwd  password required       pam_cracklib.so retry=3
passwd  password required       pam_pwdb.so use_authtok
```

또하나의 예로(/etc/pam.d/passwd 형태에서) md5 패스워드 암호화를 사용하길 원하는 경우이다.

```
#
# 이 줄들은 md5시스템이 적어도 14자이고, 숫자2, 다른문자2에 대한 크레딧값을
# 갖도록 하고, 예전 패스워드에 존재하지 않는 문자들을 적어도 3개이상 갖도록
# 하는 것이다.
#
password  required pam_cracklib.so di-
fok=3 minlen=15 dcredit= 2 ocredit=2
password  required pam_pwdb.so use_authtok nullok md5
```

6.4. 잠금 모듈(The locking-out module)

6.4.1. 개요

모듈 이름:

pam_deny

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

현재의 Linux-PAM 유지보수자

제공되는 관리 그룹:

account; authentication; password; session

암호학적 민감성:

보안 등급:

컴파일시 메시지:

에러메시지 없음

시스템 의존성:

네트워크 인식:

6.4.2. 모듈의 전반적인 설명

이 모듈은 접근을 거부하는데 사용된다. 이것은 PAM구조를 통해 응용프로그램에게 항상 실패를 리턴한다. 앞서(3.개요) 말했듯이 이 모듈은 디폴트(OTHER) 셋팅으로 적당하다.

6.4.3. 계정관리 구성요소

인식하는 인자들:

설명:

오직 실패만을 리턴한다. 실패type은 PAM_ACCT_EXPIRED.

예제/추천하는 사용법:

이 모듈을 account 타입에 설정하면 Linux-PAM의 account management함수인 pam_acct_mgmt()를 사용하는 응용프로그램들을 통해 시스템에 접근하려는 모든 사용자들을 막을 수 있다.

다음 예는 로그인을 불가능하게 만드는 것이다.

```
#
#모든 계정을 막기 위해서 이 줄을 추가하라.
#
login   account  required       pam_deny.so
```

6.4.4. 인증관리 구성요소

인식하는 인자들:

설명:

아무것도 하지 않고 실패를 리턴한다. 실패type은 pam_authenticate()가 불리는 경우에는(사용자 인증을 시도하는 경우) PAM_AUTH_ERR이고 pam_setcred()가 불리는 경우에는(사용자에게 권한을 주는 경우 -- 실제의 경우에 이 함수는 절대 안불릴것이다.) PAM_CRED_UNAVAIL이다.

예제/추천하는 사용법:

별도의 설정이 없는 응용프로그램들에 대한 접근을 이 pam_deny모듈을 사용해 막을 수 있다. 아래와 같이 Linux-PAM 설정파일에 추가하면 된다.

```
#
# OTHER 란에 이 줄을 추가해서 기본적으로 응용프로그램들의 인증이 성공하지
# 못하도록 만들어라.
#
OTHER   auth     required       pam_deny.so
```

6.4.5. 패스워드관리 구성요소

인식하는 인자들:

설명:

사용자에게 패스워드를 변경하지 못하게 한다. 항상 PAM_AUTHTOK_ERR를 리턴한다.

예제/추천하는 사용법:

이 모듈은 응용프로그램이 사용자의 패스워드를 변경하는 것을 막는데 사용한다. 예를 들어 login이 옛날 패스워드가 만료되어 새 패스워드를 자동적으로 묻게 하는 것을 막으려면 아래 내용을 설정파일에 추가해야 한다.

```
#
# 다른 로그인 항목에 이 줄을 추가해서 로그인 프로그램이 사용자 패스워드를
# 변경할 수 없도록 막아라.
#
login   password required       pam_deny.so
```

6.4.6. 세션관리 구성요소

인식하는 인자들:

설명:

호스트 컴퓨터에서 응용프로그램이 세션을 시작하는 것을 막는다.

예제/추천하는 사용법:

날짜같은 메시지를 보여주는 세션 모듈(XXX -이런 모듈은 누가 만들어야 한다)을 같이 사용하고, 이 모듈은 사용자가 shell을 시작하지 못하게 막는다. pam_motd모듈이 있다면, 아래 설정처럼 해서 사용자에게 시스템 시간을 알려주게 할 수 있다.

```
#
# 사용자의 로그인을 (정중하게) 거부하는 설정 예
#
login   session  required       pam_motd.so file=/etc/system_time
login   session  required       pam_deny.so
```

6.5. 환경 변수 설정/해제(Set/unset environment variables)

6.5.1. 개요

모듈 이름:

pam_env

저자:

Dave Kinchlea <kinch@kinch.ark.com>

유지보수:

저자

제공되는 관리 그룹:

Authentication (setcred)

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

/etc/security/pam_env.conf

네트워크 인식:

6.5.2. 모듈의 전반적인 설명

이 모듈은 환경변수를 설정/해제하는 것을 가능하게 한다. 이전에 설정한 환경변수나 PAM_RHOST같은 PAM_ITEM들을 사용하여 환경변수를 설정 하는 것을 가능하게 한다.

6.5.3. 인증관리 구성요소

인식하는 인자들:

debug; conffile=configuration-file-name; envfile=env-file-name; readenv=0|1

설명:

이 모듈은 고정 크기의 스트링(이것의 내용은 이미 설정된 환경변수나 PAM_ITEM이 될 수 있다.)을 사용하는 임의의 환경변수값을 설정/해제하게 해준다. 모든 것은 설정파일(디폴트는 /etc/security/pam_env.conf인데 conffile인자로 바꿀 수 있다.)에 의해 조정된다. 각 줄은 변수이름으로 시작하고 각 변수에 대해 DEFAULT와 OVERRIDE 두가지 옵션이 가능하다. DEFAULT는 관리자가 변수의 값을 어떤 디폴트 값으로 지정하게 하는 것이다. DEFAULT가 없으면, 빈 스트링으로 간주된다. OVERRIDE는 pam_env에게 사용되는 값이 있다면 그 값을(default값을 엎어쓰게) 넣게 하는 것이다. OVERRIDE가 사용되지 않으면, ""로 간주되고 변수값을 덮어쓰지 않는다.

```
VARIABLE   [DEFAULT=[value]]  [OVERRIDE=[value]]
```

(아마 존재하지 않는)환경 변수들이 ${string}문법을 사용해 쓰일수 있고, (아마 존재하지 않는)PAM_ITEM들이 @{string}문법을 사용해 쓸 수 있다. $, @ 문자는 백슬래쉬(\)를 앞에 붙여서 일반 문자로 사용될 수 있다. (예를 들어 \$) 값안에 큰따옴표를 사용할 수 있다.(환경 변수 이름에는 사용할 수 없다.) 값안에 공백이 필요한 경우에는 따옴표를 값 전체의 앞뒤에 붙여 사용할 수 있다. 이때에 따옴표를 값안에 넣는 방법(escaped quotes)은 지원하지 않는다.

이 모듈은 각각의 줄에 KEY=VAL형태를 갖는 단순한 환경변수 파일을 읽을 수 있다. (디폴트 파일은 /etc/environment)이다.) envfile 인자를 사용해 읽을 파일을 바꿀 수도 있고, readenv인자를 1이나 0으로 사용해 이 기능을 켜거나(1) 끌(0) 수 있다.

이 모듈의 동작은 다음의 인자들로 조정할 수 있다:

- debug - syslog(3)에 좀더 많은 정보를 기록하도록 한다.
- conffile=filename - 디폴트 설정파일은 /etc/security/pam_env.conf인데, 이 옵션을 사용해 설정파일을 바꿀 수 있다. full path를 써야한다.
- envfile=filename - 디폴트 파일 /etc/environment는 KEY=VAL형태의 변수를 바로 로드하는데 사용된다. 이 옵션은 디폴트 파일을 바꿀 수 있다. full path를 써야 한다.
- readenv=0/1 - envfile을 읽어드리는 것을 켜거나 끈다.(0은 끄기, 1은 켜기). 디폴트는 1이다.

예제/추천하는 사용법:

pam_env.conf에 더 많은 정보와 예가 있다.

6.6. The filter module

6.6.1. 개요

모듈 이름:

pam_filter

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

저자

제공되는 관리 그룹:

account; authentication; password; session

암호학적 민감성:

아직 없음.

보안 등급:

컴파일시 메시지:

리눅스 기반 시스템에서는 에러메시지 없이 깨끗하게 빌드됨.

시스템 의존성:

작동하기 위해서 시스템에 filter가 설치되어 있어야 함.

네트워크 인식:

6.6.2. 모듈의 전반적인 설명

이 모듈은 ttysnoop(XXX - 레퍼런스 필요)과 같은 프로그램의 대안으로 플러그인을 제공하기 위해 작성되었다. 이 기능을 수행하는 필터가 아직 작성되지 않았기 때문에 현재로서는 단지 장난감에 불과하다. 모듈과 함께 제공되는 유일한 필터는 간단하게 입출력 스트림에서 대문자와 소문자간에 변환하는 기능을 한다.(이 일은 termcap에 근간을 둔 에디터들에서 는 매우 귀찮은 일이 될 수 있다.)

6.6.3. 계정관리 + 인증관리 + 패스워드관리 + 세션관리 구성요소

인식하는 인자들:

debug; new_term; non_term; runX

설명:

모듈의 각 요소는 원하는 필터를 호출하는 잠재적 기능을 가진다. 필터는 항상 사용자의 권한이 아닌 호출하는 응용프로그램의 권한을 가진 execv(2)로 호출된다. 이 때문에 세션을 닫을 때를 제외하면 보통 사용자가 필터를 kill할 수 없다.

Linux-PAM 환경파일에서 전달되는 인자에 의해 모듈의 행동은 대단히 바뀔 수 있다.

- debug - 이 옵션은 모듈이 실행될 때 syslog(3)에 기록되는 정보의 양을 늘린다.

- new_term - 필터의 기본동작은 응용프로그램에 연결하기 위해 사용하고 있는 터미널을 나타내는 PAM_TTY 아이템을 설정하는 것이다. 이 매개변수는 필터된 가상 터미널로 PAM_TTY를 설정하도록 한다.
- non_term - PAM_TTY 아이템을 설정하려고 시도하지 않는다.
- runX - 모듈이 필터를 호출할 수 있도록 하기 위해 모듈이 언제 필터를 호출 할 것인지를 알아야만 한다. 이 인자는 필터가 언제 이일을 하는지 알리는데 필요하다. 뒤따르는 인자는 각각 실행되는 필터의 전체 경로명과 필터가 인식하는 명령 인자이다.

  X에 해당하는 값은 1과 2이다. 이는 필터가 실행되는 정확한 시간을 나타낸다. 이 개념을 이해하기 위해서 'Linux-PAM 모듈 개발자를 위한 가이드'를 읽는 것이 유용할 것이다. 기본적으로 각각의 관리그룹에는 모듈함수를 호출하는 두가지 방식이 있다.

  인증관리와 세션관리의 경우에는 실질적으로 분리된 두가지 함수가 있다. 인증관리의 경우 이 함수들은 _authenticate와 _setcred이다. -- 여기서 run1은 _authenticate 함수로부터 필터를 실행한다는 것을 의미하고 run2는 _setcred 함수로부터 필터를 실행한다는 것을 의미한다. 세션관리 모듈의 경우 run1은 필터가 _open_session 단계에서 호출되고, run2는 _close_session단계에서 호출됨을 나타낸다.

  계정관리의 경우 run1, run2 둘다 사용이 가능하다.

  패스워드관리의 경우 run1은 _chauthtok가 실행되는(PAM_PRELIM_CHECK상태) 첫번째 경우에 실행되는 것을 나타내는데 사용되고, run2는 두번째 경우 (PAM_UPDATE_AUTHTOK 상태)에 필터가 실행되는 것을 나타내는데 사용된다.

예제/추천하는 사용법:

글을 쓰고 있는 현재 이 모듈을 실제로 사용해 볼 만한 곳이 별로 없다. 재미삼아 다음행들을 login 환경파일에 추가해볼 수도 있다.

