

Web 网站统一口令认证系统的设计与实现

黄叶珏¹, 陈 勤²

(1. 浙江工业职业技术学院, 浙江 绍兴 312000;

2. 杭州电子科技大学, 浙江 杭州 310018)

摘 要:分析了现有 Web 网站认证系统由于管理分散导致用户认证烦琐以及采用的后续认证方法存在安全隐患等问题。针对这些问题,设计了一个 Web 网站统一口令认证系统,具体给出了系统体系结构以及协议的设计,并对系统的关键实现技术进行了探讨。系统中,用户通过一次动态口令认证,即可访问网站权限范围内所有的应用服务,同时系统也实现了对用户的安全后续认证。

关键词:身份认证;动态口令认证;散列函数;重放攻击

中图分类号:TP393.08

文献标识码:A

文章编号:1673-629X(2007)0163-03

Design and Implementation of Web Site Universal Password Authentication System

HUANG Ye-jue¹, CHEN Qin²

(1. Zhejiang Industry Polytechnic College, Shaoxing 312000, China;

2. Hangzhou University of Electronic Science and Technology, Hangzhou 310018, China)

Abstract: The problems on Web site authentication system including repeated user authentication and security leak in the method that implement the persisting authentication, is analyzed in detail. For the purpose of resolving the problems, a universal password authentication system including the system model and the design of protocol is given. In system, through the universal dynamic password authentication, the user can visit all the services in the visiting range on Web site, the system also implements the secure persisting authentication.

Key words: identity authentication; dynamic password authentication; hash function; replay attack

0 引言

网站安全是网站建设中面临的一个极为重要的问题,身份认证^[1]是保障网络安全的一种重要手段之一。通常网站都设有认证系统,对访问者进行身份确认后,才允许其进行权限范围内的相应访问。一般情况下网站提供了各式各样的应用服务,比如某校园网站提供的应用服务有电子邮箱、选课系统以及教务管理系统等。由于各个应用服务单独管理自己的一套认证系统,导致用户认证极其烦琐,访问每个应用服务都需要进行重复认证。

此外,由于 HTTP 协议^[2]的无状态性,必须提供

一种方法保持 HTTP 的相关状态信息,为用户提供后续认证,以免用户从一个页面跳到另一个页面时重复输入认证信息。根据请求中携带的 Cookie 数据或者发出请求的客户端 IP 地址信息对用户进行后续认证是目前常用的方法,但方法存在安全隐患,攻击者通过 IP 欺骗或者窃取请求中的 Cookie 值等手段即可冒充合法登录用户发送 HTTP 请求。

针对以上分析的问题,设计了一个 Web 网站统一口令认证系统。系统中,用户通过一次口令认证,即可访问网站权限范围内(用户的权限在注册阶段确定)所有的应用服务,系统同时也实现了对用户的安全后续认证。

1 系统体系结构

Web 网站统一口令认证系统的体系结构如图 1 所示。系统主要有五部分组成:浏览器、客户端代理、认证服务器、服务器应用服务以及服务器端代理。浏览器以及服务器应用服务的功能不再详述,前者为发送

收稿日期:2006-08-16

基金项目:浙江省自然科学基金重点项目(ZD0101);现代通信国家重点实验室基金项目(51436040103DZ0401);浙江省教育厅高校科研计划项目(20030636)

作者简介:黄叶珏(1978-),女,浙江嵊州人,硕士研究生,研究方向为信息安全;陈 勤,教授,硕士,研究方向为密码学理论与应用、信息安全。

HTTP 请求和接受响应,后者则为网站提供的各式各样的应用服务,接受 HTTP 请求,并给以相应的服务。系统设计主要解决的是认证服务器、客户端代理以及服务器端代理的功能设计,以达到系统统一认证、安全性高的目标。

