Kullanıcı Kimlik Kanıtlama NASIL

Yazan:
Peter Hernberg

Düzenleyen:
Floris Lambrechts
Dil değişiklikleri, küçük düzeltmeler (v0.8).

Çeviren:
Necdet Yücel
<nyucel (at) comu.edu.tr>

Aralık 2005

Özet

Bu belgede kullanıcı ve grup bilgilerinin nasıl saklanıldığı, Linux sisteminde (PAM) kullanıcı kimlik kanıtlamasının nasıl yapıldığı ve nasıl daha güvenli hale getirilebileceği anlatılmaktadır.

Konu Başlıkları

Bu çevirinin sürüm bilgileri:

1.0	Aralık 2005	ny
İlk çeviri		

Özgün belgenin sürüm bilgileri:

0.9	2004-04-03	fl
belge dışına verilen bağlar güncellendi		
0.8	2003-02-20	fl
dil değişikliği ve bazı küçük düzeltmeler yapıldı		
0.5	2000-05-15	ph
pam'ın güvenli hale getirilmesi ve kaynaklar bölümleri eklendi.		
0.1	2000-05-02	ph
ilk sürüm		

1. Giriş

1.1. Bu belge nasıl oluştu

Ev ağıma (çoğu gereksiz :)) yeni ağ servisleri eklemeye çalıştığımda bir takım kimlik doğrulama problemleriyle karşılaştığımdan Linux sistemlerinde kimlik doğrulamanın NASIL yapılacağını anlatan bu belgeyi bitirme projem olarak hazırlamaya karar verdim. Umarım bu belge sistem yönetiminin genellikle unutulan ama önemli bu yönünü anlamanıza yardımcı olur.

1.2. Yeni Sürümler

Kendime bir alan alıp çalıştırıncaya kadar bu belgenin son sürümüne http://www.tldp.org adresinden erişilebilirsiniz.

1.3. Geri Bildirim

Yorumları, düzeltmeleri ve önerileri <petehern (at) yahoo.com> adresine gönderebilirsiniz.

1.4. Lisans

1.5. Teşekkürler

Beni 18 yaşıma kadar getiren aileme teşekkür ederim. Debian geliştiricilerine hazırladıkları güzel dağıtım için teşekkür ederim. Beni bir guru (geek – yazılım hatalarıyla beslenen kişi) olarak tanımlayarak onurlandıran CGR[B2]'ye teşekkür ederim. Sandy Harris'e yararlı önerileri için teşekkür ederim. Son olarak onsuz nasıl yaşayacağımı bilmediğim ramen noodles'ı yaratanlara teşekkür ederim.

1.6. Okuyucu Hakkındaki Kabuller

Bu belgenin hedefine ulaşabilmesi için okuyucunun komut satırından komut çalıştırmakla ve yapılandırma dosyalarını düzenlemekle bir sorunu olmadığı kabul edilmiştir.

2. Sisteminizde Kullanıcı Bilgileri Nasıl Saklanır

2.1. /etc/passwd

Neredeyse tüm Linux dağıtımlarında (ve ticari *nix'lerde) kullanıcı bilgisi /etc/passwd dosyasında saklanır. Bu metin dosyasında kullanıcının kullanıcı adı, şifrelenmiş parolası, benzersiz sayısal kullanıcı kimliği (uid), sayısal grup kimliği (gid), seçimlik yorum alanı (burada genellikle kullanıcının gerçek adı, telefon numarası gibi bilgiler bulunur), ev dizini ve tercih ettiği kabuk bilgileri bulunur. /etc/passwd dosyasındaki girdiler aşağıdaki gibidir:

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

Gördüğünüz gibi anlaşılması oldukça kolaydır. Her girdi için yukarıda anlatılan altı alan vardır ve alanlar birbirinden : ile ayrılırlar. Kullanıcı kimlik denetimi sadece bu kadarcık karışık olsaydı bu NASIL belgesine ihtiyaç olmazdı.

2.2. Gölgelenmiş Parolalar

Sisteminizdeki /etc/passwd dosyasına bakarsanız aşağıdakine benzer olduğunu görürsünüz :

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

Şifrelenmiş parola nereye gitti? Buna yanıt vermeden önce biraz açıklama yapmak gerekiyor.

