

SSL Kullanımı ve Sayısal sertifika, imza işleri

1. SSL Giriş
2. Şifreleme algoritmaları
3. Sayısal imza kavramı
4. Sertifikalar
5. Sertifika Otoritesi
6. Özgür bir SSL sürümü: OpenSSL
7. Temel OpenSSL Kullanımı
8. Kendi Sertifika Otorite(CA)mızı oluşturmak
9. Yeni Bir sertifika isteği oluşturmak
10. CA Tarafından sertifika imzalama isteğinin gerçekleştirilmesi
11. Kendinden imzalı(Self-signed) sertifika oluşturmak
12. Gizli anahtardan(Private Key)'den açık anahtarı(Public key) elde etmek oluşturmak
13. HTTPS Nasıl Çalışır?
14. Apache için HTTPS Ayarları
15. Neden name-based sanal host tanımlarında SSL kullanırken her host başına bir IP gerekiyor?

Huzeyfe ÖNAL

huzeyfe@enderUNIX.org

<http://www.enderunix.org/huzeyfe>

SSL Giriş

Temelleri Netscape firması tarafından 1994 yılında atılan SSL aynı yılda ticari olarak piyasaya sürüldü ve bir sonraki yıl IETF tarafından standart olarak Kabul edildi. Aslında standartın asıl ismi TLS olmasına rağmen genellikle SSL kullanımı tercih edilmektedir. Yani TLS 1 ile SSL 3.1 aynı özelliklere sahiptir. İlk zamanlar sadece HTTP trafiğini şifreleme amaçlı geliştirilmiş olsa da günümüzde TCP tabanlı tüm servisleri şifreleme amaçlı kullanılabilir.

Şifreleme algoritmaları

Simetrik ve asimetrik olmak üzere iki çeşittir. Simetrik anahtarlı şifreleme algoritmalarında bir veriyi şifrelemek ve şifreli veriden orjinal veriyi elde etmek için aynı anahtar kullanılır. Burada bir anahtar değişim problemi vardır, ortak anahtar nasıl iki tarafa ulaştırılacak?

Asimetrik (Açık anahtarlı şifreleme) Algoritmalar

Bu tip algoritmalarda veriyi şifrelemek ve çözmek için iki anahtar kullanılır. Veriyi şifrelemek için bir anahtar(public key), çözmek için diğer anahtar(private) kullanılır.

Benim public anahtarım kullanılarak şifrelenmiş bir veri ancak benim private anahtarım kullanılarak çözülebilir. Bu yüzden asimetrik algoritmalarda açık anahtar(public key) dağıtılır, gizli anahtar(private key) saklanır.

Bir de bunların haricinde mesaj özeti(message digest) olarak adlandırılan ve veri bütünlüğünü koruma amaçlı kullanılan tek yönlü çalışan hash algoritmaları vardır, md5 , sha gibi.

Sayısal İmza Kavramı

Verinin içeriğinin değiştirilme durumu ve gönderenin gerçekliğini ispat söz konusu ise sayısal imza kullanılır. Sayısal imza kısaca yazılan mesajın özetinin gizli anahtar ile şifrelenmesi ve bir sıra numarası eklenmiş halidir.

Gönderici;

- 1) Mesajı gönderecek mesajın özetini alır
- 2) gizli anahtarını kullanarak özeti şifreler ve bir sıra numarası ekler.

Alıcı;

- 1) Alıcı göndericinin açık anahtarını kullanarak şifreyi çözer
- 2) Şifre çözüldükten sonra ortaya gönderilen mesajın özeti çıkar
- 3) Aynı algoritma kullanılarak mesaj özet işlemine tabi tutulur ve doğruluğu control edilir.

Sertifikalar

Sayısal imzaları kullanarak bir verinin gerçekten beklenen kişi tarafından gönderildiği ve iletim esnasında değişikliğe uğramadığını anlayabiliriz peki gönderilen verinin gerçekten beklediğimiz insana ulaştığından nasıl emin olabiliriz? Yani açık anahtarını kullanarak verileri şifrelediğimiz kişi gerçekte düşündüğümü kişi midir? Nasıl emin olabiliriz.

Burada sertifika tanımı ortaya çıkıyor. Bir sertifika basitce kişinin açık anahtarının yetkili bir sertifika otoritesi tarafından imzalanmış halidir diyebiliriz.

Sertifika Otoritesi

Sertifika isteğinde bulunan şahıs/kurumların gerçekte belirttikleri kişiler/kurumlar olduklarını(bunun yanında belirtilen diğer hususları da) doğrulayan ve onaylayan kurumdur. Verisign, Globalsign gibi..

Eğer her iki tarafta ortak güvenilen bir sertifika otoritesi tarafından imzalanmış sertifika kullanıyorsa birbirlerinin public keylerine güvenebilirler.