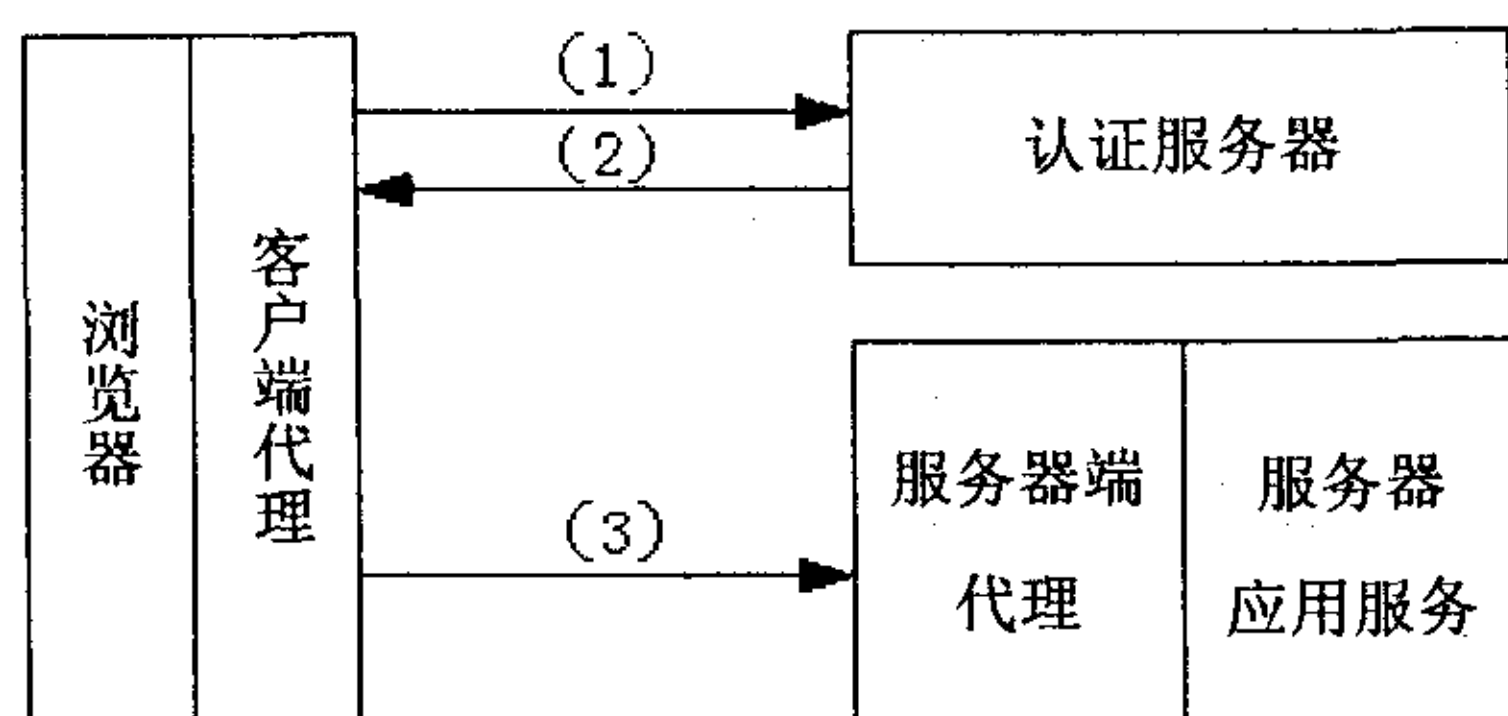


图 1 系统体系结构

1.1 客户端代理的设计

客户端代理主要分以下三个功能模块来实现:用户认证模块、票据管理模块和 HTTP 请求处理模块。

(1)用户认证模块。提供有关用户初始认证的服务,包括提取用户输入的身份标志符号和口令,向认证服务器发送认证请求,并完成与其之间的认证信息交互。

(2)票据管理模块。一旦用户通过初始认证,将接受从认证服务器返回的票据信息(包括该用户权限范围内可以访问的所有应用服务的票据,一个应用服务对应一个票据信息),并安全存放这些票据信息。

