

Doküman Kodu: BGT-5002

MySQL VERİTABANI GÜVENLİĞİ KILAVUZU

SÜRÜM 1.00

09 ŞUBAT 2009

Hazırlayan: Gökhan ALKAN

ÖNSÖZ

Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü (UEKAE)'nin misyonu, "bilgi güvenliği, haberleşme ve ileri elektronik alanlarında Türkiye'nin teknolojik bağımsızlığını sağlamak ve sürdürmek için nitelikli insan gücü ve uluslararası düzeyde kabul görmüş altyapısı ile bilimsel ve teknolojik çözümler üretmek ve uygulamaktır". Bu ana hedef göz önünde bulundurularak belirlenen "bilgi güvenliği, haberleşme ve ileri elektronik alanlarında yeni teknolojilerin geliştirilmesine öncülük eden uluslararası bilim, teknoloji ve üretim merkezi olmak" vizyonuna ulaşılabilmesi ve ülkenin ihtiyacı olan teknolojilerin geliştirilmesi için Enstitü'nün akredite test ortam ve laboratuvarlarında temel ve uygulamalı araştırmalar yapılmakta ve ihtiyaç sahiplerine teknik destek sağlanmaktadır.

Bu doküman "Ulusal Bilgi Sistemleri Güvenlik Projesi" kapsamında hazırlanmış olup ihtiyaç sahiplerini bilgi sistemleri güvenliği konusunda bilinçlendirmeyi hedeflemektedir. Tüm kurum ve kuruluşlar bu dokümandan faydalanabilir.

Bu dokümanda bahsi geçen belirli ticari marka isimleri kendi özgün sahiplerine aittir. Burada anlatılanlar tamamen tavsiye niteliğinde olup değişik ürünler/yapılandırmalar için farklılık gösterebilir. UEKAE, yapılan uygulamalardan doğabilecek zararlardan sorumlu değildir. Bu doküman UEKAE'nin izni olmadan değiştirilemez.

BİLGİLENDİRME

Bu dokümanın oluşturulmasında emeđi geçen Ađ Güvenliđi personeline ve dokümanı gözden geçirip fikirlerini öne sürerek dokümanın olgunlaşmasına katkıda bulunan Tahsin TÜRKÖZ'e teşekkürü bir borç biliriz.

İÇİNDEKİLER

1. GİRİŞ	5
1.1 Amaç ve Kapsam.....	5
1.2 Hedeflenen Kitle.....	5
1.3 Dokümanda Kullanılan Semboller	6
2. MySQL VERİTABANI.....	7
2.1 MySQL Veritabanı Genel Güvenlik Açıklık Görünümü	7
2.2 MySQL Sunucu Servisi Yapılandırma Dosyası Genel Formatı	8
3. MySQL VERİTABANI GÜVENLİK YAPILANDIRMALARI.....	9
3.1 MySQL Veritabanı Çalışma Ortamı Güvenliğinin Sağlanması	9
3.2 MySQL Veritabanını Süper Kullanıcı Harici Bir Kullanıcı Hakları ile Çalıştırmak	9
3.3 MySQL Veritabanı Dosya ve Dizin Erişim Güvenliği.....	10
3.4 MySQL Veritabanı Servisinin Çalışacağı IP Adres Bilgisinin Belirlenmesi	11
3.5 Yerel Erişim Güvenliğinin Artırılması	12
3.6 Ağ Trafikinin Gözlemlenmesi	12
3.7 Kullanıcıları Erişim Güvenliği	19
3.8 Parola Güvenliği	21
3.9 MySQL Süper Kullanıcı Güvenliği	22
3.10 Örnek Veritabanı ve Tabloların Güvenliği	23
3.11 Yama Kontrolü	23
3.12 DNS Güvenliği	23
4. KAYNAKÇA	24

1. GİRİŞ

Bu kılavuzda yaygın olarak kullanılan açık kaynak kodlu veritabanı MySQL'in güvenlik yapılandırması anlatılmaktadır.

1.1 Amaç ve Kapsam

Bu dokümanın amacı, MySQL veritabanlarının güvenli kabul edilmesi amacıyla uygulanması gereken politika, prensip ve esasların belirlenmesidir. Dokümanda, belirlenen adımların uygulanmasına yönelik yöntemler de yer almaktadır.

1.2 Hedeflenen Kitle

Bu doküman MySQL veritabanlarının yönetilmesinden ve güvenli olarak yapılandırılmasından sorumlu kişiler tarafından kullanılabilir.

1.3 Dokümanda Kullanılan Semboller

Örnekler	Açıklaması
<code>chmod 440 my.cnf</code>	Yapılandırma ya da komut alıntıları
<i>mysqladmin</i>	Metin içerisinde komut ya da dosya ismi