```
#
# 사용자가 로그인 한 후 대소문자를 바꾸도록 로그인을 설정한 예
#
login   session  required       pam_filter.so \
                        run1 /usr/sbin/pam_filter/upperLOWER
```

6.7. Anonymous access module

6.7.1. 개요

모듈 이름:

pam_ftp

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

저자

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

네트워크 인식:

사용자의 이메일 주소를 입력받음, 쉽게 속일 수 있음(XXX - 보완 작업이 필요함)

6.7.2. 모듈의 전반적인 설명

이 모듈의 목적은 쉽게 삽입가능한 익명 ftp모드 접속을 제공하는 것이다.

6.7.3. 인증관리 구성요소

인식하는 인자들:

debug; users=XXX,YYY,...;ignore

설명:

이 모듈은 사용자이름과 비밀번호를 가로챈다. 만일 이름이 "ftp"나 "anonymous" 이면, 사용자의 비밀번호는 '@' 구분자로 분리되어 PAM_RUSER와 PAM_RHOST부분에 설정된다. 이들 pam-항목들은 각각에 해당하는 값으로 설정된다. 사용자이름은 "ftp"로 설정된다. 이경우에는 모듈이 성공으로 처리하고 다른경우에는 모듈이 입력된 비밀번호로 PAM_AUTHTOK 아이템을 설정하고 실패로 처리한다.

다음의 플래그에 의해 모듈의 동작을 고칠 수 있다.

- debug - syslog(3)로 더 많은 정보를 기록한다.
- users=XXX,YYY,... - "ftp"나 "anonymouse"대신 "XXX,YYY,..."과 같이 쉼표로 분리된 사용자목록에 의해 익명 로그인을 제공한다. 만약 사용자가 이 중 하나의 이름을 입력했다면 사용자명은 그 목록의 첫번째에 나오는 "XXX"로 설정된다.
- ignore - 사용자의 이메일 주소(입력되었다면)를 무시한다.

예제/추천하는 사용법:

이 모듈 사용의 예제는 환경설정 파일 부분(4. Linux-PAM 설정 파일)에 있다. 이 모듈은 새로운/임시적인 익명로그인 계정을 제공하는데 사용될수 있다는 것을 조심하기 바란다.

6.8. The group access module

6.8.1. 개요

모듈 이름:

pam_group

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

저자

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

사용자가 접근 가능한 파일시스템의 setgid 상태에 민감하다.

컴파일시 메시지:

시스템 의존성:

/etc/security/group.conf파일이 필요하다. libpwdb가 있거나 없거나 컴파일 가능하다.

네트워크 인식:

올바르게 설정된 PAM_TTY 항목을 통해서만 인식함

6.8.2. 모듈의 전반적인 설명

이 모듈은 주어진 서비스를 요구하고 있는 사용자명과 터미널에 기반한 그룹세팅을 제공한다. 이는 time of day의 기록을 갖는다.

6.8.3. 인증관리 구성요소

인식하는 인자들:

설명:

이 모듈은 사용자를 인증하지는 않는 대신 인증모듈의 보증서(credential) 설정상태에 따라 사용자에게 그룹 멤버쉽을 부여한다. 멤버쉽은 사용자가 요구하는 서비스에 기반한다. 그룹 멤버쉽은 /etc/security/group.conf 파일에 텍스트 형태로 나열되어 있다.

예제/추천하는 사용법:

이 모듈이 올바르게 동작하기 위해서는 올바르게 설정된 /etc/security/groups.conf 파일이 존재해야 한다. 이 파일의 포맷은 다음과 같다. 그룹 멤버쉽은 설정파일 행의 조합을 만족하는 서비스 응용프로그램에 기반하여 부여된다. 각 줄('#'으로 시작하는 주석문을 제외)은 다음과 같은 문법을 갖는다.

```
services ; ttys ; users ; times ; groups
```

처음 네개의 필드는 pam_time 설정파일(/etc/security/pam_time.conf)의 문법과 같다. 마지막 필드인 groups는 쉼표(혹은 빈칸)로 구분된 그룹 선택의 텍스트 이름 목록이다. 만일 서비스에 대한 사용자 응용프로그램이 처음 네개의 필드를 만족한다면 사용자는 목록에 나열된 그룹의 멤버쉽을 부여받는다. 앞에서 언급한바와 같이 이 모듈의 유용성은 사용자가 접근가능한 파일시스템에 달려있다. 그룹의 멤버쉽을 부여받은 시점에서 사용자는 제한된 그룹의 소유권을 가진 setgid 바이너리를 만들려고 시도할 수도 있다. 후에 사용자가 이 그룹에 대한 멤버쉽을 부여받지 못했을 때 그들은 미리 컴파일 된 바이너리를 이용해 그룹 멤버쉽을 복원하려고 할 수 있다. 사용자가 접근 가능한 파일 시스템이 매우 중요한 이유는 시스템이 nosuid로 마운트 되었을 때 사용자는 그런 바이너리 파일을 만들거나 생성할 수 없다는 것이다. 이 모듈이 보안 기능을 제공하길 바란다면 사용자가 쓰기권한을 갖는 모든 파일 시스템은 nosuid로 마운트 시켜야 한다.

pam_group 모듈은 /etc/group 파일과 함께 동작한다. 사용자가 이 모듈의 동작방식에 기반하여 어떤 그룹을 부여받으면 그들은 /etc/group 파일의 내용 (또는 동등한 내용의 다른 파일)에 덧붙여 추가로 그룹의 권한을 부여받는다.

6.9. Add issue file to user prompt

6.9.1. 개요

모듈 이름:

pam_issue

저자:

Ben Collins <bcollins@debian.org>

유지보수:

저자

제공되는 관리 그룹:

Authentication(pam_sm_authenticate)

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

네트워크 인식:

6.9.2. 모듈의 전반적인 설명

이 모듈은 사용자명을 입력대기하고 있을 때 issue 파일(기본적으로 /etc/issue 파일)을 앞에 출력한다.

6.9.3. 인증관리 구성요소

인식하는 인자들:

issue=issue-file-name; noesc;

설명:

이 모듈은 사용자명을 입력대기하고 있을 때 issue파일을 앞에 출력한다. 또한 기본적으로 issue 파일에 있는 이스케이프 코드(escape code)를 파싱한다. 이 이스케이프 코드는 일반적인 getty의 것(\x 형태로 사용)과 비슷하다.

인식하는 탈출코드

- d - 현재 시간
- s - 운영체제 이름
- l - 현재 tty 이름
- m - 시스템 아키텍처(i386, sparc, powerpc, ...)
- n - 시스템의 호스트명
- o - 시스템의 도메인명
- r - 운영체제 릴리즈 번호(예. 2.2.12)
- t - 현재 시간
- u - 현재 로그인한 사용자 숫자
- U - "user"나 "users"가 뒤에 붙는 것을 제외하고는 u와 같다.(예. "1 user" or "10 users")
- v - 운영체제의 버전/빌드 날짜(예. "#3 Mon Aug 23 14:38:16 EDT 1999" on Linux)

이 모듈의 동작은 다음과 같은 플래그에 의해 변경될 수 있다.

- issue - 기본 파일을 사용하지 않으면 이 파일을 출력한다.
- noesc - 이스케이프 코드 파싱을 하지 않는다.

예제/추천하는 사용법:

login auth pam_issue.so issue=/etc/issue

6.10. The Kerberos 4 module

6.10.1. 개요

모듈 이름:

pam_kbr4

저자:

Derrick J. Brashear <shadow@dementia.org>

유지보수:

저자

제공되는 관리 그룹:

authentication; password; session

암호학적 민감성:

API를 사용함

보안 등급:

컴파일시 메시지:

시스템 의존성:

라이브러리 - libkrb, libdes, libcom_err, libkadm; 일련의 Kerberos include 파일들

네트워크 인식:

네트워크를 통해 접속한 Kerberos 키 배포 센터로부터 Kerberos 티켓을 주는 티켓을 받는다.

6.10.2. 모듈의 전반적인 설명

이 모듈은 사용자 비밀번호의 Kerberos 검증을 하고, Kerberos 티켓발급 서비스를 받기 위해서 Kerberos 티켓을 주는 티켓을 받고, 로그아웃할 때 사용자의 티켓을 파괴하고, Kerberos 비밀번호를 변경하는 인터페이스를 제공한다.

6.10.3. 세션관리 구성요소

인식하는 인자들:

설명:

모듈의 이 구성요소는 사용자의 KRBTKFILE 환경변수를 설정할 뿐 아니라 (현재로서는 이 변수를 export하는 방법은 없다) 로그아웃시에 (login에 의해 PAM_CRED_DELETE가 유지될 때까지)사용자의 티켓을 삭제한다.

예제/추천하는 사용법:

우리가 Linux-PAM 모듈 내부로부터 환경설정(environment)을 변경할 수 있을 때까지 모듈의 이 구성요소는 그리 유용하지 못할 것이다.

6.10.4. 패스워드관리 구성요소

인식하는 인자들:

use_first_pass; try_first_pass

설명:

모듈의 이 구성요소는 먼저 사용자의 옛날 패스워드를 받아 패스워드 변경 서비스의 세션키를 받는데 사용하고 다음에 새 패스워드를 그 서비스에 보내서 사용자의 Kerberos 패스워드를 변경한다.

예제/추천하는 사용법:

이것은 real Kerberos v4의 kadmind하고만 사용해야 한다. 특별한 부분이 만들어지지 않으면 AFS kaserver와 함께 사용할 수 없다. 더 많은 정보를 위해서는 모듈 작성자에게 문의하기 바란다.

6.10.5. 인증관리 구성요소

인식하는 인자들:

use_first_pass: try_first_pass

설명:

모듈의 이 구성요소는 Kerberos 서버에 티켓발급 티켓을 요청하거나 선택적으로 로컬컴퓨터의 호스트키를 얻고 만약 로컬컴퓨터에 키 파일이 존재한다면 그것을 검사함으로써 사용자의 Kerberos 패스워드를 검사한다.

이것은 또한 사용자가 나중에 사용할 티켓파일을 쓰고, 로그아웃을 할 때 티켓파일을 삭제한다. (login에서 PAM_CRED_DELETE가 불릴때까진 삭제하지 않는다)

예제/추천하는 사용법:

이 모듈은 MIT v4 Kerberos 키를 사용하는 real Kerberos서버와 함께 사용할 수 있다. 모듈이나 시스템 Kerberos 라이브러리는 AFS 방식의 Kerberos키를 지원하도록 수정할 수 있다. 현재로서는 암호학적 제한을 피하기 위해 이것을 지원하지 않고 있다.

6.11. The last login module

6.11.1. 개요

모듈 이름:

pam_lastlog

저자:

Andrew G. Morgan <morgan@kernel.org>

유지보수:

저자

제공되는 관리 그룹:

auth

암호학적 민감성:

API를 사용함

보안 등급:

컴파일시 메시지:

시스템 의존성:

/var/log/lastlog파일에 있는 내용을 사용한다.

네트워크 인식:

6.11.2. 모듈의 전반적인 설명

이 session 모듈은 /var/log/lastlog 파일을 관리한다. pam_open_session()함수를 통해 호출될 때 접속 열림 기록을 추가하고 pam_close_session()이 호출될 때 그것을 완성한다. 이 모듈은 사용자의 마지막 로그인에 관한 정보를 한줄로 표시할 수 있다. 만일 응용프로그램이 이미 이러한 작업을 수행중이라면 이 모듈을 사용할 필요는 없다.

6.11.3. 세션관리 구성요소

인식하는 인자들:

debug; nodate; noterm; nohost; silent; never

설명:

PAM 라이브러리를 사용하는 어떤 프로그램을 사용하던지 사용자가 시스템에 로그인할 때 "Last login on ..." 메시지를 제공하는데 이 모듈을 사용할 수 있다. 추가로 모듈은 /var/log/lastlog 파일을 관리한다.

이 모듈의 동작은 다음과 같은 플래그에 의해 제어할 수 있다.

- debug - syslog(3)에 더 많은 정보를 기록한다.
- nodate - 시스템에 마지막 로그인한 정보를 출력할 때 날짜 정보를 생략한다.
- noterm - 마지막으로 로그인 시도한 터미널 이름을 출력하는 것을 생략한다.
- nohost - 마지막으로 로그인 시도한 호스트명을 나타내는 것을 생략한다.
- silent - 사용자의 이전 로그인에 대한 어떠한 정보도 출력하지 않는다. 단지 /var/log/lastlog 파일에 기록하기만 한다.
- never - 만일 /var/log/lastlog 파일에 사용자의 옛날 정보가 존재하지 않으면 사용자는 이전에 로그인 하지 않았다는 것을 나타내고 "welcome..."아라는 메시지를 출력한다.

예제/추천하는 사용법:

이 모듈은 시스템에 로그인 했을 때 새로운 메일이 있다는 것을 나타내는데 사용될 수 있다. 다음은 /etc/pam.d/XXX 파일의 예제이다.