(3)HTTP 请求处理模块。透明处理浏览器发送的每个 HTTP 请求。具体处理步骤为:分析 HTTP 请求访问是哪个应用服务(根据请求中的目标 URL 来判断),从票据管理模块中查找是否有对应该应用服务的票据信息,如果存在,则生成验证字段,加到 HTTP 请求上,否则快速转发该请求(没有对应的票据信息表明用户无权访问该应用服务,或者对该应用服务的访问无需通过认证)。

1.2 认证服务器的设计

认证服务器的功能模块主要包括以下几部分:用户认证模块、票据发放模块、应用服务注册模块和用户注册模块。

(1)用户认证模块。处理客户端代理发送的认证请求,完成与其之间的认证信息交互,最终确认用户身份的合法性。

(2)票据发放模块。一旦通过对用户的身份确认,发放其可以访问的所有应用服务对应的票据信息。

(3)应用服务注册模块。网站提供的任何一个应用服务,只要该应用服务的访问是受限制的(即只有相应的合法用户才能访问),那么,该应用服务就需在认证服务器进行注册。该模块具体提供该方面服务。

(4)用户注册模块。所有用户都需要注册成为网

站的合法成员,才可以访问网站一些访问所限制的应用服务。该模块具体提供有关用户注册方面的服务。

1.3 服务器端代理的设计

网站的每个 Web 应用服务器(网站可能有多个 Web 应用服务器)都需要加上一个软件代理,即服务器端代理。每个服务器端代理管理着一张表信息,存放着该 Web 应用服务器上所有访问受限制的应用服务的 URL 等相关信息(一个 Web 应用服务器上可能有多个应用服务)。

每个服务器端代理透明处理发送给 Web 应用服务器的所有 HTTP 请求,具体处理步骤为:分析该请求需要访问的是哪个应用服务(根据请求的目标 URL 来确定);检查该应用服务的访问是否受到限制(根据管理的表信息来确定);如果该应用服务的访问受到限制,则检查 HTTP 请求是否携带验证字段,并检查验证字段的合法性,通过认证,则把请求交给 Web 应用服务器进行相应的处理,否则,则认为该请求是非法的,返回给用户无权访问的错误信息;如果 HTTP 请求访问的应用服务无需通过认证也能提供服务,则直接快速把请求转交给 Web 应用服务器。

2 系统协议设计

系统协议设计主要包括两个部分:

(1)认证服务器如何安全确认用户身份的合法性,该部分协议的设计综合考虑到安全性和操作的方便性,采用动态口令认证方式,参考并改进了文献[3]中的非对称口令认证协议,该动态口令认证协议的安全性分析可具体参考文献[4];

(2)服务器端代理如何确认每个 HTTP 请求的合法性,该部分协议的设计吸收了 Kerberos 协议^[5]基于票据来实现统一认证以及一次认证可多次访问的设计思想。

2.1 用户注册

用户选择身份标志符 ID_C 和秘密口令 PW 。认证服务器的用户注册模块首先检查该身份标志符 ID_C 是否已经被注册,如果已被注册,则要求用户重新选择身份标志符,否则就计算得到 g^{PW} (文中有关 g 的运算都在 Z_p^* 上,为便于描述,省略了 $\text{mod } p$),并按照角色给用户分配访问权限(不同的角色对应不同的访问权限),最后把该用户相应的 ID_C , g^{PW} 以及访问权限等信息存放在用户信息数据库。