Kullanıcılar hakkındaki, şifrelenmiş parolaları dahil, tüm bilgilerin tutulduğu /etc/passwd dosyası tüm kullanıcılar tarafından okunabildiğinden herhangi bir kullanıcı sistemdeki tüm kullanıcıların şifrelenmiş parolalarını elde edebilir. Parolalar şifrelenmiş olsalar bile parola–kırma programları yaygın olarak bulunmaktadır. Bu güvenlik tehditiyle mücadele edebilmek için gölgelenmiş parolalar geliştirilmiştir.

Bir sistemde gölgelenmiş parolalar etkin kılındığında /etc/passwd dosyasındaki parola alanına x yazılır ve kullanıcının gerçek şifrelenmiş parolası /etc/shadow dosyasında saklanır. /etc/shadow dosyası sadece root tarafından okunabildiği için kötü niyetli kullanıcılar başkalarının parolalarını kıramazlar. /etc/shadow dosyası her girdi için kullanıcı adı, şifrelenmiş parola ve parolanın geçerliliği ile ilgili bir kaç alan içerir. Örnek bir girdi aşağıdaki gibidir:

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

2.3. /etc/group ve /etc/gshadow

Grup bilgisi /etc/group dosyasında saklanır. Dosya biçimi /etc/passwd'e benzer. Her girdi için grup adı, parolası, grup numarası (gid) ve birbirinden virgülle ayrılmış grup üyelerinin yeraldığı alanlar bulunur. /etc/group içindeki bir girdi aşağıdaki gibidir:

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

Parola alanındaki "x" ifadesinden anlaşıldığı gibi grup parolaları da gölgelenebilir. Neredeyse hiçbir grubun parolası olmasa da gölgelenmiş grup parolalarının /etc/gshadow dosyasında saklandığını hatırlatmakta fayda var.

2.4. MD5 şifreli parolalar

Geleneksel olarak Unix parolaları **crypt()** işlevi kullanılarak şifrelenir. (**crypt()** işlevi hakkında daha fazla bilgi almak için **crypt(3)** [B3] kılavuz dosyasını okuyabilirsiniz.) Bilgisayarların hızlı gelişimi bu işlevle şifrelenmiş parolaların kolayca kırılabilir hale gelmesine yol açtı. Internet'in ortaya çıkmasıyla parola–kırma görevini çok sayıda düğüme dağıtabilmek mümkün hale geldi. Bir çok 'güncel' dağıtım parolaların daha güçlü MD5 çırpılama algoritmasıyla şifrelenebilmesi seçeneğiyle gelmektedir (MD5 çırpılama algoritmasıyla ilgili

ayrıntılı bilgi RFC 1321[B4] adresinden alınabilir). MD5 şifreli parolalar kullanarak parola–kırmanın tamamen önüne geçilemese bile oldukça zorlaştırılır.

2.5. Karışıklığı önlemek

Gördüğünüz gibi sisteminizde kullanıcı kimlik kanıtlamasında kullanılacak bilgileri saklamak için farklı yöntemler mevcuttur (MD5 şifrelemesi olmadan gölgelenmiş parolalar, parolaların MD5 ile şifrelenip `/etc/passwd` dosyasında saklanması, vs.). **`login`** veya **`su`** gibi programlar parolaları nasıl doğrulayacaklarını nereden biliyorlar? Daha kötüsü, sisteminizde parolaların saklanma biçimini değiştirirseniz ne olur? Parolanıza ihtiyaç duyan programlar parolaların artık farklı biçimde saklandığını nasıl bilecekler? Bu soruların yanıtı bizi PAM'e götürür.

3. PAM (Eklenebilir Kimlik Kanıtlama Modülleri)

Eklenebilir kimlik kanıtlama modülleri günümüzdeki tüm linux dağıtımlarında kimlik kanıtlamanın esasını oluşturur.

3.1. Neden

Linux'un eski güzel günlerinde **`su`**, **`passwd`**, **`login`** ya da **`xlock`** gibi bir program bir kullanıcının kimlik doğrulamasını yapmak istediğinde gerekli bilgiyi basitçe `/etc/passwd` dosyasından okurdu. Kullanıcının parolasını değiştirmek için `/etc/passwd` dosyasını düzenlemek yeterliydi. Bu basit ama beceriksiz yöntem sistem yöneticileri ve uygulama geliştiricilerini sorunlarla karşı karşıya bırakıyordu. MD5 şifreli ve gölgelenmiş parolalar popüler oldukça kullanıcı kimlik kanıtlamasına ihtiyaç duyan her programın doğru bilgiyi hangi yöntemle alacağını bilmesi gerekliliği ortaya çıktı. Kullanıcı kimlik kanıtlaması şemanızı değiştirmek isterseniz tüm programları yeniden derlemeniz gerekliydi. PAM bu karmaşayı kullanıcı bilgisinin nasıl saklandığından bağımsız olarak, programların kimlik kanıtlamasını şeffaf bir biçimde yapmalarına izin vererek ortadan kaldırdı.

