1. HTTPS
2. HTTP的缺点
3. 通信使用明文

HTTP协议通信使用的是明文，没有经过加密处理，容易被窃听。HTTP协议不具有加密的功能，无法对通信整体进行加密。有两种加密方式，一种是对通信过程加密，另一种是对内容加密。

通信过程加密通过SSL（Secure Socket Layer）安全套接层或TLS（Transport Layer Security）安全传输层协议组合使用，建立安全通信线路，与SSL组合使用的HTTP为HTTPS（超文本传输安全协议）。

对内容本身加密是通过客户端的加密和服务器端的解密过程实现的。

1. 没有身份验证机制

HTTP协议不验证通信方的身份，因此可能遭遇伪装。通过使用SSL可以实现身份的认证，SSL不仅提供加密处理，还是用了证书，证书由信任的第三方机构颁发，用以证明服务器和客户端是实际存在的，同时客户端持有证书也可以完成个人身份的证明。

1. 无法验证报文的完整性

HTTP协议无法证明报文是完整的，因此可能遭遇篡改。

1. HTTPS介绍
2. 概述

HTTPS就是添加了加密，认证机制以及完整性保护的HTTP，他并不是应用层的一种新的协议，只是HTTP通信接口部分使用SSL和TLS协议代替。通常HTTP直接和TCP通信，当使用SSL时，则先和SSL通信，再由SSL和TCP通信，即SSL协议在HTTP协议的外层。

采用了SSL之后，HTTP就拥有了HTTPS的加密，证书，完整性保护功能。SSL是独立的协议，所以其他应用层协议如SMTP，Telnet都可以和SSL协议配合使用。且SSL的通信速度比HTTP要慢2到100倍，同时SSL还要处理加密与解密，更慢了。

1. 加密技术
2. 共享密钥加密

加密和解密使用同一个密钥的方式称为共享密钥加密，也叫对称密钥加密。该方式在加密之后需要将密钥发送给该对方，如何保证密钥能够安全地传输给对方，且如果密钥传输是安全的，那么通信也就是安全的，密钥就成为多余的了。

1. 公共密钥加密

公共密钥加密解决了共享密钥加密的问题，使用一对非对称的密钥，一个是私钥，本人保存；一个是公钥，可以公开。发送密文的一方使用对方的公开密钥进行加密，对方接收到被加密的信息后，再使用自己的私钥解密。这种方式不需要发送解密的私钥，私钥丢失概率降低。

1. HTTPS的加密机制

HTTPS采用两者混合的加密机制，由于公共密钥加密处理更为复杂，因此通信速率较低。在交换密钥阶段采用的是公共密钥加密，而在需要大量流量通信的交换报文阶段使用的是共享密钥加密方式。

1. 证书机制

但这样还不够，交换密钥阶段无法证明发送的公钥就是服务器的公钥，为了解决这个问题，使用由数字证书认证机构（CA，Certificate Authority）颁发的公开密钥证书证明公钥的有效性。服务器向机构申请公钥证书，机构判明申请者的身份后，使用机构的私钥对服务器的公钥做数字签名，然后将已签名的服务器公钥放入公钥证书中，服务器再将该证书发送给客户端，客户端中一般内置了常见的机构的公钥，客户端根据机构的公钥解开得到服务器的公钥，就可以通过该公钥解开得到服务器通过私钥加密的共享通信的密钥，接下来就可以进行共享密钥的通信了。

1. HTTPS通信步骤
2. 服务器向证书机构申请服务器公钥的证书，机构验证了申请者的身份后，就会使用机构自己的私钥对服务器公钥进行签名，和服务器公钥一起放入公钥证书中；
3. 服务器将公钥证书发送给客户端，客户端中一般都内置了常见机构的公钥，接受到公钥证书后，通过机构的公钥解开得到了服务器公钥；
4. 客户端通过服务器公钥加密了最后HTTP通信使用的共享密钥后发送给服务器，服务器接收到之后用服务器的私钥解密得到了HTTP通信使用的共享密钥；
5. 客户端和服务器端都得到了共享密钥，就可以通过共享密钥来进行HTTP通信了。
6. 客户端证书

HTTPS还可以使用客户端证书，证明客户端的身份，作用和服务器证书一致，但是要想获取证书，需要用户自己安装客户端证书，且该证书通常是付费的，很少使用，一般用在银行的网上银行中，用以确保用户从特定的客户端登录。

1. HTTP的认证机制
2. 认证机制分类

HTTP/1.1使用的认证机制有四种，Basic基本认证，Digest摘要认证，SSL客户端认证，FormBase表单认证。

1. Basic认证

基本认证是HTTP/1.0定义的认证方式，认证步骤如下：

1. 当请求的资源需要BASIC认证时，服务器发送401并携带WWW-Authenticate首部字段的响应，该字段内包含着认证的方式（BASIC）和安全域字符串；
2. 客户端接受到响应后将用户名和密码通过冒号：连接后，经过Base64编码处理，放入Authorization首部字段中发送给服务器；
3. 服务器验证Authorization字段的正确性。

Base64编码方式并不是加密，任何人都可以解码Base64的编码结构，因此容易造成信息泄露，同时一般的浏览器无法实现认证注销，因此一般只能进行一次BASIC认证，非常麻烦，安全性也不高，因此Basic认证很少使用。

1. Digest认证

HTTP/1.1有了Digest认证，同样使用质询/响应的方式。客户端发送认证要求给服务器，服务器返回质询码，客户端接受质询码，计算生成响应码，最后将响应码返回给服务器。由于客户端发送的只是响应摘要和响应码，比起Basic认证，密码泄露可能性降低了。

1. SSL客户端认证

通过HTTPS的客户端证书完成认证，确认客户端的身份。该认证方式一般和下面的表单认证一同使用。

1. 表单认证

根据Web程序的要求由客户端向服务器发送登录信息，根据Web程序的验证逻辑验证用户之前登录的账号和密码。一般使用的是Cookie来管理Session会话。

认证步骤：

1. 客户端首先将密码等信息放入报文的实体部分，通常以POST方法将请求发送给服务器，使用HTTPS传输数据；
2. 服务器根据密码等信息发放用以识别用户的Session ID，通过验证客户端发送的登录信息进行身份验证，把用户的认证状态和Session ID绑定记录在服务器端；
3. 再向客户端返回响应，在首部字段Set-Cookie内写入Session ID；
4. 客户端收到Session ID后，将其作为Cookie保存在本地，在下次发送请求时，浏览器会自动发送Cookie，所以Session ID也会发送到服务器；
5. 服务器匹配发送来的Session ID和服务器中的Session ID，认证用户的身份；
6. 安全漏洞
7. 点击劫持

利用透明的按钮或链接做成陷阱，覆盖在web页面上，诱导用户点击。

1. Dos攻击

Denial of Service attack，让运行中的服务停止的攻击，通过集中访问请求造成资源过载，资源用尽，服务也就停止了，这些请求一般都是合法的，服务器很难分辨。DDos就是多个计算机发起的Dos攻击。

1. 后门程序

开发设置的隐藏入口，可不按照正常步骤使用受限的功能，利用后门程序就可以使用原本受限的功能。通常的后门程序有三种：

1. 开发阶段的Debug调用的后门程序；
2. 开发者植入的后门程序；
3. 攻击者设置的后门程序；