2.2 应用服务注册

网站管理员输入应用服务的 URL 信息,应用服务注册模块首先给其分配一个私钥 K_V ,然后把该应用服务对应的记录包括 URL 以及 K_V 等信息,分别存放

进认证服务器管理的应用服务信息数据库,以及服务器端代理(该应用服务所在的Web应用服务器对应的服务器端代理)管理的应用服务信息数据库。认证服务器管理的数据库包括网站所有应用服务的注册信息,而服务器端代理管理的数据库只包括对应Web应用服务器(网站可能包括多个Web应用服务器,而一个Web应用服务器对应一个服务器端代理)上所有应用服务的注册信息。两个数据库的信息都必须保证安全性,可以采用加密存放的方式。

2.3 用户认证

Step1 用户输入身份标志符 ID_C 和口令 PW 。客户端代理的用户认证模块 Client_AS 提取 ID_C 和 PW , 生成一个随机数 N_1 , 向认证服务器发送信息 $g^{N_1} + g^{PW}$, 请求认证。

Step2 认证服务器的用户认证模块 Server_AS 接受认证请求, 根据 ID_C 检查用户信息数据库是否有对应的记录, 如果找不到相应的记录, 则结束认证, 否则, 取出记录信息 g^{PW} , 生成一个随机数 N_2 , 计算得到 $g^{N_1 N_2 + N_2 PW} = ((g^{N_1} + g^{PW} - g^{PW})g^{PW})^{N_2}$, 最后向 Client_AS 发送信息 $g^{N_2} + g^{PW}$, $H_1(g^{N_1 N_2 + N_2 PW})$ ($H_1()$, 以及下文提到的 $H_0()$ 和 $H_2()$ 表示三个不同的散列函数)。

Step3 Client_AS 首先根据信息 $g^{N_2} + g^{PW}$ 、 g^{PW} 、 N_1 以及 PW 求得 $g^{N_1 N_2 + N_2 PW} = (g^{N_2} + g^{PW} - g^{PW})^{N_1 + PW}$, 然后得到 $H_1(g^{N_1 N_2 + N_2 PW})$, 把此值与 AS 发送的 $H_1(g^{N_1 N_2 + N_2 PW})$ 进行比较, 相等则通过对 Server_AS 的认证, 否则, 结束认证。通过对 Server_AS 的认证后, 发送信息 $H_2(g^{N_1 N_2 + N_2 PW})$ 给 Server_AS, 同时求得本次认证的会话密钥 $H_0(g^{N_1 N_2 + N_2 PW})$ 。

Step4 Server_AS 首先计算得到 $H_2(g^{N_1 N_2 + N_2 PW})$, 并把此值与 Client_AS 发送的 $H_2(g^{N_1 N_2 + N_2 PW})$ 进行比较, 相等则通过对用户的身份认证, 同时求得本次认证的会话密钥 $H_0(g^{N_1 N_2 + N_2 PW})$, 否则, 认证失败。

2.4 发放票据

认证服务器通过对用户的身份认证后, 剩下的工作作为发放票据。票据发放模块首先根据用户信息数据库中的记录信息得到该用户的访问权限, 即该用户具体可以访问网站哪些应用服务, 并对每个应用服务都生成相应的一张票据信息, 某个应用服务对应票据信息的生成步骤为: 随机生成用户与该应用服务的共享密钥 $K_{C,v}$, 然后用该应用服务的私钥 K_v 加密 $K_{C,v}$ 、 ID_C 、用户主机的 IP 地址 AD_C 、该应用服务的 URL 以及票据有效时间 Lifetime 等信息生成 $Ticket_v = E_{K_v}(K_{C,v}, ID_C, AD_C, URL, Lifetime)$, 最后用会话密

钥 $H_0(g^{N_1 N_2 + N_2 PW})$ 加密 $Ticket_v$ 和 $K_{C,v}$ 得到一张票据信息 $E_{H_0(g^{N_1 N_2 + N_2 PW})}(Ticket_v, K_{C,v}, URL)$ 。票据发放模块生成所有的票据信息后, 发送这些信息给客户端代理。