3.2. Nedir

Linux–PAM Sistem Yöneticisinin Kılavuzu[B5]'ndan alıntı:

> Linux–PAM projesinin amacı, ayrıcalık verme yazılımları ile güvenli ve uygun kimlik kanıtlama şemalarının gelişimini birbirinden ayırmaktır. Bu, uygulamaların kimlik kanıtlamasında kullanabilecekleri bir işlevler kütüphanesi sağlanarak gerçekleştirilir.

PAM sayesinde parolalarınızı `/etc/passwd` dosyasında veya Hong Kong'daki bir sunucuda tutmanız farketmez. Bir program kullanıcı kimlik kanıtlamasına ihtiyaç duyduğunda, PAM uygun kimlik kanıtlama şeması için gereken işlevleri içeren bir kütüphane sunar. Bu kütüphane dinamik olarak yüklendiği için kimlik kanıtlama şemasını değiştirmek için basitçe yapılandırma dosyasını düzenlemek yeterli olur.

Esneklik PAM'in en önemli güçlerinden birisidir. PAM belirli programların kullanıcı kimlik kanıtlaması yapamayacağı, sadece belirli kullanıcıların kimlik kanıtlaması yapabileceği, bazı programlar kimlik kanıtlaması yapmak istediğinde uyarı verecek şekilde ve hatta tüm kullanıcıları oturum açma ayrıcalıklarından mahrum bırakacak şekilde yapılandırılabilir. PAM'in modüler tasarımı kimlik denetimi üzerindeki bütün kontrolü elinize almanıza izin verir.

3.2.1. PAM destekleyen dağıtımlar

Neredeyse tüm tanınmış dağıtımlar bir süredir PAM destekliyorlar. Aşağıda PAM destekleyen dağıtımların bir kısmının listesi bulunmaktadır:

- Redhat, 5.0 sürümünden itibaren

- Mandrake, 5.2 sürümünden itibaren
- Debian, 2.1 sürümünden itibaren (2.1 sürümde kısmi destek — 2.2 sürümünde tam destek)
- Caldera, 1.3 sürümünden itibaren
- Turbolinux, 3.6 sürümünden itibaren
- SuSE, 6.2 sürümünden itibaren

Bu liste tam olmadığı gibi hatalı da olabilir. Düzeltmelerinizi ve eklemelerinizi <petehern (at) yahoo.com> gönderirseniz memnun olurum.

3.2.2. PAM Kurulumu

Sıfırdan PAM kurulumu bu NASIL'ın kapsamını aşan uzun bir süreçtir. Eğer sisteminizde PAM kurulu değilse, büyük olasılıkla kullandığınız dağıtımın çok eski bir sürümünü kullanıyorsunuz. Dağıtımınızı güncellemeniz için başka nedenler de bulunmasına rağmen güncel bir dağıtım kullanmak yerine PAM kurulumunu kendiniz yapmak istiyorsanız kesinlikle benim yardımıma ihtiyacı olan birisi değilsiniz. Bu nedenlerle sisteminizde PAM kurulu olduğunu kabul edeceğim.

3.3. Nasıl

Yeterince konuştum, çalışma zamanı.

3.3.1. PAM yapılandırma dosyaları

PAM yapılandırma dosyaları `/etc/pam.d/` dizininde bulunur. (Eğer sisteminizde `/etc/pam.d/` dizini yoksa dert etmeyin, sıradaki bölümde ne yapacağınızı anlatacağım) Şimdi bu dizine geçelim ve neler olduğuna bakalım.

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

Bu dizinin içerdiği dosyalar sisteminize neler kurduğunuza bağlı olarak üç aşağı beş yukarı böyledir. Ayrıntılar ne olursa olsun, sisteminizde kullanıcı kimlik kanıtlamasına ihtiyaç duyan her program için bir dosya görüyor olmalısınız. Tahmin ettiğiniz gibi her dosya bir program için PAM kimlik kanıtlaması yapılandırmasını içerir (`other` dosyası bir istisnadır, ondan birazdan bahsedeceğiz). Login için PAM yapılandırma dosyasının içeriğine bakalım (dosyanın içeriğini basitleştirdim):

```
/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 nullok
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
```

Dosyanın içeriğine geçmeden önce biraz bilgi vermem gerekir.

