传输层安全协议详解 PDF转换可能丢失图片或格式，建议阅读原文
https://www.100test.com/kao_ti2020/140/2021_2022__E4_BC_A0_E8_BE_93_E5_B1_82_E5_c100_140072.htm 传输层安全协议的目的是为了保护传输层的安全，并在传输层上提供实现保密、认证和完整性的方法。 1．SSL（安全套接字层协议） SSL(Secure Socket Layer)是由Netscape设计的一种开放协议；它指定了一种在应用程序协议(例如http、telnet、NNTP、FTP)和TCP/IP之间提供数据安全性分层的机制。它为TCP/IP连接提供数据加密、服务器认证、消息完整性以及可选的客户机认证。 SSL的主要目的是在两个通信应用程序之间提供私密信和可靠性。这个过程通过3个元素来完成：l. 握手协议。这个协议负责协商被用于客户机和服务器之间会话的加密参数。当一个SSL客户机和服务器第一次开始通信时，它们在一个协议版本上达成一致，选择加密算法，选择相互认证，并使用公钥技术来生成共享密钥。l. 记录协议。这个协议用于交换应用层数据。应用程序消息被分割成可管理的数据块，还可以压缩，并应用一个MAC（消息认证代码）；然后结果被加密并传输。接受方接受数据并对它解密，校验MAC，解压缩并重新组合它，并把结果提交给应用程序协议。l. 警告协议。这个协议用于指示在什么时候发生了错误或两个主机之间的会话在什么时候终止。 下面我们来看一个使用WEB客户机和服务器的范例。WEB客户机通过连接到一个支持SSL的服务器，启动一次SSL会话。支持SSL的典型WEB服务器在一个与标准HTTP请求（默认为端口80）不同的端口（默认为443）上接受SSL连接请求。当客户机连接到这个端口上时，它将

启动一次建立SSL会话的握手。当握手完成之后，通信内容被加密，并且执行消息完整性检查，知道SSL会话过期。SSL创建一个会话，在此期间，握手必须只发生过一次。 SSL握手过程步骤： 步骤1：SSL客户机连接到SSL服务器，并要求服务器验证它自身的身份。 步骤2：服务器通过发送它的数字证书证明其身份。这个交换还可以包括整个证书链，直到某个根证书权威机构(CA)。通过检查有效日期并确认证书包含有可信任CA的数字签名，来验证证书。 步骤3：然后，服务器发出一个请求，对客户端的证书进行验证。但是，因为缺乏公钥体系结构，当今的大多数服务器不进行客户端认证。 步骤4：协商用于加密的消息加密算法和用于完整性检查的哈希函数。通常由客户机提供它支持的所有算法列表，然后由服务器选择最强健的加密算法。 步骤5：客户机和服务器通过下列步骤生成会话密钥： a. 客户机生成一个随机数，并使用服务器的公钥（从服务器的证书中获得）对它加密，发送到服务器上。 b. 服务器用更加随机的数据（从客户机的密钥可用时则使用客户机密钥；否则以明文方式发送数据）响应。 c. 使用哈希函数，从随机数据生成密钥。 SSL协议的优点是它提供了连接安全，具有3个基本属性：l. 连接是私有的。在初始握手定义了一个密钥之后，将使用加密算法。对于数据加密使用了对称加密（例如DES和RC4）。 l. 可以使用非对称加密或公钥加密（例如RSA和DSS）来验证对等实体的身份。 l. 连接时可靠的。消息传输使用一个密钥的MAC，包括了消息完整性检查。其中使用了安全哈希函数（例如SHA和MD5）来进行MAC计算。 对于SSL的接受程度仅仅限于HTTP内。它在其他协议中已被表明可以使用，但还没有被广泛应用。 注

