PostgreSQL İstemci Yetkilendirmesi ve Güvenliği

Bu makale de PostgreSQL 8.0 (8.0.7) için istemci yetkilendirmesi (client authentication) ve PostgreSQL güvenliği hakkında bilgi verilecektir.
Bu belgenin en son haline
http://www.enderunix.org/docs/postgresql/postgresql_security.pdf adresinden ulaşabilirsiniz.

Yazar: İsmail YENİGÜL
EnderUNIX Çekirdek Takım Üyesi
ismail at enderunix dot org
ismail.yenigul at endersys dot com
http://www.enderunix.org
Değişiklikler
21 Mar 2006 Sal EET 13:52:35 - İlk Yazım
22 Mar 2006 Çar EET 14:57:55 - Kullanıcı Haklarını Kısıtlama kısmı eklendi.

İçindekiler

1. PostgreSQL'e Erişim

Ön tanımlı olarak PostgreSQL sadece localhost'a bağlanır (bind).

```
$ netstat -na|grep LISTEN |grep 5432
tcp4       0      0  127.0.0.1.5432      *.*     LISTEN
```

Uzaktan erişim için öncelikle postgresql.conf dosyasındaki
`listen_addresses = 'localhost'`
Değerini sunucunun IP adresini dinleyecek şekilde değiştirilmesi gerekir. Bunun için pgsql kullanıcısı ile listen_addresses ifadesinin başındaki # kaldırıldıktan sonra bu değişkenin değeri '*' yapılır.

```
listen_addresses = '*'
```

Değişikliğin etkinleşmesi için PostgreSQL servisi kapatılıp tekrar açılır. PostgreSQL'i FreeBSD port ağacından kurduysanız aşağıdaki komutlarla servisi kapatıp açabilirsiniz.

```
# /usr/local/etc/rc.d/010.pgsql.sh stop
postmaster stopped
# /usr/local/etc/rc.d/010.pgsql.sh start
# netstat -na|grep LISTEN|grep 5432
tcp4       0      0  *.5432         *.*                LISTEN
```

Artık PostgreSQL'e uzaktan erişim sağlanabilir.

2. pg_hba.conf

PostgreSQL'e erişim hakları pg_hba.conf dosyasında tanımlanır. Bu dosya ön tanımlı olarak PostgreSQL data dizininde (*/usr/local/pgsql/data*) bulunur. HBA, **H**ost **B**ased **A**uthentication manasına gelmektedir. pg_hba.conf dosyasında erişim izinlerinin formatı aşağıdakiler gibidir.

```
local       database  user  authentication-method  [authentication-option]
host        database  user  CIDR-address  authentication-method  [authentication-option]
hostssl     database  user  CIDR-address  authentication-method  [authentication-option]
hostnossl   database  user  CIDR-address  authentication-method  [authentication-option]
host        database  user  IP-address  IP-mask  authentication-method  [authentication-
option]
hostssl     database  user  IP-address  IP-mask  authentication-method  [authentication-
option]
hostnossl   database  user  IP-address  IP-mask  authentication-method  [authentication-
option]
```

local: UNIX domain soket üzerinden erişim kuralını belirler.
Not: Ön tanımlı olarak PostgreSQL Unix domain soket için */tmp/.s.PGSQL.5432* dosyasını oluşturur.

host: TCP/IP üzerinden bağlantı yapabilecek istemci listesini belirler. Hem SSL hem de SSL olmayan bağlantıyı kabul eder.

hostssl: Sadece SSL bağlantılarını kabul eder.
hostnossl: Sadece SSL olmayan bağlantıları kabul eder.
database: İzin verilecek veritabanını belirtir. Tüm veritabanlarını ifade etmek için all ifadesi kullanılır. Bu alanın değerinin sameuser olması bağlantı yapacak olan kullanıcı adı ile bağlantı yapılacak veritabanının aynı olmasını şart koşar. samegroup değeri ise bağlantı yapacak olan kullanıcı grubunun bağlanacağı veritabanı ile aynı olmalıdır.

user: Erişim yapacak kullanıcı adını belirler. Tüm kullanıcıları belirtmek için all ifadesi kullanılır. Virgül kullanarak birden fazla kullanıcı belirtilebilir. Grubu ifade etmek için ise grup adının başına + konulması yeterlidir. Listenin dosyadan alınması için ise @dosyası biçiminde bir format kullanılabilir.

