User Authentication HOWTO

Peter Hernberg

petehern (at) yahoo.com

도운이

성윤기

yune (at) netian.com

2000/05/02(번역: 2001/09/14)

이문서는 사용자와 그룹정보가 저장되는 방법, 사용자가 리눅시스템(PAM)에서 인증되는 방법 그리고 시스템의 사용자 인증을 안전하게 하는 방법을 설명한다.

1. 소개

1.1. 어떻게 이문서가 만들어졌나?

내집에 있는 홈네트워크에 많은 네트워크 서비스(대부분 필요없는 서비스)를 추가하려고 하니까, 나는 계속 인증문제에 부딪히게 되었다. 그래서 나는 리눅스 시스템에서 어떻게 인증이 동작하는 지 알아보기로 하였다(즉 HOWTO 문서를 작성). 그리고 이것을 나의 졸업프로젝트로 잡았다. 나는 이 문서가 여러분들에게 시스템을 관리할 때 종종 잊고 있지만 아주 중요한 부분을 이해하도록 도움이 되기를 기대한다.

1.2. 새로운 버젼

내가 내 도메인을 정상적으로 가동시킬 때, 여러분들은 이 문서의 가장 새로운 버전을 찾을 수 있을 것이다. 그 때까지는 http://www.linuxdoc.org/면 충분하다.

1.3. 의견

의견, 수정, 제안, 열정, 번쩍이는 생각은 petehern@yahoo.com 로 보내주세요.

1.4. 버전 역사

v0.1 (May 13, 2000) 첫 판(발표되지 않았음).

v0.3 (May 14, 2000) 개정판 (발표되지 않았음).

v0.5 (May 15, 2000) PAM 보안 추가, 자료관련 절 추가(발표되지 않았음).

v0.7 (May 15, 2000) 개정판 (발표 준비).

1.5. 저작권과 상표권

1.6. 감사의 말

18년동안 나를 인내해주신 내 가족에 감사한다. 이렇게 멋진 배포판을 만들도록 도와준 데비안 친구들에게 감사한다. 나를 괴짜 geek로 인정해준 CGR에 대해서 감사한다. Sansy Harris의 도움이 되는 제안에 감사한다. 마지막으로 라면을 만든사람에게 감사하고 싶다. 왜냐하면 그것없이 어떻게 살지 모르기 때문이다.

1.7. 독자에 관한 가정

이문서의 목적에 맞게, 독자들은 커맨드라인에서 명령어를 실행할 수 있고 텍스구성파일을 편집할 수 있다고 가정한다.

2. 어떻게 사용자 정보가 여러분의 시스템에 저장되는가

2.1. /etc/passwd

대부분의 리눅스 배포판(그리고 상용 *nixes도 마찬가지)에서는, 사용자정보는 /etc/passwd에 저장이 된다. /etc/passwd는 사용자 로그인과 암호화된 패스워드, 유일한 사용자 ID, 그룹ID(gid라고 부른다), 선택사항인 코멘트 필드(보통 사용자의 실제 이름과 전화번호등이 포함된다), 사용자의 홈디렉토리, 사용자가 사용하는 쉘들을 포함하는 텍스트 파일이다. 전형적인/etc/passwd 한 줄은 아래와 같다:

```
pete:K3xcO1Qnx8LFN:1000:1000:Peter Hernberg,,,1-800-
FOOBAR:/home/pete:/bin/bash
```

여러분들이 보듯이 이것은 꽤 솔직하다. 각각의 사용자들은 내가 위에서 설명한 여섯 개의 필드로 구성되어 있고, 각 필드는 콜론으로 구분되어 있다. 만약에 이것이 사용자 인증만큼 복잡하다면 이 HOWTO 문서는 필요가 없을 것이다.

2.2. Shadow passwords

/etc/passwd를 봐라. 여러분들은 실제 아래와 같은 것을 보았을 것이다:

```
pete:x:1000:1000:Peter Hernberg,,,1-800-FOOBAR:/home/pete:/bin/bash
```

암호화된 패스워드는 어디로 갔지? 그것의 위치를 말하기 전에 약간의 설명이 필요하다.

