一种 B/S 模式一次性口令系统①

方 俊

(浙江越秀外国语学院 网络传播学院, 绍兴 312000)

摘 要: 由于 B/S 模式下客户机对服务器的验证困难, 提出了使用.NET Remoting 程序对网站的真伪进行验证, 并使用神经网络方式实现了服务器端的认证信息的安全存储, 客户端和服务器端的相互认证和每次认证产生不同的会话密钥, 安全分析表明, 所提的方案能有效防范钓鱼网站等的常见攻击, 增加了系统的安全性. 方案使用 ASP.NET 实现, 证明是可行的.

关键词: 身份认证; .NET Remoting; Hash 函数; 神经网络

One-Time Password Authentication System Based on B/S

FANG Jun

(College of Network Communication, Zhejiang Yuexiu University of Foreign Languages, Shaoxing 312000, China)

Abstract: In the condition of the B/S mode, the authentication server is difficult implemented by a client. This paper puts forward to use .NET Remoting program to verify the authenticity of Web site and reaches storage safety of authentification message by useing neural network weight and also realizes mutual authentication between the client and the server, every time the authentication produce different session keys so on. Safety analysis shows that the proposed scheme can effectively prevent fishing websites against common attack, increase the system security. The scheme is achieved by using ASP.NET, It proved to be feasible.

Key words: authentication; .NET Remoting; Hash function; neural network

身份认证技术是信息系统安全中的重要技术, 基于口令的身份认证技术具有简单、易用的特点. 为了解决口令传送的安全性问题, 在上世纪 80 年代初, 美国科学家 Leslie Lamport[1]首次提出了利用哈希(Hash)函数产生一次性口令(简称 OTP)的设想. 到 90 年代, Bellcore 建立了"S/Key"一次性口令系统[2], 但"S/Key"一次性口令系统存在多种安全缺陷[3,4]. 此后人们提出基于令牌、IC 卡、口令卡和软件方式的一次性口令系统, 但这些方案大部分都是基于 C/S 应用而提出的, 而基于 B/S 的方案, 客户端一般是使用浏览器, 实现方式上有自己的特点. 文献[5, 6]实现了使用 B/S 方式进行口令认证, 但这些方案均未对服务器的真伪进行验证, 存在着较大的安全隐患. 目前国内流行的一次性口令的应用方式是使用动态口令牌, 包括硬件令牌、短信密码、手机令牌. 这些动态口令牌无需在 PC 端安装软件, 也无需和 PC 机连接, 动态口令生成是根据专门算法, 基于时间同步、基于事件同步和基于挑战/应答模式. 但这种方式容易遭到钓鱼网站通过中间人的方式攻击[7].

B/S 的特点是客户端一般是使用浏览器, 无需设计专门的客户端. 在使用时, 只需要打开浏览器, 输入正确的域名地址, 便可从服务器端下载可与用户交互的页面, 客户端的浏览器只是作为一个交互界面, 在客户端一般不进行复杂计算. 而一次性口令的很多计算任务是需要在客户端完成, 因此可以使用 JavaScript 实现. 由于 B/S 模式下客户机对服务器的验证困难, 这给网络钓鱼攻击者带来了可趁之机, 虽然可以使用证书颁发机构向用户颁发证书, 但这需要公

① 收稿时间:2012-01-17;收到修改稿时间:2012-05-12

钥基础设施, 并且其设置和维护成本较高. 网站一旦遭受到网络钓鱼攻击, 即使用户使用一次性口令也难以防范, 用户输入的个人信息将全部被攻击者获取. 本文使用.NET Remoting 技术先对网站的真实性进行验证, 再输入用户口令, 这样可以避免口令的泄漏和网络钓鱼的攻击, 同时所使用的口令一次有效, 实现了双向认证, 安全性分析表明本方案是安全的. 方案同时使用了神经网络技术对认证信息进行存储, 提高了安全性. 最后, 对所提方案使用 ASP.NET 技术和神经网络技术加以实现, 证明是可行的.