3.3.2. Ek bilgi

Okuyucuların küçük bir kısmı şöyle düşünüyor olmalı; "Olamaz! Sistemimde `/etc/pam.d` dizini yok! Yukarıdaki listenizde dağıtımımın PAM içerdiği söyleniyor ama bulamıyorum. PAM olmadan hayatım boş ve anlamsız! Ne

yapabilirim?" Telaşlanmayın, herşey bitmiş değil. Dağıtımınızın PAM içerdiğini bildiğiniz halde `/etc/pam.d/` dizini yoksa PAM yapılandırmanız `/etc/pam.conf` dosyasında saklanıyordur. Birçok dosya kullanmak yerine tüm PAM yapılandırması için tek bir dosya kullanılıyordur. Bu PAM yapılandırmasını biraz karmaşıklaştırır ama uygun ayarlamalar *pam.conf yapılandırması* (sayfa: 9) bölümünde anlatılacaktır.

3.3.3. Yapılandırma sözdizimi

PAM yapılandırma dosyaları aşağıdaki sözdizimine sahiptir:

tür denetim modul–yolu modul–argümanları

Yukarıdaki örnekte `login` dosyası için verilen yapılandırma dosyasını kullanarak PAM yapılandırma sözdizimine bakalım:

PAM yapılandırma özellikleri

tür

tür özelliği PAM'e bu modül için hangi tür kimlik kanıtlamasının kullanılacağını söyler. Aynı türden modüller "istiflenebilir". PAM dört farklı *tür* tanır:

`account`
: Kullanıcının servise erişmeye izni olup olmadığını, parolasının süresinin geçip geçmediğini tespit eder.

`auth`
: Kullanıcının iddia ettiği kişi olup olmadığını denetler, bunu genellikle parola ile yapar ama örneğin biyometri gibi daha karmaşık yöntemler de kullanabilir.

`password`
: Kullanıcının kimlik kanıtlaması için kullandığı şeyi değiştirmesi için bir mekanizma sağlar. Bu genellikle paroladır.

`session`
: Kullanıcının kimlik kanıtlaması yapıldıktan sonra ve/veya önce yapılması gerekenler. Bunlar kullanıcının ev dizininin bağlanması/çözülmesi, açıp kapattığı oturumların kaydının tutulması ve kullanıcının kullanabileceği servislerin kısıtlanması gibi şeyler olabilir.

`login` yapılandırma dosyasında her *tür* için en az bir girdi olduğunu gördük. Bu program kullanıcıların oturum açmalarına izin verdiğinden (adından da anlaşıldığı gibi :), kimlik kanıtlamasının her türlüsüne erişebilmesinin gerekmesi anlaşılabilir bir şeydir.

denetim

denetim özelliği PAM'e bir modül kimlik kanıtlamasında başarısız olduğunda ne yapması gerektiğini söyler. PAM dört farklı *denetim* türü tanır:

`requisite`
: Bu modül yoluyla kimlik kanıtlaması başarısız olursa kimlik kanıtlaması derhal reddedilir.

`required`
: PAM kimlik kanıtlamasını reddetmeden önce bu servis için listelenmiş diğer modülleri çağırmaya devam etse de başarısızlık yine kimlik kanıtlamasının reddi ile sonuçlanır.

`sufficient`
: Bu modül ile kimlik kanıtlaması başarılı olursa, PAM kimlik doğrulamasını daha önceki gerekli bir modülde başarısız olsa bile kabul edecektir.

`optional`
: Bu modülün başarılı olması veya olmaması ancak bir servis için kendi türünde tek modül olması halinde önemlidir.