모든 사용자의 정보와 암호화된 패스워드를 가지고 있는 /etc/passwd 파일은 모든 사용자에 의해 읽혀질 수 있고 어떤 사용자라도 시스템의 모든사용자의 암호화된 패스워드 파일을 가져갈 수 있게 되어있다. 패스워드는 암호화되어 있더라도 패스워드 푸는 프로그램들은 굉장히 많이 널려있다. 이렇게 증가하는 보안위협에 대처하기 위해 쉐도우패스워드가 개발되었다.

시스템이 쉐도어 패스워드를 사용한다면 /etc/passwd의 패스워드필드는 "x"로 교체되고 사용자의 실제 암호화된 패스워드는 /etc/shadow 파일안에 저장된다. /etc/shadow 는 root사용자만 읽을 수 있기 때문에 악의적인 사용자는 다른 사용자의 패스워드를 크랙할 수 없다. /etc/shadow파일의 각각의 목록들은 사용자의 로그인 ID, 암호화된 패스워드, 그리고 패스워드 변경과 관련된 필드를 포함하고 있다. 전형적인 하나의 목록은 아래와 같다:

```
pete:/3GJllg1o4152:11009:0:99999:7:::
```

2.3. /etc/group 과 /etc/gshadow

그룹정보는 `/etc/gorup`파일에 저장이 된다. 형식은 `/etc/passwd`와 형식과 마찬가지로 그룹명, 숫자로 된 그룹 ID, 그리고 콤마로 구분된 그룹멤버들의 리스트들을 포함하고 있다. `/etc/group`파일의 하나의 목록은 다음과 같다:

```
pasta:x:103:spagetti,fettucini,linguine,vermicelli
```

패스워드 필드에 "x"가 보이듯이 그룹패스워드는 마찬가지로 쉐도우파일을 사용한다. 그룹은 대개 거의가 자신들의 패스워드를 가지고 있지 않지만, 쉐도우된 그룹패스워드정보가 /etc/gshadow 파일에 저장되고 있다는 것을 알릴만한 가치는 있다.

2.4. MD5로 암호화된 패스워드

전통적으로 유닉스의 패스워드는 표준 crpyt()함수를 이용해서 암호화하였다. (crypt()에 대한 더 많은 정보는 crypt(3) manpage를 보면된다). 컴퓨터가 더 빨리 성장함에 따라 이 함수로 암호화된 패스워드는 쉽게 깨졌다. 인터넷이 등장하면서 다중호스트를 사용한 패스워드 크랙킹을 위해 분산 작업도구들이 유용하게 되었다. 많은 더욱강한 MD5 해쉬알고리즘으로 패스워드를 암호화는 옵션들을 가지는 많은 새로운 리눅스 배포판이 나오고 있다(MD5에 대해서 더 알고싶으면 RFC 1321을 참고하라). MD5 패스워드가 패스워드 크래킹의 위협을 제거해주지는 않지만 패스워드파일을 크래킹하는 것을 더욱 어렵게 만들것이다.

2.5. 엉망진창에서 걸러내기

여러분들도 아시다시피, 서로 다른 많은 방법으로 사용자 인증정보들이 여러분들의 시스템에 저장될 수 있다(MD5 암호화 없는 쉐도우 패스워드, MD5로 암호화한 /etc/passwd의 패스워드). login이나 su 같은 프로그램들이 어떻게 당신의 패스워드를 확인할까? 더 나쁜경우에 여러분들의 시스템에 패스워드가 저장되는 방법을 바꾸기를 원한다면 어떻게 될까? 여러분들의 패스워드가 필요한 프로그램들이 패스워드가 다르게 저장되어 있다는 것을 어떻게 알까? PAM이 그 정답이다.

3. PAM (Pluggable Authentication Modules)

PAM은 현대 리눅스 배포판에서 사용자 인증의 핵심이다.

3.1. 왜

옛날 리눅스가 좋은 시절에는 su, passwd, login, xlock와 같은 프로그램들은 사용자 인증이 필요할 경우 간단히 `/etc/passwd` 파일에서 정보를 읽었다. 만약에 사용자 패스워드를 변경할 필요가 있을 경우, 간단히 `/etc/passwd`를 편집했다. 이러한 간단하나 투박한 방법은 시스템 관리와 응용프로그램 개발자들에 많은 문제점을 제공했다. MD5와 쉐도우 패스워드가 굉장

히 유행함에 따라 다른 방법을 다루고자 할 때 사용자 인증을 필요로 하는 각각의 프로그램 적절한 정보를 얻는 방법을 알아야 했다. 만약에 여러분들이 사용자 인증 방법을 변경하고자 한다면 모든 프로그램들은 새로 컴파일 해야 했다. PAM은 사용자정보의 저장방법과 관계없이 프로그램들이 투명하게 사용자를 인증하게 함으로써 이러한 혼잡함을 제거했다.