Özgür bir SSL sürümü: OpenSSL

Temel OpenSSL Kullanımı

OpenSSL dosyalarının hangi dizinde bulunduğunu öğrenmek için;

```
# openssl version -d
```

```
OPENSSLDIR: "/usr/share/ssl"
```

Komutu kullanılabilir. Bu dosyaları daha akılda kalıcı bir dizinden yönetmek istersek aşağıdaki komut işlemi görecektir. Bu komut sonrasında openssl ile ilgili dosyalar /etc/ssl dizini altından da erişilebilir olacaktır.

```
#ln -s /usr/share/ssl/ /etc/ssl
```

Hangi OpenSSL sürümü ile çalıştığınızı öğrenmek için ,

```
# openssl version
```

OpenSSL 0.9.7a Feb 19 2003

Komutunu kullanabilirsiniz. Detaylı bir çıktı almak istenirse **openssl version -a** komutu da kullanılabilir.

OpenSSL ile yardım alma.

OpenSSL kullanırken parametrelerin neler olduğunu ve bunların detaylarını öğrenmek için **-h** parametresini kullanılabilir. Mesela genel openssl kullanımı için

```
#openssl -h
openssl:Error: '-h' is an invalid command.

Standard commands
asn1parse  ca      ciphers  crl      crl2pkcs7
dgst      dh      dhparam  dsa      dsaparam
enc       engine  errstr   gendh    gensa
genrsa    nseq    ocspl    passwd   pkcs12
pkcs7     pkcs8   rand     req      rsa
rsautl    s_client s_server s_time   sess_id
smime     speed   spkac    verify   version
x509

Message Digest commands (see the `dgst' command for more details)
md2      md4      md5      rmd160   sha
sha1

Cipher commands (see the `enc' command for more details)
aes-128-cbc aes-128-ecb aes-192-cbc aes-192-ecb aes-256-cbc
aes-256-ecb base64      bf          bf-cbc      bf-cfb
bf-ecb      bf-ofb     cast        cast-cbc    cast5-cbc
cast5-cfb   cast5-ecb  cast5-ofb   des          des-cbc
des-cfb     des-ecb    des-ede     des-ede-cbc des-ede-cfb
des-ede-ofb des-ede3   des-ede3-cbc des-ede3-cfb des-ede3-ofb
des-ofb     des3       desx        rc2          rc2-40-cbc
rc2-64-cbc rc2-cbc    rc2-cfb     rc2-ecb     rc2-ofb
rc4         rc4-40
```

Standart komutlardan biri ile ilgili yardıma ihtiyaç duyarsak `openssl komut_adi -h` ile detay bilgi edinebiliriz.

openssl gendsa -h

```
usage: gendsa [args] dsaparam-file
-out file - output the key to 'file'
-des      - encrypt the generated key with DES in cbc mode
-des3     - encrypt the generated key with DES in ede cbc mode (168 bit key)
-aes128, -aes192, -aes256
            encrypt PEM output with cbc aes
-engine e - use engine e, possibly a hardware device.
-rand file:file:...
            - load the file (or the files in the directory) into
              the random number generator
dsaparam-file
a DSA parameter file as generated by the dsaparam command
```

Kendi Sertifika Otorite(CA)mızı oluşturmak

Kendi sertifika otoritemizi oluşturmak ancak bize bağlı çalışan istemciler/sunucular arası gizliliği sağlayabilir. Bunun için gerekmedikçe kendi CA yapımızı kullanmaktan kaçınmalıyız. Tekrar hatırlatmakta fayda var: bir CA'nin görevi güven ilişkisini sağlamaktır.

Kendi yönettiğimiz CA'yi oluşturmak için sistemde bir dizi komut çalıştırmamız gerekir.

```
#cd /etc/ssl
#touch index.txt
#echo '01' >serial
```

```
# openssl req -new -x509 -extensions v3_ca -keyout private/cakey.pem -out
cacert.pem -days 3650 -config ./openssl.cnf
```

Generating a 1024 bit RSA private key

.....++++++

.....++++++

writing new private key to 'private/cakey.pem'

Enter PEM pass phrase:**test123**

Verifying - Enter PEM pass phrase:**test123**

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]:**Istanbul**
Locality Name (eg, city) [Newbury]:**Uskudar**
Organization Name (eg, company) [My Company Ltd]:**Snort A.S**
Organizational Unit Name (eg, section) []:**IT**
Common Name (eg, your name or your server's hostname) []:**snort-home**
Email Address []:**huzeyfe@enderunix.org**

-new -x509 yeni bir sertifika oluştur
-extensions v3_ca CA sertifikası oluşturmak için
-days 3650 3650 sertifika geçerlilik süresi
-keyout, -out çıktıları göstermek için
-config ./openssl.cnf belirtilen konfig dosyasını kullanacağını belirtir.

Bu komut sonrasında bulunduğumuz dizinde(/etc/ssl) **cacert.pem** ve private dizininde **cakey.pem** adlı iki dosya oluşacaktır. Buradaki cakey.pem dosyası saklanması gereken dosyadır bize(CA olduk artık)herhangi bir sertifika imzalama isteği geldiğinde bu dosyayı kullanarak imzalama yapacağız.