Login için yapılandırma dosyasında neredeyse tüm *denetim* türlerini gördük. En çok ihtiyaç duyulan modül `pam_unix.so` (temel kimlik kanıtlama modülü), zorunlu tek modül `pam_securetty.so` (kullanıcının güvenli konsola oturum açtığından emin olmayı sağlar) ve seçimlik tek modül `pam_lastlog.so` (kullanıcının en son açtığı oturum ile ilgili bilgileri getiren modül).

modül–yolu
: PAM hangi modülü kullanacağını ve modülleri nerede bulacağını *modül–yolu* sayesinde bilir. Çoğu yapılandırma `login` örneğinde olduğu gibi sadece modülün adını içerir. Böyle durumlarda, PAM öntanımlı PAM modül dizinine bakar, bu normalde `/usr/lib/security` dizinidir. Bununla birlikte eğer linux dağıtımınız Dosyasistemi Hiyerarşisi Standardına (FHS) uygun ise PAM modülleri `/lib/security` dizininde bulunur.

modüle–argümanları
: *modüle–argümanları* modüllerin parametreleridir. Her modülün kendi parametresi vardır. Örneğin bizim `login` yapılandırmasında "nulok" (pam_unix.so modülüne "null ok" parametresi gönderilmesi "boş" parolaların "geçerli" olduğu anlamındadır).

3.3.4. `pam.conf` yapılandırması

Eğer PAM yapılandırmanız `/etc/pam.d/` dizini yerine `/etc/pam.conf` dosyasında saklanıyorsa PAM yapılandırma satırları biraz farklıdır. Her servisin kendi yapılandırma dosyası olması yerine tüm yapılandırmalar `/etc/pam.conf` dosyasında servisin adı ile başlayan satırlardan oluşur. Örneğin `/etc/pam.d/login` dosyasındaki aşağıdaki satır:

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

`/etc/pam.conf` dosyasında şu hale gelir:

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

Bu basit farklılıkların dışında, yapılandırmanın geri kalanında PAM sözdizimi uygulanır.

3.4. Daha fazla bilgi edinmek

PAM yapılandırması hakkında daha fazla bilgi edinmek ve bütün PAM modüllerinin teknik açıklamalarına ulaşmak için Linux–PAM Sistem Yöneticisinin Kılavuzu[B7]'nu kullanabilirsiniz. Bu kılavuz size PAM yapılandırması hakkındaki en güncel bilgileri sağlar.

4. Kullanıcı Kimlik Denetimini Güvenli Hale Getirmek

Birçok linux dağıtımı kullanıcı kimlik kanıtlamasını yeterince güvenli gerçekleştirmez. Bu bölümde sisteminizde kimlik denetimini nasıl daha güvenli hale getirebileceğinizi tartışacağız. Burada anlatılanları yapmanız sisteminizi daha güvenli hale getirir ama kırılamaz yapmaz.

4.1. Güçlü `/etc/pam.d/other` dosyası

`/etc/pam.d/` dizinindeki dosyaların tümü özel bir servis için yapılandırmaları içerir. Bu kuralın tek istisnası `/etc/pam.d/other` dosyasıdır. Bu dosya kendisi için bir yapılandırma dosyası bulunmayan

servisler için yapılandırmaları içerir. Örneğin, eğer (hayali) **xyz** servisi kimlik kanıtlamasına ihtiyaç duyarsa PAM /etc/pam.d/xyz dosyasını arar. Bulamayınca **xyz** uygulaması için kimlik denetiminde /etc/pam.d/other dosyasını kullanır. /etc/pam.d/other sahipsiz PAM servislerinin yapılandırma dosyası olduğundan onun güvenli olması önemlidir. Burada /etc/pam.d/other dosyasının birisi neredeyse paranoyak diğeri biraz daha mantıklı olan iki farklı güvenli yapılandırmasını tartışacağız.

4.1.1. Paronayak Yapılandırma

/etc/pam.d/other dosyasının paronayak yapılandırması aşağıdaki gibidir:

```
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
```

Bu yapılandırma ile bilinmeyen bir servis dört yapılandırma türünden birine erişmeye çalışırsa PAM kimlik denetimini reddeder (pam_deny.so modülü ile) ve sistem günlüklerine bir uyarı mesajı (pam_warn.so modülü ile) kaydeder. Bu yapılandırma ile PAM kabaca güvenlidir. Bu kabalık ile ilgili tek problem eğer bir servisin yapılandırma dosyasını silerseniz ortaya çıkar. Örneğin eğer /etc/pam.d/login dosyasını kazara silerseniz kimse oturum açamaz!

4.1.2. Daha nazik yapılandırma

O kadar da kaba olmayan bir yapılandırma aşağıdaki gibidir:

```
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
```

Bu yapılandırma ile bilinmeyen bir servis için kullanıcı kimlik kanıtlamasına (pam_unix.so modülü ile) izin verilir ama parolasını değiştirmesine izin verilmez. Bilinmeyen servislerin kimlik kanıtlaması yapmalarına izin verilse bile sistem günlüklerine bir uyarı mesajı kaydeder.

