



Zaman Damgası

Teknik Destek Grubu ODTÜ BİDB

Ekim 2010



Gündem

- Mevzuat
- Zaman Damgası
- Akış şeması
- OpenSSL ile Log İmzalama
 - OpenSSL Kurulumu
 - Sertifika Oluşturma
 - Logları Damgalama
 - Logların Doğruluğunu Kontrol Etme



Mevzuat

- 5651 Sayılı Kanun ve İlgili Yönetmelikler
 - İnternet Ortamında Yapılan Yayınların Düzenlenmesi ve Bu Yayınlar Yoluyla İşlenen Suçlarla Mücadele Edilmesi Hakkında Kanun (Kanun No: 5651, Resmi Gazete Tarih / Sayı: 23 Mayıs 2007 / 26530)
 - Telekomünikasyon Kurumu Tarafından Erişim Sağlayıcılara ve Yer Sağlayıcılara Faaliyet Belgesi Verilmesine İlişkin Usul ve Esaslar Hakkında Yönetmelik (Resmi Gazete Tarih / Sayı : 24 Ekim 2007 / 26680)



Mevzuat

- İnternet Toplu Kullanım Sağlayıcıları Hakkında Yönetmelik (Resmi Gazete Tarih / Sayı : 1 Kasım 2007 / 26687)
- İnternet Ortamında Yapılan Yayınların Düzenlenmesine Dair Usul ve Esaslar Hakkında Yönetmelik (Resmi Gazete Tarih / Sayı : 30 Kasım 2007 / 26716)



Mevzuat

- Temel Ceza Kanunlarına Uyum Amacıyla Çeşitli Kanunlarda ve Diğer Bazı Kanunlarda Değişiklik Yapılmasına Dair Kanun (Resmi Gazete Tarih / Sayı : 8 Şubat 2008 / 26781, 23 Ocak 2008 tarihinde kabul edilmiş olan bu kanunun 256. maddesi bilişim suçları ile ilişkilendirilmiştir.)



Mevzuat

“Trafik erişim bilgilerin doğruluğu, verilerin dosya bütünlük değerleri zaman damgası ile birlikte saklamalı ve gizliliğini temin edilmelidir.”



Zaman Damgası

- Zaman damgası nedir?
 - Zaman damgası, elektronik ortamda log, doküman ve sözleşme gibi elektronik verilerin, belirli bir zamandan önce var olduğunu kanıtlamak için kullanılır. Mesela bir log dosyasının, kayıt altına alındığı tarihte orjinal haliyle var olduğunu, sonradan değiştirilmediğini ispatlamak amacıyla zaman damgasından yararlanılabilir.