3.2. 무엇을

리눅스-PAM 시스템 관리자 설명서 (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html)에 설명하기를 "리눅스-PAM프로젝트의 목적은 안전하고 적합한 인증방법의 개발과 소프트웨어에게 부여한 특권의 개발을 분리하는 것이다"라고 되어 있다. 이것은 사용자가 인증되어야 한다는 요청하기 위해 어떤 응용프로그램이 사용할 수 있는 함수의 라이브러리를 제공함으로써 가능하다. PAM을 사용하면 여러분들의 패스워드가 /etc/passwd 에 저장되었든지 또는 홍콩의 서버에 저장되었든지 중요하지 않다. 프로그램이 사용자 인증을 필요로 하면, PAM이 적절한 인증방법에 대한 함수를 포함하는 라이브러리를 제공한다. 이 라이브러리는 동적으로 할 당되기 때문에 인증스킴을 변경하는 것은 간단히 구성파일을 편집하면 된다.

PAM은 가장 큰 장점은 유동성이다. PAM은 사용자를 인증하는 사용자를 인증하는 어떤 프로그램의 권리를 부정하도록 구성될 수 있다. 어떤 사용자들에게 인증되도록 허가할 수 있고, 어떤 프로그램이 인증을 시도할 때 경고를 줄 수 있고, 모든 사용자들의 로그인 권한을 없앨 수도 있다. PAM의 모듈 디자인으로 인해 사용자 인증방법에 대해 확실한 통제를 할 수 있도록 한다.

3.2.1. PAM을 지원하는 리눅스 배포판

거의 모든 유행하는 배포판은 당분간 PAM을 지원한다. 여기에 PAM을 지원하는 불충분한 리눅스 배포판의 리스트가 있다:

- 버전 5.0 이후의 RedHat
- 버젼 5.2 이후의 Mandrake
- 버젼 2.1이후의 Debian (부분적으로 2.1도 지원 -- 2.2에는완벽하게 지원)
- 버젼 1.3 이후의 Caldera
- 버젼 3.6 이후의 Turbolinux
- 버젼 6.2 이후의 SuSE

이 목록은 확실히 불충분하고 아마도 부정확하다. 여러분들이 이목록에 대해 수정또는 추가할 사항을 `<petehern (at) yahoo.com>`로 보내주면 고맙겠다.

3.2.2. PAM 설치하기

아무생각없이 PAM을 설치하면 HOWTO의 범위를 벗어나서 시간이 오래 걸린다. PAM이 여러분의 시스템에 설치되어 있지 않다면 여러분들은 아마 오래된 리눅스를 운영하고 있으며

이것을 업그레이드 되어야 한다. 여러분들이 혼자선 해보고 싶다면 나의 도움이 필요한 사람이 아니다. 이러한 이유로 나는 여러분들은 이미 PAM이 설치되어 있다고 가정한다.

3.3. 어떻게

할 이야기가 충분하다. 그것을 해보자

3.3.1. PAM 구성파일

PAM 구성파일은 `/etc/pam.d` 디렉토리에 저장되어 있다(여러분이 `/etc/pam.d/` 디렉토리가 없다면 걱정하지 마라. 다음 절에서 다룰 것이다.) 거기로 가서 살펴보자.

```
~$ cd /etc/pam.d
/etc/pam.d/$ ls
chfn chsh login other passwd su xlock
/etc/pam.d/$
```

