文章编号:1005－0523(2005)02－0055－05

基于 SHA 的一次性口令认证技术

汤鹏志[1],李黎青[1],左黎明[2]

(1.华东交通大学,江西 南昌 330013;2.江西师范大学,江西 南昌 330027)

摘要:针对网络信息系统的口令验证的安全问题,提出一种基于 SHA 和一次性口令验证方案,实践证明该方案具有很高的安全性和实用性.

关 键 词:SHA;一次性口令;信息安全工程
中图分类号:TP393.08　　**文献标识码**:A

1 引 言

目前,随着人们生活信息化水平的提高,网上支付、网上划帐、网上金融交易行为随着电子商务的展开越来越普及,大量重要的数据存储在网络数据库中,并通过网络共享,为人们的生活提供了方便,但是也带来了巨大的信息安全隐患和金融风险.为了保证网络信息系统的安全性,我们除了要规范信息系统的管理,同时还需要对现有的系统进行安全加固.通过参阅近年来大量的黑客教程、黑客技术文章和攻击日志,我们分析了针对网络数据库系统的黑客攻击行为的主要手法和技巧,黑客攻击主要技术方法有以下几种:缓冲区溢出技术、木马技术、计算机病毒(主要是宏病毒和网络蠕虫)、分布式拒绝服务攻击技术、暴力破解－穷举攻击(Brute Force)、sniffer 报文截获.

在大部分的黑客技术文献和攻击日志中我们发现一个很重要的相似特征:几乎没有多少攻击行为是针对于协议和密码学算法的,窃取系统口令文件和偷听网络连接获取用户 ID 和口令是最常见的攻击方式,大部分攻击的主要方向和目的是得到用户 ID 和用户密码,其中以各种形式的木马对用户的口令安全威胁最大.目前黑客的木马编写技术水平比我们想像的要更加高明的多,尤其是基于多线程的反弹端口式木马.而且现在这类木马做得越来越隐蔽,与系统进程捆绑,一般的杀毒程序和入侵检测系统对其无效.当数据库服务器和数据库应用服务端被注入木马后,现有的所有安全保密措施形同虚设,木马将把键盘输入的用户名和密码记录下来以电子邮件或其它方式发送给攻击者.只要获得了用户 ID 和密码,所有敏感数据将暴露无遗,因此用户登陆验证方法有待于改进.但遗憾的是目前大多数网络信息工程都没有很重视口令的安全性,而且相关的工程实践文献和理论文献都比较少.在文献[1]中,Ross J.Anderson 比较系统地归纳了一些常见的问题,例如:口令劫持、端口监听、中间人攻击、木马技术、缓冲区溢出、分布式拒绝服务攻击等,并提出了一些解决办法:采取客户端加密、随机密钥、端到端的加密等方法.另外 Roger Needham 等人也提出了一些与口令验证协议相关的理论:例如挑战－应答机制、零知识证明.

2 基于 SHA 的一次性口令认证技术

消息摘要是一种验证消息完整性的技术.消息

收稿日期:2004－10－20
作者简介:汤鹏志(1961－),男,江西九江人,副教授,主要研究方向:信息安全工程.

摘要获取消息作为输入并生成位块(通常是几百位长),该位块表示消息的指纹,是一个定长的输出序列.消息中很小的更改(比如说,由闯入者和窃听者造成的更改)将引起指纹发生显著更改.消息摘要函数是单向函数(Hash 函数).从消息生成指纹是很简单的事情,但生成与给定指纹匹配的消息却很难.大多数强函数使用散列法,比较常见的消息摘要算法有 MD^5 和 SHA(包括 SHA－1、SHA－256、SHA－512 等).