1 .NET Remoting和Matlab C#编程技术概述

.NET Remoting[8]是 Microsoft .NET 进行分布式开发面向对象的应用体系, 是.NET 平台上在不同应用程序域之间通信的技术, 是 DCOM 的升级. 在一个应用程序域中的进程与另外一个应用域中的进程通信可以使用 Remoting 技术. Remoting 的分布式处理程序主要包括三个部分: 远程服务对象、服务端和客户端. .NET Remoting 体系结构如图 1 所示.

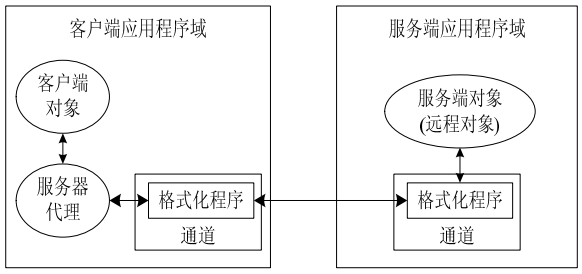

图 1 .NET Remoting 体系结构

Matlab 是矩阵实验室(Matrix Laboratory)的简称. 在国际学术界, Matlab 已经被确认为准确、可靠的科学计算标准软件. 本文的 RBF 神经网络计算使用 Matlab 编程实现, 把在 Matlab 中编写的 m 文件编译成动态链接库 dll. Matlab 可以把 m 代码文件编译成 dll, 则函数的输入、输出参数都是 MWArray 对象. 因此在使用中必须引入 MWArray.dll. 具体步骤[8]:

① 编写函数形式的 M 文件;

② 将 m 文件生成 dll 文件;

③ 在网站中添加 m 文件生成的 dll 和 MWArray.dll 的引用.

这样在网页的业务逻辑层就可以脱离 Matlab 环境, 直接引用 dll 文件实现所需用的功能.

2 方案描述

2.1 符号约定:

U: 客户端;

S: 服务器端;

ID: 用户的身份标识;

PW: 用户口令;

Se: 服务器选择的种子;

R: 随机数; 其中 R_u 为客户端产生的随机数, R_s 为服务端产生的随机数

//: 联结运算符;

⊕: 异或运算符;

H(x): 安全哈希函数;

$H^2(x)$: 对 x 连续进行 2 次哈希运算;

U→S:x: U 向 S 发送消息 x;

2.2 口令方案:

本方案包括注册和登录认证两个阶段.

2.2.1 注册过程

用户在安全的环境下请求注册, 用户选择ID、PW; 服务器选择 Se, 计算 H(ID)、H2(ID)、H(PW//Se), Se⊕H(ID//PW)、H(Se//ID//PW)、H(ID//PW//Se), 将计算结果进行 Reed-Solomon 编码[9], 以 H(ID)、H^2(ID)、H(PW//Se) 作为神经网络的输入, 对应的 Se⊕H(ID//PW)、H(Se//ID//PW)、H(ID//PW//Se)作为神经网络的期望输出, 使用 MATLAB 的 newrb()构建神经网络, 并将文件以 H(ID).mat 保存在服务器端的专门文件夹中. 客户端则需安装简单的 Remoting 验证程序, 完成注册.

2.2.2 认证过程

本方案的身份认证结构图如图 2 所示.

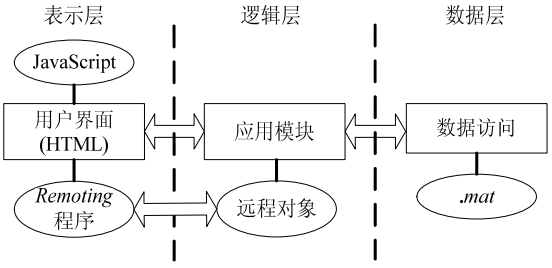

图 2 身份认证结构图

① 用户在浏览器中输入 ID, 计算 H(ID)、Ru、$H(ID//R_u)$, 并将 $H(ID//R_u)$ 保存在客户端.