NOT:CA sertifikasını sıkı bir parola ile korumak gerekir ve her imzalama isteği geldiğinde bu parola ile yapılmalıdır.. Zira bu sertifikayı ele geçiren biri bizim adımıza sertifika imzalayabilir.

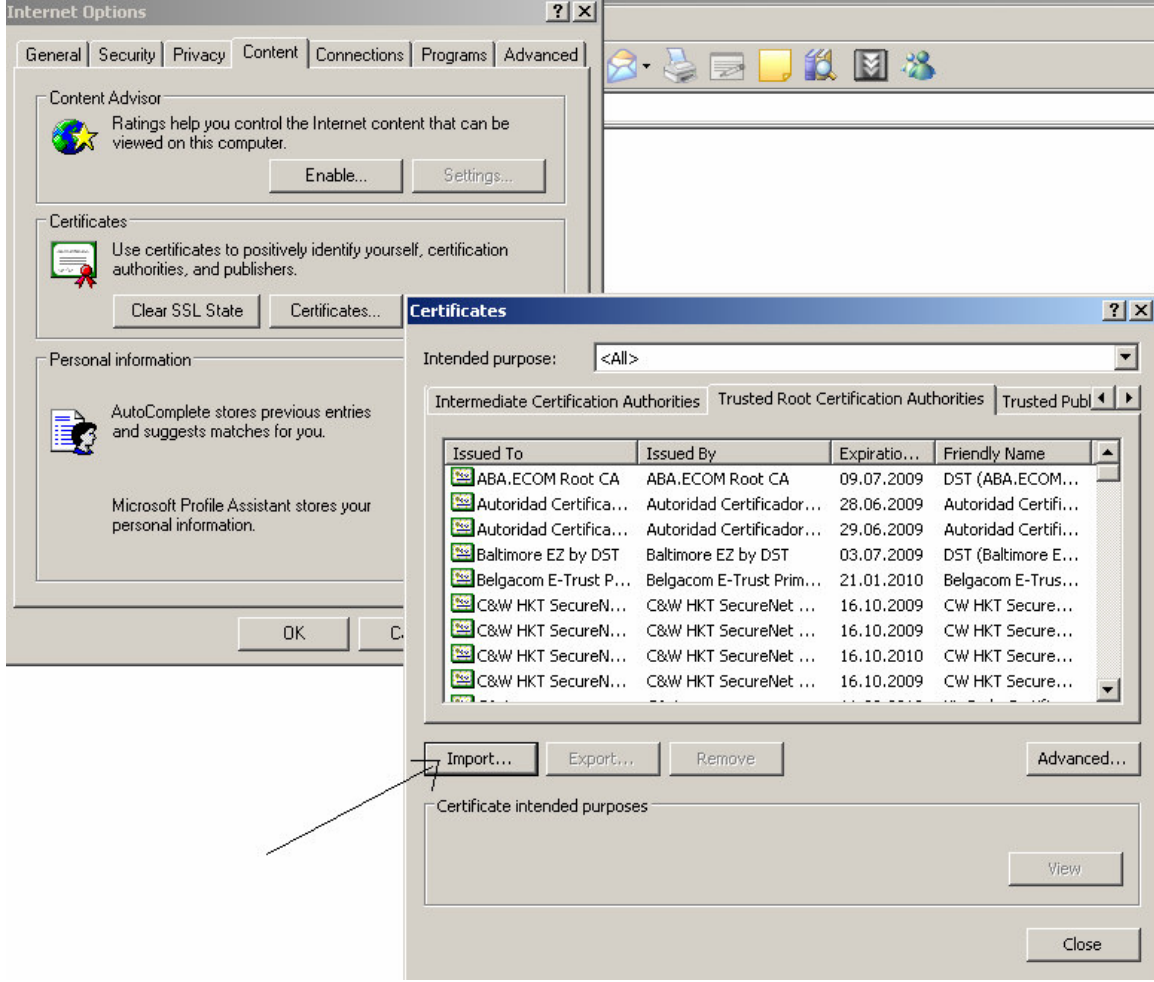
Diğer dosya ise bize güvenecek istemcilerde bulunması gereken dosyadır. Bu dosyayı cacert.crt olarak yeniden adlandırılın

#mv cacert.pem cacert.crt

Sonra bu dosyayı istemcilerimize yüklemek için onların ulaşabileceği bir yere koymamız gerekir. İstemcilerimiz bu dosyayı edinip kullandıkları browserlara'a(ya da ssl istemcisi programlara) import etmeleri gerekir.

Internet Explorer için:

"Tools > Internet Options > Content > Certificates > Trusted Root Certificate Authorities"



Buradan ekleme yerine cacert.crt dosyasının üzerine çift tıkladığımızda size sertifikayı yüklemek istediğinize dair bir soru soracaktır buradan da ekleyebilirsiniz.

NOT: Bir root sertifikanın geçerlilik süresi dolduğunda bu sertifika ile imzalanmış tüm sertifikalarda geçersiz olacağı için root sertifikaların geçerlilik süreleri olabildiğince uzun verilmelidir.

