安全 FTP系统的设计与实现

马 燕 刘海涛 白英彩
(上海交通大学信息安全学院 上海 200030)

摘 要 FTP协议是一种简单易用的文件传输协议,应用十分广泛,但是其安全问题不容忽视。在当前常用的基于 SSL/TLS协议的 FTP应用的基础上,设计并实现了一系列安全措施,从而大大提高了 FTP系统中的用户认证、传输和文件存储安全性。

关键词 FTP 安全 SSL 加密 OTP BASE64

DESIGN AND IMPLEMENTATION OF A SECURE FTP SYSTEM

Ma Yan Liu Haitao Bai Yingcai
(*School of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200030, China*)

Abstract FTP is a protocol which can transfer files between computers conveniently and has been widely used. However there are lots of security flaws with it. Based on the most commonly used FTP which adopts SSL/TLS protocol, a series of measures to enhance the security of authentication, data transfer and storage in an FTP system is designed and implemented.

Keywords FTP Security SSL Encryption OTP BASE64

0 引 言

文件传输协议 FTP是基于 TCP/IP的应用层协议,其主要功能是提供文件的共享、支持远距离计算机间接或直接连接、保护用户不因各类主机文件存储器系统的差异而受影响、进行可靠且有效的数据传输等[1],应用非常广泛。但是传统的 FTP有不少的安全漏洞,例如明文传输、缺乏对数据的机密性和完整性保护,对通信双方也没有可靠的认证措施等。针对 FTP的安全漏洞,近年来也出现了一些不需要对 FTP协议自身做完全更改的协议扩展模块,如 FTP SSL/TLS Extension。

SSL(Secure Sockets Layer)是用于对 TCP/IP数据流进行加密的协议,同时还包括了身份认证和数据完整性校验等内容。显然,基于 SSL/TLS的 FTP克服了明文传输的致命弱点,但是无可否认的是,在开放式的互联网环境下 FTP服务器受到恶意攻击的可能性还是很大,而且协议数据的安全性还是未得到保障。安全的本质是在信息的安全期内保证其在网络上流动的或者静态存放时不被非授权用户非法访问,但授权用户却可以访问[5]。基于这一概念,本文在 SSL的基础上设计了一个安全 FTP系统,从认证、传输、存储三个方面大大提高了 FTP的安全性。

1 系统设计

1.1 系统模型

1.1.1 网络模型

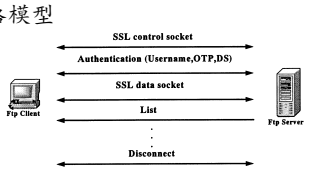

图 1 网络模型

如图 1所示,Ftp Client与 Ftp Server之间的交互过程如下所述:

(1) 建立 Ftp Control Connection的 SSL通道,接下来所有的传输都在 SSL通道里进行。

(2) 用户认证,通过(Username, OTP, DS)三元组进行,其中 OTP(One Time Password)是指一次性密码,DS(Digital Signature)指数字签名。整个认证过程需要事先在服务器端对每个用户名存储有种子、迭代次数 N以及根据秘密通行证短语(即用户的真正 Password)产生的 OTP(64位),这在创建用户及其口令时即保存为用户记录的内容。另外每个用户在客户端要产生自己的 RSA公/私钥对,其中私钥秘密保存、用于加密文件用,公钥上传至服务器以验证数字签名用。

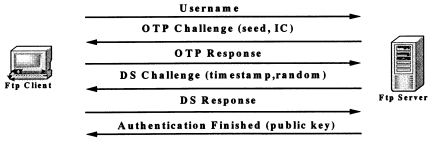

图 2 认证过程

认证过程如图 2所示,如果服务器接收到的 Username合法的话,将发出 OTP Challenge(seed, IC),客户端收到以后将根据 Challenge进行计算,得到 OTP短语(即一次性口令)返回。如果 OTP验证通过的话,服务器将产生一个时间戳以及一个大随机数并传送给客户端,要求数字签名。客户端使用用户自己的私钥对接收到的时间戳和随机数进行签名,并将结果返回服务器。服务器端收到签名后用保存的用户公钥验证数字签名,通过即验证完毕,同时返回用于对服务器上公共目录中文件加/解密的公共密钥。此密钥并非用于验证数字签名的用户个人的公钥,而是为所有合法用户所共有的用于访问公共目录的公共密钥。