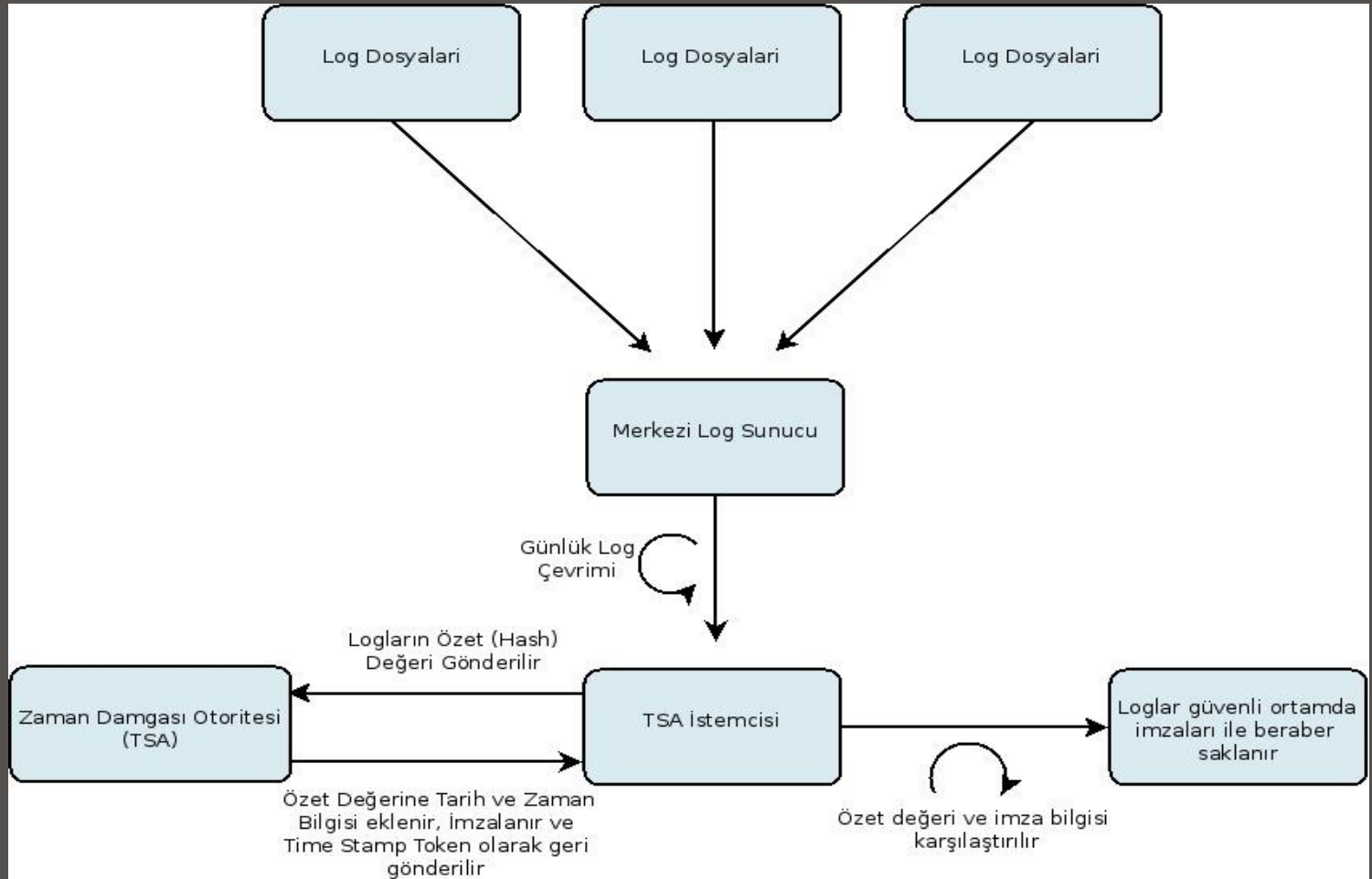


Zaman Damgası

- ODTÜ BİDB TSG sorumluluğunda bulunan sunucuların ürettiği günlük log dosyaları merkezi bir log sunucusunda toplanır ve OpenSSL ile imzalama yöntemi kullanılarak damgalanır.



Akış Şeması





OpenSSL ile Log İmzalama

- OpenSSL Kurulumu

- openssl'in řu anki debian repositorylerinde bulunan kararlı surumu olan 0.9.8f'te TS desteđi olmadığı iin openssl elle derlendi.

/usr/local/ssl altına su komutlar verilerek kuruldu:

```
#wget http://openssl.org/source/openssl-1.0.0a.tar.gz
#tar -zxvf openssl-1.0.0a.tar.gz
#./config
#make
#make install
```



OpenSSL ile Log İmzalama

/usr/local/ssl/openssl.cnf dosyasında aşağıdaki değişiklikler yapıldı:

```
dir                = /certificates          # Where everything
  is kept

# These are used by the TSA reply generation only.

dir                = /certificates          # TSA root
  directory

# This is typical in keyUsage for a client certificate.
keyUsage = nonRepudiation, digitalSignature

# This is required for TSA certificates.
extendedKeyUsage = critical,timeStamping
```



OpenSSL ile Log İmzalama

- Sertifika Oluşturma

- Aşağıdaki komut verilerek sertifika oluşturuldu:

```
#!/usr/local/ssl/bin/openssl req -config \  
/usr/local/ssl/openssl.cnf -days 1825 -x509 -newkey \  
rsa:2048 -out cacert.pem -outform PEM
```



OpenSSL ile Log İmzalama

- Komut verildikten sonra bizden bir takım parametreler isteniyor

```
Password: <cok_gizli>
```

```
Country Name (2 letter code) [AU]:TR
```

```
State or Province Name (full name) [Some-State]:Ankara
```

```
Locality Name (eg, city) []:Ankara
```

```
Organization Name (eg, company) [Internet Widgits Pty  
Ltd]:METU
```

```
Organizational Unit Name (eg, section) []:CC
```

```
Common Name (eg, YOUR name) []:TSG
```

```
Email Address []:tsg-sys@metu.edu.tr
```



OpenSSL ile Log İmzalama

- Sonrasında sertifikaların tutulacağı dizinleri ayarlamak gerekli

Not: /certificates dizinini kendimiz yaratıyoruz. Yeter şart openssl.cnf dosyasındaki ile aynı olmasıdır.

```
#mkdir /certificates
#mkdir /certificates/private
#mkdir /certificates/certs
#mkdir /certificates/newcerts
#mv privkey.pem /certificates/private/cakey.pem
#cp /usr/local/src/openssl-1.0.0a/apps/demoCA/index.txt \
/certificates
#cp /usr/local/src/openssl-1.0.0a/apps/demoCA/serial \
/certificates
```

OpenSSL ile Log İmzalama

- Şimdi de zaman damgası için anahtar oluşturmak gerekiyor

```
#!/usr/local/ssl/bin/openssl genrsa -aes256 -out \
    tsakey.pem 2048
```

```
Password: <cok_gizli>
```

Not: Aslında password'lerin aynı olup olmaması gerektiği tarafımızca meçhul. Biz kolaylık olsun diye hepsini aynı yaptık.

