上课笔记：

TLS：主流基于1.2版本

重放保护：将数据重新发送给对方来获取账户和密码

1. 通过https访问网站
2. 服务器通过证书例的签名过的公钥
3. 客户段
4. 有效生成随机的加密密钥，通过公钥加密
5. 服务通过私钥解密
6. 以后传输多通过公钥加密

OpenSSL：开源项目

Bshe64内审不回家你

非对称的加密算法

搭建私有CA

加密与安全

墨菲定律：

1. 任何事情都没有表面看起来那么简单
2. 任何的事队会比你预计的时间长
3. 会出错的事总会出错
4. 如果你担心某种事情发生，那么他总会发生

安全机制：

信息安全的防护目标：

1. 保密性
2. 完整性
3. 可用性
4. 可控制性
5. 不可否认性

保护安全的环节：

1. 物理安全：保证与机器接触的安全
2. 系统安全：操作系统的安全
3. 应用安全： 一些应用也有一定的安全
4. 网络安全：对一些网络访问的控制和一些防火墙的规则
5. 数据安全：数据的加密，数据的备份
6. 管理安全：各种规范和方法

攻击安全的方法：

1. 冒充：例冒充别人发邮件：

可以使用：telnet IP 子网

来冒充别人发送邮件

查看公司邮件服务器地址：dig -t 域名

1. 篡改 否认 拒绝服务 信息泄露 提升权限

安全设计的基本原则：

使用成熟的安全系统 

以小人之心度输入数据 

外部系统是不安全的 

最小授权 

减少外部接口 

缺省使用安全模式 

安全不是似是而非 

从STRIDE思考 

在入口处检查 

从管理上保护好你的系统

对称加密算法;

没有加密过的数据叫明文，加密过的叫密文

对称加密：加密和解密多使用同一个密钥

对称加密传输过程：明文🡪加密🡪密文🡪解密🡪明文

使用的密钥是：以前是用：DES：56位

现在多使用AES：（128，192，256位）可以转换

非对称加密算法：

每个用户都有自己的公钥和私钥

公钥是公开的，私钥是自己的每个机器的私钥都不一样

存放公钥的位置：/etc/ssh/ssh\_host\_rsa\_key.pub

存放私钥的位置：/etc/ssh/ssh\_host\_rsa\_key

公钥只能用自己的私钥解密

非对称加密算法的传输：

a🡪b

a🡪用b的公钥加密🡪密文🡪b用b的私钥解密🡪明文🡪b

确认发送源：

1. 》b

a🡪用b的公钥加密🡪用a的私钥加密🡪密文🡪用a的公钥解密（来确定数据来自a）🡪用b的私钥解密🡪明文 顺序反了

确认来源的过程叫：数字签名

算法使用的有：RSA等

私钥一般才用对称的算法加密

单向散列：haxi算法： 把不同长度的数据用相同的算法加密得到的二进制密文长度是相同的

不同的数据使用单向散列算法得出的结果不同（结果又叫摘要），摘要不能出算出数据

数据相同摘要相同，数据不同摘要不同

常见算法：MD5：128位，sha512

Md5的单向散列可以使用md5sum +数据，计算摘要

Sha1的单向散列可以使用：sha1sum +数据 计算摘要

加密传输优化：数据asd

1. asd+{【haxi的结果】+a的私钥签名+b的公钥加密}🡪🡪🡪🡪🡪🡪b的私钥解密+确认来源a的公钥解密+haxi运算数据有无变化+asd

因为haxi的结果是固定的所以只加密haxi值会提升效率

1. 对数据asd对称加密+非对称加密+haxi值

对称加密+{asd+a的私钥签名+asd的haxi}+b的公钥加密{对称密钥}

1. 解密过程：
2. B的私钥解密获得对称加密=对称加密+{asd+a的私钥签名+asd的haxi}
3. 对称解密={asd+a的私钥签名+asd的haxi}
4. 确认来源a的公钥解密=asd+ads的haxi
5. 确认数据有无变化=asd

Rpm中使用公钥检查安装包是否被篡改

Rpm –-import rpm-gpg-key-centos7导入公钥钥

Rpm -k tree-1.6………。rpm 检查公钥钥

证书和申请

加密文件（对称加密）

Gpg -c 文件名

解密：gpg -d 文件名

生成新的文件：gpg -o 要改成的文件名 ：-o要放在-d前

非对称加密传输

以34.6向34.7传输

1. 在34.6上生成公钥

：查看公钥gpg –-list-keys

生成公钥：gpg –gen-key

1. 生成算法
2. 加密位数
3. 制定有效期
4. 确定？
5. 生成的密钥是谁的？：输入名称（最少5位）
6. 邮箱 （可选）
7. 描述 （可选）
8. 确定？
9. 对私钥输入口令
10. 生成随机数