客户端代理收到票据信息后, 票据管理模块处理所有的票据信息, 具体处理其中一张票据信息的步骤为: 首先利用会话密钥 $H_0(g^{N_1 N_2 + N_2 PW})$ 解密 $E_{H_0(g^{N_1 N_2 + N_2 PW})}(Ticket_v, K_{C,v}, URL)$ 得到 $Ticket_v$ 、 $K_{C,v}$ 以及 URL, 最后把 URL、 $Ticket_v$ 和 $K_{C,v}$ 作为一条记录存放到临时数据库。

2.5 客户端代理处理 HTTP 请求和服务器端代理验证 HTTP 请求

客户端代理处理完所有的票据信息后, 监听浏览器发送的每个 HTTP 请求, 具体处理 HTTP 请求的步骤: 分析得到请求访问的目标应用服务的 URL, 检查临时数据库是否有对应该 URL 的记录, 没有则快速转发该请求, 如果存在对应记录, 则取出记录信息, 用 $K_{C,v}$ 加密 ID_C 、 AD_C 以及即时时间 TS 生成 $E_{K_{C,v}}(ID_C, AD_C, TS)$, 把验证字段 ($Ticket_v, E_{K_{C,v}}(ID_C, AD_C, TS)$) 加到 HTTP 请求中, 再转发该请求。

服务器端代理收到 HTTP 请求后, 得到请求访问的应用服务的 URL, 查看应用服务信息数据库中是否有对应该 URL 的记录, 如果没有则快速把请求转交给 Web 应用服务器(表示该应用服务的访问不受限制), 存在相应的记录则取出记录信息, 同时从请求中取出验证字段(如果请求中没有验证字段, 则直接返回给用户无权访问的错误信息), 然后利用 K_v 解密 $Ticket_v$ 得到信息 $K_{C,v}$ 、 ID_C 、 AD_C 、URL、Lifetime, 检查这些信息的正确性, 包括 IP 地址是否相符, 根据 Lifetime 查看 $Ticket_v$ 是否在有效期内等, 接着用 $K_{C,v}$ 解密信息 $E_{K_{C,v}}(ID_C, AD_C, TS)$, 根据 TS 检查验证字段的即时性, 防止重放攻击。一切检查正确, 则去掉请求的验证字段, 把请求快速转交给 Web 应用服务器, 否则, 返回给用户无权访问的错误信息。

3 系统实现的关键技术

系统基于 Java 语言实现^[6], 实现过程中使用的部分关键技术简单介绍如下。

认证服务器的用户认证模块可能需同时处理多个认证请求, 所以其实现基于线程的方式: 即一旦接受到一个认证请求, 则创建一个单独的线程进行相应的处理。由于每个线程的创建、启动以及退出后的清理工作都需要一定的时间, 为了提高系统对认证请求的响

(下转第 169 页)

商务、电子证券、电子银行等方面,而用户对这些概念的普遍接受尚需一个时间过程;三是目前对于黑客攻击、网站被黑等报道太多,有的几乎是言过其实,这无疑给用户造成了一定的误导,也在一定程度上造成了对 PKI 的不信任。

3)现有 PKI 使用对个人隐私有较大的侵害。PKI 的设计往往忽略了这些问题,CA 在为用户颁发证书时,必须要对此人的信息进行登记,利用个人证书,PKI 可以追踪到每一个用户的每一步操作和通信,尤其是属性证书引入到 PKI 中以后,更增加了个人隐私暴露的危险性。PGP,PEM 将证书放在公共域中,证书的发布可能导致消息的泄漏。证书中的任何唯一性消息都可能被滥用,从而对证书持有者造成极大的伤害。

4)赔付机制的缺失造成用户利益遭受损失时,却没有一种有效的赔付机制来理赔和索保,这是国内认证机构在建立信誉时面临的问题。在国外,认证机构的赔付机制大多是依靠保险,许多国家都开设有网络保险业务,但中国目前还没有这一险种。

2 结束语

任何事物的发展都有正反两个方面,PKI 技术亦是如此。PKI 是网络社会秩序的维护者,目前的网络

(上接第 165 页)