意： IETF正在定义一种新的协议，叫做“ 传输层安全 ”(Transport Layer Security,TLS)。它建立在Netscape所提出的SSL3.0协议规范基础上；对于用于传输层安全性的标准协议，整个行业好像都正在朝着TLS的方向发展。但是，在TLS和SSL3.0之间存在着显著的差别（主要是它们所支持的加密算法不同），这样，TLS1.0和SSL3.0不能互操作。 2. SSH（安全外壳协议） SSH是一种在不安全网络上用于安全远程登录和其他安全网络服务的协议。它提供了对安全远程登录、安全文件传输和安全TCP/IP和X-Window系统通信量进行转发的支持。它可以自动加密、认证并压缩所传输的数据。正在进行的定义SSH协议的工作确保SSH协议可以提供强健的安全性，防止密码分析和协议攻击，可以在没有全球密钥管理或证书基础设施的情况下工作的非常好，并且在可用时可以使用自己已有的证书基础设施（例如DNSSEC和X.509）。 SSH协议由3个主要组件组成： l. 传输层协议，它提供服务器认证、保密性和完整性，并具有完美的转发保密性。有时，它还可能提供压缩功能。 l. 用户认证协议，它负责从服务器对客户机的身份认证。 l. 连接协议，它把加密通道多路复用组成几个逻辑通道。 SSH传输层是一种安全的低层传输协议。它提供了强健的加密、加密主机认证和完整性保护。SSH中的认证是基于主机的；这种协议不执行用户认证。可以在SSH的上层为用户认证设计一种高级协议。 这种协议被设计成相当简单而灵活，以允许参数协商并最小化来回传输的次数。密钥交互方法、公钥算法、对称加密算法、消息认证算法以及哈希算法等都需要协商。 数据完整性是通过在每个包中包括一个消息认证代码(MAC)来保护的，这个MAC是根据一个共

享密钥、包序列号和包的内容计算得到的。 在UNIX、Windows和Macintosh系统上都可以找到SSH实现。它是一种广为接受的协议，使用众所周知的建立良好的加密、完整性和公钥算法。 3．SOCKS协议“ 套接字安全性 ”(socket security,SOCKS)是一种基于传输层的网络代理协议。它设计用于在TCP和UDP领域为客户机/服务器应用程序提供一个框架，以方便而安全的使用网络防火墙的服务。 SOCKS最初是由David和Michelle Koblas开发的；其代码在Internet上可以免费得到。自那之后经历了几次主要的修改，但该软件仍然可以免费得到。SOCKS版本4为基于TCP的客户机/服务器应用程序（包括telnet、FTP,以及流行的信息发现协议如http、WAIS和Gopher）提供了不安全的防火墙传输。SOCKS版本5在RFC1928中定义，它扩展了SOCKS版本4，包括了UDP；扩展了其框架，包括了对通用健壮的认证方案的提供；并扩展了寻址方案，包括了域名和IPV6地址。 当前存在一种提议，就是创建一种机制,通过防火墙来管理IP多点传送的入口和出口。这是通过对已有的SOCKS版本5协议定义扩展来完成的，它提供单点传送TCP和UDP流量的用户级认证防火墙传输提供了一个框架。但是，因为SOCKS版本5中当前的UDP支持存在着可升级性问题以及其他缺陷（必须解决之后才能实现多点传送），这些扩展分两部分定义。l. 基本级别UDP扩展。l. 多点传送UDP扩展。 SOCKS是通过在应用程序中用特殊版本替代标准网络系统调用来工作的（这是为什么SOCKS有时候也叫做应用程序级代理的原因）。这些新的系统调用在已知端口上（通常为1080/TCP）打开到一个SOCKS代理服务器（由用户在应用程序中配置，或在系统配置文件中指定

）的连接。如果连接请求成功，则客户机进入一个使用认证方法的协商，用选定的方法认证，然后发送一个中继请求。SOCKS服务器评价该请求，并建立适当的连接或拒绝它。当建立了与SOCKS服务器的连接之后，客户机应用程序把用户想要连接的机器名和端口号发送给服务器。由SOCKS服务器实际连接远程主机，然后透明地在客户机和远程主机之间来回移动数据。用户甚至都不知道SOCKS服务器位于该循环中。 使用SOCKS的困难在于，人们必须用SOCKS版本替代网络系统调用（这个过程通常称为对应用程序SOCKS化---SOCKS-ification或SOCKS-ifying）。幸运的是，大多数常用的网络应用程序（例如telnet、FTP、finger和whois）都已经被SOCKS化，并且许多厂商现把SOCKS支持包括在商业应用程序中。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com