CIDR-address: IP adresi aralığını belirler. Örnek yazım 192.168.0.1/32, 212.156.115.0/24

IP-address IP-mask: CIDR-address formatı yerine IP adresi ve netmask kullanılarak gösterimi sağlar. Örnek yazım: 192.168.0.1 255.255.255.0

authentication-method: Bağlantıyı gerçekleştirmek için hangi tür yetkilendirme yapılacağını belirler. Bu kısım trust, reject, md5, crypt, password, krb4, krb5, ident,pam değerlerinden birini alabilir.

trust: Kullanıcıların veritabanına parolasız bağlanmasını sağlar.
reject: Erişimi reddeder.
md5: md5 formatında şifrelenmiş parola ile giriş gerekir.
crypt: bağlantı için crypt formatında şifrelenmiş parola girmesi gerekir.
password: Düz metin parola ile girişe izin verir.
pam: PAM kullanarak yetkilendirme sağlar.

pg_hba.conf dosyasındaki ön tanımlı kurallar aşağıdaki gibidir.

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only
local   all         all                               trust
# IPv4 local connections:
host    all         all         127.0.0.1/32          trust
# IPv6 local connections:
host    all         all         ::1/128               trust
```

Bu dosya PostgreSQL servisinin başlatıldığı zaman veya postmaster prosesi SIGHUP sinyali aldığı zaman okunur. Yapılan değişikliklerin etkinleşmesi için "*pg_ctl reload*" komutu kullanılabilir.

Yukarıdaki erişim kurallarından "local" ile başlayan satırın sonunda "trust" değeri ile Unix kullanıcılarının PostgreSQL veritabanlarına parolasız olarak erişimi sağlanır. Eğer sunucudaki tüm sistem kullanıcılarının veritabanına parolasız erişmesini istemiyorsanız trust yetkilendirme yöntemini diğer yöntemlerle değiştirmeniz gerekir.

Örneğin yerel kullanıcıların yetkilendirilmesi için md5 kullanılmasını sağlamak için bu satır aşağıdaki gibi değiştirilir.

```
local   all         all                               md5
```

```
Değişiklik etkinleştirilmeden (pg_ctl reload) önce sistem
kullanıcılarının veritabanına erişimi için daha önceden bir parola
atanmadıysa aşağıdaki komutlar verilerek parola atama yapılabilir.
```

```
# su - pgsql
$ psql template1
Welcome to psql 8.0.7, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# ALTER USER pgsql with password 'deneme1';
ALTER USER
#\q

Değişikliği etkinleştirmek için
$ pg_ctl reload
postmaster signaled
$
```

Tekrar bağlantı denemesi yapıldığında

```
 $ psql template1
Password:
Welcome to psql 8.0.7, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \q
```

Örnek erişim kuralları

```
# enderunix kullanıcısına 192.168.1.5'den tüm veritabanlarına
parolasız erişim.
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
host   all      enderunix       192.168.1.5/32        trust
```

```
# acikkod kullanıcısına 192.168.1.0 ağından deneme ve bsd
veritabanına parola ile erişim.
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

host    deneme,bsd     acikkod      192.168.1.0/24        md5
```

3. Güvenlik

3.1. Parola Güvenliği

PostgreSQL 7.3'den itibaren ön tanımlı olarak PostgreSQL'de parolaların şifreli olarak saklanmaktadır. Bu değerin etkin olup olmadığını anlamak için data/postgresql.conf dosyasında *password_encryption* değişkeninin değeri *"true"* olup olmadığına bakılır veya aşağıdaki SQL sorgusu çalıştırılabilir.

```
template1=# SHOW password_encryption ;
 password_encryption
---------------------
 on
(1 row)

template1=#
```

```
Kullanıcı oluşturma şablonu aşağıdaki gibidir.