应速度,采用了线程池的实现技术:即主程序开始启动时预先创建一定数目的线程集合,接受到一个认证请求,则从线程池中取出空闲的线程进行相应的处理,处理完毕后,再把线程放进线程池。这样就节省了线程创建、启动以及退出等需要的一系列开销。

认证服务器用户信息数据库的结构为 userinformation(name, password, exponent, visitrange),其中 visitrange 表示用户的访问权限,网站所有应用服务是根据其 URL 来区分的,如果在该数据库中存放所有用户可以访问的应用服务的 URL,那么无疑需要大量的存储空间。系统具体实现时,visitrange 表示为二进制 512 位,其中每个二进制位对应一个应用服务,如果该应用服务在用户的访问权限范围内,则设置为 1,否则为 0,这样 512 位就可以表示 512 个不同的应用服务,对于一般的小型网站是足够的。

4 结 论

针对目前网站身份认证系统中存在的问题,设计了一种新 Web 网站统一口令认证系统,使用统一认证

安全应用已越来越多地使用或离不开 PKI 技术的支持,正因为如此,尤其应该客观地分析和想方设法去面对和克服 PKI 的缺陷。通过对 PKI 安全性实事求是的分析,希望能引起人们对 PKI 的全面认识。

参考文献:

- [1] Nash A, Duane W, Joseph C, et al. 公钥基础设施(PKI):实现与管理,电子安全[M]. 张玉清,陈建奇,杨波,等译. 北京:清华大学出版社,2002.
- [2] 沈昌祥. 谈我国的 PKI 建设[J]. 中国信息导报,2002(9): 53-54.
- [3] 于洋,王戟,陈晓桦. PKI 的使用脆弱性及对策[J]. 计算机工程与科学,2003,25(2):27-30.
- [4] Fox B, LaMacchia B. Certificate Revocation: Mechanics and Meaning[C]// Proc: Financial Cryptology - CRYPTO'98. LNCS 1465. Berlin: Springer-Verlag,1998:158-164.
- [5] 王建业,周振国,陈森发. Internet X. 509 PKI 深入讨论与分析[J]. 计算机应用研究,2003(2):97-99.
- [6] 管海明,任朝荣. PKI 缺陷分析及新一代 PKI 的要求[J]. 计算机安全,2004(1):13-15.
- [7] 赵富强. PKI 技术的发展与隐忧[J]. 计算机安全,2004(3): 14-15.
- [8] 张秋余,梁爽,王利娜. PKI 的发展及问题分析[J]. 微机计算机信息,2006,22(2):39-41.

技术之后,用户登录一次即可访问网站权限范围内所有的应用服务,消除了多次登录带来的烦琐。另外,由于用户发送的每个 HTTP 请求,服务器端代理都根据其的身份验证字段检验其合法性,相比传统的根据 IP 地址或者 Cookie 值识别请求的认证方式,安全性高,可有效抵御 IP 地址欺骗和截取重放攻击(窃取 Cookie 信息)。

参考文献:

- [1] William S. 密码编码学与网络安全——原理与实践(英文影印版)[M]. 第4版. 北京:电子工业出版社,2006.
- [2] Fielding R. Hypertext Transfer Protocol——HTTP/1.1(rfc 2068)[EB/OL]. 1997. <http://rfc.net/rfc2068.html>.
- [3] 陈开渠. 基于口令的认证:协议和应用[D]. 北京:中国科学院软件研究所,2000.
- [4] 黄叶珏,陈勤. 一个新的动态口令认证方案[J]. 计算机工程与设计,2005,26(7):1735-1736.
- [5] 赖溪松,韩亮,张真诚. 计算机密码学及其应用[M]. 北京:国防工业出版社,2001.
- [6] Bruce E. Java 编程思想[M]. 第3版. 北京:机械工业出版社,2005.