```
#
# 우리가 언제 마지막으로 여기 있었는가?
#
session optional pam_lastlog.so
```

몇몇 응용프로그램은 이 기능을 자체적으로 수행할 수도 있다. 그런 경우에 이 모듈은 필요하지 않다.

6.12. The resource limits module

6.12.1. 개요

모듈 이름:

pam_limits

저자:

Cristian Gafton <gafton@redhat.com> 이 모듈의 개선을 위해 조언해준 Elliot Lee <sopwith@redhat.com>에게 감사합니다.

유지보수:

Cristan Gafton - 1996/11/20

제공되는 관리 그룹:

session

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

/etc/security/limits.conf 파일이 필요하고 커널이 리소스 제한을 지원해야한다. 또한 libpwdb 라이브러리를 사용한다.

네트워크 인식:

6.12.2. 모듈의 전반적인 설명

이 모듈은, Linux-PAM open-session을 가로채는 것을 통해 user-session에서 얻을 수 있는 시스템 자원을 제한한다. 이 동작에 대해서는 아래에서 얘기할 설정파일을 통해 더 자세히 설명할 것이다.

6.12.3. 세션관리 구성요소

인식하는 인자들:

debug; conf=/path/to/file.conf

설명:

/etc/security/limits.conf 설정파일의 내용에 따라서 사용자 세션에서 자원제한이 설정된다. uid=0인 사용자는 이러한 제한에 영향을 받지 않는다.

이 모듈의 동작은 다음의 매개변수들에 의해 제어할 수 있다.

- debug - syslog(3)의 기록을 자세하게 한다.
- conf=/path/to/file.conf - 디폴트 파일이 아닌 제한 설정 내용을 가지는 다른 파일의 위치를 나타낸다.
- change_uid - 실제 uid를 제한사항이 설정된 사람으로 변경한다. 프로세스가 없는 사용자가 login시 shell을 포크하지 않는 것과 같은 문제가 있을 때 이 옵션을 사용하라. 이렇게 하면 누군가가 침입할 수도 있음을 주의하라.

예제/추천하는 사용법:

이 모듈을 사용하기 위해서 시스템 관리자는 먼저 root만 읽기 가능한 파일을 만들어야 한다. (기본적으로 /etc/security/limits.conf이다) 이 파일은 수퍼유저가 제한하려고 하는 사용자와 그룹의 자원제한 사항을 갖고 있다. uid=0인 계정은 어떠한 제한도 할 수 없다.

설정파일의 각 줄은 다음과 같은 형식으로 사용자별 제한을 나타낸다.

```
<domain>  <type>  <item>  <value>
```

위에 나열된 필드들은 다음과 같은 내용을 갖는다.

<domain>

- 사용자명
- @group 형태의 그룹명
- 기본엔트리를 위한 와일드카드 *

<type>은 세가지 값을 가질 수 있다.

- hard - hard 리소스 제한을 실시한다. 이 제한은 수퍼유저에 의해 설정되고 리눅스 커널에 의해 실시된다. 사용자는 이런 값을 가진 시스템 자원에 대한 요구량을 올릴 수 없다.
- soft - soft 리소스 제한을 실시한다. 이러한 제한은 존재하는 hard제한 범위내에서 사용자가 올리거나 내릴 수 있는 제한이다. 이 식별자로 표시된 값은 일반적인 시스템 사용법의 기본값으로 간주될 수 있다.
- - soft, hard 리소스 제한을 동시에 걸 때 사용한다.

<item> 다음 값들 중 하나가 될 수 있다.

- core - core파일 크기를 제한한다.(KB)
- data - 최대 데이타 크기(KB)
- fsize - 최대 파일크기(KB)
- memlock - 최대 locked-in-memory 주소공간(KB)
- nofile - 최대 열린 파일 개수

- rss - 최대 resident set 크기(KB)
- stack - 최대 스택 크기(KB)
- cpu - 최대 CPU time(MIN)
- nproc - 최대 프로새스 개수
- as - 주소공간 제한
- maxlogins - 이 사용자를 위한 로그인 최대 개수
- priority - 사용자 프로세스가 실행되는 우선순위

만일 "-" 타입을 사용하면서 item과 value값을 주지 않으면 모듈은 user/group 멤버 등에 해당하는 어떤 제한도 시행하지 않음을 주의하라. 사용자 인증이 적용되는 첫번째 엔트리는 설정파일의 다른 모든 제한 사항의 엔트리에 우선한다는 것을 주의하라. 그런 경우에 pam_limits는 항상 PAM_SUCCESS를 리턴한다.

일반적으로 사용자별 제한은 그룹의 제한에 우선한다. 그래서 만일 admin 그룹에 아무 제한을 두지 않고, 이 그룹의 멤버중 한명에 대한 제한을 등록했다면 그 사용자는 이 줄에 의해 제한을 받을 것이다.

또한 모든 제한설정 사항은 login 단위로 설정된다는 것을 주의하라. 이것들은 글로벌한 것도 아니고 영원한 것도 아니다. 단지 세션이 유지되는 동안만 존재한다.

제한 설정파일에서 '#' 문자로 시작하는 문장은 주석문이다. 그 줄의 나머지 부분은 모두 무시된다.

pam_limits 모듈은 설정파일에 문제가 생겼을 때 syslog(3)을 통해 설정문제를 보고한다.

다음은 설정파일의 예제이다.

```
# EXAMPLE /etc/security/limits.conf file:
# =======================================
# <domain> <type> <item> <value>
*               soft    core            0
*               hard    rss             10000
@student        hard    nproc           20
@faculty        soft    nproc           20
@faculty        hard    nproc           50
ftp             hard    nproc           0
@student        -       maxlogins       4
```

주의, 같은 자원(@faculty를 보라)에 soft와 hard 제한을 사용하는 것은 주어진 서비스 세션에서 사용자가 얻을 수 있는 자원의 기본값과 최대 레벨의 값을 설정한다.

리소스 제한을 필요로 하는 서비스(예를들어 login)는 /etc/pam.conf의 해당 서비스의 설정 마지막줄에 다음과 같은 줄을 추가하면 된다.(보통 pam_unix session줄 다음에)

```
#
# pam_limits를 통해서 로그인 세션에 자원 제한을 설정
#
login   session    required     pam_limits.so
```

6.13. The list-file module

6.13.1. 개요

모듈 이름:

pam_listfile

저자:

Elliot Lee <sopwith@cuc.edu>

유지보수:

Red Hat Software: Michael K. Johnson <johnsonm@redhat.com> 1996/11/18 (이게 연락이 안 되면 Elliot Lee<sopwith@cuc.edu>로 연락하라)

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

깨끗하게 컴파일됨.

시스템 의존성:

네트워크 인식:

6.13.2. 모듈의 전반적인 설명

list-file 모듈은 임의의 파일에 대해 서비스를 허가하거나 거부하는 방법을 제공한다.

6.13.3. 인증관리 구성요소

인식하는 인자들:

onerr=succeed|fail; sense=allow|deny; file=filename; item=user|tty|rhost|ruser|group|shell apply=user|@group

설명:

모듈은 다음과 같이 지정된 타입에 해당하는 항목의 값을 얻는다. -- user - 사용자명; PAM_USER tty - 요청된 터미널의 이름을 나타낸다; PAM_TTY rhost - 요청이 들어온 원격 호스트의 이름(가능하면)을 나타낸다; PAM_RHOST ruser - 요청이 들어온 원격 사용자의 이름을 나타낸다; PAM_RUSER -- 또한 file=filename에 해당하는 파일을 읽어서 나타난 항목의 인스턴스를 찾는다. filename은 나열된 아이템당 한 줄을 포함한다. 만일 아이템이 발견될때 sense=allow이면 PAM_SUCCESS가 리턴되고 권한요구는 성공하고, sense=deny이면 PAM_AUTH_ERR가 리턴되고 권한요구는 실패한다.

만일 오류가 발생하고(예를들어 filename이 존재하지 않는다거나 잘못된 인자를 만났을때) onerr=succeed이면 PAM_SUCCESS를 리턴하고 그렇지 않을때 즉, onerr=fail이면 PAM_AUTH_ERR 또는 PAM_SERVICE_ERR(적절하게)이 리턴된다.

추가적인 매개변수로 'apply='는 특정 사용자(apply=사용자명) 또는 주어진 그룹(apply=@그룹명)으로 적용을 제한하는데 사용할 수 있다. 이는 tty, rhost, shell 아이템과 함께 사용될 때만 의미가 있는 제한이다.

마지막것을 제외하면 모든 매개변수는 필수적이다. 이 모듈이 기본적으로 무엇을 해 줄것이라고 기대하지 마라. 이 모듈의 기본 동작은 언제나 변할 수 있다.

이 모듈은 어떤 보증서(credential)도 줄 수 없다.

예제/추천하는 사용법:

전통적인 'ftpusers' 인증은 /etc/pam.conf에 아래의 예를 사용하여 가능하다:

```
#
# /etc/ftpusers 파일에 존재하는 사용자는 ftp 접속을 거부
#
ftp auth     required   pam_listfile.so \
    onerr=succeed item=user sense=deny file=/etc/ftpusers
```

/etc/ftpusers파일에 있는 사용자는 (직관과는 반대로) ftp서비스를 사용할 수 없음을 유의하라.

특정 사용자만 login을 하게하려면 pam.conf를 아래와 같이 하라.