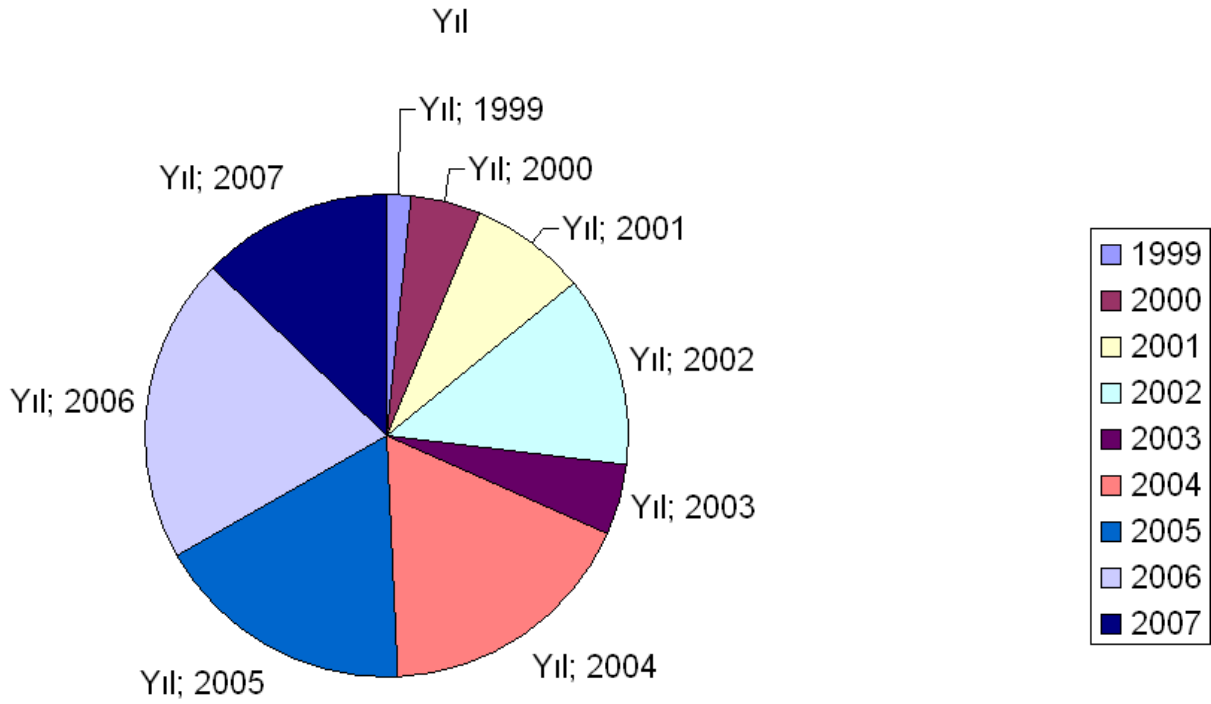
2. MySQL VERİTABANI

MySQL sunucu servisi hem Windows hem de Linux/Unix sistemler üzerinde koşan, genellikle php ile uygulama geliştiren geliştiriciler tarafından tercih edilen, kullanım oranı yüksek bir veritabanı uygulamasıdır. Burada Unix/Linux sistemler üzerinde çalışan MySQL veritabanı uygulamasının güvenli hale gelmesi için gerekli adımlardan bahsedilecektir. Burada anlatılanlar ne kadar Unix/Linux sistemlere has bilgiler olsada temel olarak diğer işletim sistemleri içinde uygulanabilmektedir. Sıkılaştırma adımlarından genel olarak aşağıda bahsedilmiştir:

- MySQL sunucu servisi çalışma ortamı güvenliği (*chroot*),
- MySQL sunucu servisinin *root* harici “kullanıcı/grup” ikilisinin hakları ile çalışacağının belirlenmesi,
- MySQL sunucusu dosya dizin erişim güvenliği,
- MySQL sunucu servisinin çalışacağı IP adresi bilgisinin belirlenmesi,
- Yerel erişim güvenliğinin artırılması,
- Ağ trafiğinin şifrenmesi,
- MySQL kullanıcıları erişim güvenliği,
- Parola güvenliği,
- MySQL *root* kullanıcısı güvenliği,
- Örnek veritabanı ve tabloların güvenliği,
- Yama Kontrolü,
- DNS Güvenliği

2.1 MySQL Veritabanı Genel Güvenlik Açıklık Görünümü

MySQL yapılandırma dosyası genel formatının belirtilmesinin ardından, MySQL sunucu servisinin nasıl daha güvenli hizmet verebileceği noktasında yapılacaklar anlatılacaktır. Gerçekleştirilecek adımların ana başlıkları yukarıda listelenmiştir. Bugüne kadar 3.3 sürümünden 5.0 sürümüne kadar MySQL veritabanında bulunan ve yayınlana açıklık listesine http://forge.MySQL.com/wiki/Security_Vulnerabilities_In_MySQL_Server adresinden ulaşılabilir. Bu durum ayrıca Şekil 2-1 ‘de gösterilmiştir. Burada belirtilen açıklıkların CVE numaraları ile belirtilmiştir. Şekil 2-1’deki grafik bu verilere göre elde edilmiştir.



Şekil 2-1 MySQL sunucu veritabanı üzerinde (1999-2007 Yılları Arası) bulunan ve çözümlenen açıklıkların yıllara göre dağılımı

2.2 MySQL Sunucu Servisi Yapılandırma Dosyası Genel Formatı

MySQL sunucu servisi genel yapılandırma dosyası olarak `“/etc/my.cnf”` kullanılmaktadır. Eğer MySQL servisi `chroot` ortamı altında çalıştırılıyor ise, yapılandırma dosyası olarak `chroot` ortamı için belirlenen dizinin altında bulunan `“/etc/my.cnf”` dosyası kullanılacaktır. Bu dosyanın genel yapısı ve basitçe yapılandırma seçeneklerinin alacağı değerler aşağıda belirtilmiştir.