Yeni Bir sertifika isteği oluşturmak(*certificate-signing request (CSR)*.)

Öncelikle bir private key oluşturup bu private keyi kullanarak CSR oluşturacağız. CSR oluşturmanın farklı yolları da var fakat burada bir tanesini kullanıyor olacağız.

Gizli anahtar(Private key) oluşturma;

#openssl genrsa -out sunucu.key 1024

#openssl req -new -key sunucu.key -out sunucu.csr

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]:.

Locality Name (eg, city) [Newbury]:.

Organization Name (eg, company) [My Company Ltd]:.

Organizational Unit Name (eg, section) []:IT

Common Name (eg, your name or your server's hostname) []:**snort-home**

Email Address []:**huzeyfe@enderunix.org**

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

A challenge password []:

An optional company name []:

NOT: sorulan sorularda boş bırakmak istediğiniz alanlar için .(nokta) kullanınız, Enter'a basarak geçtiğiniz alanlar default değerler ile doldurulacaktır.

NOT: Sunucu sertifikası oluştururken parola kullanırsak sertifikayı kullandığımız servisi her yeniden başlatmımızda parolayı girmemiz gerekecektir.

Oluşan .csr dosyasının içeriği;

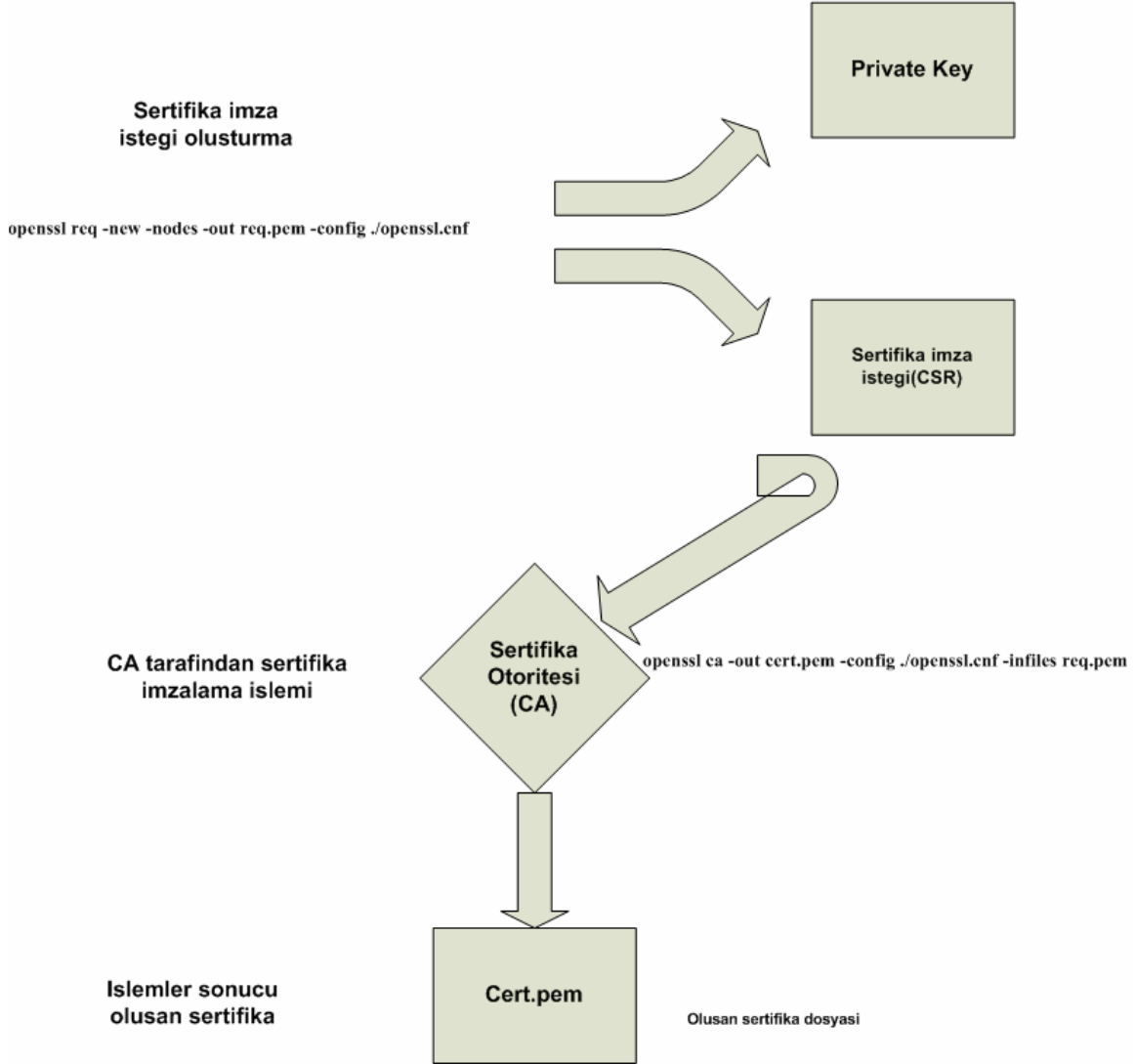
cat sunucu.csr

-----BEGIN CERTIFICATE REQUEST-----

```
MIIB1TCCAT4CAQAwgZQxCzAJBgNVBAYTAiRSMREwDwYDVQQIEwhJU3RhbWJ1bDEQMA4GA1UEBxMHVXNrdWRhcjESMBAGA1UEChMJRW5kZXJVTklYMREwDwYDVQQLEwhTZWN1cm10eTETMBEGA1UEAxMKc25vcnQtaG9tZTEkMCIGCSqGSIb3DQEJARYVaHV6ZXlmlmZlbnRlcnVuaXgub3JnMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC7zfpHyTPgT3jeVwafDkbohxEHiaWksDdWGfxCDrSHT1IqqePLylqHBRzf6anU3qoX8chJQ7gKS8nwEWAKONRpoycCUkFIolq67K9cgVzpQrIrBC0An0O9zcjht9u6ivDUMulARPyA0gQw2423+Dhw776Rg1Iaz2uFjUB1I+gN6wIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEASnVKLebYGBOfvo85jkD1dZo/BwrSbQ3d1dBtmmjWK5n4M1gf3FM1lkWREcFL/AshdK7bM68QJgXO4F0uKbzRmrABZmM4fhCle0feUrqh8AGLuoVfqFwcA2D1mdf4HXkVy5Hw4lqctFUnB6+3e8ktHEeCrnmMQmKoVV6+oZrmZk=
```

-----END CERTIFICATE REQUEST-----

Sertifika istegi olusturma ve Sertifika imzalama



CSR oluşturduktan sonra `sunucu.csr` dosyasını ya bilinen bir CA'ya gönderip imzalamalarını isteyeceğiz –ki bu iş oldukça prosedürel bir iştir, şirketinizin varlığını ispat edip gerekli tüm belgeleri CA'ya ya da onun için bu işleri yapan firmaya teslim etmek zorundasınız- ya da kendi oluşturduğumuz sertifika otoritesi (CA) ile imzalayacağız.