```
#
# /etc/loginusers 파일에 있는 사용자만 로그인을 허용
#
login   auth     required   pam_listfile.so \
    onerr=fail item=user sense=allow file=/etc/loginusers
```

이 예제가 제대로 동작하기 위해서는, login서비스를 허용하려는 모든 사용자는 /etc/loginusers에 열거되어 있어야 한다. 일부러 root를 login하지 않게 하는게 아니라면, root가 로그인할 수 있는 방법을 열어 놓았는지 확인해라. /etc/loginusers에 root를 넣어 놓

던지, root로 su를 할 수 있는 사용자를 넣어 놓던지 해서 root가 login할 방법을 열어 두어라.

6.14. The Mail module

6.14.1. 개요

모듈 이름:

pam_mail

저자:

Andrew G. Morgan <morgan@linux.kernel.org>

유지보수:

저자

제공되는 관리 그룹:

authentication (credential) Session(open)

암호학적 민감성:

보안 등급:

컴파일시 메시지:

깨끗함

시스템 의존성:

기본 mail 디렉토리 /var/spool/mail/

네트워크 인식:

6.14.2. 모듈의 전반적인 설명

이 모듈은 사용자의 메일 디렉토리를 보고 메일이 들어 있는지 알려준다.

6.14.3. 세션관리 구성요소

인식하는 인자들:

debug; dir=directory-name; nopen; close; noenv; empty; hash=hashcount; standard; quiet;

설명:

이 모듈은 사용자에게 "you have new mail" 서비스를 제공한다. 이는 보증서를 가로채는 어떤 응용프로그램에도 끼워 넣을 수 있다. 이는 사용자의 메일 폴더에 어떤 새로운 메일이 있음을 나타내는 간단한 메시지를 보낸다. 이 모듈은 Linux-PAM 환경변수인 MAIL을 사용자의 메일 디렉토리로 설정한다.

이 모듈의 동작은 다음 플래그들에 의해 변경할 수 있다.

- debug - syslog(3)로 더 많은 정보를 기록한다.
- dir=pathname - 기본값 대신 지정된 pathname에서 사용자의 메일을 찾는다. 메일의 기본 위치는 /var/spool/mail이다. 주의할 점은 만일 주어진 pathname이 '~'로 시작한다면 디렉토리는 사용자 홈 디렉토리에 있는 파일을 나타내는 것으로 해석될 것이라는 것이다.
- nopen - 모듈이 사용자의 보증서를 얻었을 때 어떤 메일에 대한 정보도 출력하지 않도록 한다. 이것은 MAIL환경변수의 값을 얻지만 정보를 표시하지 않을때 유용하다.
- close - 사용자의 보증서가 취소되었을 때 모듈이 사용자에게 온 메일이 있는지 알려준다.
- noenv - MAIL 환경변수를 설정하지 않는다.
- empty - 만일 사용자의 메일디렉토리가 비었다면 사용자에게 그 사실을 알린다.
- hash=hashcount - 메일디렉토리 해쉬 깊이. 예를들어 hashcount가 2라는 것은 메일파일은 /var/spool/mail/u/s/user가 되도록 만든다.
- standard - 예전 방식의 "You have..." 형태의 메시지를 보여준다. 이것은 사용되는 메일 스풀을 보여주지 않는다. 또한 이것은 'empty'를 포함한다.
- quiet - 새로운 메일이 있을 때만 알려준다.

예제/추천하는 사용법:

이 모듈은 사용자가 시스템에 로그인 했을 때 새로운 메일이 있음을 나타내는 데 사용될 수 있다. 여기 /etc/pam.conf 파일의 예제가 있다.

```
#
# 새 메일이 왔는가?
#
login session optional pam_mail.so
```

주의할 점은, 만일 메일 스풀파일(/var/spool/mail/$USER 또는 dir=parameter로서 주어지는 경로가 될 수 있다)이 디렉토리라면 pam_mail은 그것이 Qmail Maildir 형태라고 생각

한다는 것이다. 몇몇 응용프로그램은 이 기능을 자체적으로 수행할 수도 있다. 그런경우에 이 모듈은 필요하지 않다는 점을 주의하라.

6.14.4. 인증관리 구성요소

인증관리 부분은 모든 동작이 pam_setcred() 상태에서 이루어진다는 것을 제외하면 session 부분과 같게 동작한다.

6.15. 첫 로그인시 홈 디렉토리 만들기(Create home directories on initial login)

6.15.1. 개요

모듈 이름:

pam_mkhomedir

저자:

Jason Gunthorpe <jgg@ualberta.ca>

유지보수:

Ben Collins <bcollins@debian.org>

제공되는 관리 그룹:

Session

암호학적 민감성:

보안 등급:

컴파일시 메시지:

깨끗함

시스템 의존성:

네트워크 인식:

6.15.2. 모듈의 전반적인 설명

인증된 사용자를 로그인시키는 중에 홈 디렉토리를 생성한다.

6.15.3. 세션관리 구성요소

인식하는 인자들:

debug; skel=skeleton=dir; umask=octal-umask;

설명:

이 모듈은 사용자가 중앙 데이타베이스(NIS, NIS+, LDAP등)에 의해서 관리되고 여러 시스템을 통해 접근하는 분산시스템에서 유용하게 사용된다. 이는 사용자가 처음으로 성공적으로 인증된 로그인을 했을 때 사용자의 기본 홈 디렉토리를 생성함으로써 관리자가 각각의 시스템에 직접 홈디렉토리를 생성하는 수고를 덜어준다. 기본 파일을 복사하기 위해 skeleton 디렉토리(보통 /etc/skel/)가 사용되고 또한 생성시에 umask값도 설정한다.

이 모듈의 동작은 다음중 하나의 플래그들에 의해 제어된다.

- skel - 새로운 홈 디렉토리에 복사할 기본파일이 있는 skeleton 디렉토리
- umask - 쉘의 umask 명령어에게 전달할 8진수의 숫자

예제/추천하는 사용법:

```
session required pam_mkhomedir.so skel=/etc/skel/ umask=0022
```

6.16. 오늘의 메시지 출력(Output the motd file)

6.16.1. 개요

모듈 이름:

pam_motd

저자:

Ben Collins <bcollins@debian.org>

유지보수:

저자

제공되는 관리 그룹:

Session(open)

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

네트워크 인식:

6.16.2. 모듈의 전반적인 설명

이 모듈은 성공적인 로그인시에 motd 파일(기본적으로 /etc/motd파일)을 출력한다.

6.16.3. 세션관리 구성요소

인식하는 인자들:

debug; motd=motd-file-name;

설명:

이 모듈은 여러분이 성공적인 로그인 후에 임의의 motd(message of the day)를 출력할 수 있게 한다. 기본적으로 /etc/motd파일이 사용되고 어떤 파일로도 설정이 가능하다.

이 모듈의 동작은 다음 플래그들에 의해 제어된다.

- motd - 기본파일을 사용하지 않을 때 출력할 파일

예제/추천하는 사용법:

```
login session pam_motd.so motd=/etc/motd
```

6.17. The no-login module

6.17.1. 개요

모듈 이름:

pam_nologin

저자:

Michael K. Johnson <johnsonm@redhat.com> (based on code taken from a module written by Andrew G. Morgan <morgan@parc.power.net>).

유지보수:

Michael K. Johnson <johnsonm@redhat.com>

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

dropping const에 관한 warning 1개

시스템 의존성:

네트워크 인식:

6.17.2. 모듈의 전반적인 설명

표준 UNIX nologin 인증을 제공한다.

6.17.3. 인증관리 구성요소

인식하는 인자들:

설명:

표준 UNIX nologin인증을 제공한다. /etc/nologin파일이 존재하면, root만 로그인할 수 있고; 다른 사용자는 에러메시지와 함께 거부된다. 모든 사용자들(root또는 일반 사용자)은 /etc/nologin의 내용에 들어갈 수 있다.

/etc/nologin파일이 없다면, 모듈은 조용히 성공을 리턴한다.

예제/추천하는 사용법:

이 모듈이 효과적이려면, 모든 login방법들이 이것을 사용하게 해야한다. 표준 UNIX nologin동작을 제대로 하려면 sufficient모듈 앞에 required로 이 모듈을 설정해야 한다.

6.18. 무조건 신뢰 모듈(The promiscuous module)

6.18.1. 개요

모듈 이름:

pam_permit

저자:

Andrew G. Morgan, <morgan@parc.power.net>

유지보수:

Linux-PAM 유지보수자

제공되는 관리 그룹:

account; authentication; password; session

암호학적인 민감성:

보안 등급:

매우 낮음. 사용시 주의요함.

컴파일시 메시지:

없음.

시스템 의존성:

네트워크 인식:

6.18.2. 개요

이 모듈은 보안상으로 매우 위험하므로 대단히 주의해서 사용해야 한다. 이것이 행하는 일은 언제나 접근을 허용하는 것 뿐이기 때문이다. 그 외의 다른 일은 전혀 하지 않는다.

6.18.3. 계정관리 + 인증관리 + 패스워드관리 + 세션관리 구성요소

인식하는 인자들:

설명:

어떤 관리그룹에 속하건간에 이 모듈이 하는 동작은 작업이 성공적이었음을 알리는 PAM_SUCCESS를 돌려주는 일 뿐이다.

인증의 경우에는 사용자의 이름을 얻는다. 만약 이 이름이 없다면 많은 프로그램들이 작업에 혼란을 겪게 되기 때문이다.

예제/추천하는 사용법:

이 모듈은 사용하는 것은 거의 바람직하지 않다. 그렇지만 이 모듈의 사용이 적절한 경우도 있다. 예를들어 만약 시스템 관리자가 계정관리를 꺼버리지만 동시에 로그인을 허용하는 경우 다음의 설정파일 내용을 이용할 수도 있다.

```
#
# 사용자 로그인을 계속허용하지만 계정관리를 꺼버리기 위해서
# 다른 로그인 등록에 이 줄을 추가하라.
#
login   account  required       pam_permit.so
```

6.19. The Password-Database module

6.19.1. 개요

모듈 이름:

pam_pwdb

저자:

Cristian Gafton <gafton@redhat.com> and Andrew G. Morgan <morgan@kernel.org>

유지보수:

저자

제공되는 관리 그룹:

account; authentication; password; session

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

올바르게 설정된 libpwdb 필요함

네트워크 인식:

6.19.2. 모듈의 전반적인 설명

이 모듈은 pam_unix_.. 모듈의 대체판이다. 이 모듈은 http://linux.kernel.org/morgan/libpwdb/index.html에 있는 Password Database library의 일반적인 인터페이스를 따른다.

6.19.3. 계정관리 구성요소

인식하는 인자들:

debug

설명:

debug 옵션을 주면 이 모듈의 추적기능을 맡은 syslog(3)에 더 많은 정보를 기록한다. (이 모듈에서 지원되는 다른 옵션들은 에러 메시지를 내보내지 않고 조용히 무시된다. 그렇지만 그외의 옵션들을 입력한 경우 syslog(3)를 통해서 에러로 기록된다.)

이 모듈은 다음의 pwdb_element들에 기반하는데 그 pwdb_element들은 다음과 같다. expire; last_change; max_change; defer_change; warn_change. 이 모듈은 사용자의 계정과 패스워드의 상태를 확인한다. 패스워드의 상태를 확인할 때에는 사용자의 패스워드를 바꾸도록 권고하거나 PAM_AUTHTOKEN_REQD를 돌려줌으로써 새 패스워드를 입력할 때까지 서비스 실행을 늦추게 할 수 있다. 앞에서 언급한 pwdb_element들에 대한 접근은 Password Database Library Guide에 기술되어 있다. 만약 사용자의 기록이 이 모듈들중 하나 이상을 포함하지 않는다면 해당하는 shadow에 대한 검사는 하지 않는다.

예제:

계정관리 모드에서는 다음과 같이 이 모듈을 사용할 수 있다.