어떤 시스템인가에 따라 다르지만 여러분의 시스템에는 이 디렉토리에 다소 많거나 적은 파일들이 있을 것이다. 자세한 사항은 어떤지 간에, 시스템의 각각의프로그램에 대한 사용자들은 인증하는 파일을 보았을 것이다. 아마 이미 눈치챗듯이, 각각의 파일들은 그 이름을 딴 프로그램에 대한 PAM 인증구성을 포함하고 있다(`other`파일은 예외다. 이것은 나중에 조금 다룰 것이다). passwd 파일에 대한 PAM 구성을 보자(아주 간단하게 하기 위해 요약을 했다.)

```
/etc/pam.d/$ cat login
# PAM configuration for login
auth       requisite  pam_securetty.so
auth       required   pam_nologin.so
auth       required   pam_env.so
auth       required   pam_unix.so nulok
account    required   pam_unix.so
session    required   pam_unix.so
session    optional   pam_lastlog.so
password   required   pam_unix.so nullok obscure min=4 max=8
```

이 파일을 알아보기 전에 조금만 언급해야 겠다.

3.3.2. 조금

아주 일부의 사람들은 이렇게 생각할 것이다." 어 나는 /etc/pam.d 디렉토리가 없어!" 리눅스 목록에는 PAM을 가지고 있다고 적혀 있는데 나는 그 디렉토리를 찾을 수 없다. PAM없이는 내인생은 공허하고 무의미해! 어떻게 해야 하지?" 걱정하지 마라. 모든 것을 잃은 것은 아니다. 여러분이 리눅스에는 PAM이 포함되어 있다고 하는 데 `/etc/pam.d` 디렉토리가 없다면

PAM구성은 `/etc/pam.conf`에 저장되어 있다. 여러개의 파일에 걸쳐 퍼져 있는 것이 아니라 모든 PAM 구성은 하나의 파일에 저장되어 있다. 이것으로 인해 PAM 구성이 약간 꼬이게 된다. 그러나 3.3.2절에 적절하게 고치는 법을 설명해 놓았다.

3.3.3. 구성문법

PAM 구성파일은 다음과 같은 문법을 가지고 있다:

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

로그인 구성파일(위를 봐라)을 예로 해서 PAM 구성파일에 대한 문법에 대해서 알아보자:

PAM 구성토큰

type

타입토큰은 PAM에 이 모듈에 어떤 타입의 인증이 사용될 것인지를 알려준다. 같은 타입의 모듈은 "쌓일" 수 있고, 사용자에 인증되기 위한 다중 요구사항을 만족하도록 요청할 수 있다. PAM은 네개의 타입을 인식한다.

account

계정은 사용자가 해당 서비스에 접근이 허용되었는지, 패스워드가 기간이 만료가 되었는지를 결정한다.

auth

주로 패스워드를 통하지만 생체인증과 같은 보다 정교한 방법을 통해서 사용자가 자신이 주장하는 사용자가 맞는지 결정한다.

password

패스워드는 사용자가 그들의 인증을 변경하도록 어떤 방법을 제공한다. 다시 이것은 주로 패스워드이다.

session

사용자가 인증받기 전에 또는/그리고 후에 되어야 할 것이다. 이것은 사용자 홈 디렉토리를 마운팅/언마우팅하는 것과 로그인/로그아웃 그리고 사용자에게 제공하는 서비스를 제한/제공하는 것과 같은 것을 포함할 수 있다.

로그인 구성 파일에는 각각의 타입하나에 적어도 한 개의 목록을 볼 수 있다. 이것 때문에 프로그램이 사용자들에 로그인을 허용한다. 모든 다른타입의 인증에 접근할 필요가 있다는 것을 이해할 수 있다.

control

통제 토큰은 이 모듈이 동작하지 않는다면 PAM에게 무엇을 해야할 지 알려주는 것이다. PAM은 네가지의 통제 형식을 인식한다.

requisite

이 모듈을 이용하는 인증이 실패할 경우, 즉시 인증을 거부하도록 한다.

required

인증이 거부되기 전에 비록 PAM이 이 서비스에 등록된 다른 모든 모듈들을 요구함에도 불구하고 실패할 경우 인증을 거부하도록 한다.

sufficient

비록 이전에 요청되어진 모듈이 실패하더라도 이 모듈에 의해서 인증이 성공할 경우, PAM은 인증을 승인한다.

optional

이 모듈이 성공 또는 실패하는 지는 그 모듈이 서비스에 대한 형식에 대한 유일한 모듈일 경우에 중요하다.