CA Tarafından sertifika imzalama isteğinin gerçekleştirilmesi

```
# openssl ca -out sunucu.pem -config /etc/ssl/openssl.cnf -infile sunucu.csr
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./private/akey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: Nov 13 07:51:41 2006 GMT
    Not After : Nov 13 07:51:41 2007 GMT
  Subject:
    countryName           = TR
    organizationalUnitName = IT
    commonName            = snort-home
    emailAddress          = huzeyfe@enderunix.org
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
      4C:D4:D7:96:B4:FD:AA:F9:10:B4:F5:84:65:6D:07:A5:25:54:D0:4D
    X509v3 Authority Key Identifier:
      keyid:D4:D3:3F:A5:5C:5C:58:0E:8E:A1:09:44:F8:C6:6F:47:61:F4:A4:E1
      DirName:/C=TR/ST=Istanbul/L=Uskudar/O=Snort A.S/OU=IT/CN=snort-
      home/emailAddress=huzeyfe@enderunix.org
      serial:00

Certificate is to be certified until Nov 13 07:51:41 2007 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

CSR: public anahtarı ve ismimizi barındıran dosya. Tek başına bir işe yaramaz. CSR, CA tarafından imzalandıktan sonra sayısal sertifika olur .

Bir sertifika da iki şey önemlidir. Biri sizing açık anahtarınız, diğeri de CA imzası. Bir sertifika, public anahtarımız, ismimiz ve CA imzasını taşır.

Kendinden imzalı(Self-signed) sertifika oluşturmak

Kullanacağımız sertifika sadece kendimiz kullanacaksak bir CA tarafından imzalanmasına gerek yoktur. Kendinden imzalı bir sertifika oluşturmak için.

```
# openssl genrsa -out server.key 1024
```

Generating RSA private key, 1024 bit long modulus

```
.....++++++
.....++++++
e is 65537 (0x10001)
```

Komutu kořturulur. Bundan sonraki adımı x509 tabanlı kendinden imzalı sertifikamızı oluşturmak. Sertifika oluşturmada bir önceki adımda oluşturduğumuz server.key'i kullanacağız.

```
# openssl req -new -x509 -nodes -sha1 -days 365 -key server.key -out server.crt
```

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]:**Istanbul**

Locality Name (eg, city) [Newbury]:**Uskudar**

Organization Name (eg, company) [My Company Ltd]:**EnderUNIX**

Organizational Unit Name (eg, section) []:**Security**

Common Name (eg, your name or your server's hostname) []:**snort-home**

Email Address []:**huzeyfe@enderunix.org**

Oluřturulan sertifikanın içeriğine bakmak isterseniz ařağıdaki komutu çalıştırmanız yeterlidir.