U→S:H(ID)、$H(ID//R_u)$, 认证请求;

② S 查找 H(ID)，若该 H(ID)不存在，S 断开与 U 会话；否则，S 将 $H(ID//R_u)$保存在服务器端，S 查找 H(ID)对应的神经网络文件(H(ID).mat)构建神经网络，并计算 H2(ID)；将 H(ID)和 H2(ID)作为输入参数调用 DllRBF.dll，则对应的输出为 $Se \oplus H(ID//PW)$、$H(Se//ID//PW)$；服务器端生成随机数 R_s，并计算 $Rs \oplus H(Se//ID//PW)$；将 $C_1= Se \oplus H(ID//PW)$、$C_2= Rs \oplus H(Se//ID//PW)$作为挑战发送给 U.

S→U:C_1、C_2.

③ 启动客户端 Remoting 程序，访问指定 URL 的远程对象．若客户端的 $H(ID//R_u)$与指定服务器上的 $H(ID//R_u)$一致，服务器为真．否则为假.

当客户端弹出窗口说明服务器为真时，用户可在浏览器中输入 PW，并计算: H(ID//PW), Se =$H(ID//PW) \oplus C_1$, H(Se//ID//PW), H(ID//PW //Se), $R_s = H(Se//ID//PW) \oplus C_2$;

根据上面的计算结果，计算: H(PW//Se)、$C_3=R_s \oplus H(PW//Se)$、$C_4=R_s \oplus H(ID//PW //Se)$;

U→S:C_3、C_4.

④ S 收到 C_3 后，计算 $C_3 \oplus R_s$ 得到认证依据 H(PW//Se)，以 H(PW//Se)为输入参数调用 DllRBF.dll，其输出为 H(ID//PW//Se)；计算 $C_4 \oplus Rs$，若结果与神经网络输出一致，则用户合法．若不相同，则为非法用户，S 断开会话.

⑤ 在接下去的 U 和 S 的相互通信中，使用双方共有的 R_s 值作为会话密钥进行加密通信.

3　系统实现

系统使用 VS.NET 开发工具，服务器端使用 Internet 信息服务器(IIS)作为 Web 服务器，客户端使用 IE6.0 或更高版本的浏览器.

用户登录系统时在浏览器中输入用户名点击”确定”后，浏览器通过嵌入在 HTML 页面中的 JavaScript 代码计算 H(ID)、$H(ID//R_u)$．数据传递使用 Querystring 方法，服务器端的业务逻辑层收到表示层的请求和数据以后，业务逻辑层生成随机数 R_s，将其保存到 Session 中，调用 DllRBF.dll，调用代码如下:

```
DllRbfCacu myDLL = new DllRbfCacu(); //定义对象
MWArray SeXORHashIDPW = myDLL.DLL_RBF((MWArray )UserHash); //调用自定义 DLL_RBF()方法，实现神经网络输出.
MWArray HashSeIDPW = myDLL.DLL_RBF((MWArray )User2Hash);
string SeXORHashIDPW = SeXORHashIDPW.ToString(); // MWArray 类型转换为字符串类型.
string HashSeIDPW = HashSeIDPW.ToString();
```

在向表示层返回结果时使用 Cookie 将 C_1、C_2 信息保存到客户端．接着提示用户在输入用户密码前启动 Remoting 客户端程序验证服务器的真伪.

本文方案用户登录系统时的使用习惯基本上没有改变，只是在输入密码前加入验证服务器的步骤，如图 3 所示．如果返回”网站为真！请继续输入密码！”的窗口时，则可以在页面继续输入密码．该窗口与 Windows 的三种对话框是完全不同的，对话框的用户界面是脚本编写者不能修改的，且现在浏览器均不支持 windows.open()方法的 titlebar 属性[10]，即该窗体无法通过脚本的方式创建，如图 4 所示．如果弹出如图 4 的窗口，则用户可以在页面上输入口令登录网站.