4.1.3. /etc/pam.d/other dosyalarından birini seçmek

Aksi yönde çok iyi bir nedeniniz olmadıkça yukarıdaki /etc/pam.d/other dosyalarından ilkini seçmenizi kuvvetle öneririm. 'Öntanımlı olarak güvenli' olmak her zaman iyi fikirdir. Eğer yeni bir servis için kimlik kanıtlamasına izin vermeniz gerekirse bunu basitçe o servis için bir PAM yapılandırma dosyası oluşturarak yapabilirsiniz.

4.2. Kullanıcıların boş parolalarla oturum açmasını engellemek

Linux sistemlerinin çoğunda ftp, web sunucusu ve mail gibi sistem servislerine ayrıcalıklar tanıyabilmek için bir takım "sözde" kullanıcı hesapları bulunur. Bu hesapların bulunması sisteminizi daha güvenli hale getirir. Çünkü bir servisin açığından faydalanan saldırgan sınırsız root yetkilerini değil sadece kısıtlı ayrıcalıkları olan sözde

hesabın yetkilerini kazanacaktır. Bununla birlikte bu sözde hesapların bulunması genellikle boş (null) parolaları olduğundan bir güvenlik açığıdır. Boş parolaların kabul edilmesine "nullok" yapılandırma seçeneği izin verir. Bu seçeneği oturum açmasına izin verilen 'auth' *tür*ündeki tüm servislerden kaldırmak isteyebilirsiniz. Bu genellikle login servisidir ama rlogin ve ssh gibi servisleri de kapsayabilir. Bu yüzden /etc/pam.d/login dosyasının aşağıdaki satırı:

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

yerine

```
auth          required        pam_unix.so
```

haline getirilmelidir.

4.3. Kullanılmayan servislerin iptal edilmesi

/etc/pam.d/ dizinindeki dosyalara baktığınızda kullanmadığınız hatta bazılarının adını bile duymadığınız programlar için yapılandırma dosyaları olduğunu göreceksiniz. Bu servislerin kimlik denetimi yapmasına izin vermek sisteminizde büyük güvenlik açıkları oluşturmasa da kimlik denetiminden reddedilmeleri daha iyidir. Bu programlar için PAM kimlik denetimine izin vermemenin en iyi yolu dosyalarını yeniden adlandırmaktır. PAM kimlik kanıtlama talebinde bulunan servis için gerekli dosyayı bulamadığında çok güvenli olan /etc/pam.d/other dosyasını kullanacaktır. Eğer ilerde bu programlardan birine ihtiyacınız olduğunu farkederseniz dosyanın adını eski haline getirmeniz her şeyin gerektiği gibi çalışması için yeterli olacaktır.

4.4. Parola–kırma araçları

Parola–kırma araçları saldırganlar tarafından sistemi ele geçirmek için kullanılabilecekleri gibi sistem yöneticileri tarafından sistemlerinde güçlü parolalar kullanıldığından emin olmak için de kullanılırlar. En yaygın kullanılan parola–kırma araçları "crack" ve "John the Ripper"'dır. Crack büyük ihtimalle kullandığınız dağıtıma dahil edilmiştir. John the Ripper ise http://www.openwall.com/john/ adresinden edinilebilir. Bu araçları parola veri tabanınızda çalıştırdığınızda sonuçlar büyük ihtimalle sizi şaşırtacaktır.

Bunlara ilave olarak, kullanıcılar parolalarını değiştirirken parolaların dayanıklılıklarını ölçen bir PAM modülü de mevcuttur. Bu modül yüklendiğinde kullanıcılar parolalarını ancak minimum dayanıklılığa sahip parolalar ile değiştirebilirler.

4.5. Gölgelenmiş ve MD5 parolalar

Bu belgenin ilk bölümünde bahsettiğimiz gibi gölgelenmiş ve MD5 parolalar sisteminizi daha güvenli hale getirir. Günümüzdeki dağıtımların çoğu kurulum aşamasında gölgelenmiş ve/veya MD5 parolaları kullanmak isteyip istemediğinizi sorar. Aksi için çok iyi bir nedeniniz yoksa onları etkin kılın. Gölgelenmemiş/MD5–lenmemiş parolaların dönüştürülmesi karmaşık bir süreçtir ve bu belgenin kapsamının dışındadır. Gölgelenmiş Parola NASIL[B9] belgesi eskimiş olsa da yardımcı olabilir.