```
# openssl x509 -noout -text -in server.crt
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 0 (0x0)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=TR, ST=Istanbul, L=Uskudar, O=EnderUNIX, OU=Security, CN=snort-home/emailAddress=huzeyfe@enderunix.org

Validity

Not Before: Nov 13 07:34:27 2006 GMT

Not After : Nov 13 07:34:27 2007 GMT

Subject: C=TR, ST=Istanbul, L=Uskudar, O=EnderUNIX, OU=Security, CN=snort-home/emailAddress=huzeyfe@enderunix.org

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

```
00:bb:cd:fa:47:c9:33:e0:4f:78:de:57:06:9f:0e:
46:e8:87:11:07:89:a5:a4:b0:37:56:19:fc:42:0e:
b4:87:4f:52:2a:a9:e3:cb:ca:5a:87:05:1c:df:e9:
a9:d4:de:aa:17:f1:c8:49:43:b8:0a:4b:c9:f0:11:
60:0a:38:d4:69:a3:27:02:52:41:48:a2:5a:ba:ec:
af:5c:81:5c:e9:42:b2:2b:04:2d:00:9f:43:bd:cd:
c8:e1:b7:db:ba:8a:f0:d4:9a:e9:40:44:fc:80:d2:
04:30:db:8d:b7:f8:38:70:ef:be:91:83:52:1a:cf:
6b:85:8d:40:75:23:e8:0d:eb
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
98:79:F1:46:32:98:2D:D3:15:E9:88:29:3B:79:3D:55:9D:D8:C7:9E
X509v3 Authority Key Identifier:
keyid:98:79:F1:46:32:98:2D:D3:15:E9:88:29:3B:79:3D:55:9D:D8:C7:9E
DirName:/C=TR/ST=Istanbul/L=Uskudar/O=EnderUNIX/OU=Security/CN=snort-
home/emailAddress=huzeyfe@enderunix.org
serial:00

X509v3 Basic Constraints:
CA:TRUE
Signature Algorithm: sha1WithRSAEncryption
22:a6:34:c0:da:cf:fe:90:64:21:a0:75:43:0c:4d:66:da:a1:
fe:df:a0:d6:8a:71:6f:3a:5f:fa:65:79:2b:43:bc:a9:bf:af:
5c:fc:65:54:50:fe:7e:2b:61:7e:14:26:2e:86:c4:af:b5:81:
a5:0f:7a:9c:c8:11:15:43:10:bb:1d:07:da:48:3d:ab:3c:b7:
61:a8:08:1d:0f:f6:66:fb:1a:bc:db:67:11:42:88:eb:9f:d2:
30:df:87:19:dc:7f:24:4c:4f:c9:ae:89:0b:b4:17:0a:30:b3:
3d:08:66:9d:c6:55:75:79:50:da:ed:56:45:83:df:f3:9b:47:
6a:85
```

Private Key'den Public keyi oluşturmak

Bazı durumlarda gizli anahtardan açık anahtar oluşturmamız gerekebilir. Bunun için aşağıdaki komutu çalıştırmak yetecektir. Buradan da gizli anahtar ve açık anahtar arasındaki ilişki anlaşılabilir.

```
#openssl genrsa -out sunucu.key 1024
Generating RSA private key, 1024 bit long modulus
.++++++
.....++++++
e is 65537 (0x10001)
```

Gizli anahtar oluşturmak

cat sunucu.key

-----BEGIN RSA PRIVATE KEY-----

```
MIICXwIBAAKBgQC3pyDVapzwTVeXjyyThOMSj2MQ/2Kv2zhaoBFJEHVn8gNcX4o6
D56r8LBRQa0oHmf+xpkrUyanBf1McrE8Lk7+cuGfqAhDoMzmNKX/jqrBsaz8dl3+
YSgVXNpuqq3fMv/i7LOKXsMxEyDCzUYpntzyJv1Zn4r9BhjJzd5p0+qQIDAQAB
AoGBAI0TNhaahmZ3+pNkjlzWbAWzMvW0IEyPfTnr+AhB+n+erwnTcCnqmHx3lz
0PbbfJlqqLsx4KQ25p/eDqWjEQv5d7HLjtb7kv3ObiQ4QOqFu4ai9PhH5LC1hCtm
AeR5o4ZCirRACSThot483LwnE05krOUmgzFDAM9wSJtxhbLNAkEA4kvABSrROMsq
GGRPIShxSADiVwLP0VwomfQKRe9ZeMIMEdk/gWAZuSYthT501SMRz4nzYFWqb6e2
yFR/tHBY3wJBAM/CcC3owi8/MQuqA/LGjwkE7ADL8CM0PjF934Ssc/h5rogZOA0
v3TCXrD9gvYSriF574g81jcoZ8zkPQKG8XcCQQCnRWzpzNmWTGSaUROOXD+W5Khp
vrdUvvV4Dy7E0Q5m7DqRJavkBSTikfdLPQRU/HnYcYXcFsiW2s9m1AmVWErAkEA
iDG8fDwDYBmauzy+zd3gUldJptQKHenXg3YymrS6aX7LLSjrFJAEGjR5AqmNumZQ
hF186uCGxS3VrfIJdWkMzQJBAMMlmeuCKt3eaiym+4lsd2UQRmSIJU69didTOyCg
5zlul1km+UfquxkP1659tjzPjcsqXXTjvVgExzE/3144T3c=
```