```
#
# 사용자의 계정과 패스워드가 아직 유효한지 확인
#
login   account  required       pam_pwdb.so
```

6.19.4. 인증관리 구성요소

인식하는 인자들:

debug; use_first_pass; try_first_pass; nullok; nodelay; likeauth

설명:

debug 옵션을 주면 이 모듈의 추적기능을 맡은 syslog(3)에 더 많은 정보를 기록한다.

이 모듈은 만약 사용자의 패스워드가 비어있다면 사용자가 서비스를 사용하는 것을 거부하는 것을 기본으로 하도록 설정되어 있다. 만약 nullok옵션을 사용한다면 이 기본동작을 무시하고 패스워드가 비어있어도 사용자가 서비스를 사용하는 것이 가능하다.

try_first_pass옵션을 사용한 경우 우선적으로 auth-module에 정의되어 있는 다른 모듈에서 사용된 패스워드를 먼저 대입해보고 실패한 경우 사용자에게 패스워드를 입력하도록 요구한다. 만약 use_first_pass를 옵션으로 사용한경우에는 이렇게 기억된 패스워드만을 사용하도록 설정되므로 사용자에게 패스워드를 입력하도록 요구하지 않는다. 만약 이 경우에 패스워드가 틀리거나 사용할 수 있는 기억된 패스워드가 아무것도 없다면 인증에 실패하게 된다.

nodelay를 옵션으로 사용한 경우에 만약 인증이 전체적으로 실패한 경우 인증모듈이 기다리지 않고 바로 처리하도록 하는 경우에 쓰일 수 있다. 기본적인 모듈의 동작은 실패한 경우 초 단위로 연기를 요청하도록 되어 있다.

이 모듈의 다른 기능에서 지원되는 남은 인자들은 조용히 무시된다. 그외의 인자들은 syslog(3)에서 에러로 기록된다.

pwdb_chkpwd프로그램은 사용자의 패스워드가 읽기를 막아놓은 database에 저장한 경우 사용자의 패스워드를 검사하기 위해서 제공된다. 이 프로그램은 매우 간단하며 이 프로그램을 실행한 사용자의 패스워드만을 검사한다. 이 프로그램은 이 모듈의 인증 부분만을 호출하며 사용자를 위해서 투명하게 실행된다. 이런 방식을 통해서 xclock같은 응용프로그램들이 setuid-root를 걸지 않고도 실행될 수 있다.

likeauth옵션을 사용하면 신임하는 모듈로 호출할때나 인증모듈로 호출할때나 같은 값을 돌려주도록 만든다. 이렇게 하도록 하면 libpam이 당신의 만든 설정파일의 설정을 인증관리 구성요소가 정상적인 경로를 따라 수행가능하다.

예제/추천하는 사용법:

이 모듈의 적절한 작동을 위해서는 /etc/pwdb.conf파일의 설정이 올바르게 되어 있어야 한다. 그 파일에 지정된 사용자 database는 인증된 사용자의 기록을 가지고 있다.

6.19.5. 패스워드관리 구성요소

인식하는 인자들:

debug; nullok; not_set_pass; use_authtok; try_first_pass; use_first_pass; md5; bigcrypt; shadow; radius; unix

설명:

pam_pwdb의 이 모듈은 사용자의 패스워드를 갱신하는 역할을 한다. libpwdb의 유연성 덕분에 이 모듈은 사용자의 패스워드를 다른 database로 옮길 수 있게 해 준다. 이것은 아마도 동적으로 사용자의 database접근을 안전하게 해주는 역할을 할 것이다. (현재 이 기능은 매우 불안정하다!) - 이것이 shadow, radius, 그리고 unix argument들의 목적이다.

보통의 unix database의 경우(암호화된 사용자 패스워드를 저장) md5옵션을 사용하는데 이것은 전통적인 crypt(3)대신 MD5 function을 사용해서 암호화한다. 이 외의 선택사항으로 DEC (Digital Equipment Cooperation)의 'C2' extension을 사용해서 8자이상을 암호화할 수 있는 bigcrypt옵션을 줄 수 있다.

nullok 모듈을 사용해서 사용자의 패스워드를 빈 것으로 만들 수 있다. 만약 nullok옵션을 사용하지 않은 경우엔 빈 패스워드는 계정을 잠그는 것과 같이 처리된다.

use_first_pass 옵션은 앞에서 사용된 패스워드 모듈에서 쓴 패스워드로 고정시킬 때 사용된다. try_first_pass옵션은 앞에서 사용되었을지도 모르는 패스워드를 다시 공유해서 사용하도록 지정해서 사용자가 패스워드를 다시 입력할 필요가 없도록 할 경우 사용할 수 있다. 만약 이전에 사용된 패스워드가 올바르지 않다면 사용자가 올바른 패스워드를

입력하도록 허용한다. use_authtok옵션은 앞서 사용된 패스워드 모듈에서 사용했던 패스워드를 새 패스워드로 지정한다. (앞서 설명된 Cracklib모듈에서 예제로 사용되었다.)

not_set_pass옵션은 현재 모듈에게 앞서 사용된 모듈이나 뒤에 사용될 모듈에서 참고로 할 옛 패스워드나 새 패스워드를 참고하거나 저장하지 않을 것임을 지정한다.

debug옵션은 이 모듈의 동작을 syslog(3)에게 더 많은 정보를 알려준다. 다른 옵션들은 syslog(3)에 의해서 에러로 기록될 수 있다.

예제:

이 모듈의 패스워드 점검 모듈인 pam_cracklib과의 적층예제는 앞선 모듈에 대한 설명에서 이미 다루어졌다.

6.19.6. 세션관리 구성요소

인식하는 인자들:

설명:

이 모듈 구성요소에서 인식하는 옵션은 아무것도 없다. 이 구성요소는 단지 사용자이름과 서비스의 종류를 syslog(3)를 통해서 기록할 뿐이다. 사용자의 세션의 시작과 끝에 대한 내용은 로그에 기록된다.

예제/추천하는 사용법:

세션 모듈에 대한 사용은 매우 직관적이다:

```
#
# pwdb - unix 와 비슷한 세션 시작과 종료
#
login   session  required       pam_pwdb.so
```

6.20. The RADIUS session module

6.20.1. 개요

모듈 이름:

pam_radius

저자:

Cristian Gafton <gafton@redhat.com>

유지보수:

저자

제공되는 관리 그룹:

session

암호학적 민감성:

이 모듈은 암호를 다루지 않음.

보안 등급:

컴파일시 메시지:

gcc 로 컴파일할 때 /usr/include/rpc/clnt.h를 컴파일하면서 Warning 1개 발생함. 이것은 내 책임이 아니다!

시스템 의존성:

네트워크 인식:

인식함. 네트워크 모듈임.(응용프로그램에 무관함)

6.20.2. 모듈의 전반적인 설명

이 모듈은 RADIUS server로 인증된 사용자들에게 세션 서비스를 제공할 목적으로 제작되었다. 현재로는 계정 서버로 RADIUS서버를 사용하는 것만 지원된다.

6.20.3. 세션관리 구성요소

인식하는 인자들:

debug - syslog(3)에 상세한 로그를 기록함.

설명:

이 모듈은 RADIUS server로 인증된 사용자들에게 세션 서비스를 제공할 목적으로 제작되었다. 현재로는 계정 서버로 RADIUS서버를 사용하는 것만 지원된다.

(PAM 프로젝트에서 이 모듈을 사용해서 RADIUS서버에 반응해 자동으로 pppd를 시작해서 사용자를 위한 PPP접속을 연다든가 다른 호스트로의 telnet접속을 한다던가

RADIUS서버에서의 인자에 따라 전화를 되걸어주는 것을 실제로 동작하게 하려면 먼저 확실하게 매듭지어야 하는 부분들이 있다. 대부분의 것들은 radius login 응용프로그램에서 처리하는 것이 보다 적합해 보인다. 이런 식으로 동작하는 Real Soon(tm) 로그인 패치를 만들기를 기대한다.)

세션을 시작할 때 "Accounting-Start" 메시지를 모듈에서 RADIUS 서버로 보내는데 이 메시지는 사용자 database에 로그기록을 하던 갱신을 하던 어떤 식으로든 작용한다. 세션을 닫을 때 "Accounting-Stop" 메시지가 RADIUS서버로 보내진다.

이 모듈은 동작하기 위해서 요구하는 선결조건이 없다. 누구든 RADIUS서버를 그냥 재미삼아 설치하고 계정관리 서버를 중앙에서 운영하면서 wtmp/last/sac같은 것들은 그냥 잊고 지낼 수 있다.

예제/추천하는 사용법:

이 모듈이 필요한 서비스(예를들어 login)를 위해서 /etc/pam.conf파일에서 그 서비스의 마지막 줄에(보통은 pam_unix session 줄 다음이다) 다음의 예를 추가하라.