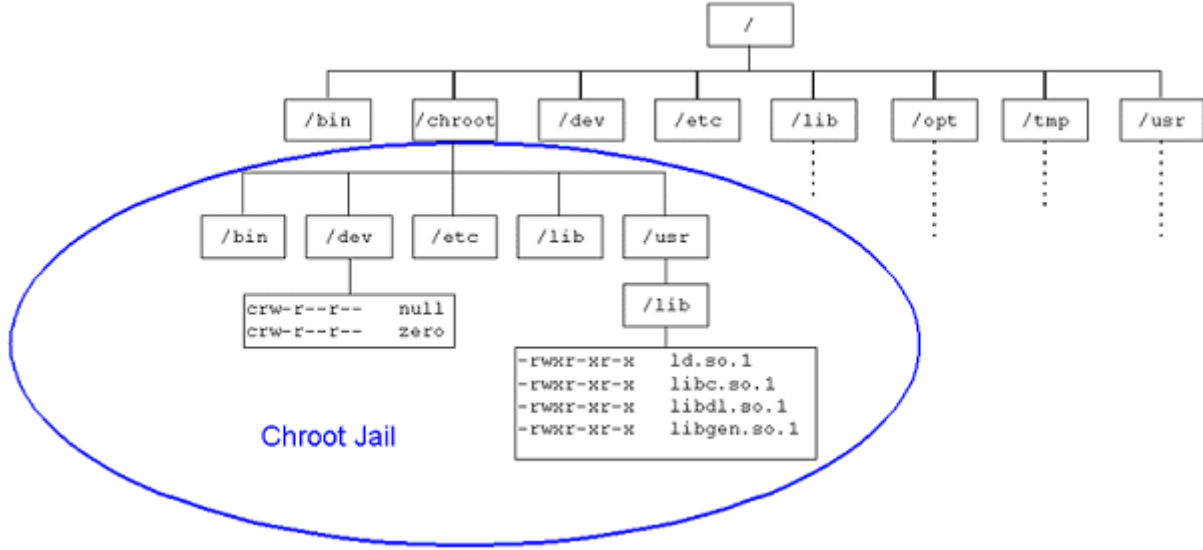
- yapılandırma_seçeneği=değer

Yukarıda belirtilen gösterim komut satırından aynı değerın `mysqld` ikilisi için belirtimi aynıdır. Örneğin `“my.cnf”` yapılandırma dosyası içerisinde belirtimi gerçekleştirilen `“port = 3306”` ile `“mysqld --port= 3306”` çalıştırılması aynı anlamdadır. `“#”` ile başlayan satırlar açıklama satırı olarak ele alınmaktadır.

3. MySQL VERİTABANI GÜVENLİK YAPILANDIRMALARI

3.1 MySQL Veritabanı Çalışma Ortamı Güvenliğinin Sağlanması

MySQL sunucu servisinin *chroot* ortamı altında çalışması sağlanarak, MySQL sunucu servisinin işletim sistemi kaynaklarının tümüne erişimi kısıtlanmalıdır. Bu şekilde MySQL sunucu servisi sadece ihtiyaç duyacağı kaynaklara erişim sağlanmış olacaktır. Bu durum Şekil 3-1'de gösterilmiştir.



Şekil 3-1 Chroot Ortamının Şematize Edilmesi

3.2 MySQL Veritabanını Süper Kullanıcı Harici Bir Kullanıcı Hakları ile Çalıştırmak

- *[group]*

group ile belirtilen değer yapılandırması gerçekleştirilmek istenen programın adı ya da *group*. Yapılandırma seçenekleri *[group]* ile başlayıp bir sonraki *[group]* adına kadar geçerli olmaktadır. Örneğin *[mysqld]* ile *mysqld* sunucu yazılımı için etkinleştirilmek istenen değerler belirtilecektir.

- “*user = kullanıcı_adi*” değeri ile MySQL sunucu servisinin hangi kullanıcı hakları ile çalıştırılacağı belirlenmektedir. Burada *root* kullanıcı haklarına sahip olmayan MySQL grubuna üye MySQL kullanıcısı belirtilmiştir.

```
# vi my.cnf

[mysqld]

...

user = mysql
```

3.3 MySQL Veritabanı Dosya ve Dizin Erişim Güvenliği

MySQL sunucu servisi tarafından kullanılan izin ve dosya sahipliği hakları *root* kullanıcıya, grup sahipliği hakları ise MySQL sunucu servisini çalıştıran kullanıcının dahil olduğu gruba ait olmalıdır. MySQL sunucu servisinin *mysql/mysql* kullanıcı, grup ikilisi hakları ile çalıştırıldığı düşünüldüğünde örnek bir dosya/dizin sahipliği hakları aşağıdaki gibi olmalıdır.

```
# chown root:mysql [dosya/dizin]
```

Yukarıda belirtilen dosya izin sahipliklerine göre, dosya erişim izinleri için sayısal notasyonda 440 (r--r----), izin erişim izinleri için ise 550 (r-xr-x---) şeklinde olmalıdır. Bu durum için istisna teşkil edecek durumlar olabilmektedir. Örneğin *log*, *var* ya da yeni oluşturulan bir veritabanı için kullanılacak olan izinler içinde MySQL kullanıcısının dahil olduğu grubun yazma hakkının olması gerekmektedir. Bunun yanında MySQL sunucu servisinin tarafından gerçekleştirilen veri değişimleri hakkında bilgileri içeren “.bin” uzantılı dosyalar içinde yazma hakkının bulunması gerekmektedir. Bu duruma örnek teşkil edecek bir kullanım aşağıda gösterilmiştir.

```
# chown root:mysql /var/mysql/{log,var}
# chmod 550 /var/mysql/{log,var}
# chmod 460 /var/mysql/var/mysql-bin.*
```