CREATE USER name [ [ WITH ] seçenek [ ... ] ]

Seçenek:

      SYSID uid
    | CREATEDB | NOCREATEDB
    | CREATEUSER | NOCREATEUSER
    | IN GROUP groupname [, ...]
    | [ ENCRYPTED | UNENCRYPTED ] PASSWORD 'password'
    | VALID UNTIL 'abstime'
```

Eğer *password_encryption* değişkeni etkinleştirilmiş ise kullanıcı oluştururken parolası otomatik olarak şifrelenmiş (ENCRYPTED) olarak saklanmak bu yüzden kullanıcı oluşturulurken ENCRYPTED parametresinin kullanılmasına gerek yoktur.

3.2. Veritabanı Kullanıcılarının Haklarını Kısıtlama

Hangi kullanıcının hangi haklara sahip olduğu aşağıdaki sorgu ile görülebilir.

```
template1=# SELECT usename,usecreatedb,usesuper,usecatupd from
pg_shadow ;
  usename  | usecreatedb | usesuper | usecatupd
-----------+-------------+----------+-----------
 enderunix | t           | t        | t
 apache    | f           | f        | f
 user1     | f           | t        | t
 ismail    | t           | t        | t
 pgsql     | t           | t        | t
(5 rows)

template1=#
```

usecreatedb alanı kullanıcının veritabanı oluşturup oluşturamayacağını belirler.
usesuper alanı kullanıcının başka kullanıcı oluşturup oluşturamayacağını belirler.
usecatupd alanı kullanıcının sistem kataloglarını güncelleyip güncelleyemeyeceğini belirler. Değerin ***t*** olması ***true f*** olması ***false*** manasına gelmektedir.
Bu bilgiler ışığında sistemdeki kullanıcıların haklarında gerekirse engellemeler yapılabilir.

3.3. SSL

3.3.1. PostgreSQL'de SSL'in Etkinleştirilmesi

PostgreSQL'de SSL özelliğini etkinleştirmek için öncelikle PostgreSQL ***--with-openssl*** parametresi ile derlenmiş olması gerekir.

Daha sonra /usr/local/data/postgresql.conf dosyasındaki "ssl" değişkeninin değeri "true" atanmalıdır.

```
ssl = true
```

Bu işlemlerden sonra SSL bağlantılarında kullanmak için SSL sertifikaları oluşturulmalıdır. Kendinden imzalı (self-signed) SSL sertifikalarını oluşturmak için aşağıdaki komutlar verilebilir. Kendinden imzalı sertifikalar test amaçlı olarak kullanılmalıdır...

```
# su – pgsql
$ cd /usr/local/pgsql/data
$ openssl req -new -text -out server.req
Generating a 1024 bit RSA private key
..............++++++
...++++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
```

```
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TR
State or Province Name (full name) [Some-State]:Marmara
Locality Name (eg, city) []:Istanbul
Organization Name (eg, company) [Internet Widgits Pty Ltd]:EnderUNIX
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:devel.enderunix.org
Email Address []:ismail.yenigul@endersys.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Yukarıdaki "*Common Name*" kısmına mutlaka sunucunun hostname'i yazılmalıdır. Yukarıdaki anahtar her açılışta parola (pass phrase) soracaktır. Bunu kaldırmak için aşağıdaki komut verilir.

```
$ openssl rsa -in privkey.pem -out server.key
$ rm privkey.pem
```

server.crt dosyasını oluşturmak için de aşağıdaki komut verilir.

```
$ openssl req -x509 -in server.req -text -key server.key -out
server.crt
$ chmod og-rwx server.key
```

Yukarıdaki sertifikaları PostgreSQL'in data dizininden (/usr/local/pgsq/data) başka bir dizinde oluşturuldu ise sertifikaların bu dizine kopyalanması gerekir.
Sunucu üzerinde root.crt dosyası olmadığı zaman istemci ile sunucu arasında güvenli SSL bağlantısı kurulacaktır fakat istemci SSL sertifikasına göre yetkilendirme yapılmayacaktır. Bunun için tam bir SSL haberleşmesi için gerçek SSL sertifikasının alınması gerekir.

