HTTPS

名词

SSL

- **SSL** - Secure Sockets Layer
  - https的加密是基于SSL的
  - SSL著名的漏洞 - *心脏出血*
- **TLS** - Transport Layer Security
  - 它的前身就是 `SSL`

证书标准

- **X.509** - 这是一种证书标准,主要定义了证书中应该包含哪些内容.其详情可以参考RFC5280,SSL使用的就是这种证书标准.

编码格式

同样的X.509证书,可能有不同的编码格式,目前有以下两种编码格式

- **PEM** - Privacy Enhanced Mail
  - 打开看文本格式,以"-----BEGIN..."开头, "-----END..."结尾,内容是BASE64编码.
  - 查看PEM格式证书的信息: `openssl x509 -in certificate.pem -text -noout`
  - Apache和*NIX服务器偏向于使用这种编码格式.
- **DER** - Distinguished Encoding Rules
  - 打开看是二进制格式,不可读.
  - 查看DER格式证书的信息:openssl x509 -in certificate.der **-inform der** -text -noout
  - Java和Windows服务器偏向于使用这种编码格式

相关的文件扩展名

这是比较误导人的地方,虽然我们已经知道有PEM和DER这两种编码格式,但文件扩展名并不一定就叫"PEM"或者"DER",常见的扩展名除了PEM和DER还有以下这些,它们除了编码格式可能不同之外,内容也有差别,但大多数都能相互转换编码格式.

- **CRT** - CRT应该是certificate的三个字母,其实还是证书的意思,常见于*NIX系统,有可能是PEM编码,也有可能是DER编码,大多数应该是PEM编码,相信你已经知道怎么辨别.

- **CER** - 还是certificate,还是证书,常见于Windows系统,同样的,可能是PEM编码,也可能是DER编码,大多数应该是DER编码.
- **KEY** - 通常用来存放一个公钥或者私钥,并非X.509证书,编码同样的,可能是PEM,也可能是DER.
  - 查看KEY的办法:openssl rsa -in mykey.key -text -noout
  - 如果是DER格式的话,同理应该这样了:openssl rsa -in mykey.key -text -noout **-inform der**
- **CSR** - Certificate Signing Request,即证书签名请求,这个并不是证书,而是向权威证书颁发机构获得签名证书的申请,其核心内容是一个公钥(当然还附带了一些别的信息),在生成这个申请的时候,同时也会生成一个私钥,私钥要自己保管好.做过iOS APP的朋友都应该知道是怎么向苹果申请开发者证书的吧.
  - 查看的办法:openssl req -noout -text -in my.csr (如果是DER格式的话照旧加上-inform der,这里不写了)
- **PFX/P12** - predecessor of PKCS#12,对*nix服务器来说,一般CRT和KEY是分开存放在不同文件中的,但Windows的IIS则将它们存在一个PFX文件中,(因此这个文件包含了证书及私钥)这样会不会不安全？应该不会,PFX通常会有一个"提取密码",你想把里面的东西读取出来的话,它就要求你提供提取密码,PFX使用的时DER编码,如何把PFX转换为PEM编码？
  - openssl pkcs12 -in for-iis.pfx -out for-iis.pem -nodes

    这个时候会提示你输入提取代码. for-iis.pem就是可读的文本.
  - 生成pfx的命令类似这样:openssl pkcs12 -export -in certificate.crt -inkey privateKey.key -out certificate.pfx -certfile CACert.crt

    其中CACert.crt是CA(权威证书颁发机构)的根证书,有的话也通过-certfile参数一起带进去.这么看来,PFX其实是个证书密钥库.
- **JKS** - 即Java Key Storage,这是Java的专利,跟OpenSSL关系不大,利用Java的一个叫"keytool"的工具,可以将PFX转为JKS,当然了,keytool也能直接生成JKS

证书编码的转换

- **PEM转为DER**
  - openssl x509 -in cert.crt -outform der -out cert.der
- **DER转为PEM**
  - openssl x509 -in cert.crt -inform der -outform pem -out cert.pem

(提示:要转换KEY文件也类似,只不过把x509换成rsa,要转CSR的话,把x509换成req...)

获得证书

- **向权威证书颁发机构申请证书**
  - 用这命令生成一个csr: openssl req -newkey rsa:2048 -new -nodes -keyout my.key -out my.csr
  - 把csr交给权威证书颁发机构,权威证书颁发机构对此进行签名,完成.保留好csr,当权威证书颁发机构颁发的证书过期的时候,你还可以用同样的csr来申请新的证书,key保持不变.
- **或者生成自签名的证书**
  - openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem
  - 在生成证书的过程中会要你填一堆的东西,其实真正要填的只有Common Name,通常填写你服务器的域名,如"yourcompany.com",或者你服务器的IP地址,其它都可以留空的. 生产环境中还是不要使用自签的证书,否则浏览器会不认,或者如果你是企业应用的话能够强制让用户的浏览器接受你的自签证书也行.向权威机构要证书通常是要钱的,但现在也有免费的,仅仅需要一个简单的域名验证即可.有兴趣的话查查"沃通数字证书".

HTTPS免费证书

windows服务器上通过`Let’s Encrypt`（教程）添加证书