- tsakey'i olması gereken yere taşıyalım

```
#mv tsakey.pem /certificates/private
```



OpenSSL ile Log İmzalama

- Akabinde zaman damgası otoritesinden sertifika isteğinde bulunuyoruz:

```
#!/usr/local/ssl/bin/openssl req -new -key tsakey.pem \  
-out tsareq.csr
```

```
Country Name (2 letter code) [AU]:TR
```

```
State or Province Name (full name) [Some-State]:Ankara
```

```
Locality Name (eg, city) []:Ankara
```

```
Organization Name (eg, company) [Internet Widgits Pty  
Ltd]:METU
```

```
Organizational Unit Name (eg, section) []:CC
```

```
Common Name (eg, YOUR name) []:TSG
```

```
Email Address []:tsg-sys@metu.edu.tr
```



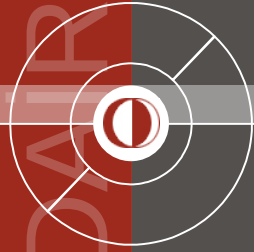
OpenSSL ile Log İmzalama

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []: <ENTER>

An optional company name []: <ENTER>

Not: Burada istenen 'extra' niteliklerin girilmesi zorunlu değil. Biz boş
geçtik.



OpenSSL ile Log İmzalama

- CSR dosyası /certificates dizini altında oluşturuldu. Şimdi bu dosyayı kullanarak TSA (time stamp authority) public key'i oluşturulacak

```
#!/usr/local/ssl/bin/openssl ca -config /usr/local/ssl/openssl.cnf -in tsareq.csr -out tsacert.pem
```



OpenSSL ile Log İmzalama

- Logları damgalama

- Artık zaman damgasi basabilir durumdayız. Bu aşamada loglarımızı zıpleyip damgalayıp 6 ay (mevzuata göre 1 yıl) bir yerlerde saklamamız gerekiyor. Bu iş için biz bir betik yazdık. Kısaca şu işleri yapıyor.

1. Gece yarısından sonra bir önceki günün loglarını `log.%Y%m%d.gz` formatında zıple

2. Her ziplenmiş loga şunları yap:

- a. query oluştur

```
openssl ts -query -data <log_dosyasi> -no_nonce -out  
<log_dosyasi>.tsq
```



OpenSSL ile Log İmzalama

b. reply oluştur

```
openssl ts -reply -queryfile <log_dosyasi>.tsq -out \  
<log_dosyasi>.tsr -token_out -config \  
/usr/local/ssl/openssl.cnf -passin pass:"<cok_gizli>"
```

c. bu aşamadan sonra *.tsq dosyasının işlevi kalmıyor gibi, silinebilir

3. Zipli dosyaları bir yere taşı
4. *.tsr dosyalarını başka yere taşı
5. 6 aylık logları ve *.tsr dosyalarını sil

OpenSSL ile Log İmzalama

- Logların doğruluğunu kontrol etme
 - Saklanan log dosyasının damgalandığı günden beri değiştirilmediğini şu komutu vererek anlayabiliriz:

```
#!/usr/local/ssl/bin/openssl ts -verify -data \  
  <log_dosyasi> -in <log_dosyasi>.tsr -token_in \  
  -CAfile /certificates/cacert.pem -untrusted \  
  /certificates/tsacert.pem
```

- Çıktı şu şekilde olacaktır:

```
Verification: OK
```



Özet

- Sunucularımızdan gelen log dosyalarını merkezi log sunucumuz üzerinde zipledik.
- Bu log dosyalarının Özet (Hash) Değerlerini Zaman Damgası Otoritesine (TSA) gönderdik.
- Özet Değerlerine tarih ve zaman bilgisi eklenip, imzalanıp, Time Stamp Token olarak geri gönderildi.



Özet

- TSA işlemcisine güvenli bir ortamda imzaları ile saklanan log dosyalarının özet değeri ve imza bilgisini karşılaştırdık.
- Detaylı bilgi için aşağıdaki linklere göz atabilirsiniz:
 - www.metu.edu.tr/5651
 - www.opentsa.org/
 - www.openssl.org/docs/apps/ts.html



TEŞEKKÜRLER

Sorular?