收稿日期:2005-04-08。马燕,硕士生,主研领域:网络安全、存储安全。

(3) 服务器动态挂载用户的虚拟目录, 建立 FTP Data Connection的 SSL通道, 向用户列出目录。

(4) 进行其他 FTP操作, 如 RETR、STOR等。

(5) 断开连接。

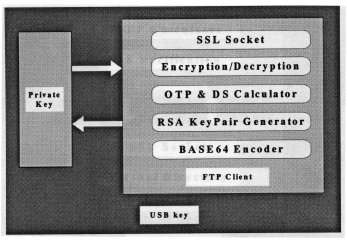

图 3 客户端模型

1.1.2 客户端模型

整个客户端应用程序被封装在可随身携带的 USB Key中, 用户可以随时将其中的 FTP Client安装到当前所使用的计算机中运行。如图 3 FTP Client中主要设计了以下几个安全模块:

(1) RSA Key pair Generator: 产生用户的 RSA 公钥 密钥对, 公钥上传到服务器在用户认证时验证数字签名; 私钥以加密方式存储到 USB Key 用于对用户私有目录下的文件进行加 /解密。

(2) SSL Socket: 建立 FTP控制连接以及数据连接的 SSL通道, 确保所有传输在加密通道里进行。

(3) OTP & DS Calculator: OTP Calculator在用户认证时根据从服务器端接收的 OTP challenge(seed IC)计算出 OTP短语(即一次性口令), 返回给服务器。 DS Calculator则根据服务器返回的时间戳和大随机数计算数字签名。

(4) Encryption /Decryption: 该模块在上传 /下载文件时根据用户选择的加密算法自动对数据及文件名进行加 解密, 密钥即上述通过 RSA Key-pair Generator产生的私钥, 从 USB Key中读取。该模块与 SSL通道一起构成了数据双重加密传输的安全机制。

(5) BASE64 Encoder: 考虑到文件名也有泄露相关信息的可能性, 如(4)所述, 在上传 /下载时除了对文件内容加 解密以外, 对文件名也同样要进行加 解密。由于加密后的数据会出现一些不可见字符, 为确保文件名可以正常显示, 故采用 BASE64编码对加密后的文件名进行编码, 转换不可见字符为可见字符。

1.1.3 服务器端模型

本文设计的 FTP系统其实是安全网络存储系统的一个子系统, 整个服务器端支持各种不同的协议, 例如 FTP、HTTP、CIFS、NFS等, FTP Server只是其中之一。因此, 设计 Server端时将各服务应用到的公共模块分离出来, 以提高系统的可扩展性。如图 4所示, 对 FTP应用来说, 主要用到服务器端的两个模块: FTP Server和 Authentication(认证层)。

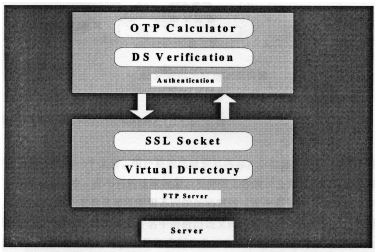

图 4 服务器端模型

FTP Client与服务器端通信时, 首先与 FTP Server建立 SSL连接。 FTP Server收到用户名后将与 Authentication模块交互, 处理接下来的用户认证工作。完成认证以后, 服务器将用户所在的目录动态地挂载到虚拟目录, 接着就向客户端传送当前目录信息(LIST)。

1.2 主要安全机制

1.2.1 认证三元组

传统的用户认证一般就是用户名加密码。如上所述, 在本系统中用户登录时需通过三元组(Username, OTP, DS)来认证。用户只有输入了正确的用户名、口令, 并用自己唯一的私钥进行数字签名, 才能成功登录服务器。一次性口令的应用可以保护用户的真实口令在任何时候都不会在网上传输, 数字签名的应用则可以更进一步地验证用户的身份, 阻止非授权用户的登录。