生成的路径：公钥：/root/.gnupg/pubrinf.gpg

私钥：/root/.gnupg/secring.gpg

因为文件是乱码，

查看使用命令：gpg -a –-export （以文件方式到处）-o（导出的位置）

使centos7有两把钥匙：6和7的

在7生成后将6的密钥导入到7中使用：gpg –-import （6的公钥在7中保存的位置）

开始传输：gpg -e -r （使用的密钥name）（要传输的文件名）

1. 是否是要使用的name
2. 是否覆盖？

使用scp 开始传输

在6上开始解密：gpg -o 传过来的文件要改的名 -d 传过来的文件

完成

删除公钥：gpg –delete-keys （公钥的name）

删除自己的公钥：

1. 先删除私钥：gpg –-delete-secret-keys （name）
2. 删除公钥：gpg –-delete-keys （name）

CA和证书：

确保真实公钥

通过证书验证： CA的私钥签名+CA的机构名称+提交申请的名称+

Ca的作用在a与b进行通信加密时

在交换公钥时通过ca来加密

在使用证书进行加密公钥：b在获得a的证书后使用本机自带的根ca的公钥来解开证书来获得a的公钥

根ca的证书是自己证明自己

吊销证书的目录：CRL

在证书的主机名显示的是：\*.1236.com 表示广域证书，是1236.com为后缀的都可使用

SSL：安全套接层

也叫TLS

功能：机密性，认证，完整性，重放保护

两阶段协议：握手协议和应用阶段

握手阶段：加密传输时需用使用的密钥协议等

应用阶段：使用握手阶段的协商好的密钥方式等

在协议层的位置：在应用层之下，传输层之上

https协议使用ssl协议

http协议使用tls协议叫https协议

在网站上登陆的保密https的工作过程：

1. 向服务端输入域名：
2. 服务端通过证书加密的公钥发送跟客户段
3. 客户段通过ca的公钥解密证书得到公钥
4. 客户段证书验证不通过显示网站不安全但也可以继续访问，通过则继续
5. 客户端生成随机的对称密钥
6. 使用服务的公钥加密客户端的公钥传输跟服务端
7. 服务端使用私钥解密自己的公钥得到客户端的公钥
8. 使用非对称的加密进行传输

Openssl开源的项目：实现ssl协议

Openssl来自openssl的包

有两种工作模式：交互模式和批处理模式

交互式模式： 最后的表现形式使用base64编码进行显示

使用openss进行加密：

Openssl enc -e -des3 -a -adlt -in 文件 -out 将结果重定向到文件

对文件进行加密 使用3des加密 对密码显示进行打乱

解密将-e改为-d

也支持 单向加密haxi算法

Openssl dgst -（哈希算法）+文件 实现文件校验

生成随机数：openssl rand -base64 3（指定的位数）

Openssl生成私钥：

Openssl genrsa -out /文件路径 （umask 077；openssl genrsa -out f1 -des 2048）

Umask 077：表示文件的权限是600

加（）：表示只在当前命令生效

-out f1：生成的私钥传输到f1文件

-des 2048：使用des进行加密 位数2048

对私钥进行解密：openssl rsa in f1 -out 创新的文件

简单方式：在 /etc/pki/tls/certs 里执行命令

Make /data/ssl/tset4.key

在/data/ssl/下生成私钥文件test4.key

原因在 /etc/pki/tls/certs下的makefiie文件定义了生成密钥的命令

从四幺中提取公钥：

加密文件 Openssl rsa -in 私钥文件 -pubout -out 要生成的文件

生成私有CA，实现证书申请

搭建私有CA，用于网站[www.mage.com](http://www.mage.com)

参考配置文件来搭建私有CA /etc/pki/tls/openssl。Conf

可以搭建多个ca ，但要有一个默认ca



定义了默认ca是那个

Dir = /etc/pki/ca 关于ca的所有文件都在该文件下

Carts =$dir/certs 旧的发布的证书的位置

Crl\_dri =$dir/crl 吊销的证书位置

Database =$dir/index.txt 存放证书编号的数据库文件（默认没有，手工添加）

New\_certs\_dir =$dir/newcarts 新颁发的证书

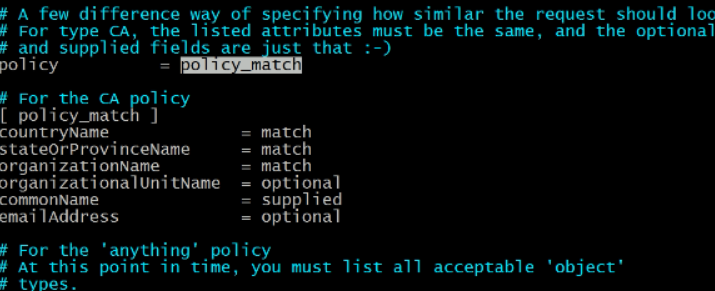
Certificate =$dri/cacert.pem ca的跟证书的位置

Serial =$dir/serial 下一个证书的编号（默认没有，手工添加）

Crlnumber =$dri/crlnumber 下一个吊销证书编号

Private\_key =$dir/private/cakey.pem ca的私钥

申请证书时与ca的拼配策略



1. 国家
2. 省（下面少一个城市）
3. 公司
4. 部门
5. 网站名
6. 邮箱

后为match的项目必须与ca服务器相拼配

后为optional的可以一样也可以不一样

具体操作：

进入：/etc/pki/ca

1. 创建私钥 （umake 077；openssl genrsa -out /etc/pki/ca/private/cakey.pem 1024）
2. 自己跟自己颁发证书（根证书）：

Openssl req -new -x509 -key /etc/pki/ca/private/cakey.pem -deys 3650 -out cacetr。Pem

注释：req 申请 -new 新的 -x509 自己跟自己颁发 -key 私钥证书

-deys 有效期 3650 天 -out 保存到

3. 输入国家等

4. 创建index。Txt文件

5. 在serial里写入开始的编号16进制数

完成后可以使用sz传到windows将文件后缀改为crt来查看

在客户段申请证书的过程：

1. 生成私钥：命令同上
2. 申请证书：

Openssl req -new -key 私钥文件 -out 存放位置app。Csr

填写国家等资料

1. 使用scp 将生成的文件传到服务器放到/etc/pki/ca

审核完成后

颁发证书：

在服务端：openssl ca -in （提交的申请文件）-out /etc/pki/ca/certs/设一个文件名 -days 3065（有效期）

SSH基于密钥的验证登陆

用户的公钥和私钥