```
login   session  required       pam_radius.so
```

이 모듈을 쓰는 서비스각각에 대해서 'login'대신 대체하라.

이 모듈은 libpwdb 0.54preB또는 그 이후에서 제공되는 API를 광범위하게 사용하고 있다. 기본적으로 이 모듈은 /etc/raddb/server파일에서 radius 서버에 대한 설정(hostname, secret)을 읽어들인다. 이것은 libpwdb에서 기본값으로 컴파일되어 있으며 현재로서는 libpwdb를 다시 컴파일하지 않고는 바꿀 수 있는 방법이 없다. 현재 libpwdb에서 runtime에 설정을 바꿀수 있는 radius지원하기 위해서 작업중이다.

libpwdb는 RADIUS dictionary도 필요로 함을 또한 기억하기 바란다. (/etc/raddb/dictionary파일이다)

6.21. The rhosts module

6.21.1. 개요

모듈 이름:

pam_rhosts_auth

저자:

Al Longyear <longyear@netcom.com>

유지보수:

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

에러메시지 없음.

시스템 의존성:

네트워크 인식:

표준 inet_addr(), gethostbyname() 함수 호출

6.21.2. 모듈의 전반적인 설명

이 모듈은 rlogin이나 rsh같은 프로그램에서 전통적으로 쓰던 방식으로 서비스의 네트워크 인증을 한다.

6.21.3. 인증관리 구성요소

인식하는 인자들:

no_hosts_equiv; no_rhosts; debug; no_warn; privategroup; promiscuous; suppress

설명:

이 모듈의 인증체계는 /etc/host.equiv (또는 #include <netdb.h>에 있는 _PATH_HEQUIV)와 ~/.rhosts두 파일에 있는 내용에 기반한다.

우선 앞의 파일에 나열되어 있는 호스트들은 localhost처럼 취급된다. 그리고 사용자가 가지고 있는 .rhosts파일은 "원격 호스트 원격 사용자"짝을 현재 호스트의 사용자 계정과 대응시키는데 쓰인다. 만약 사용자의 호스트가 /etc/host.equiv파일에 등록되어 있고 그들의 계정이 원격호스트와 localhost에서 일치하는 경우에는 접근이 허용된다. 또한 그들의 원격계정이 개인적인 설정파일에 등록되어 있는 경우에도 접근이 허용된다.

사용자의 개인적인 설정파일에는 몇가지 속성의 제한이 따른다. 먼저 파일은 POSIX.1의 S_ISREG(x)에서 정의된 정규파일이어야 한다. 그리고 설정파일은 superuser나 사용자가 소유한 파일이어야 하며 소유자외에는 아무도 쓰기 권한을 가지면 안된다.

이 모듈은 원격 호스트(PAM_RHOST에서 내부적으로 지정한 호스트)에서 접속한 사용자(PAM_RUSER에서 내부적으로 지정한 사용자)를 인증하는 역할을 한다. 따라서 이 인증 모듈과 호환성을 가지기 위해서는 응용프로그램 또한 pam_authenticate()를 호출하기 전에 이 항목들을 먼저 확인해야 한다. 이 모듈은 그런 정보를 확인하기 위해서 독자적으로 네트워크 접속을 확인할 수 있는 능력이 없다.

root로 접속하는 경우 hosts_equiv_rootok옵션이 사용되지 않으면 /etc/host.equiv파일은 무시된다. 대신 superuser는 올바르게 설정된 개인적인 설정파일이 있어야 한다.

모듈의 동작은 다음의 플래그에 의해서 제어할 수 있다.

- debug - syslog(3)에 더 많은 정보를 기록한다. (XXX - 실제로 이 모듈은 현재 전혀 로그를 남기지 않는다. 이것을 고치기 위한 자원자를 구하는 중이다!)
- no_warn - 사용자에게 실패등에 대해서 에러메시지를 보내지 않는다. (XXX - 이 모듈은 현재 아무런 경고도 출력하지 않는다. 역시 자원자를 구한다!)
- no_hosts_equiv - /etc/hosts.equiv 파일의 내용을 무시한다.
- hosts_equiv_rootok - superuser에 대해서도 /etc/hosts.equiv 파일을 참고한다. 이 옵션이 없는 경우 superuser에 대해서는 /etc/hosts.equiv를 참고하지 않는다. 만약 no_host_equiv옵션이 사용된 경우 host_equiv_rootok옵션은 무시된다.
- no_rhosts - 모든 사용자의 ~/.rhosts설정파일의 내용을 무시한다.
- privategroup - 정상적인 경우 ~/.rhosts 파일은 소유자외에는 아무도 쓰기 권한을 가져서는 안된다. 이 옵션은 인증되는 사용자의 이름과 그룹 소유권자의 이름이 같은 경우 그룹이 쓰기 권한을 가진 경우에도 접근을 허용한다. 이 옵션으로 인해 발생할지 모르는 보안 문제를 줄이기 위해서 이 모듈은 사용자가 그 그룹의 유일한 구성원인지도 검사한다.
- promiscuous - 호스트 등록에서 '+' 로 된 경우 모든 호스트에서의 접근이 허용된다. 이 옵션이 없다면 '+'로 쓴 등록은 무시된다. 후자의 경우 debug옵션이 켜져있으면 syslog에 경고메시지를 남기게 된다.
- suppress - 만약 인증에 실패했을 경우에도 syslog(3)에 경고메시지를 남기지 않는다. 이 옵션은 모듈이 충분한 제어 flag로 동작하는 상황에서 의미없는 에러로 가득찬 로그를 만들지 않기 위해서 사용된다.

예제/추천하는 사용법:

신뢰하는 원격 컴퓨터에서 사용자의 접속을 허용하기 위해서는 /etc/pam.conf파일에서 사용자에서 암호를 입력하기를 요구하기 전에 다음의 줄들을 추가해야 한다.

```
#
```

```
# 위에 나열된 호스트의 사용자는 패스워드를 요구하지 않는다.
#
login  auth  sufficient  pam_rhosts_auth.so no_rhosts
```

이 예제에서 시스템 관리자는 모든 사용자의 개인 rhosts 설정파일을 무시하도록 설정했음을 주목하라. 그리고 위의 예에서 sufficient를 required로 바꿈으로써 /etc/host.equiv파일에 등록된 서버들로부터만 접속할 수 있게 되어 있음을 주의해서 보기 바란다.

6.22. The root access module

6.22.1. 개요

모듈 이름:

pam_rootok

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

Linux-PAM 유지보수자

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

깨끗함

시스템 의존성:

네트워크 인식:

6.22.2. 모듈의 전반적인 설명

이 모듈은 superuser가 암호의 입력없이 서비스에 대한 접근을 하도록 허용할 때 사용하도록 만들어졌다.

6.22.3. 인증관리 부분

인식하는 인자들:

debug

설명:

이 모듈은 사용자의 uid가 0인 경우 인증에 성공한 것으로 처리한다. ustuid-root로 처리된 프로그램들은 사용자의 uid를 유지하고 있지만 euid에 있는 사용자의 권한을 가진채로 실행된다. 실제로 사용자에 대한 권한 검사가 이루어지는 것은 이 euid이다.

예제/추천하는 사용법:

su 프로그램의 경우 전통적으로 superuser가 일반 사용자의 암호를 입력하지 않고도 일반 사용자로 작업을 수행할 수 있도록 하는 기능을 사용해 왔다. Linux-PAM에서 이 기능을 쓰기 위해서는 다음의 두 줄이 설정파일에 포함되어야 한다.

```
#
# su 인증. Root 는 기본적으로 접근 허용
#
su      auth     sufficient     pam_rootok.so
su      auth     required       pam_unix_auth.so
```

주의사항) superuser에 의해서 수행되는 프로그램들이나 시스템 부팅시에 시작되는 프로그램들에는 사용자 인증에 이 모듈을 사용하면 안된다.

6.23. The securetty module

6.23.1. 개요

모듈 이름:

pam_securetty

저자:

Elliot Lee <sopwith@cuc.edu>

유지보수:

Red Hat Software: 현재 Michael K. Johnson <johnsonm@redhat.com> (만약 연락이 안될 때는 Elliot Lee <sopwith@cuc.edu>).

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

/etc/securetty 파일

네트워크 인식:

정상적으로 동작하기 위해서는 응용프로그램에서 PAM_TTY항목을 정확히 기록해야 함.

6.23.2. 모듈의 전반적인 설명

표준적인 UNIX의 securetty 검사를 제공한다.

6.23.3. 인증관리 구성요소

인식하는 인자들:

설명:

표준의 UNIX securetty검사를 해서 /etc/securetty파일에 기록된 내용과 PAM_TTY항목의 내용과 일치하는 경우에만 root계정에 대한 인증이 성공하게 된다. 그 외의 다른 사용자에 대해서는 항상 인증이 성공한 것으로 처리한다.

예제/추천하는 사용법:

다른 suffient 인증 방법보다 앞서 required 인증 방법으로 등록하는 것이 표준 사용방식이다.

6.24. The login counter(tallying) module

6.24.1. 개요

모듈 이름:

pam_tally

저자:

Tim Baverstock

유지보수:

제공되는 관리 그룹:

auth; account

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

faillog 파일(/var/log/faillog가 기본 위치다)이 필요

네트워크 인식:

6.24.2. 모듈의 전반적인 설명

이 모듈은 총 접근 시도회수를 관리하며 로그인 성공시에는 접근 시도 회수를 0으로 만들 수 있고 너무 많은 실패한 시도횟수에 도달하면 접근을 차단할 수 있다.

pam_tally는 pam_tally.so와 pam_tally의 두 개로 나뉘어 제공된다. 전자는 PAM 모듈이고 후자는 단독실행형(stand-alone) 프로그램이다. pam_tally는 (선택적인) 프로그램으로 카운터 파일을 조회할 수도 있고 관리할 수도 있다. 이 프로그램은 사용자의 카운터를 보여주고 개인별 카운트 회수를 수정하거나 모든 카운트 회수를 지우는 것도 가능하다. 이 카운트를 인위적으로 높게 설정하면 사용자의 암호를 고치지 않고도 사용자 로그인을 막을 수 있다. 예를들어 cron job으로 매일밤 자정에 모든 카운트 회수를 지워버리는 것도 유용한 설정일 수도 있다.

카운트 파일은 uid로 인덱스된 바이너리 형식의 배열이다. 만약 제공되는 프로그램을 사용하지 않을 생각이라면 od프로그램으로도 이 파일을 이해할 수 있을 것이다.

주의사항) 현재 이 모듈에는 몇가지 문제점이 남아있다. pam_tally모듈은 getpw*()에 매우 의존적이다. - 사용자의 계정에 대한 database는 훨씬 더 다양할 것이다. '현재 로그인에 대한 카운트 유지'비트는 #ifdef로 처리되어 현재는 기본적으로 지원하지 않는다. 따라서 현재는 인증에 성공하는 것만이 카운터 숫자를 초기화하는 유일한 방법이다.

6.24.3. 두 구성요소에서 모두 사용되는 일반적인 옵션

- onerr=(succeed|fail): 만약 파일을 열 수 없다든가 하는 이상한 일이 생긴다면 모듈이 어떻게 반응할 것인가?
- file=/where/to/keep/counts: 카운트를 저장할 파일 위치를 지정한다. 기본위치는 /var/log/faillog이다.

6.24.4. 인증관리 구성요소

인식하는 인자들:

onerr=(succeed|fail); file=/where/to/keep/counts; no_magic_root

설명:

이 모듈의 인증 구성요소는 시도된 로그인 카운트를 증가시킨다.

예제:

no_magic_root옵션이 쓰이면 만약 uid가 0인 사용자가 이 모듈을 호출한 경우 카운트를 증가시킨다. 시스템 관리자는 이것을 telnet/rsh/login같이 데몬이 실행시키는 서비스에 사용해야 한다. 사용자가 실행시키는 su같은 프로그램에서는 이 옵션이 빠져야 한다.

설명을 더 하자면 이미 root권한으로 돌고 있는 프로세스가 다른 서비스에 접근할 경우 이 접근은 magic이 되고 pam_tally의 검사를 받지 않는다. 이것은 root로 부터 다른 방법으로는 접근이 막혀있는 계정으로 su할 때 편리하다. 그러나 telnet이나 login같이 언제나 root권한으로 실행되는 서비스들의 경우 앞서 기술한 그대로 root(즉 전원)는 이런 magic상태가 허용되어서는 안되므로 no_magic_root옵션이 켜져있어야 하다.

6.24.5. 계정관리 구성요소

인식하는 인자들:

onerr=(succeed|fail); file=/where/to/keep/counts; deny=n; no_magic_root; even_deny_root_account; reset; no_reset; per_user; no_lock_time

설명:

계정관리 구성요소는 시도횟수에 대한 접근을 막거나 시도횟수를 초기화 할 수 있다. 또한 카운트 파일이 plain파일인지 쓰기 권한이 모두에게 주어지지 않고 적절히 유지되고 있는지를 검사한다.

예제:

deny=n 만약 사용자별 tally회수가 n을 넘어선 경우 그 사용자의 접근을 막는다. 이 옵션이 사용되면 접근 권한을 요구하는 사용자가 root이고 no_magic_root옵션이 지정되지 않았던 경우가 아니라면 reset/no_reset옵션의 기본값이 reset으로 바뀐다.

no_magic_root root로 부터의 접근을 무시하지 말고 거부하도록 만든다. 이 옵션을 telnet/rsh/login같은 데몬이 실행시키는 서비스들에 적용하라.

even_deny_root_account 옵션은 root계정을 사용불가능하게 만든다. root를 얻으려는 magic root는 이 옵션이 설정되어 있어도 통과하고 일반 유저들만 막힌다는 것을 기억해 두기 바란다.

reset 인자는 magic root의 경우이거나 아니거나 상관없이 인증에 성공했을 경우 모듈이 카운트를 0으로 초기화하도록 한다. no_reset옵션은 성공적인 로그인에도 카운터값을 초기화하지 않는다. deny가 없고 접근하는 사용자가 magic root가 아닌 상태에서는 이것이 기본으로 적용된다.

만약 /var/log/faillog 파일에서 이 사용자에 대해서 .fail_max필드가 0이 아닌 값을 가지고 있다면 per_user옵션이 전체에 공동으로 해당되는 deny=n값 대신 이 값을 사용하도록 한다.

no_lock_time 옵션은 /var/log/failog파일에 있는 .fail_locktime필드값을 사용하지 않도록 한다.

정상적인경우 Denial Of Service공격을 막기 위해서 root계정에 대한 접근 실패는 누적되더라도 root계정은 사용정지되지 않는다. 만약 사용자들은 shell계정이 주어지지 않고 root계정은 단지 su명령을 통해서나 console login을 통해서만 로그인할 수 있다면(telnet, rsh등을 통하지 않고) 이 정책은 안전하다고 할 수 있다. 만약 당신이 진정으로 특정서비스에 대해서 root가 접근하는 것을 허용하길 원하지 않는다면 even_deny_root_account옵션을 사용하도록 하라.

6.25. Time control

6.25.1. 개요

모듈 이름:

pam_time

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

저자

제공되는 관리 그룹:

account

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

/etc/security/time.conf 설정 파일 필요

네트워크 인식:

PAM_TTY 항목을 통해서만 인식

6.25.2. 모듈의 전반적인 설명

잘 정비된 시스템은 때때로 특정 서비스에 대해서는 선택적으로 접근을 허용하도록 만들어 놓기도 한다. 이 모듈은 설정파일에 있는 내용을 바탕으로 시스템에서 제공되는 서비스에 접근할 수 있는 시간을 제어할 수 있게 해 준다. 이 모듈은 사용자가 접근하는 로그인 이름, 하루 중의 시간, 요일, 접근을 요구하는 서비스, 사용자의 터미널에 대한 정보를 받아서 접근을 허가할 것인지를 판단한다.

6.25.3. 계정관리 구성요소

인식하는 인자들:

설명:

이 모듈은 /etc/security/pam.conf파일에 들어 있는 내용을 근거로 어떤 행동을 할 것인지를 결정한다. 각 규칙은 다음의 형태를 따른다.