“*my.cnf*” dosyasının sahipliği *root* kullanıcısı, grup sahipliği olarak ise MySQL sunucu servisini çalıştıran kullanıcının (Örneğin “*mysql/mysql*” kullanıcı/grup ikilisi) dahil olduğu grup olarak yapılandırılmalı ve erişim izinleri için ise sayısal notasyon olarak 440 olarak belirlenmelidir.

```
# chown root:mysql my.cnf
# chmod 440 my.cnf
```

3.4 MySQL Veritabanı Servisinin Çalışacağı IP Adres Bilgisinin Belirlenmesi

“*bind-address= ip adres bilgisi*” değeri ile MySQL sunucu servisinin bağlantıları dinleyeceği ip adres bilgisi tanımlanmaktadır. Ön tanımlı olarak MySQL sunucu yazılımı bütün ağ arayüzlerinden gelen bağlantıları kabul edecek şekilde yapılandırılmaktadır. Eğer MySQL sunucu servisi ve web uygulaması aynı sunucu makine üzerinde çalışacak ise bu değer “127.0.0.1” olarak belirtilmelidir. Aslında bu durumda “*skip-networking*” değeri kullanılabilir. “*skip-networking*” değeri ile MySQL sunucu yazılımı hiçbir şekilde TCP/IP bağlantılarını dinlemez. Bütün etkileşim (Linux/Unix) unix soketleri aracılığı ile gerçekleştirilir. Ancak burada Apache ve MySQL sunucu servisleri için ayrı ayrı *chroot* dizinleri oluşturulmuş ise, bu seçenek kullanılamamaktadır. Bu seçeneğin yerine “*bind-address=127.0.0.1*” değeri ile sadece belirtilen ip adreslerinden gelen bağlantıları kabul edecek şekilde yapılandırılmalıdır. Eğer Apache ve MySQL sunucu servisleri aynı sunucu makine üzerinde değil iseler, güvenlik duvarı ya da *tcp_wrapper* gibi uygulamalar aracılığı ile ip adres kısıtlaması gerçekleştirilmelidir. Bu şekilde sadece belirtilen ip adreslerinden bağlantı kabul edilecektir. Bunun yanında iletişimin açık bir şekilde değil, şifreli olarak ağ üzerinden gerçekleştirilmesi sağlanmalıdır.

```
# vi my.cnf
[mysqld]
...
bind-address=127.0.0.1
...
#
```

netstat komutu ile MySQL sunucu servisinin hangi ağ ara yüzünden gelen bağlantıları kabul edeceği görülebilmektedir.

```
# netstat -plunat | grep "mysqld"
tcp        0  0  127.0.0.1:3306      0.0.0.0:*           LISTEN      5010/mysqld
#
```

3.5 Yerel Erişim Güvenliğinin Artırılması

“*LOAD DATA LOCAL INFILE*” komutları engellenerek, yerel dosyaların yetkisiz erişimlerden korunması sağlanmalıdır. Bu şekilde özellikle *php* ile geliştirilen web uygulamalarında bulunabilecek olan, sql enjeksiyonu saldırılarından kaynaklanan yetkisiz yerel dosyaların okunmasını engellemektedir. Bunun için MySQL sunucu servisi yapılandırma dosyasında aşağıdaki değer bulunmalıdır.

```
# vi my.cnf
...
local-infile      = 0
...
#
```

3.6 Ağ Trafikinin Gözlemlenmesi

Aksi belirtilmedikçe MySQL sunucu istemci arasındaki ağ trafiği açık metin olarak gerçekleştirilmektedir. Aslında bu sadece MySQL’e has bir özellik değil, sunucu istemci mimarisi ile çalışan bütün uygulamalar için geçerli bir durumdur. Kötü niyetli kişiler sunucu istemci arasında gerçekleştirilen açık metin trafiği dinlererek hassas bilgilere erişim sağlayabilirler. Aşağıda MySQL sunucusu ile istemcisi arasındaki ağ trafiğinin gözlemlenmesi halinde nelerin yapılabileceğini göstermek açısından MySQL sunucu sistemi üzerinde *tcpdump* ve *strings* komutları kullanılarak gerçekleştirilen sorgu işlemleri gösterilmiştir.

```
# tcpdump -l -i eth0 -w - src or dst port 3306 | strings
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes
5.0.45
wRuuJ5g,
LLBPALES
show databases
SCHEMATA
Datab
show tables
TABLE_NAMES
select @@version_comment
@@version_com
show databases
```

SCHEMATA

Datab

Görüldüğü gibi MySQL istemcisinden gerçekleştirilen sorgular açık olarak görülmektedir. Aynı işlem MySQL sunucu istemci arasındaki trafiğin şifreli olarak iletildiği durumda aşağıda görüldüğü gibi olmaktadır.

```
# tcpdump -l -i eth0 -w - src or dst port 3306 | strings
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96
bytes
gLOY
%DgLOY
gLOY
lnp@
%HgLOY4
5.0.45
bA[ 'P;x1
gLO\
gLO_
4nq@
%KgLO_
gLOo
4nr@
```

Yukarıda görüldüğü gibi MySQL sunucu ve istemcisi arasındaki ağ trafiği şifreli bir biçimde gerçekleştirildiğinde, sorgular açık olarak görüntülenememektedir.