5. Tümünü birden denemek

Bu bölümde bir önceki bölümde anlatılanların anlaşılmasına yardımcı olacak basit bir örnek vereceğim.

5.1. Apache + mod_auth_pam

Örneğimizde PAM kullanarak web sunucunuzun kullanıcılarının kimlik denetimini yapmak üzere bir Apache modülü olan mod_auth_pam'i kurup yapılandıracağız. Örneğin hedefine ulaşabilmesi için kurulu bir Apache'niz olduğunu kabul edeceğim. Eğer kurulu değilse dağıtımınızın kurulum paketlerinden yararlanabilirsiniz.

5.2. Örnek

Kullanıcılarımızın kimlik denetimini PAM ile yapabilmek için hedefimiz web sunucumuzda kısıtlı bir `aile/` dizini yapılandırmak olacak. Bu dizin özel aile bilgileri içerecek ve sadece "aile" grubunun üyeleri tarafından erişilebilir olacak.

5.3. `mod_auth_pam` kurulumu

İlk olarak mod_auth_pam[B10] paketini indirmek isteyeceksiniz. Aşağıdaki komutlarla (root olarak) `mod_auth_pam` derlenebilir:

```
~# 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
```

`mod_auth_pam` modülünü yüklerken bir hata ile karşılaşırsanız dağıtımınızın `apache-dev` paketini kurup kurmadığınızı kontrol edin. `mod_auth_pam` kurulduktan sonra apache'yi yeniden başlatmanız gerekir. Bunu aşağıdaki komutla (yine root olarak) yapabilirsiniz:

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

5.4. PAM Yapılandırması

Apache için PAM yapılandırması `/etc/pam.d/httpd` dizininde saklanır. Öntanımlı yapılandırma (`mod_auth_pam` kurulumunda yapılan yapılandırma) güvenlidir ama birçok sistemde kurulu olmayabilen `pam_pwdb.so` modülünü kullanır. (Ayrıca, sıfırdan yapılandırmak eğlencelidir!) Bu nedenle `/etc/pam.d/httpd` dosyasını silin ve yeni bir tanesiyle başlayın.

5.4.1. PAM'in nasıl yapılandırılacağına karar vermek

Eğer PAM Apache'nin kimlik kanıtlama isteklerine yanıt verecek şekilde yapılandırılacaksa PAM'in tam olarak neyi denetlemesine ihtiyacımız olduğunu bilmeliyiz. İlk olarak PAM kullanıcının parolasının standart unix parola veritabanındaki parola ile aynı olup olmadığına bakmalıdır. Bu '`auth`' *tür*üne ve `pam_unix.so` modülüne benzer. Modülün *denetim* türünü '`required`' olarak atayacağız. Bu sayede doğru parola girilmez ise kimlik denetimi başarısız olacaktır. `/etc/pam.d/httpd` dosyamızın ilk satırı aşağıdaki gibi olmalıdır:

```
auth       required        pam_unix.so
```

İkinci olarak, kullanıcı hesabının geçerli olduğundan (yani parolasının süresinin geçmediğinden veya bunun gibi bir uyumsuzluk olmadığından) emin olmalıyız. Bu '`account`' *tür*üdür ve `pam_unix.so` modülü ile sağlanır. Yine bu modülün *denetim* türünü '`required`' olarak atayacağız. Bu satırı da ekledikten sonra `/etc/pam.d/httpd` yapılandırma dosyamız aşağıdaki hale gelir:

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

Çok karışık değildir ama görevini yapar. PAM servislerini nasıl yapılandıracağınızı öğrenmek için iyi bir başlangıç olabilir.

5.5. Apache'nin Yapılandırılması

Artık PAM, apache isteklerinin kimlik denetimini yapacak şekilde yapılandırıldı. Bundan sonra apache'yi `aile/` dizinine erişimi kısıtlaması için PAM kimlik kanıtlamasını kullanacak şekilde yapılandıracağız. Bunu yapabilmek

için, aşağıdaki satırları `httpd.conf` dosyanıza (genellikle `/etc/apache/` ya da `/etc/httpd` dizininde bulunur) ekleyin:

```
<Directory /var/www/family>
  AuthPAM_Enabled on
  AllowOverride None
  AuthName "Aile Sirlari"
  AuthType "basic"
  require group aile
</Directory>
```

`/var/www/` ifadesini web sunucunuzun kök dizini ile değiştirmelisiniz. Bu dizin bazen `/home/httpd/` olabilmektedir. Her nerede olursa olsun, içinde `aile` dizinini oluşturmalısınız.