2.1 基于 SHA 的分权式一次性口令认证基本原理

为防止入侵者得到用户 ID 和口令,一个比较强的口令系统必须能够防范以下几类攻击:

第一类:通过木马得到用户 ID 和口令的攻击;第二类:通过报文截获得到用户 ID 和口令的攻击;第三类:各种形式的重放攻击和中间人攻击.针对以上要求,我们提出了基于 SHA 和一次性口令验证技术,主要应用于网络数据库系统平台,其主要原理如下:

在客户端用户输入用户 ID 和密码 UserPass 的时候,客户端自动产生一个足够长度的随机序列 RndStr,同时获取当前客户端系统时间 TimeMark,然后按照的一定格式(用户 ID＋用户密码 UserPass＋随机序列 RndStr＋时间戳 TimeMark)组合成一个新的字符序列 IDInfo,将该序列通过 SHA 算法得到一定长的值输出 IDInfoMark,然后将用户 ID 、定长的值输出 IDInfoMark、随机序列 RndStr 和时间戳 TimeMark 按照格式(ID ＋IDInfoMark ＋RndStr ＋TimeMark)利用 AES 算法加密,加密后通过网络发送给远程数据管理系统主机,远程主机对得到的数据进行解密.解密后分离出用户 ID 、定长的值输出 IDInfoMark、随机序列 RndStr 和时间戳 TimeMark,通过用户 ID 从数据库得到真正的用户密码 password,将用户 ID、真正密码 password、随机序列 RndStr、时间戳 TimeMark 组合成一个新的字符序列 PassInfo,将该序列通过 SHA 算法得到一定长的值输出 PassInfoMark,比较 PassInfoMark 和 IDInfoMark 的值,如果值相同则验证成功,否则失败.客户端部分原理如图 1,服务器端如图 2.

2.2 安全性分析

登陆时我们采用两段口令技术(一部分用键盘输入,另一部分利用经过随机置乱后的数字键盘输入),从而可以有效地防范第(1)类攻击(通过各种木马得到用户 ID 和口令的攻击).同时由于在整个登录过程中加入不确定因素,使每次登录过程中传送的信息都不相同,网上传送的口令只使用一次,攻击者就无法用窃取的口令来访问系统,并且传输时我们采用 AES 算法加密,这样可以有效地防范第(2)类攻击(通过报文截获得到用户 ID 和口令的攻击)和第(3)类攻击(各种形式的重放攻击和中间人攻击).除了能防止这三类攻击外,还能有效的防范字典攻击.

3 实现细节与关键代码

登陆系统时,数字键盘字符次序经过随机置乱,每次的顺序是不同的,两段口令由 PIN^1 和 PIN^2 组成,其中 PIN^2 只能由数字键盘输入.这样可以保证具有足够的强度保证登陆系统在木马存在的情况下也能安全的使用,同时也能防范字典攻击.

3.1 客户端的关键技术细节与说明(JAVA 描述)

(1)声明变量并进行初始化

```
String ID="";  //用户 ID
String UserPass="";  //用户密码,由 PIN1 和 PIN2 组合而成
Random rand=new Random();
Date da1=new Date();
String TimeMark="";  //时间戳
String IDInfoMark="";  //消息摘要
byte[] RandStr=new byte[20];  //随机序列
```

(2)将 ID＋UserPass＋RandStr＋TimeMark 生成消息摘要,保存在 IDInfoMark

```
rand.nextBytes(RandStr);  //将随机数传入随机序列
TimeMark=da1.toString();  //通过 Date 对象实例产生时间戳
//生成 MessageDigest 对象,选用 SHA－512 算法
MessageDigest m = MessageDigest.getInstance("SHA－512");
//将 ID 和 UserPass 转化成 UTF8 类型,并传递给 m 的 update 方法
m.update((ID＋UserPass).getBytes("UTF8"));
m.update(RandStr);  //将 RandStr 传递给 m 的 update 方法
m.update(TimeMark.getBytes("UTF8"));//将 TimeMark 传递给 m 的 update 方法
byte s[]=m.digest();//计算消息摘要
for( int i=0;i<s.length;i＋＋)
{
```

```
//得到 IDInfoMark
IDInfoMark+=Integer.toHexString((0x000000ff&s[i])|0xffffff00).substring(6);
}
```

(3)将 IDInfoMark、用户 ID、随机序列 RandStr 和时间戳加密后发送给服务器，需要注意的一点是 IDInfoMark、用户 ID、随机序列 RandStr 和时间戳之间必须用一定的符号隔开，例如：(IDInfoMark)000000000000(ID)000000000000(RandStr)000000000000(TimeMark)，该部分的相关代码在文献[2]中有专门论述，文献[3]中也有详细的说明.