1.2.2 数据的双重加密传输

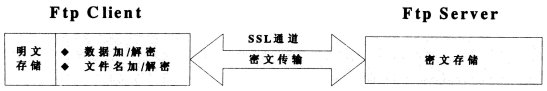

图 5 数据的双重加密传输

在所有的系统中, 数据本身的安全是尤为关键的, 因此在本文设计的系统中采用了 SSL通道以及密文传输的双重加密手段来增强数据传输和存储的安全性。如图 5所示, 为了减轻服务器端的负担, 将数据加 解密的工作放在客户端进行。文件上传时, 首先在客户端由用户选择加密算法并用相应的密钥(如果访问的是公共目录, 则用服务器传回的公共密钥; 如果访问的是用户的私有目录, 则用用户个人唯一的私钥)对文件内容及文件名进行加密, 然后通过建立的 SSL数据通道传输, 在服务器端存储的数据是密文。反之, 文件下载时, 在网络上通过 SSL通道传输的也是密文, 在客户端进行解密。由此可见, 在整个系统中, 除了客户端存放的数据是明文以外, 在网络上传输以及在服务器端存储的数据始终都是密文, 这样不但可以实现传输时的双重加密, 还可以防止数据在服务器端的非授权泄密, 极大地提高了数据的安全性。

1.2.3 文件存取四元组

相应地, 对应于服务器上的每一个文件, 用户要能读取到其内容, 必须正确提供四元组(UID, GID, Key Algorithm)。每个用户在不同的 group中, 有不同的读取权限。不同的文件夹(公共目录和私有目录)有不同的密钥(公共密钥和用户私钥), 而且对于每个文件又可以采用不同的加密算法(算法在文件上传前由用户动态指定, 在文件名上加一算法标志), 下载时客户端程序自动从文件名中解析出该文件所使用的加密算法, 然后用相应的密钥解密。因此, 服务器端数据存储的安全性非常高, 即使服务器受到攻击, 攻击者也很难获取真正需要的内容。

1.2.4 虚拟目录挂载

当用户认证通过以后, 服务器会动态地将用户的个人目录挂载到一个虚拟目录下, 每次登录时挂载的位置都不同, 以提高存储安全性。

2 系统实现

考虑到跨平台的因素, 系统采用 Java语言编写, Java提供了很好的安全编程的接口。本系统主要用到了 JSSE(Java Secure Sockets Extension即 Java安全套接字扩展)、JCA(Java

(下转第 218页)

4 结 论

从海量数据中提取有用信息,并利用这些信息作出有利于自己企业发展的决策是在现代信息社会中立于不败之地的关键所在,而 Analysis Services就是专门用于帮助决策者理解大量数据含义的有力工具。利用 Analysis Services支持多维分析的功能建立 OLAP分析系统模型,客户端应用程序利用 ADO MD结合 MDX 查询语句以及 Visual Basic功能强大的工具库进行开发,建立适合企事业的 OLAP多维分析系统,有助于管理者对市场作出及时的反应,正确地预测市场变化趋势,使企业实现商业智能和利润最大化。

实践证明运用基于 MS Analysis Services的 OLAP系统模型构建的决策支持系统实现了良好的数据多维分析的功能,为企事业创造了价值,具有较好的应用前景。

参 考 文 献

[1] 彭木根.数据仓库技术与实现[M].北京:电子工业出版社,2002.

[2] Inmon W H. Building the Data Warehouse[M]. Third Edition 北京:机械工业出版社,2003.

[3] Tony Bain. SQL Server 2000数据仓库与 Analysis Services[M]. 北京:中国电力出版社,2003.

[4] luca cabibbo the design and development of a logical system of OLAP. lecture notes in computer science 2004.

[5] 张涛. OLAP技术在数据仓库中的研究与应用[D].黑龙江:哈尔滨工业大学,2003.

[6] 康博创作室. SQL Server 2000数据仓库设计和使用指南[M].北京:清华大学出版社,2001.

[7] 庞康.数据仓库与 OLAP在决策支持系统中的设计与实现[D].广州:中山大学,2004.

(上接第 176页)

Cryptography Architecture即 Java密码架构)、JCE(Java Cryptography Extention即 Java加密扩展)等提供的各种安全类 API.