MySQL sunucu istemci trafiğinin şifrelenmesi için uygulanabilecek yöntemlerden bir tanesinde *openssl* uygulamasının kullanılmasıdır. MySQL sunucu istemci trafiğinin şifrelenmesi için kullanılacak açık kaynak kodlu yöntemlerden bazıları olarak *OpenSSH*, *OpenVPN* ve *OpenSSL* gösterilebilir. Burada *OpenSSL* kullanılarak nasıl sunucu istemci arasındaki trafiğin şifreleneceği anlatılacaktır. Öncelikle mevcut MySQL sunucu servisinde SSL desteğinin aktif olup olmadığına bakılmalıdır. Şekil 3-2’de görüldüğü gibi “*SHOW VARIABLES LIKE 'have_openssl'*” SQL sorgusu ile görüntülenebilir.

```
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.45 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SHOW VARIABLES LIKE '%openssl%';
+-----+-----+
| Variable_name | Value      |
+-----+-----+
| have_openssl  | DISABLED  |
+-----+-----+
```

Şekil 3-2 MySQL SSL desteğinin kontrol edilmesi

“Value” değerinin “DISABLED” olması, veritabanı sunucusunun SSL desteğinin olduğunu fakat doğru parametreler ile başlatılmadığını göstermektedir. SSL desteğinin kullanılabilmesi için MySQL sunucu yazılımının derlenmesi esnasında “*--with-vio --with-openssl*” parametrelerinin kullanılması gerekmektedir.

MySQL sunucu servisine SSL desteğinin verilmesinin ardından sunucu ve istemci için gerekli sertifikalar oluşturulmalı ve son olarak sunucu ve istemci yapılandırma dosyalarında gerekli işlemler gerçekleştirilmelidir. Burada sertifika otoritesi için gerekli sertifikalar öncelikle bizim tarafımızdan üretilir. Daha sonra sunucu istemci için gerekli sertifikalar üretilerek bu sertifika otoritesi yardımı ile imzalanır. Bu işlemler için öncelikle uygun bir dizin belirlenmeli ve gerekli sertifikalar oluşturulmalıdır.

```
# mkdir /usr/local/SQL-certs
# cd /usr/local/MySQL-certs
```

Sertifika için gerekli dizinin oluşturulmasının ardından öncelikle sertifika otoritesi için gerekli sertifikalar oluşturulmalı ardından sunucu ve istemci için gerekli sertifikalar oluşturulmalıdır. Sertifika otoritesi için gerekli sertifikayı oluşturmak için aşağıdaki adımlar sırası ile uygulanmalıdır.

```
# openssl genrsa -out ca-key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+
+++++ e is 65537 (0x10001)
# openssl req -new -x509 -nodes -days 1000 -key ca-key.pem -out ca-cert.pem
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) [GB]:TR
State or Province Name (full name) [Berkshire]:Kocaeli
Locality Name (eg, city) [Newbury]:Gebze
Organization Name (eg, company) [My Company Ltd]:Tubitak/Uekae
Organizational Unit Name (eg, section) []:Bilgi Guvenligi
Common Name (eg, your name or your server's hostname) []:Uekae/Bilgi
Guvenligi
Email Address []:bilgi@bilgiguvenligi.gov.tr
```

Ardından sunucu için gerekli sertifikalar oluşturulmalıdır. Bunun için aşağıdaki adımların sırası ile takip edilmesi gerekmektedir.

```
# openssl req -newkey rsa:2048 -days 1000 -nodes -keyout server-key.pem -out
server-req.pem
```

```
Generating a 2048 bit RSA private key
```

```
.....
.....+++
```

```
.....+++
```

```
writing new private key to 'server-key.pem'
```

```
-----
```

```
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) [GB]:TR
State or Province Name (full name) [Berkshire]:Kocaeli
Locality Name (eg, city) [Newbury]:Gebze
Organization Name (eg, company) [My Company Ltd]:Tubitak/Uekae
Organizational Unit Name (eg, section) []:Bilgi Guvenligi
```

```
Common Name (eg, your name or your server's hostname) []:Uekae/Bilgi
Guvenligi
Email Address []:bilgi@bilgiguvenligi.gov.tr

Please enter the following 'extra' attributes to be sent with your
certificate request
A challenge password []:
An optional company name []:
# openssl x509 -req -in server-req.pem -days 1000 -CA ca-cert.pem -CAkey ca-
key.pem -set_serial 01 -out server-cert.pem
Signature ok
subject=/C=TR/ST=Kocaeli/L=Gebze/O=Tubitak/Uekae/OU=Bilgi
Gvenligi/CN=Uekae/Bilgi Guvenligi/emailAddress=bilgi@bilgiguvenligi.gov.tr
Getting CA Private Key
```