```
services;ttys;users;times
```

설명하자면 각 규칙은 한줄로 되어 있고, newline 캐릭터가 나오거나 '#'로 표시하는 주석이 시작하는 시점에서 끝난다. 이 규칙은 세미콜론(';')으로 분리되는 네 개의 항목으로 분리된다. 각 항목은 다음과 같다.

- services - 이 규칙이 적용될 서비스의 이름
- ttys - 규칙을 적용할 터미널의 이름
- user - 적용할 사용자이름의 목록

이 목록은 다음의 규칙을 따르는 토큰의 나열인데 이들은 적절한 PAM_ 항목과 연결되게 된다. - 적어도 하나 이상의 wildcard character '*'를 가진다. - 부정을 나타내는 '!'를 선택적으로 앞에 쓸 수 있다. - 이 나열에는 다음의 두 가지 논리 연산자를 쓸 수 있다. 논리적 AND인 '&'와 논리적 OR인 '|'이다. 두 가지 예를 들어보자. !morgan&!root - 이 규칙은 morgan에게도 root에게도 적용되지 않는다. tty*&!ttyp* - 이 규칙은 콘솔 터미널에만 적용되고 가상 터미널에는 적용되지 않음을 나타낸다.

- times - 언제 이 규칙을 적용할 것인지를 지정한다. 각 부분은 날짜/시간 범위이다. 날짜는 두개의 글자의 조합으로 나타낸다. 예를들어 'MoTuSa'는 월, 화, 토요일을 나타낸다. 반복되는 날은 효력을 상쇄시킴을 주의하라. 'MoTuMo'라고 쓴 경우 실제로는 화요일만 유효하게 처리된다. 'MoWk'라고 쓴 경우 월요일을 제외한 평일이라는 뜻이다. 다음의 두글자 조합이 유효하게 처리된다.

```
Mo Tu We Th Fr Sa Su Wk Wd Al
```

마지막에서 두번째는 주말을 뜻하고 마지막 항목은 한 주의 모든 날에 해당한다.

시간 범위는 24시간 표시로 나타낸 HHMM형식에다가 하이픈'-'표시로 시작범위와 끝범위를 지정한다. 만약 끝시간이 시작시간보다 작다면 다음날의 해당시간으로 해석된다. 'Mo1800-0300'이라고 쓰면 허용되는 시간은 월요일 저녁 6시부터 다음날 새벽 3시까지라는 뜻이다.

주어진 시간제한조건은 앞의 세개의 필드의 조건이 만족되는 경우에만 적용된다.

읽기 쉽고 편리하도록 규칙은 newline chararacter로 분리해서 다음줄에 계속 쓰는 것이 가능하다.

예제/추천하는 사용법:

이 모듈의 사용은 Linux-PAM 설정 파일에서 다음과 같이 시작된다.

```
#
# pam_time 을 로그인 요청에 적용
#
login   account  required       pam_time.so
```

여기서의 예는 login 프로그램에 이 모듈을 사용하는 것이다.

/etc/security/time.conf설정파일에서 사용할 수 있는 규칙의 예는 다음과 같다.

```
login ; tty* & !ttyp* ; !root ; !Al0000-2400
```

root를 제외한 모든 사용자는 언제나 콘솔 로그인을 하는 것이 거부된다.

```
games ; * ; !waster ; Wd0000-2400 | Wk1800-0800
```

games (Linux-PAM을 사용하기 위해서 설정됨) 서비스는 근무시간대에는 사용이 불가능하다. 이 규칙은 waster라는 사용자에게는 적용되지 않는다.

현재 세션을 강제로 종료하는 데몬은 없다. 이것은 나중에 고쳐져야 할 부분이다.

잘못 설정된 규칙은 syslog(3)에 의해서 에러로 기록된다.

6.26. The Unix Password module

6.26.1. 개요

모듈 이름:

pam_unix

저자:

유지보수:

제공되는 관리 그룹:

account; authentication; password; session

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

네트워크 인식:

6.26.2. 모듈의 전반적인 설명

이것은 표준 UNIX인증 모듈이다. 계정의 정보를 얻고 쓰고 인증처리를 하기 위해서 시스템의 라이브러리를 사용한다. 보통 이 정보는 /etc/passwd파일이나 shadow가 사용되는 경우 /etc/shadow파일에서 얻어온다.

6.26.3. 계정관리 구성요소

인식하는 인자들:

debug; audit

설명:

debug 옵션의 경우 syslog(3)에 모듈의 동작에 대해서 더 많은 정보를 기록한다. 다른 구성요소가 지원하는 옵션은 기록을 하지 않고 무시하고 그 외의 옵션은 로그에 에러로 기록된다. audit옵션을 사용하는 경우 보다 더 많은 정보를 기록한다.

expire; last_change; max_change; min_change; warn_change의 shadow구성요소들에 기반하고 있는 이 모듈은 사용자의 계정과 패스워드를 확인한다. 패스워드를 확인하는 경우 사용자의 패스워드를 변경하도록 권유하거나 PAM_AUTHTOKEN_REQD를 돌려줌으로써 사용자가 새 패스워드를 입력할 때까지 서비스의 실행을 유보한다. 앞에서 기술된 내용은 GNU Libc info문서에서 기록되어 있다. 만약 이 항목중 하나이상의 기록을 읽어들일 수 없다면 해당되는 shadow의 체크를 행하지 않는다.

예제/추천하는 사용법:

계정관리 모드에서 다음과 같이 이 모듈을 실행시키는 것이 가능하다.

```
#
# 사용자의 계정과 패스워드가 유효한지 확인
#
login   account  required       pam_unix.so
```

6.26.4. 인증관리 구성요소

인식하는 인자들:

debug; audit; use_first_pass; try_first_pass; nullok; nodelay

설명:

debug 옵션의 경우 syslog(3)에 모듈의 동작에 대해서 더 많은 정보를 기록한다. 다른 구성 요소가 지원하는 옵션은 기록을 하지 않고 무시하고 그 외의 옵션은 로그에 에러로 기록된다. audit옵션을 사용하는 경우 보다 더 많은 정보를 기록한다.

이 모듈의 기본적으로 사용자의 암호가 비어있다면 사용자가 서비스를 사용하지 못하게 한다. nullok옵션을 사용하면 이 기본동작을 무효화시킨다.

try_first_pass옵션을 사용하면 사용자에게 암호를 입력하는 것을 요구하기 전에 앞선 인증모듈에서 사용된 암호를 먼저 사용해서 대입해본다. use_first_pass옵션을 사용하는 경우 이렇게 저장된 암호만을 사용하며 사용자에게 암호를 입력할 것을 요구하는 메시지를 내보내지 않는다. 만약 사용가능한 암호가 없거나 틀리다면 사용자가 서비스를 사용하는 것을 막는다.

nodelay옵션을 사용한 경우 인증전체가 실패하게 되는 경우 잠시 기다리는 것을 취소시킨다. 기본적으로 각 묘듈은 초 단위로 delay-on-failure를 요청하게 되어 있다.

이 모듈의 다른 구성 요소에서 지원되는 옵션들은 조용히 무시되고 그 외는 syslog(3)를 통해서 에러로 기록된다.

unix_chkpwd 라는 프로그램은 사용자의 암호가 읽을수 없는 장소에 보관되는 경우 사용자의 암호를 검사해주는 역할을 한다. 이 프로그램은 매우 단순하며 단지 이 프로그램을 호출한 사용자의 암호를 검사해주는 역할만 할 뿐이다. 이 프로그램은 사용자를 위해서 투명하게 작동되며 이 모듈의 인증 구성요소에 의해서 수행된다. 이러한 방식으로 xlock같은 프로그램이 setuid-root를 하지 않고도 실행되는 것이다.

예제/추천하는 사용법:

이 모듈의 정상적인 동작을 위해서는 /etc/nsswitch.conf파일이 올바르게 설정되어 있어야 한다. 그 파일에 지정된 사용자의 database는 인증된 사용자의 기록을 보관하고 있다.

인증 모드에서는 다음과 같이 사용할 수 있다.

```
#
# 사용자를 인증
#
login   auth  required       pam_unix.so
```

6.26.5. 패스워드관리 구성요소

인식하는 인자들:

debug; audit; nullok; not_set_pass; use_authtok; try_first_pass; use_first_pass; md5; bigcrypt; shadow; nis; remember

설명:

pam_unix모듈의 이 구성요소는 사용자의 암호를 갱신하는 역할을 한다.

통상적으로 쓰이는 암호화된 사용자암호를 저장하는 UNIX database의 경우 암호화 처리를 할 때 전통의 crypt(3)를 사용하지 않고 md5옵션을 사용해서 MD5 함수를 호출한다. 8자 이상을 암호화 처리해서 저장하기 위해서는 이 외에도 bigcrypt옵션을 사용하여 표준 UNIX의 crypt 알고리즘을 확장한 DEC의 'C2'확장함수를 이용할 수 있다.

nullok 옵션을 사용하면 암호를 공란으로 비워두도록 바꾸는 것을 허용한다. 이 옵션이 없는 경우 암호가 없는 것은 계정을 잠그는 것으로 해석된다.

use_first_pass옵션은 옛날 암호와 새로운 암호를 앞선 모듈에서 사용된 암호로 고정시키는 역할을 한다. try_first_pass옵션이 사용된 경우 앞서 사용된 사용자의 옛 암호를 이용해서 사용자가 옛날 암호를 다시 입력할 필요가 없도록 한다. 만약 이 옛날 암호가 바르지 않다면 사용자는 옛날 암호를 입력하라는 메시지를 받게 된다. use_authtok옵션을 사용한 경우 이 모듈은 새 암호를 앞서 사용된 암호처리 모듈에서 사용한 암호로 만든다. (이 경우는 앞서 Cracklib모듈을 적층하는 예제에서 설명했다.)

not_set_pass옵션은 다른 암호처리 모듈과 옛날 암호나 새 암호를 주거나 받는 것에 대해서 전혀 신경쓰지 않도록 지시한다.

debug 옵션은 이 모듈의 암호처리 부분의 동작에 대해서 syslog(3)에 더 많은 정보를 기록하도록 한다. 다른 옵션들은 syslog(3)에 의해서 에러로 기록될 수도 있다. audit옵션을 사용하면 더 많은 정보를 기록한다.

nis옵션은 pam_unix모듈은 새 암호를 설정하는데 NIS RPC를 이용해서 시도할 것을 지시한다.

remember옵션은 하나의 값을 받아들인다. 이 값은 각 사용자별로 가장 최근에 사용된 암호의 갯수다. 이 암호들은 /etc/security/opasswd에 저장되며 사용자의 암호 변경 이력을 조사해 같은 암호를 너무 자주 바꾸어서 사용하는 것을 막는다.

예제/추천하는 사용법:

표준 사용방법:

```
#
# 사용자 암호를 바꿈
#
passwd   password   required   pam_unix.so
```

pluggable 암호 검사 모듈인 pam_cracklib과 함께 적층해서 사용하는 예이다.