Değişikliğin etkinleşmesi için PostgreSQL yeniden başlatılmalıdır.

```
$ pg_ctl –D /usr/local/pgsql/data  stop
$ pg_ctl –D /usr/local/pgsql/data –m fast start –l /tmp/logfile
```

PostgreSQL çalışmamışsa hatanın sebebini bulmak için postgresql.conf dosyasındaki *silent_mode* değerini false yaparak PostgreSQL tekrardan başlatılır.

```
silent_mode = false
```

Muhtemel bir hata aşağıdaki gibi olabilir

```
$ tail /tmp/logfile
FATAL:  could not load server certificate file
"/usr/local/pgsql/data/server.crt": No such file or directory
$
```

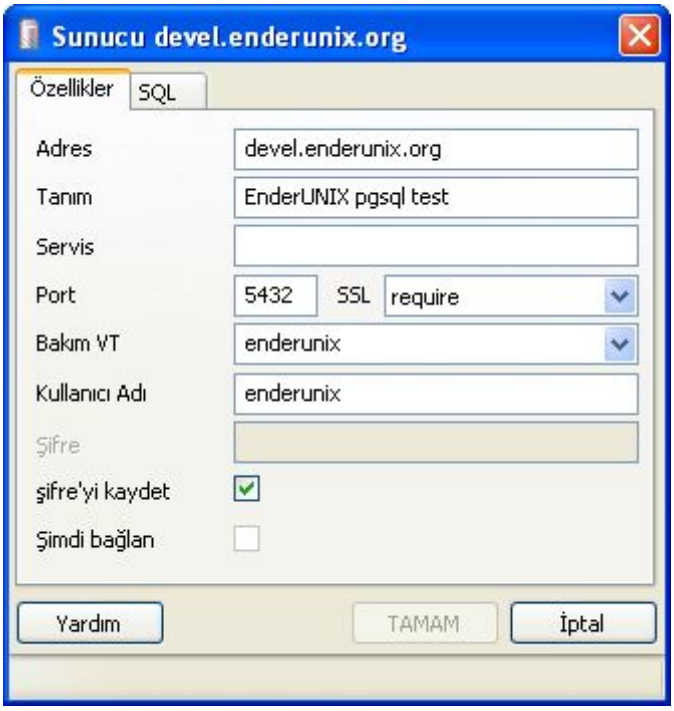
Eğer prosesler başarı ile çalışmış ve slient_mode değeri false yapılmış ise /tmp/logfile dosyasında aşağıdaki gibi uyarı mesajı görebilirsiniz.

```
$ tail /tmp/logfile
LOG:  could not load root certificate file
"/usr/local/pgsql/data/root.crt": No such file or directory
DETAIL:  Will not verify client certificates.
$
```

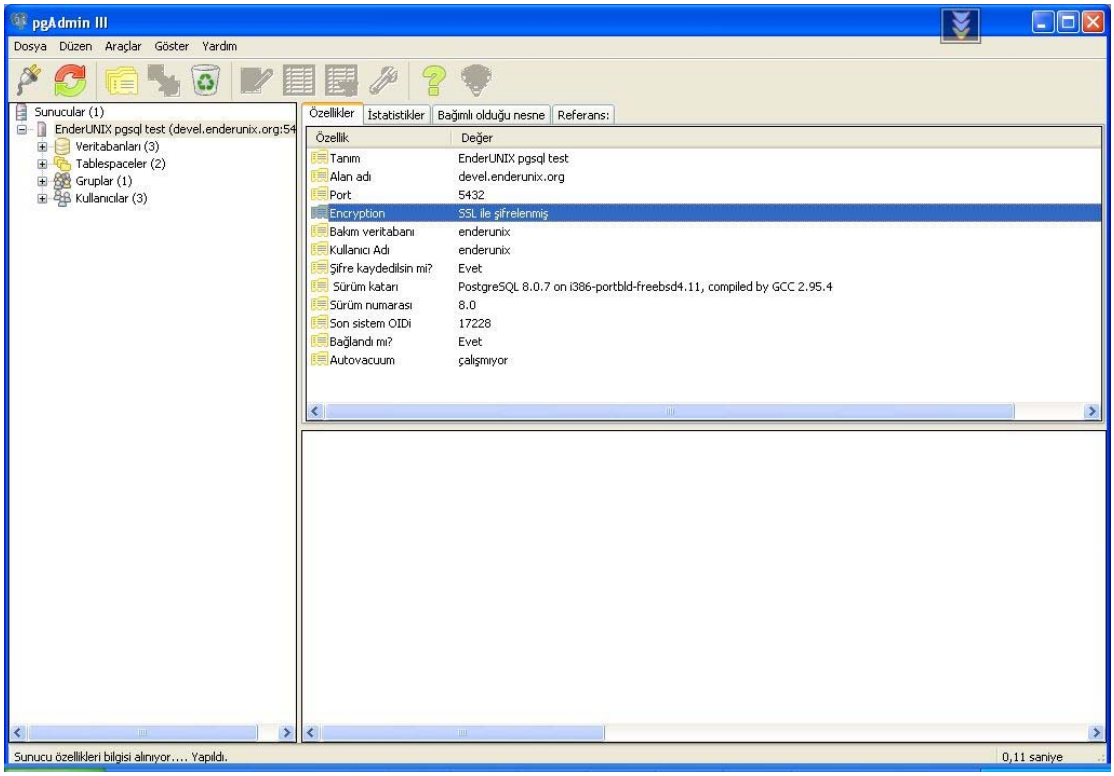
pılma Bu mesaja göre PostgreSQL'in SSL özelliği ile çalıştığını fakat root.crt dosyasını yükleyemediğini ifade eder.