Son olarak MySQL istemcisi için gerekli sertifikalar oluşturulmalıdır. Bunun için aşağıdaki adımların sırası ile takip edilmesi gerekmektedir.

```
# openssl req -newkey rsa:2048 -days 1000 -nodes -keyout client-key.pem -out
client-req.pem
Generating a 2048 bit RSA private key
.....+++.....
.....+++ writing new private key to 'client-key.pem'
-----
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) [GB]:TR
State or Province Name (full name) [Berkshire]:Kocaeli
Locality Name (eg, city) [Newbury]:Gebze
Organization Name (eg, company) [My Company Ltd]:Tubitak/Uekae
Organizational Unit Name (eg, section) []:Bilgi Guvenligi
```

```
Common Name (eg, your name or your server's hostname) []:Uekae/Bilgi
Güvenligi
Email Address []:bilgi@bilgiguvenligi.gov.tr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
# openssl x509 -req -in client-req.pem -days 1000 -CA ca-cert.pem -CAkey ca-
key.pem -set_serial 01 -out client-cert.pem
Signature ok
subject=/C=TR/ST=Kocaeli/L=Gebze/O=Tubitak/Uekae/OU=Bilgi
Güvenligi/CN=Uekae/Bilgi Güvenligi/emailAddress=bilgi@bilgiguvenligi.gov.tr
Getting CA Private Key
```

Sertifikaların oluşturulmasının ardından kullanım için MySQL sunucusu üzerinde, MySQL yapılandırma dosyası içerisinde gerekli belirtilmelerinin gerçekleştirilmesi gerekmektedir. Bunun için yapılandırma dosyasında, sunucu taraflı yapılandırma için **[mysqld]** bölümü içerisine aşağıda belirtilen satırların eklenmesi gerekmektedir.

```
[mysqld]
ssl-ca=/usr/local/mysql-certs/cacert.pem
ssl-cert=/usr/local/mysql-certs/server-cert.pem
ssl-key=/usr/local/mysql-certs/server-key.pem
```

MySQL sunucu üzerinde sunucu taraflı gerekli yapılandırmanın gerçekleştirilmesinin ardından, MySQL sunucu servisi yeniden başlatılmalıdır ve ardından MySQL sunucu servisi üzerinde SSL kullanımının aktif olduğu görülebilecektir.

```
[root@mysql mysql-certs]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.45 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SHOW VARIABLES LIKE 'have_openssl';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_openssl  | YES   |
+-----+-----+
1 row in set (0.01 sec)

mysql>
```

Şekil 3-3 Sunucu servisi için SSL kullanımının aktif hale getirilmesi

Ardından MySQL istemcisi üzerinde sertifika kullanımı için gerekli yapılandırma değişikliklerinin gerçekleştirilmesi gerekmektedir. Burada uzak MySQL istemcisi ile komut satırından MySQL sunucusu ile iletişim sağlanıp, sorguların gerçekleştirileceği göz önünde bulundurularak aşağıdaki adımlar sırası ile takip edilmelidir. Kullanılmak istenen MySQL istemcisine göre sertifika kullanımı ile ilgili yapılandırma farklılık göstermektedir. Öncelikle sertifika yetkilisi ve istemcisi için gerekli sertifikalar MySQL istemcisi üzerinde belirlenen uygun bir dizine taşınmalıdır. Burada “/usr/local/mysql-certs” dizini olarak belirlenmiştir. İsteğe göre farklı bir dizin seçilebilir.

```
mysql> SHOW STATUS LIKE 'Ssl_cipher';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Ssl_cipher    |       |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Şekil 3-4 MySQL istemcisinde SSL kullanımı aktif olmadan önce

MySQL istemcisi için SSL sertifika kullanımının durumunu “*SHOW STATUS LIKE 'Ssl_cipher'*” sorgusu çalıştırılarak elde edilebilir. Bu duruma örnek bir kullanım Şekil 3-4’de gösterilmiştir.

MySQL komut satırı istemcisinde SSL kullanımını aktif hale getirmek için yapılandırma dosyasında sertifikaların tam yolları belirtilmelidir.

[client]

```
ssl-ca=/usr/local/mysql-certs/cacert.pem  
ssl-cert=/usr/local/mysql-certs/client-cert.pem  
ssl-key=/usr/local/mysql-certs/client-key.pem
```

İlgili ayarların ardından MySQL istemcisinde “SHOW STATUS LIKE “Ssl_cipher” sorgusu tekrar çalıştırılarak şifreleme için kullanılacak olan anahtar görüntülenebilir.

```
mysql> SHOW STATUS LIKE 'Ssl_cipher';  
+-----+-----+  
| Variable_name | Value  
+-----+-----+  
| Ssl_cipher     | DHE-RSA-AES256-SHA  
+-----+-----+  
1 row in set (0.00 sec)  
mysql>
```

Şekil 3-5 Şifreleme için kullanılacak olan anahtarın durumunun görüntülenmesi

SSL kullanımının aktif hale getirilmesinin ardından MySQL kullanıcılarının SSL kullanımı ve MySQL sunucu servisi ile iletişime geçmelerini zorlama işlemi, GRANT ifadesinde “*REQUIRE SSL*” kullanılarak gerçekleştirilebilir. Bu duruma örnek bir ifade aşağıda yer almaktadır.