-----END RSA PRIVATE KEY-----

Gizli anahtarın içeriği

openssl rsa -in sunucu.key -pubout

writing RSA key

-----BEGIN PUBLIC KEY-----

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC3pyDVapzwTVeXjyyThOMSj2MQ
/2Kv2zhaoBFJEHVn8gNcX4o6D56r8LBRQa0oHmf+xpkrUyanBf1McrE8Lk7+cuGf
qAhDoMzmNKX/jqrBsaz8dl3+YSgVXNpuqq3fMv/i7LOKXsMxEyDCzUYpntzyJv
1Zn4r9BhjJzd5p0+qQIDAQAB
```

-----END PUBLIC KEY-----

Gizli anahtardan oluşturulan açık anahtar

Digest(Özet) şifreleme yöntemleri.

Bir dosyanın parmak izini almak

MD5

openssl dgst -md5 dosya_ismi

SHA1

openssl dgst -sha1 dosya_ismi

openssl dgst -md5 cacert.pem

MD5(cacert.pem)= 1aed568811bde9231305aa0ee970be0

Base64 encode etme

#echo -n "deneme yazisi" | openssl enc -base64

ZGVuZW11IHlhemlzaQ==

```
#echo "ZGVuZW11IHlhemlzaQ==" | openssl enc -base64 -d  
deneme yazisi
```

herhangi bir dosyanın içeriğini base64 ile kodlamak isterseniz `-in dosya_ismi` parametresini eklemeniz yeterlidir.

/etc/shadow tipi parola hash'i oluşturmak

Linux sistemlerde `/etc/shadow` dosyası system kullanıcılarının parolalarının şifreli bir şekilde saklandığı dosyadır. Bu dosyaya bakacak olursak

```
# grep huzeyfe /etc/shadow
```

```
huzeyfe:$1$47JagHAu$6HvvYMo4maDHziTGdB6WT/:13465:0:99999:7::
```

kırmızı ve yeşilli alan huzeyfe kullanıcısının parolasının şifreli halidir. Burada ikinci ve üçüncü \$ işaretleri arasındaki ifade salt olarak olarak adlandırılır. Burada gizli olan parolayı bulmanın bir yolu yoktur zaten Linux sistemlerde kullanıcı adı-parola kontrolü yaparken kullanıcının girdiği değeri alıp salt ile işleme soktukten sonra çıktığı bu dosyadaki ilgili satır ile karşılaştırıyor ve sonucu olumlu ya da olumsuz olarak dönüyor.

OpenSSL kullanarak benzer şifreleli hash çıktıları oluşturabiliriz.

```
# openssl passwd -1 -salt 47JagHAu test123  
$1$47JagHAu$6HvvYMo4maDHziTGdB6WT/
```

HTTPS Nasıl Çalışır?

Testdomain.com isimli bir alan adımız olsun ve biz www.testdomain.com için yetkili bir sertifika otoritesinden(Verisign, Globalsign vs) sertifikamızı almış olalım.

Istemci , ssl siteye bağlanan tarafı belirtir.

Sunucu, ssl çalışan sistemi gösterir olsun, bu ikisi arasındaki güvenli iletişim aşağıdaki adımlardan oluşur.

Istemci sunucuya, *ClientHello* olarak adlandırılan ve istemcinin desteklediği şifreleme algoritmaları, sıkıştırma algoritmaları gibi özellikleri belirten bir mesaj gönderir.

Sunucu bu mesajı alır ve istemciye uygun seçtiği algoritmaları vs bildiren bir mesaj gönderir. Bu mesaj *ServerHello* olarak geçer.

https testi

```
$ openssl s_client -connect localhost:443 -state -debug  
GET / HTTP/1.0
```

Apache için HTTPS Ayarları

Apache de SSL kullanmak istiyor ve yetkili CA(certificate authority)lara para vermek istemiyorsanız kendi sertifikanızı kendiniz oluşturup->imzalayıp kullanabilirsiniz.

Sonuç olarak web trafiğinizi şifrelemiş olursunuz fakat oluşturduğunuz sertifika Browserlar(Ms Explorer, Mozilla Firefox vs) tarafından tanınmayacağı için siteye her girişte kullanıcıyı uyaran bir yazı çıkacaktır, kullanıcı bu uyarıyı "yes" olarak geçerse şifreli alana adım atmış olur, "no" derse giriş yapamaz "view" diyerek de sertifika hakkında daha detaylı bilgi alabilir.

tmp dizinine geçerek sertifikamızı oluşturalım.

```
#cd /tmp
```

```
#openssl genrsa -out hostname.key 1024
```

```
#openssl req -new -key hostname.key -out hostname.csr
```

bu 2 komut sonrası oluşan sertifikanızı herhangi bir yetkili CA ya yollayıp imzalamalarını isteyebilirsiniz(parali) ya da aşağıdaki komutla kendiniz imzalayabilirsiniz.