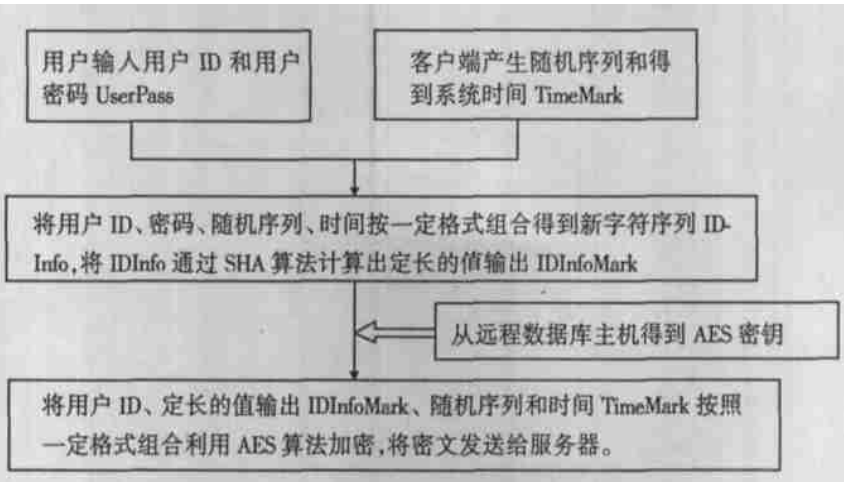

图 1　客户端原理

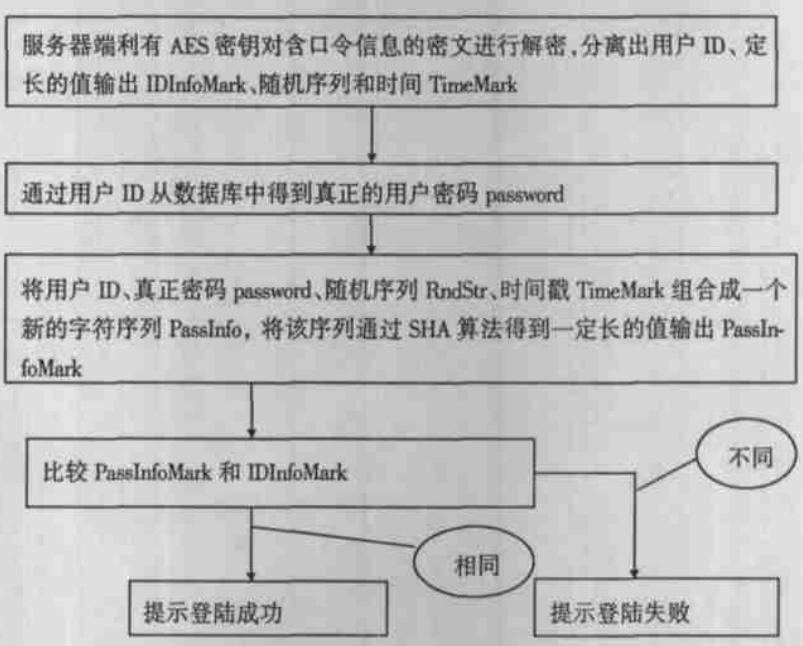

图 2　服务器端原理

3.2　服务器端的关键技术细节与说明(JAVA 描述)