```
mysql> GRANT ALL ON deneme.* TO mysqluser@192.168.6.105 IDENTIFIED BY  
'sifre' REQUIRE SSL;
```

Bu şekilde *mysqluser* kullanıcısının *deneme* veritabanına 192.168.6.105 ip adresinden SSL kullanımı aktif olacak şekilde bağlanması zorlanmış olur. Eğer SSL kullanımı aktif olmadan bağlanmaya çalışılırsa aşağıda görülen hata mesajı ile karşılaşılacaktır

```
[root@localhost mysql-certs]# mysql -u mysqluser -psifre -h 192.168.6.114 deneme  
ERROR 1045 (28000): Access denied for user 'mysqluser'@'192.168.6.105' (using password: YES)
```

Şekil 3-6 SSL kullanımına zorlanmış MySQL kullanıcısının, SSL kullanımı olmaksızın bağlantı gerçekleştirilmesi esnasında karşılaşılan hata mesajı

3.7 Kullanıcıları Erişim Güvenliği

MySQL kullanıcıların erişim izinleri MySQL veritabanındaki db tablosuna göre belirlenmektedir. Öncelikle bu tabloya MySQL *root* kullanıcısı haricinde kimsenin erişim izni olmamalıdır.

MySQL erişim kontrolleri *GRANT* ve *REVOKE* deyimleri ile belirlenmektedir. *GRANT* deyimini ile hangi kullanıcının hangi veritabanına hangi ip adresi ya da ip adreslerinden erişim sağlayabileceği belirlenmektedir. MySQL *root* kullanıcısının uzak erişimine izin verilmemeli, sadece yerel erişimlere tam makine adı ya da ip adres bilgisi belirtilerek izin verilmelidir. Örneğin “kullanıcı1” MySQL kullanıcısının 192.168.8.98 ip adresinden veritabanı1 veritabanına “şifre” şifresi ile erişim sağlaması için aşağıdaki SQL sorgusu çalıştırılmalıdır.

```
mysql > GRANT ALL ON veritabanı1.* TO kullanıcı1@192.168.8.98 IDENTIFIED BY 'şifre';
```

Aynı şekilde “kullanıcı2” MySQL kullanıcısının yerel olarak “veritabanı2” isimli veritabanı üzerinde sadece *INSERT*, *SELECT* ve *DELETE* yetkilerine sahip olabilmesi için aşağıdaki sql sorgusu çalıştırılmalıdır.

```
mysql > GRANT SELECT,INSERT,DELETE ON veritabanı2.* TO kullanıcı2@localhost IDENTIFIED BY 'şifre';
```

Yukarıda belirtilen kullanıcıların ilgili veritabanları üzerindeki erişim izinlerinin yapılandırılmasının ardından, *Db* ve *User* tablosundaki ilgili alanları ortak olarak görüntülemek için aşağıda belirtilen SQL sorgusu çalıştırılmalıdır.

```
mysql > select db.Host,db.Db,db.User,user.Password from db,user where db.Host=user.Host and db.User=user.User ;
```

Sorgunun çıktısı Şekil 3-7’de görüldüğü gibi olmaktadır.

```
mysql> select db.Host,db.Db,db.User,user.Password from db,user where db.Host=user.Host and db.User=user.User ;
+-----+-----+-----+-----+
| Host      | Db      | User      | Password      |
+-----+-----+-----+-----+
| 192.168.8.98 | veritabanı1 | kullanıcı1 | 3405e3e35d1c79a9 |
| localhost  | veritabanı2 | kullanıcı2 | 3405e3e35d1c79a9 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Şekil 3-7 Db ve User tablolarından MySQL kullanıcıların görüntülenmesi

Bu çıktı sistem güvenlik politikası doğrultusunda incelerken, gereksiz yetkilere sahip kullanıcıların yetki erişimleri düzenlenmelidir.

3.8 Parola Güvenliği

Hemen hemen her uygulamada olduğu gibi MySQL sunucu servisi için de kullanıcı parolalarının güvenliği çoğu zaman en kritik durum olarak göze batmaktadır. MySQL kullanıcıları MySQL veritabanı içerisinde bulunan *User* tablosunda tutulmaktadır. MySQL kullanıcı şifreleri açık olarak bu tabloda yer almazlar. Şifreler tek yönlü geri döndürülemez bir algoritma ile şifrelenerek tutulmaktadır.

Şifrelenmiş olarak *User* tablosunda tutulan özet (hash) bilgilerinden, deneme yanılma (brute force) yolu ile ya da farklı yöntemler uygulanarak parola bilgisi elde edilebilmektedir. Şekil 3-8’de görüldüğü gibi MySQL *root* kullanıcısının şifrelenmiş parola bilgisi elde edilmiş ve ardından deneme yanılma (brute force) yolu ile parola bilgisi, görüldüğü gibi açıkça görüntülenmiştir.

```
[root@localhost ~]# mysql -u root -psifre
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.0.45 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select User,Password from user;
+-----+-----+
| User | Password |
+-----+-----+
| root | 3405e3e35d1c79a9 |
+-----+-----+
1 row in set (0.00 sec)

mysql> quit
Bye
[root@localhost ~]# ./mysql_brute_force 3405e3e35d1c79a9
Hash: 3405e3e35d1c79a9
Trying length 3
Trying length 4
Trying length 5
Found pass: sifre
[root@localhost ~]#
```

Şekil 3-8 MySQL kullanıcı şifrelerinin hash bilgilerinin elde edilmesi

Belirli aralıklar kullanıcı şifrelerinin hash (özet) bilgileri brute force (deneme yanılma) yolu ile açık olarak görüntülenmeye çalışılmalı, eğer parola bilgileri kolayca elde edilebilecek parolalar varsa bu parola bilgisine sahip kullanıcıların tahmin edilmesi zor parola seçimleri yapmaları sağlanmalıdır.