```
#
# 사용자 암호를 바꿈
#
passwd   password   re-
quired   pam_cracklib.so retry=3 minlen=6 difok=3
passwd   password   required   pam_unix.so use_authtok nullok md5
```

6.26.6. 세션관리 구성요소

인식하는 인자들:

설명:

이 구성요소에서 받아들이는 옵션은 아무것도 없다. 단지 사용자의 이름과 서비스의 종류를 syslog(3)에 기록하는 역할을 할 뿐이다. 사용자의 세션 시작과 끝을 기록한다.

예제/추천하는 사용법:

세션 모듈을 사용하는 것은 직관적이다.

```
#
# 세션 시작과 종료
#
login   session  required       pam_unix.so
```

6.27. The userdb module

6.27.1. 개요

모듈 이름:

pam_userdb

저자:

Cristian Gafton <gafton@redhat.com>

유지보수:

저자

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

Berkeley DB가 필요함.

네트워크 인식:

6.27.2. 모듈의 전반적인 설명

.db database에서 사용자를 검색하고 그 database에서 사용자의 암호를 검사함

6.27.3. 인증관리 구성요소

인식되는 인자들:

debug; icase; dump; db=XXXX;

설명:

이 모듈은 사용자명과 암호를 Berkely DB database에서 검사하는 역할을 한다. database는 사용자이름으로 정렬되어 있으며 사용자명의 키에 해당하는 자료는 암호화되지 않은 상태로 보관되고 있는 사용자 암호이다. 그러므로 DB database자체의 접근 권한에 대해서 주의를 기울여야 한다.

이 모듈은 사용자의 암호를 전통적인 방식으로 읽어들인다. 만약 당신이 이 모듈을 다른 인증 모듈(pam_pwdb같은 모듈)보다 먼저 사용하는 경우에는 그 모듈이 입력된 암호를 이 모듈에서 만드는 항목인 PAM_AUTHTOK로 부터 읽어들이도록 만들어야 한다.

이 모듈의 동작은 /etc/pam.d/<service>파일에 있는 설정에 따라 제어된다.

- debug - syslog(3)에 더 많은 정보를 제공한다.
- icase - 암호를 대소문자 구별하지 않고 검사한다.
- dump - database의 모든 내용을 log에 기록한다. (이것을 기본동작으로 만들지 말라!)
- db=XXXX - XXXX라는 위치에 있는 database파일을 사용한다. 보통 Berkeley DB는 필요한 확장자를 자동으로 붙여주므로 /etc/foodata.db대신 /etc/foodata만 써 주어야 한다.

예제/추천하는 사용법:

이것은 /tmp/dbtest.db파일에 저장된 사용자의 로그인이름과 암호를 검사해 맞으면 로그인을 허용하는 보통의 ftp설정 파일이다. (이 파일은 보통 대부분의 시스템에서 /etc/pam.d/ftp에 있다)

```
#%PAM-1.0
auth       re-
quired     pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed
auth       sufficient   pam_userdb.so icase db=/tmp/dbtest
auth       required     pam_pwdb.so shadow nullok try_first_pass
auth       required     pam_shells.so
account    required     pam_pwdb.so
session    required     pam_pwdb.so
```

6.28. Warning logger module

6.28.1. 개요

모듈 이름:

pam_warn

저자:

Andrew G. Morgan <morgan@parc.power.net>

유지보수:

저자

제공되는 관리 그룹:

authentication; password

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

네트워크 인식:

원격 사용자와 호스트에 대한 정보를 로그에 기록(만약 pam-item을 알고 있으면)

6.28.2. 모듈의 전반적인 설명

이 모듈은 제안된 인증에 대한 정보를 기록하거나 프로그램이 암호를 갱신하도록 하는 것을 목적으로 만들어졌다.

6.28.3. 인증관리+패스워드관리 구성요소

인식하는 인자들:

설명:

서비스, 터미널, 사용자, 원격사용자, 원격호스트를 syslog(3)에 기록한다. 각 항목들은 조사해서 얻는 것이 아니라 표준의 pam-item들을 읽어들인다.

예제/추천하는 사용법:

앞의 설정 파일부분(4. Linux-PAM 설정파일)에서 설명되어 있음

6.29. 휠 그룹 모듈(The wheel module)

6.29.1. 개요

모듈 이름:

pam_wheel

저자:

Cristian Gafton <gafton@redhat.com>

유지보수:

저자

제공되는 관리 그룹:

authentication

암호학적 민감성:

보안 등급:

컴파일시 메시지:

시스템 의존성:

libpwdb 필요함.

네트워크 인식:

6.29.2. 모듈의 전반적인 설명

root권한을 얻을 수 있는 자격을 wheel(gid=0) 그룹의 사용자에게만 준다.

6.29.3. 인증관리 구성요소

인식하는 인자들:

debug; use_uid; trust; deny; group=XXXX

설명:

이 모듈은 소위 wheel그룹을 사용하도록 한다. 이 모듈은 기본적으로는 root권한을 얻을 수 있는 자격을 사용자가 wheel그룹에 속한 경우에만 가능하도록 만든다. (우선 이 모듈은 'wheel'그룹이 존재하는지를 먼저 검사한다. 만약 없다면 이 모듈은 group id가 0인 그룹을 wheel 그룹으로 지정한다)

이 모듈의 동작은 /etc/pam.conf파일에 정의된 내용에 따라서 제어할 수 있다.

- debug - syslog(3)에 더 많은 정보를 제공한다.
- use_uid - 이 옵션을 사용하면 사용자의 getlogin(3)이름을 이용하지 않고 현재 프로세스의 uid를 이용한다. 이 옵션은 예를 들어 'su'명령어 같은 경우와 같이 한 계정에서 다른 계정으로 뛰어 넘는 경우 유용하다.
- trust - 이 옵션이 사용된 경우 root권한을 요구한 사용자가 wheel그룹에 속한 경우 PAM_SUCCESS를 돌려주도록 한다. 기본동작은 이런 경우에 PAM_IGNORE를 돌려주는 것이다. 이 옵션을 사용하면 wheel group사용자들이 암호를 입력하지 않고도 root권한을 획득할 수 있게 만든다. 주의해서 사용하기 바란다!
- deny - 모듈의 동작을 반대로 하도록 만든다. 만약 사용자가 wheel그룹에 속한 사용자가 uid=0을 얻기 위한 접근을 한다면 접근을 거부한다. (아마도 이것은 터무니없는 짓일 것이다.) 이 옵션은 'group= '옵션과 같이 사용되기 위해서 만들어졌다.
- group=XXXX - gid=0 인 그룹을 검사하는 대신 인증을 위해서 XXXX그룹을 검사한다. 여기서 XXXX는 그룹의 이름이지 숫자로 된 그룹의 id가 아니다.

예제/추천하는 사용법:

superuser 권한을 얻을 수 있는 권한을 wheel group 에만 허용하려는 경우 다음의 내용을 설정파일에 기록하면 된다.

```
#
# root 는 기본적으로 접근이 허용된다 (rootok), 오직 wheel 그룹의 사용자만
# root (wheel) 가 될 수 있지만 root가 아닌 사용자는 Unix 인증을 거친다.
#
su      auth     sufficient     pam_rootok.so
su      auth     required       pam_wheel.so
su      auth     required       pam_unix_auth.so
```

7. 파일들(Files)

/usr/lib/libpam.so.*

응용프로그램들이 **Linux-PAM**에 접근하게 해 주는 공유 라이브러리

/etc/pam.conf

Linux-PAM 환경설정 파일

/usr/lib/security/pam_*.so

Linux-PAM의 동적으로 로드가능한 오브젝트파일(모듈)들의 기본 위치

8. 관련 참고(See Also)

- The Linux-PAM Application Writers' Guide.
- The Linux-PAM Module Writers' Guide.
- The V. Samar and R. Schemers (SunSoft), "UNIFIED LOGIN WITH PLUGGABLE AUTHENTICATION MODULES", Open Software Foundation Request For Comments 86.0, October 1995. 이 url을 보라: **http://www.kernel.org/pub/linux/libs/pam/pre/doc/rfc86.0.txt.gz**

9. 주의, 참고사항(Notes)

원래 여기에는 개발 참고사항들을 넣고자 했었다... '현재 이것은 실제로는 지원되지 않습니다.' 같은 것들 말이다. 문서의 발표시 여기 있던 내용들은 아래에 있는 버그란(Bugs)에 옮겨질 것이다! :)

use_mapped_pass 모듈인자를 지원하는 것이 가능할 것인가? 누구 싼(혹은 공짜로 쓸 수 있는) 좋은 변호사 아는 사람 없는가?!

- Sun에서 새로운 관리 그룹을 추가하는 것을 검토했기 때문에 이 내용은 사라질지도 모른다. 이 방식으로는 libpam이 강력한 암호화를 거쳐서 안전하게 패스워드를 보관하는 모듈을 사용할 수 있고, 이런식으로 되면 이 모듈들은 Linux-PAM과 같이 배포할 필요가 없을 것이다.

10. 저자/감사인사(Author/acknowledgments)

이 문서는 Andrew G. Morgan (morgan@kernel.org)가 썼고 다음과 같은 사람들이 제공한 내용들이 포함되었다:

Sun Microsystems에 감사를 드리며 특히 Vipin Samar 와 Charlie Lai가 충고해 준 것에 대해서 감사한다. **Linux-PAM**의 개발 초기 단계에서, Sun은 고맙게도 그들의 PAM 구현에 대한 문서를 볼 수 있게 해 주었다. 이것으로 **Linux-PAM**의 개발이 크게 앞당겨질 수 있었다.

11. 버그/탈자(Bugs/omissions)

언제나 PAM모듈들이 개발되고 있다. 이 문서가 최신정보가 되는 것은 거의 불가능하다!

이 메뉴얼은 완성되지 않았다. 좋은 일을 한 많은 사람들중 단지 일부만이 여기에 나열되었다.

12. 이 문서에 대한 저작권(Copyright information for this document)

Copyright (c) Andrew G. Morgan 1996-9. All rights reserved. Email: **<morgan@linux.kernel.org>**

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Alternatively, this product may be distributed under the terms of the GNU General Public License (GPL), in which case the provisions of the GNU GPL are required **instead of** the above restrictions. (This clause is necessary due to a potential bad interaction between the GNU GPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.