로그인에 대한 구성파일에서 거의 모든 통제타입이 다르다는 알 수 있다. 대부분의 요청되어지는 모듈들은 pam_unix.so(주요 인증 모듈)이고, 단 한 개의 requitsite 모듈은 pam_securitty.so 이다(사용자가 안전한 콘솔에 로그인한다는 것을 확인하는 것이다). 그리고 유일한 선택모듈은 pam_lastlogin.so 이다(사용자의 가장 최근 로그 정보를 가지고 오는 모듈)

모듈-경로

모듈경로는 PAM에게 어떤 모듈을 사용할 것인지(선택적으로) 그리고 그것을 어디서 찾을 지를 알려준다. 대부분 구성은 로그인 구성파일의 경우와 마찬가지로 모듈의 이름만 가지고 있다. 이와 같은 경우, PAM은 기본 PAM 모듈의 디렉토리에서(보통 `/usr/lib/security`) 모듈을 찾는 다. 그러나 여러분의 리눅스가 리눅스 파일시스템의 표준을 따른다면 PAM모듈은 `/lib/security`에 있다.

모듈-인수

모듈-인수는 모듈에게 전달되는 인수이다. 각각의 모듈은 각각의 인수를 가지고 있다. 예를 들어 로그인 구성에서 "nulok"가 그것이다. ("null ok", pam_unis.so 모듈로 전달되어 지는 인수로서 공백(null)패스워드를 허용한다는 것이다("ok")).

3.3.4. `pam.conf` 구성

여러분의 PAM구성은 `/etc/pam.d`가 아니라 `/etc/pam.conf`에 저장되어 있다면 PAM 구성줄은 조금 다르다. 각각의 서비스들이 각자의 구성파일을 가지고 있지 않고, 한줄의 구성에

첫 번째 토큰으로 서비스 이름을 가지는 모든 구성들이 `/etc/pam.conf`에 저장되어 있다. 예를 들어 /etc/pam.d/login에 다음과 같은 줄이:

```
auth        required    pam_unix.so nulok
```

`/etc/pam.conf`에는 다음과 같은 줄로 나타나 있다.

```
login auth        required    pam_unix.so nulok
```

이러한 사소한 차이를 제외하고는 PAM의 나머지 문법은 같이 적용된다.

3.4. 다른 정보 얻기

PAM구성과 완벽한 PAM 모듈에 대한 참고에 관한 정보라면 Linux-PAM 시스템 관리자 설명서 (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html)를 찾아봐라. 이 설명서는 PAM 구성에 관한 완벽하고 최신의 참고를 제공한다.

4. 사용자 인증 보안

많은 리눅스 배포판은 적절하게 보안성을 갖추지 않은 사용자 인증을 가지고 판매한다. 이 섹션에서서는 여러분들의 시스템에서 사용자인증을 안전하게 만드는 방법에 대해서 논의한다. 이러한 일을 하는 동안에 여러분들의 시스템을 더욱 안전하게 만들 긴 하겠지만 취약하지 않게 만든다는 순진한 생각은 하지 마라.

4.1. 강한 `/etc/pam.d/other`

`/etc/pam.d`의 모든 파일은 특정한 서비스에 의한 구성을 포함하고 있다. /etc/pam.d/other파일이 있는 데 이것은 이러한 규칙에 대한 아주 중요한 예외이다. 이 파일은 자신들의 구성 파일을 가지고 있지 않은 모든 서비스에 대한 구성을 가지고 있다. 예를 들어 (가상의)xyz 서비스가 인증을 시도한다면 PAM은 `/etc/pam.d/xyz`파일을 찾을 것이다. 하나라도 찾아지지 않는다면 xyz에 대한 인증은 `/etc/pam.d/other` 파일에 의해서 결정이 날 것이다.
`/etc/pam.d/other` 파일은 PAM 서비스가 의지하는 구성파일이므로 안전이 중요한 문제이다. 여기서 `/etc/pam.d/other` 의 안전한 구성을 위해 2가지를 논의한다. 하나를 굉장히 편집증적인 것이고 하나는 적절한 것이다.