图 3　用户登录界面

图 4　服务器验证成功时弹出的提示窗体

点击网页中的”验证服务器”按钮，则网页中的 Javascript 代码即可启动 Remoting 程序．启动 Remoting 程序的 Javascript 代码如下:

```
//启动 Remoting 程序
function exec ()
{
var command = document.getElementsByName("path")[0].value  //path 为 Remoting 程序的路径
```

```
var wsh = new ActiveXObject('WScript.Shell'); //创建客户端程序执行对象
if (wsh)
wsh.Run(command);      //调用 Run 命令执行 exe 程序, 参数是一个 exe 文件的路径
}
```

Remoting 程序包含客户端程序和远程服务对象, 宿主服务器选择用 IIS 来承载远程对象. Remoting 远程对象是 Remoting 系统的核心, 这里所实现的功能只是简单的读取服务器端的 H(ID//Ru)值并删除. Remoting 客户端与远程对象之间的通信是一个相互传递数据的过程, 数据从客户端传递到远程对象和获取远程对象的数据都是通过客户端调用远程对象的方法实现.

4 安全性分析

方案在安全的环境下注册, 并将认证所需要的信息以神经网络的 mat 文件保存. 由于 RBF 神经网络的优良性能, 训练网络时设置的目标误差小, 对于神经网络的特定输出值是不可能构造其输入值. 网上传输的认证信息是通过随机数 R_s 加密的, 用户名和口令等未在网络中传输, 即使攻击者窃取了 mat 口令文件也无法构造出神经网络的输入输出的值, 也无法进行穷举攻击, 因此比单纯的哈希比对验证具有更高的安全性.

重放和小数攻击: 浏览器和服务器之间的数据传递除 H(ID)外均有随机数加密, 每次登录时所传递的值均不相同. 其中, U 发送给 S 用于认证的 C_3、C_4 中含有随机数 R_s 且每次的值均不同, 认证依据 H(PW//Se)没有在网上直接传输, 因此本方案能抵御重放攻击. 同时, 由于本方案不是基于哈希的口令序列, 因此本方案不存在小数攻击.

冒充攻击: 冒充合法用户. 当攻击者截获了用户登录时的信息 H(ID), 再发送给服务器, 服务器返回 C_1 和 C_2. 由于用户口令不在网上传输, 也不在任何系统中保存, 攻击者无法计算 H(ID//PW)获取 Se 和 R_s, 因此, 就无法计算认证依据 H(PW//Se), 也无法从 C_3、C_4 获得该值, 冒充用户失败.

攻击者冒充服务器. 用户登录假冒的网站后, 用户可能输入用户标识 ID, 此时用户启动 Remoting 程序, 该程序将与真实的服务器建立通道读取 H(ID//R_u), 并从客户端读取 H(ID//Ru)进行比较, 但此时真实的服务器端并无 H(ID//R_u)值存在, Remoting 程序无法完成对网站的验证, 同时, 假冒的网页也无法通过脚本代码方式制作出”提示窗体”提示用户继续输入密码, 冒充服务器失败, 因此本文所提的方案可以有效地防止钓鱼网站的攻击.

如果攻击者窃听了认证过程中的 C_3、C_4, 此时攻击者使服务器崩溃, 则服务器重启后恢复到认证前状态, 攻击者可重放 C_3、C_4 登录系统. 但在其后的通信中必须使用会话密钥 R_s 进行加密通信, 而攻击者无法获得 R_s 值, 而且每次认证产生的随机数 R_s 值不同, 通信无法继续. 因此本方案可以抵御服务器崩溃攻击和会话劫持攻击.

在本方案中, 服务器端的认证依据 H(PW//Se)是存储在神经网络的权值中, 攻击者即使攻破服务器获取 mat 文件也是无法获得认证依据 H(PW//Se). 由于认证过程中有随机数 R_u、R_s 参与, 没有明文信息在网络上传输, 并且在合法的通信连接建立以后进行加密通信, 因此, 攻击者无法实施中间人攻击.