3.9 MySQL Süper Kullanıcı Güvenliği

MySQL *root* kullanıcısının adı tahmin edilmesi zor, farklı bir isim ile değiştirilmelidir. Bu şekilde MySQL *root* kullanıcısının parola bilgisi deneme yanılma yolu ile tahmin edilmesi zorlaşacaktır. Bu şekilde sadece MySQL *root* kullanıcısının sadece şifresini değil aynı zamanda kullanıcı adını da tahmin etmelidir. MySQL *root* kullanıcısının adını değiştirmek için aşağıdaki adımlar sırası ile uygulanmalıdır.

```
mysql > update user set user="dbadmin" where user="root";
```

MySQL *root* kullanıcısının parola bilgisi ön tanımlı olarak boş gelmektedir. Yani hiçbir parola bilgisi olmaksızın MySQL *root* kullanıcısı ile erişim hakkı elde edilebilmektedir. MySQL *root* kullanıcısının parola bilgisi tahmin edilmesi zor bir parola ile değiştirilmelidir. Bunun için aşağıdaki adımlar uygulanmalıdır.

```
mysql > SET PASSWORD FOR root@localhost=PASSWORD('yenisifre');
```

MySQL *root* kullanıcı hesabının komut satırından değiştirilmesinin doğuracağı güvenlik açıklıkları olabilmektedir. Örneğin yukarıdaki işlem sonrası “*.mysql_history*” dosyasında bu işlemin kayıtları aşağıdaki şekilde görülmektedir.

```
# cat .mysql_history
...
UPDATE\040mysql.user\040SET\040Password=PASSWORD('yenisifre')\040WHERE\040User='root';
#
```

Bu dosyanın erişim izinlerinden kaynaklanan bir durumda MySQL *root* kullanıcısının parolası görülebilmektedir. Ya da parolanın “*mysqladmin*” gibi komut satırı araçları ile değiştirilmesi sırasında, sistem üzerinde oturum açmış bir kullanıcı *ps* komutu yardımı ile şifreyi elde edebilir. Bu duruma örnek bir durum aşağıda gösterilmiştir. *User* kullanıcısının sistemde “*Mysqladmin*” komutunun çalışmasını gözlemlediği sırada, *root* kullanıcısının MySQL *root* şifresi *mysqladmin* komutu yardımı ile değiştirilmektedir.

```
$ while [ 1 ]; do ps aux | grep "mysqladmin" | grep -v "grep"; done
```

MySQL *root* kullanıcılarına yeni şifre atama işleminin gerçekleştirildiği sırada *mysqladmin* komutunu gözlemlene olayının sonucu aşağıdaki şekilde gerçekleşmiştir.

```
...
root      15823  0.0  0.2   7748  1740 pts/0    S+   09:55   0:00 MySQLAdmin
-u root -p password sifre
...
```

Görüldüğü gibi MySQL *root* kullanıcılarına atanan yeni şifre açık olarak görülmektedir.

3.10 Örnek Veritabanı ve Tabloların Güvenliği

Kurulum ile birlikte ön tanımlı olarak gelen örnek veritabanı ve MySQL *root* kullanıcısı haricindeki kullanıcıların silinmesi gerekmektedir. Bunun için aşağıdaki adımların sırası ile uygulanması gerekmektedir. Bu şekilde anonim kullanıcıların da sistemden kaldırılması sağlanmış olacaktır.

```
mysql > drop database test;
mysql > use MySQL;
mysql > delete from user where not (host="localhost" and user="root");
```

3.11 Yama Kontrolü

Herhangi bir uygulama için en önemli güvenlik açığı, uygulama için yayınlanmış güvenlik yamalarıdır. Düzenli olarak kullanılmak istenen uygulama için yayınlanmış güvenlik yamaları takip edilmeli ve uygulanmalıdır. MySQL sunucu servisi içinde bu tehlike Şekil 2-1’de bugüne kadar MySQL sunucu servisi için yayınlanmış güvenlik yamaları ile gösterilmiştir. İlgili güvenlik yamaları <http://www.mysql.com/> adresinden ya da bu konu üzerinde çalışma gösteren sitelerin e-posta listelerinden düzenli olarak takip edilmeli ve uygulanmalıdır.

3.12 DNS Güvenliği

DNS ön bellek zehirlenmesi gibi DNS saldırılarını önlemek için, kullanıcı yetkilendirmesi esnasında kullanılan DNS isimleri yerine ip adres bilgisi kullanılmalıdır. Bu duruma örnek bir kullanım aşağıda gösterilmiştir.

```
mysql> select Host,User from db;
+-----+-----+
| Host          | User          |
+-----+-----+
| 192.168.8.98 | kullanıcı1    |
+-----+-----+
```

Şekil 3-9 DNS isimleri yerine IP adres bilgisinin kullanılması

4. KAYNAKÇA

- [1] <http://dev.mysql.com/doc/>
- [2] <http://www.securityfocus.com/infocus/1726>
- [3] <http://www.securityfocus.com/infocus/1706>
- [4] http://forge.mysql.com/wiki/Security_Vulnerabilities_In_MySQL_Server