4.1.1. 편집증적인 구성

`/etc/pam.d/other`파일의 편집적인 구성은 아래와 같다.

```
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
```

이렇게 구성하면 모르는 서비스가 네 개의 구성타입중 어떤곳에 접근을 시도하더라도 PAM 은 인증을 거부하다(pam deny.so 모듈을 통해서)그리고 syslog warning(pam warrn.so 모듈을 통해서)로그를 남긴다. 이러한 잔인한 방법에 대한 유일한 문제점은 당신이 유연히 다른 서비스의 구성을 지워버리면 문제가 발생할 수 있다는 것이다. 당신의 `/etc/pam.d/login` 파일이 실수로 지워진다면 어떤 사람도 로그인할 수 없을 것이다!

4.1.2. 적절한 구성

여기에 그래도 좀 괜찮은 구성이 있다:

```
auth required pam_unix.so
auth required pam_warn.so
account required pam_unix.so
account required pam_warn.so
password required pam_deny.so
password required pam_warn.so
session required pam_unix.so
session required pam_warn.so
```

이구성은 모르는 서비스가 인증하도록 허용한다. 하지만 사용자의 패스워드를 바꾸도록 허용하지 않는다. 이 구성은 모르는 서비스에 의해 인증을 허용하지만, 그러한 서비스가 인증을 시도할 때마다 syslog warning 로그를 남긴다.

4.1.3. `/etc/pam.d/other` 선택하기

I would strongly reccomend that you implement the first `/etc/pam.d/other` configuration unless you have a very good reason not to. It always a good idea to be 'secure by default'. If you ever do need to grant a new service authentication privileges, you can simply create a PAM configuration file for that service. 특별한 이유가 없는 한 여러분이 먼저 /etc/pam.d/other 구성을 구현하기를 나는 강력히 권한다. "무조건적 보안"을 실현하기 위해 가장 좋은 생각이다. 여러분들이 새로운 서비스 인증권한을 부여할 필요가 있을 때면 여러분들은 간단히 그 서비스에 대해 PAM 구성을 만들면 된다.

4.2. Null 패스워드로 사용자 로그인 금지하기

대부분의 리눅스 시스템에서는 ftp나 웹서버, 메일게이트웨이와 같이 어떤 시스템의 서비스에 권한을 부여할 사용되는 "가짜"사용자계정이 많이 있다. 이러한 계정을 가짐으로써 시스템은 더 안전해질 수 있다. 왜냐하면 이러한 서비스들이 해킹당하면 공격자들은 root로 운영되는 서비스의 모든 권한을 얻지 못하고 "가짜" 계정의 제한된 권한만을 얻을 수 있기 때문이다. 그러나 이러한 가짜 계정은 주로 null 패스워드를 가지기 때문에 로그인 권한 보안의 위험요소이다. 로그인을 허용하도록 하는 서비스에 대한 여러분들은 'auth' 타입의 모듈에서 이러한 아규먼트를 제거하기 원할 것이다. 이것은 주로 로그인 서비스에 해당하고 또는 rlogin이나 ssh 서비스도 해당한다. 그래서 `/etc/pam.c/login` 파일을

```
auth required pam_unix.so nullok
```

아래와 같이 바꾸어야 한다:

```
auth required pam_unix.so
```

4.3. 사용하지 않는 서비스 제거하기

`/etc/pam.d`의 파일들을 보면, 사용하지 않거나 거의 들어도 보지 못한 많은 프로그램의 구성파일들이 찾을 수 있을 것이다. 이러한 서비스에 인증을 허용하는 것은 큰 보안허점을 제공하지는 않지만 여러분들은 인증을 거부하는 것이 좋다. 이러한 프로그램의 PAM인증을 제거하는 가장 좋은 방법은 이 파일들의 이름을 바꾸는 것이다. 인증을 요청하는 서비스의 이름을 가지는 파일을 찾지 못하면 PAM은 매우 (기대하건대)안전한 `/etc/pam.d/other`을 참조할 것이다.여러분들이 나중에 이러한 프로그램중 하나가 필요하다면 간단히 원래 파일이름으로 바꾸면 되고 그러면 모든 것이 예상한대로 동작할 것이다.

4.4. 패스워드 크래킹 도구

패스워드 크래킹도구가 공격자들에 의해서 시스템을 해킹하기위해 사용되고 있는 반면 오히려 그러한 도구를 이용해 시스템관리자들은 시스템의 패스워드 강도를 보장할 수 있는 혁신적인 도구로 사용할 수 있다. 가장 널리 사용되고 있는 패스워드 크래킹도구는 "crack"과 "John the Ripper"이다. 크랙은 여러분의 facorite 배포판에 포함되어 있다. John the Ripper는 http://www.false.com/security/john/index.html에서 구할 수 있다. 여러분들의 패스워드 데이터베이스를 상대로 이러한 도구를 돌려봐라. 결과를 보고 나면 깜짝 놀랄 것이다.