5 结语

本文所提的方案实现了使用 Remoting 技术对网站真伪的鉴别和使用神经网络的方式存储认证所需要的信息, 实现了客户端和服务器端的双向认证, 能有效地防止了重放攻击和钓鱼网站的欺骗攻击等常见攻击. 方案的计算开销小, 认证依据 H(PW//Se) 的神经网络输出值只需与服务器端的计算值 H(ID//PW//Se)进行比对就可完成认证, 用户名和密码不在系统存储, 而由神经网络权值方式存储认证所需的信息, 提高了系统的安全性. 由于认证时只进行简单哈希计算和神经网络的加法乘法运算, 避免了目前一些方案在认证时使用的幂指数运算, 减少了系统的开销.

参考文献

1 Lamport L. Password authentication with insecure communication. Communications of the ACM, 1981,24(11):770–772.

2 Haller NM. The S/ Key One-time Password System. Proceedings of the Internet Society Symposium on Network and Distributed System Security, San Diego,CA,USA:[s.n.], 1994.

(下转第 78 页)

本一中存在的问题全部消除，取得了良好的效果.

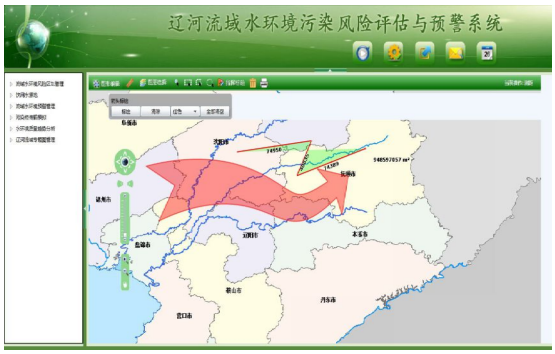

图 2　WebGIS 系统功能执行测试效果

4　结语

针对 WebGIS 系统开发过程中遇到的事件管理问题，本文提出了一种对事件进行自动化管理的事件管理器结构，通过在辽河流域水环境管理信息化系统中的应用表明，使用该事件管理器对地图事件进行管理，简化了程序开发过程中的复杂事件操作，减少了错误事件响应的发生，提高了系统运行效率. 实践证明，本文事件管理器的设计思想对 WebGIS 的开发具有极大的意义，同时，该事件管理器设计思想对于桌面 GIS 和普通应用程序开发的事件管理也有很好的参考作用. 由于作者本身水平所限，设计的事件管理器结构必然存在很多不足，今后作者会继续深入研究事件管理机制，争取使其得到更好的改进.

参考文献

1 玉伟,刘艳艳.基于 WebGIS 的居民信息系统的设计与实现.电脑应用技术,2010,77:18–24.
2 兰天,曲鹏东,孙高飞,等.Flex 企业应用开发实战.北京:机械工业出版社,2010.1–10.
3 丘威.基于 X3D 的富客户端 WebGIS 应用研究.微电子学与计算机,2011,1(1):157–159.
4 陈谦,余江峰,潘森,等.基于 RIA 方式的 WebGIS 构建.遥感信息,2009,8(4):89–94.
5 Watson K, Nagel C. 齐立波,等译. C#入门经典. 4th ed.,北京:中国书籍出版社,2008.298–300.
6 王颖..NET 中对象序列化方法.计算机与信息技术,2008,5:54–57.

(上接第 127 页)

3 Mitchell CJ, Chen L. Comments on the S /KEY User Authentication Scheme. ACM Operating System Review, 1996,30(4):12–16.
4 李世平,李凤霞,战守义.S/key 认证系统的安全缺陷及改进.计算机工程,2003,29(20):18–19.
5 杨明,罗军舟.基于 OTP 的安全 Web 登录系统的设计与实现.计算机工程,2003,(29):56–58.
6 雷超,雷劲.基于脚本级的一次性口令系统的实现.计算机应用,2001,21(7).
7 王贵智.网银动态口令牌应用的安全性.中国金融电脑,2010,11:29–31.
8 Nagel C, Evjen B, Glynn J. C# 高级编程.北京:清华大学出版社,2006.
9 Reed IS, Solomon G. Polynomial Codes Over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics, Jun. 1960,8(2):300–304.
10 Goodman D, Morrison M. JavaScript 宝典.第 6 版.北京:人民邮电出版社,2009.