Kurulumu denetlemeden önce Apache yapılandırmasına yukarıda eklediklerinizi açıklayayım. `<Directory>` ifadesi yapılandırmanın sadece bu dizin için geçerli olması için kullanılır. Bu ifadenin içinde PAM kimlik denetimini etkinleştirdik (`AuthPAM_enabled on`), başka bir yapılandırmanın önceliği olmasını engelledik (`AllowOverride none`), bu kimlik denetim alanını "Aile Sirlari" olarak adlandırdık (`AuthName "Aile Sirlari"`), http kimlik kanıtlamasını (PAM değil) öntanımlı olarak atadık (`AuthType "basic"`) ve gerekli kullanıcı grubu olarak aile'yi atadık (`require group aile`).

5.6. Kurulumun Denetlenmesi

Herşeyi gerektiği gibi kurduk, artık kutlama zamanıdır. Tercih ettiğiniz web tarayıcısını çalıştırın ve `http://`*sizin–alaniniz*`/aile/` adresini (*sizin–alaniniz* yerine kendi alanınızın adını yazın) açın. Tebrikler, başardınız!

6. Kaynaklar

Kullanıcı kimlik kanıtlaması ile ilgili bilgi bulabileceğiniz çevrimiçi ve çevrim dışı pek çok kaynak bulunmaktadır. Aşağıdaki listeye eklenebilecek bildiğiniz kaynaklar varsa <petehern (at) yahoo.com> adresine gönderebilirsiniz.

6.1. PAM

- Linux–PAM Sistem Yöneticisinin Kılavuzu[B11]
- Linux–PAM Modül Yazıcısının Kılavuzu[B12]
- Linux–PAM Uygulama Geliştiricisinin Kılavuzu[B13]

6.2. Genel Güvenlik

- linuxsecurity.com[B14]
- securitywatch.com[B15]
- Güvenlik NASIL
- Packetstorm[B17]

6.3. Çevrimdışı Belgeler

Sisteminizdeki kılavuz sayfalarından çokça bilgi edinebilirsiniz. Aşağıda kullanıcı kimlik kanıtlaması ile ilgili kılavuz dosyalarının bir listesi bulunmaktadır. Parantez içindeki sayılar kılavuz sayfalarının bölümlerini

göstermektedir. `passwd(5)` kılavuz sayfasını görüntülemek için konsoldan **`man 5 passwd`** komutunu çalıştırmalısınız.

- **`passwd(5)`** [B18]
- **`crypt(3)`** [B19]
- **`pam.d(5)`**
- **`group(5)`** [B21]
- **`shadow(5)`** [B22]

7. Sonuç

Umarım bu NASIL yardımcı olmuştur. Sorularınızı, yorumlarınızı ve önerilerinizi <petehern (at) yahoo.com> adresine gönderebilirsiniz.

Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

[B2] http://www.cgr.org/

[B3] ../man/man3/man3-«crypt.pdf

[B4] http://www.faqs.org/rfcs/rfc1321.html

[B5] http://www.kernel.org/pub/linux/libs/pam/Linux-«PAM-«html/pam.html

[B7] http://www.kernel.org/pub/linux/libs/pam/Linux-«PAM-«html/pam.html

[B9] http://www.tldp.org/HOWTO/Shadow-«Password-«HOWTO.html

[B10] http://pam.sourceforge.net/mod_auth_pam/

[B11] http://www.kernel.org/pub/linux/libs/pam/Linux-«PAM-«html/pam.html

[B12] http://www.kernel.org/pub/linux/libs/pam/Linux-«PAM-«html/pam_modules.html

[B13] http://www.kernel.org/pub/linux/libs/pam/Linux-«PAM-«html/pam_appl.html

[B14] http://www.linuxsecurity.com/

[B15] http://www.securitywatch.com

[B17] http://www.packetstormsecurify.org

[B18] ../man/man5/man5-«passwd.pdf

[B19] ../man/man3/man3-«crypt.pdf

[B21] ../man/man5/man5-«group.pdf

(B22) ../man/man5/man5-«shadow.pdf

Bu dosya (user-auth-howto.pdf), belgenin XML biçiminin TEXLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

23 Ocak 2007