2.1 SSL通道的建立

FTP协议中采用了两条单独的 TCP连接通道,一条专用于发送控制命令,一条专用于传输数据。控制连接是客户端发出连接命令到关闭连接一直保持的,而数据连接则是每次需要传输数据时建立连接,传输完毕后立即释放。两条通道都使用 implicit的 SSL通道,从初始化连接开始的数据将全部加密以保证安全性。

2.2 数据的加解密

如上所述,加密算法可以由用户在上传文件时自行选择,因此在实现时提供了多种加密算法,如 DES、TripleDES、Blowfish、AES等,适当时还可以扩充其他算法。

对文件名加密以后可能出现不可见字符,因此采用 BASE64编码进行转换,它可以把二进制数据转换为可显示的 ASCII字符,如图 6所示。

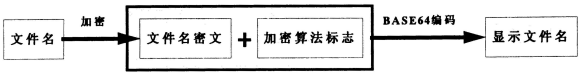

图 6 文件名的加密编码

按照 RFC2045的定义,BASE64编码被设计用来把任意序列的 8位字节描述为一种不易被人直接识别的形式。BASE64编码要求把 3个 8位字节(3*8=24)转化为 4个 6位的字节(4*6=24),之后在 6位的前面补两个 0,形成 8位一个字节的形式。Base64有其自身的编码表,表中有 64个可见字符(A~z,0~9,+、/),这也是 BASE64名称的由来。由于 BASE64是将 3个字节转换为 4个字节,所以当被编码字符不是 3的整数倍时,另外还有一个填充字符'='来解决这一问题。

在实现中发现 BASE64标准在对文件名的编码转换中有两个问题:(1)字符的转换中带有'/'字符,而该字符出现在文件名中则会被解析为路径而出错。(2)标准默认在转换出来的 76字符后自动添加回车符,这使得一些加密后字符数大于 76的长文件名在传输中出现了问题。为适应本文系统中文件名编码的需要,对 BASE64编码做了稍加修改,用'-'替换了原先的'/',同时去掉默认添加的回车符,从而保证文件名的加密编码并能正确传输显示。

2.3 OTP实现

在客户端和服务器端都有 OTP计算器,客户端的认证实现如下:

(1)连接到服务器 向服务端发出登录请求,将用户名送到服务端以便服务端核对用户是否合法。

(2)处理接收到的信息 收到服务器应答信息,例如 331 Response to otp-md5 999 rock376 required for skey.,从中取出挑战信息,则获得 IC=999,Seed=rock376,算法为 MD5。同时在 IC小于 5时提醒用户修改口令。

(3)进行 OTP计算 把所得 IC、Seed和用户自己的 Password作为计算器的输入参数,用 IC控制运算的次数,最后产生一个由六个英语单词组成的 OTP短语,并把得到的 OTP短语回送给服务器。

(4)修改口令 与服务端连接后输入旧口令,待服务端应答认可后输入新口令,并将新口令和服务端回传来的新挑战一起输入 OTP计算器处理产生新 OTP送服务端更新。

3 结束语

本文基于加密、身份认证、数字签名、SSL等技术提出一系列安全机制,设计并实现了一个安全的 FTP系统,从认证、传输、存储三大方面提高系统的安全性,可以满足较高的数据保密性要求。

参 考 文 献

[1] RFC959 (FILE TRANSFER PROTOCOL, FTP). http://rfc.net/rfc959.html

[2] RFC2577 (FTP Security Considerations). http://rfc.net/rfc2577.html

[3] RFC2045 (6.8 Base64 Content-Transfer-Encoding). http://rfc.net/rfc2045.html#p24.

[4] RFC1938 (A One-Time Password System). http://rfc.net/rfc1938.html

[5] 袁津生,吴砚农.计算机网络安全基础.人民邮电出版社.

[6] Chuck Cavaness Geoff Friesen Brian Keetoon. JAVA完全探索.师夷工作室,译.中国青年也版社.

[7] Rich Helton Johennie Helton JAVA安全解决方案.袁泉,吴静,等译.北京:清华大学出版社.

[8] Jess Garms Dainel Somerfield JAVA安全性编程指南.庞南,管和昌,陈立志,等译.电子工业出版社.