3.4. PgAdmin ile SSL Bağlantı Testi

PgAdmin Sunucu özelliklerine tıklanır ve SSL özelliği etkinleştirilir.



Bağlantı kurulduktan sonra SSL bağlantı bilgisi aşağıdaki ekrandan görülebilir. (**Encryption: SSL ile şifrelenmiş** ibaresi)



3.5. SSH Üzerinden İstemci Bağlantısı

Bu tür bir bağlantı yöntemi SSL bağlantısını desteklemeyen (psql gibi) uygulamaların PostgreSQL'e güvenli bir şekilde bağlanmasını sağlamak için kullanılır. Öncelikle pg_hba.conf dosyasında PostgreSQL sunucunun kendi IP adresinden erişim izni verilmelidir. PostgreSQL'in IP adresi 192.168.1.10 ise aşağıdaki gibi bir satırı pg_hba.conf dosyasına eklemek yeterlidir.

```
host    all         all         192.168.1.10/32       md5
```

Değişikliği etkinleştirmek için

```
$ pg_ctl reload
postmaster signaled
$
```

PostgreSQL sunucuya bağlanmak istediğimiz istemciden aşağıdaki gibi bir komutla SSH tünel oluşturulur.

```
istanbul[ismail]$ ssh -L 2000:192.168.1.10:5432 ismail@192.168.1.10
Password:
Last login: Tue Mar 21 09:57:38 2006 from 212.X.X.X
$
```

Yukarıdaki komutta 2000 ile belirtilen port tünelin kendi bilgisayarımız üzerindeki port numarasıdır. 1024'den büyük ve herhangi boş bir port seçilebilir. 192.168.1.10 ise PostgreSQL sunucunun IP adresi 5432 PostgreSQL servisinin port numarası. Bu

erişimin gerçekleşmesi için aynı zamanda PostgreSQL sunucu üzerinde ismail adlı sistem kullanıcısının tanımlı olması gerekir.

PostgreSQL'e bağlandığımız istemcide başka bir terminalde daha açılarak aşağıdaki komut verilir. Karşı taraftaki veritabanına bağlanmak için yerel makinenin 2000. portuna bağlanmak yeterlidir. Bu porta gelen tüm istekler tünel üzerinden veritabanı sunucunun 5432 nolu portuna gönderilecektir.

```
istanbul[ismail]$ psql -p 2000 -h localhost -U ismail template1
Password:
Welcome to psql 8.0.6, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)

template1=#
```

4. Kaynaklar

Practical PostgreSQL kitabı - http://www.commandprompt.com/ppbook/book1
PostgreSQL El kitabı: http://www.postgresql.org/docs/8.0/interactive/index.html