추가로 사용자의 패스워드가 바뀔 때 마다 패스워드 강도를 검사하는 데 필요한 크랙 라이브러리를 이용하는 PAM 모듈이 있다. 이 모듈이 설치되면 사용자는 패스워드를 변경시 최소 패스워드 강도를 만족하는 패스워드로만 변경할 수 있다.

4.5. Shadow와 MD5 패스워드

이문서의 처음 섹션에서 논의가 되었듯이, 쉐도우와 MD5 패스워드는 여러분들의 시스템을 더욱 안전하게 만들 수 있다. 설치절차중에 대부분의 현재 리눅스배포판은 MD5 와 쉐도우 패스워드를 설치할 것인 지를 질문할 것이다. 그것을 선택하지 않을 충분한 이유가 없다면 MD5와 쉐도우를 사용해야 한다. 비쉐도우와 비MD5에서 바꾸는 절차는 복잡하다. 그리고 이 문서에서는 다루지 않는다. Shadow 패스워드 HOWTO (http://www.linuxdoc.org/HOWTO/Shadow-Password-HOWTO.html)는 좀 오래 되었지만 도움이 될 것이다.

5. 함께 묶기

이번 섹션에서는 지난 섹션에 다루었던 내용을 묶는 예제를 보여주겠다.

5.1. 아파치 + mod_auth_pam

앞의 예와 같이 PAM을 사용해서 웹서버의 사용자 인증을 할 때 사용하는 아파치 모듈인 mod_auth_pam을 설치 및 구성할 것이다. 이 예제의 목적에 맞게 여러분들은 아파치가 설치되어 있어야 한다. 만약에 아파치가 설치되어 있지 않다면 판매자로 부터 설치패키지를 구할 수 있을 것이다.

5.2. 예

우리들의 목적은 PAM을 이용해서 사용자인증을 하기 위해`family/` 디렉토리인 웹서버의 제한된 영역을 구성하는 것이다. 이 디렉토리는 개인적인 패밀리 정보를 담고 있고 사용자 그룹 패밀리의 멤버들에게 접근이 허용되어야 한다.

5.3. mod_auth_pam 설치하기

먼저 여러분들은 http://blank.pages.de/pam/mod_auth_pam/에서 mod_auth_pam을 다운받아야 한다. 다음 명령어들은 mod_auth_pam을 컴파일하는 것이다.(반드시 root 로 로그인해야 한다.):

```
~# tar xzf mod_auth_pam.tar.gz
~# cd mod_auth_pam-1.0a
~/mod_auth_pam-1.0a# make
~/mod_auth_pam-1.0a# make install
```

만약에 mod_auth_pam 모듈을 설치할 때 문제가 발생하면 여러분들은 반드시 아파치-dev 패키지 배포판을 설치해야 한다. 여러분들이 mod_auth_pam을 설치한 후에, 아파치를 재시작해야 한다. 아파치는 아래 명령어를 치면 재시작된다.(root 이어야 한다):

```
~# /etc/init.d/apache restart
```

5.4. PAM 구성하기

아파치 PAM구성파일은 `/etc/pam/d/httpd`에 저장되어 있다. 기본구성(mod_auth_pam을 설치하면 기본구성이 설치된다)은 안전하다. 그러나 기본구성파일은 많은 시스템에서 없는 모듈(`pam_pwdb.so`)을 사용한다.(게다가 그 파일을 처음부터 구성해보면 재미있을 것이다). 그래서 `/etc/pam.d/httpd` 파일을 삭제하고 새로 시작해보자.