```
#openssl x509 -req -days 710 -in hostname.csr -signkey hostname.key -out  
hostname.crt
```

oluşan hostname.crt ve hostname.key dosyalarını apache konfigürasyon dizinine taşıyarak, httpd.conf ta bunların yerini belirtiniz.

httpd.conf taki tanımlar aşağıdakine benzerl olmalı

```
SSLCertificateFile /etc/httpd/conf/hostname.crt
```

```
SSLCertificateKeyFile /etc/httpd/conf/hostname.key
```

***Tüm bunları kullanabilmek için sisteminizde OpenSSL paketi olmalıdır ve apache mod_ssl desteği ile derlenmiş olmalıdır.

*** red hat sistemleti için ssl ayarları

/etc/httpd/conf.d/ssl.conf tur, buradan gerekli düzenlemeleri yapabilirsiniz.

Neden name-based sanal host tanımlarında SSL kullanırken her host başına bir IP gerekiyor?

isim tabanlı sanal host kavramı (name-based virtual hosting) bir IP üzerinden birden fazla hostu sunma için kullanılan bilindik bir yöntem. Fakat bu yöntem eğer host edilen sunucularda SSL kullanılacaksa ise yaramıyor. Neden mi?

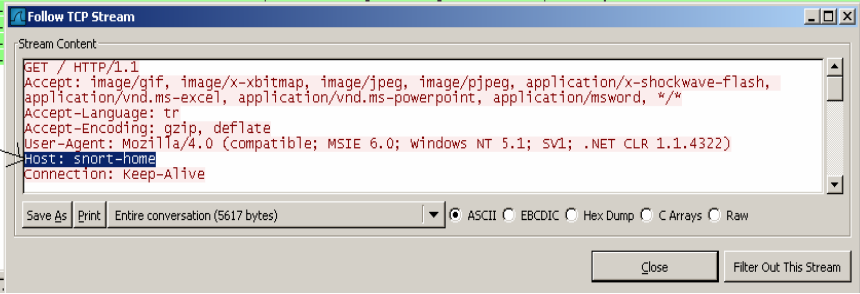
Bir IP üzerinden birden fazla host sunma Http isteklerinde taşınan Host başlığına bağlıdır.

Bir http isteği oluşturulduğunda host başlığı hedef IP'deki hangi hostu istediğini belirtir. Bizim web sunucumuza gelen istek apache(ya da başka bir web sunucu) tarafından yorumlanarak uygun site kullanıcıya gösterilir.

SSL destekli bir web sunucuda bir hosta gelen HTTPS isteğinde Host headeri şifreli olarak geleceği için(OSI katmanında SSL HTTP'den önce, yani http istekleri şifrelenmiş olduğu için) web sunucu Host alanını okuyamaz ve o alan adına özel işlemler gerçekleştiremez, dolayısı ile istenilen siteyi gösteremez, bu yüzden virtual host ile birlikte SSL kullanılmaz , bunun yerine Ip tabanlı sanal host kullanılmalıdır.

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.206.1	192.168.206.128	TCP	1159 > http [SYN] Seq=0 Len=0 MSS=1460 WS=0
2	0.000235	192.168.206.128	192.168.206.1	TCP	http > 1159 [SYN, ACK] Seq=0 Ack=1 Win=23360 Len=0 MSS=1460 WS=2
3	0.000253	192.168.206.1	192.168.206.128	TCP	1159 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.014758	192.168.206.1	192.168.206.128	HTTP	GET / HTTP/1.1
5	0.015028	192.168.206.128	192.168.206.1	TCP	http > 1159 [ACK] Seq=1 Ack=376 Win=6912 Len=0
6	0.019624	192.168.206.128	192.168.206.1	TCP	[TCP segment of a reassembled PDU]
7	0.019711	192.168.206.128	192.168.206.1	TCP	[TCP segment of a reassembled PDU]
8	0.019732	192.168.206.1	192.168.206.128	TCP	1159 > http [ACK] Seq=376 Ack=2921 Win=64240 Len=0
9	0.019807	192.168.206.128	192.168.206.1	TCP	[TCP segment of a reassembled PDU]
10	0.019830	192.168.206.1	192.168.206.128	TCP	1159 > http [ACK] Seq=376 Ack=4381 Win=64240 Len=0
11	0.020522	192.168.206.128	192.168.206.1	HTTP	HTTP/1.1 403 Forbidden (text/html)
12	0.020812	192.168.206.1			
13	0.020835	192.168.206.1			
14	0.023894	192.168.206.1			
15	0.024486	192.168.206.1			

istenilen site ismi burada belirtiliyor..



```
GET / HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
Accept-Language: tr
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Host: snort-home
Connection: Keep-Alive
```

Tipik bir HTTP isteği ve Host Alanı

Kaynaklar:

<http://www.cacert.org>.

<http://www.openca.org>

Apache SSL Documentation