ÖN BİLGİ

Bu belgenin resmi adresi bulunamamıştır. Alternatif adreste yedeklenmiştir. Bu belge

- https://www.includekarabuk.com/kitaplik/indirmeDeposu/Siber_Guvenlik_Teknik_Makaleler/Teori/BaskalarinaAitMakaleler/SSL%20Say%C4%B1sal%20Sertifikalar.pdf

adresindeki makaleye çalışılarak elde edilen notlarımı kapsamaktadır. Bu çıkarılan notlar belgemde alıntılar ve/veya kişisel ilavelerim mevcuttur.

1)

Sifreleme algoritmaları
Simetrik ve asimetrik olmak üzere iki çesittir. Simetrik anahtarlı sifreleme algoritmalarında bir veriyi sifrelemek ve sifreli veriden orjinal veriyi elde etmek için aynı anahtar kullanılır. Fakat asimetrik algoritmalarda veriyi sifrelemek ve çözmek için iki anahtar kullanılır. Veriyi sifrelemek için bir anahtar(public key), çözmek için diger anahtar(private) kullanılır. Benim public anahtarım kullanılarak sifrelenmis bir veri ancak benim private anahtarım kullanılarak çözülebilir. Bu yüzden asimetrik algoritmalarda açık anahtar(public key) dagıtılır, gizli anahtar(private key) saklanır. Bir de bunların haricinde mesaj özeti(message digest) olarak adlandırılan ve veri bütünlügünü koruma amaçlı kullanılan tek yönlü çalısan hash algoritmaları vardır, md5 , sha gibi.

2)

Sayısal İmza Kavramı
Verinin içeriginin degistirilme durumu ve göndereninin gerçekliligini ispat söz konusu ise sayısal imza kullanılır. Sayısal imza kısaca yazılan mesajın özetinin gizli anahtar ile sifrelenmesi ve bir sıra numarası eklenmis halidir.

Gönderici;
i) Mesajı gönderen mesajın özetini alır
ii) gizli anahtarını kullanarak özeti sifreler ve bir sıra numarası ekler.

Alıcı;
i) Alıcı göndericinin açık anahtarını kullanarak sifreyi çözer
ii) Sifre çözüldükten sonra ortaya gönderilen mesajın özeti çıkar
iii) Aynı algoritma kullanılarak mesaj özet islemine tabi tutulur ve dogrulugu kontrol edilir.

3)

Sertifikalar
Sayısal imzaları kullanarak bir verinin gerçekten beklenen kişi tarafından gönderildigi ve iletim esnasında degisiklige ugramadıgını anlayabiliriz. Peki gönderilen verinin gerçekten bekledigimiz insana ulastığından nasıl emin olabiliriz? Yani göndereceğimiz kişinin açık anahtarını kullanarak sifrelediğimiz veriler gerçekte düsündügümüz kisinin açık anahtarıyla mı

şifrelendi? Ve dolayısıyla gerçekte düşündüğümüz kişiye mi gidiyor? Nasıl emin olabiliriz. Burada sertifika tanımı ortaya çıkıyor. Bir sertifika basitce kisinin açık anahtarının yetkili bir sertifika otoritesi tarafından tescillenmiş halidir diyebiliriz.

Sertifika Otoritesi
Sertifika isteginde bulunan sahıs/kurumların gerçekte belirttikleri kisiler/kurumlar olduklarını dogrulayan ve onaylayan kurumdur. Verisign, Globalsign gibi.. Eger her iki taraf da ortak güvenilen bir sertifika otoritesi tarafından imzalanmıs sertifika kullanıyorsa birbirlerinin public key'lerine güvenebilirler.

4)

Kendi sertifakamızı özgür bir ssl sürümü olan OpenSSL ile oluşturabiliriz. Detaylı bilgi için sayfa 3 – 13.

5)

 # grep huzeyfe /etc/shadow
huzeyfe**:$1**$47JagHAu$6HvvYMo4maDHzlTGdB6WT/:13465:0:99999:7::
kırmızı ve yesilli alan huzeyfe kullanıcısının parolasının sifreli halidir. Burada kırmızı alan içerisindeki birinci ve ikinci $ isaretleri arasındaki ifade salt olarak adlandırılır. Linux sistemlerde kullanıcı adı-parola kontrolü yaparken kullanıcının girdigi degeri alıp salt ile isleme soktuktan sonra çıktıyı bu dosyadaki ilgili satır ile karsılastırıyor ve sonucu olumlu ya da olumsuz olarak dönüyor.

6)

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

Istemci ssl siteye baglanan tarafı belirten, Sunucu ise ssl çalısan sistemi belirten taraf olsun. Bu ikisi arasındaki güvenli iletisim asagıdaki adımlarla gerçekleşir.

Istemci sunucuya *ClientHello* olarak adlandırılan istemcinin destekledigi sifreleme algoritmaları, sıkıstırma algoritmaları gibi özellikleri belirten bir mesaj gönderir.
Sunucu bu mesajı alır ve istemciye uygun seçtigi algoritmaları vs bildiren bir mesaj gönderir. Bu mesaj *ServerHello* olarak geçer.

7)

İsim tabanli sanal host kavramı (name-based virtual hosting) bir IP üzerinden birden fazla hostu sunma için kullanılan bilindik bir yöntem. Fakat bu yontem eger host edilen sunucularda SSL kullanilacaksa ise yaramiyor. Neden mi?

Bir IP uzerinden birden fazla host sunma http isteklerinde tasınan Host baslıgına baglidir. Bir http istegi olusturuldugunda host basligi hedef IPdeki hangi hostu istedigini belirtir. Bizim web sunucumuza gelen istek apache (ya da baska bir web sunucu) tarafindan yorumlanarak uygun site kullaniciya gosterilir.

SSL destekli bir web sunucuda bir hosta gelen HTTPS isteginde Host headeri sifreli olarak gelecegi için (OSI katmaninda SSL HTTP'den oncedir) web sunucu Host alanını okuyamaz ve o alan adina ozel islemler gerceklestiremez, dolayisi ile istenilen siteyi gösteremez. Bu yüzden virtual host ile birlikte SSL kullanilamaz. Bunun yerine Ip tabanli sanal host kullanilmalidir.