5.4.1. PAM 구성방법 결정하기

PAM이 아파치의 인증요청을 처리하는 방법을 구성하려고 하면, PAM이 무엇을 체크해야 하는 지를 정확히 알 필요가 있다. 먼저 반드시 PAM은 사용자 패스워드는 표준유닉스 패스워드 데이터베이스에 사용자의 패스워드와 일치시킨다는 것을 확인해야 한다. 이말은 'auth' 타입과 `pam_unix.so` 모듈와 같은 말이다. 모듈의 통제 타입이 'required'로 설정되어 있어야 하고, 그래서 정확한 패스워드 없이는 인증은 실패한다.아래는 `/etc/pam.d/httpd`의 첫 번째줄을 보여주고 있다:

```
auth required pam_unix.so
```

두 번째 사용자 계정은 반드시 유효해야 한다.(즉 사용자의 패스워드는 사용기간 만료되지 않았다). 이것은 'account'의 타입이고 `pam_unix.so` 모듈에 의해서 제공된다. 다시 이 모듈의 통제 타입을 'required'로 설정하고, 이 줄을 추가한 후에 `/etc/pam.d/httpd` 구성은 아래와 같다:

```
auth required pam_unix.so
account required pam_unix.so
```

이것은 엄청 정교한 것은 아니다. 그러나 잘 된다. PAM서비스를 구성방법을 배울려면 이렇게 시작해야 한다.

5.5. 아파치 구성하기

PAM이 아파치 요청을 인증하도록 구성했기 때문에, `family/` 디렉토리에 접근을 제한하는 PAM 인증을 적절히 사용하기 위해 아파치를 구성해야 한다. 그렇게 하기 위해서는 `httpd.conf`(`/etc/apache/`나 `/etc/httpd`에 저장되어 있다)에 다음과 같은 줄을 추가해라:

```
<Directory /var/www/family>
AuthPAM_Enabled on
AllowOverride None
AuthName "Family Secrets"
AuthType "basic"
require group family
</Directory>
```

여러분들은 /var/www/을 /home/httpd/인 웹문서의 기본저장장소로 바꾸는 것이 필요하다. 저장장소가 어디든 family 디렉토리를 만들어야 한다. 설치을 시험하기 전에 여러분들이 지금 막 들어간 아파치 구성을 잠깐 설명하겠다. <Directory>명령은 이 디렉토리의 구성데이터를 포함하는 데 사용된다. 이 명령어 안에 PAM인증을 활성화한다("AuthPAM_enabledon"), 이 구성내용은 변경되는 것을 막고("AllowOverride none"), 이 인증 영역을 "Family Secrets"라고 이름을 붙이고, 기본적인 http 인증형태(PAM 인증이 아니다)로 설정하고(AuthType "basic"), 사용자 그룹패밀리를 요구한다.("Require group family")

5.6. 설치한 것 시험하기

전부다 설치가 잘 되었기 때문에, 성공을 축하할 때다. 가장좋아하는 웹브라우저를 열어라. 그리고 http://your-domain/family/을 쓰라(여러분의 도메인으로 바꿔어라). 여러분들은 이제 사용자 인증자이다

6. 자료

사용자인증에 관해 더 많은 정보들이 온라인 오프라인 많은 자료들이 있다. 이 리스트에 추가할 필요가 있는 자료를 알고 있으면 <petehern (at) yahoo.com>으로 메일보내주기 바란다.

6.1. PAM

- Linux-PAM System 관리자 설명서 (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html)
- Linux-PAM 모듈 작성자 매뉴얼 (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_modules.html)
- Linux-PAM 응용프로그램 개발자 매뉴얼 (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_modules.html)

6.2. 일반적인 보안

- linuxsecurity.com (http://www.linuxsecurity.com/)
- securitywatch.com (http://www.securitywatch.com)
- Security HOWTO (http://www.linuxdoc.org/HOWTO/Security-HOWTO.html)
- Packetstorm (http://packetstorm.securify.com)

6.3. 오프라인 문서

시스템 매뉴얼 보면 많은 정보들을 모을 수 있다. 다음에 열거한 것이 사용자 인증과 관계된 맨페이지이다. 괄호안의 숫자는 맨페이지 섹션이다. passwd(5) 맨페이지를 보고싶다면 **`man 5 passwd`**를 치면 된다. passwd(5)

- passwd(5)
- crypt(3)
- pam.d(5)
- group(5)
- shadow(5)

7. 결론

나는 여러분들이 이 HOWTO 문서가 도움이 되길 바란다. 여러분들이 질문, 의견, 제안이 있으면 언제든지 `<petehern (at) yahoo.com>`로 메일을 보내주기 바란다.