(1)对客户端传送过来的用户登陆信息密文解密后分离出 IDInfoMark、用户 ID、随机序列 RandStr 和时间戳 TimeMark，我们在服务器端也要声明相应的变量.

```
/*由于服务器端可能有多个用户同时登陆，因此变量一般采用数组和向量类型，
为说明方便，我们仍然采用一般变量*/
String ID="";  //用户 ID
String password="";  //用户真正密码，从服务器中得到
String TimeMark="";  //时间戳
String IDInfoMark="";  //客户端生成的消息摘要
```

```
String PassInfoMark="";  //服务器端生成的消息摘要
String TxtSQL="";  //SQL 语句
Boolean FLAG;  //登陆是否成功标志
byte[] RandStr=new byte[20];//随机序列
(2)执行 SQL 操作,获得 password 的值.
/*在实际应用中,一般我们可以采用三层网络结构,关于用户的身份验证与数据操作可以单独用一个中间层应用服务来实现,为说明方便起见,我们采用 C/S 结构*/
String url="jdbc:odbc:fortune";  // fortune 是 ODBC 数据源
Connection con=null;
Statement sm=null;
ResultSet rs=null;
try
{ //加载 JDBC-ODBC bridge 驱动程序
Class.forName("com.sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(Exception e)
{
System.out.println("无法装载 JDBC-ODBC Bridge 驱动程序");
return;
}
//与数据库建立连接
try
{
con=DriverManager.getConnection(url);
sm=con.createStatement();  //创建对象

//执行 SQL 语句
TxtSQL="SELECT * FROM user-info WHERE user-id='"+ID+"'";
rs=sm.executeQuery(TxtSQL);
password=rs.getString(2);//获取用户真正密码
}
catch(SQLException e){ }
finally{
try{ //关闭对象
rs.close();
sm.close();
//关闭连接
con.close();}
catch(SQLException e){}
}
(3)将 ID+password+RandStr+TimeMark 生成消息摘要,保存在 PassInfoMark
//生成 MessageDigest 对象,选用 SHA-512 算法
MessageDigest m = MessageDigest.getInstance("SHA-512");
//将 ID 和 password 转化成 UTF8 类型,并传递给 m 的 update 方法
m.update((ID+password).getBytes("UTF8"));
m.update(RandStr);  //将 RandStr 传递给 m 的 update 方法
m.update(TimeMark.getBytes("UTF8"));//将 TimeMark 传递给 m 的 update 方法
byte s[]=m.digest();//计算消息摘要
for( int i=0;i<s.length;i++)
{
//得到 PassInfoMark
PassInfoMark+=Integer.toHexString((0x000000ff&s[i])|0xffffff00).substring(6);
}
(4) 比较 PassInfoMark 和 IDInfoMark
if (PassInfoMark.compareTo(IDInfoMark)==0)
FLAG=true;//如果相同,则登陆成功标志为 ture
else
FLAG=false;  //如果不相同,则登陆成功标志为 false
```

4 结束语

我们针对网络信息系统的口令验证的安全问题,提出一种基于 SHA 和一次性口令验证技术的解决方案,在远程实时股票信息管理系统证明了该方案具有很高的安全性和实用性,同时该方案抛开了抽象难懂的密码学和协议理论,易于被软件开发者掌握并应用.同时,这种方法可以应用至其它安全信息工程中去.

参考文献:

[1] (英)Ross J. Anderson 著,蒋 佳,等译. 信息安全工程[M]. 北京:机械工业出版社,2003.

[2] (美)Jess Garms 著,庞 南,等译. Java 安全性编程指南

[M]·北京:电子工业出版社,2002.
[3] Jianxin Yan, Alan Blackwell, Ross Anderson, Alasdair Grant· The memorability and security of passwords · some empirical results·UCAM—CL—TR—500[R]·University of Cambridge Computer Laboratory, September 2000.
[4] (美)Joseph J·Bambara 等,著,刘　　,等译· J2EE 技术内幕[M]·北京:机械工业出版社,2002.
[5] (美)Li Gong 著·JAVA 2 平台安全技术——结构、API 设计和实现[M]·北京: 机械工业出版社,2000.
[6] (美)John Bell Tony Loton· Java Servlets 2.3 编程指南[M]·北京: 电子工业出版社,2002.

A New Verification Technology Based on SHA and OTP

TANG Peng-zhi[1], LI Li-qing[1], ZUO Li-ming[2]

(1·East China Jiaotong University, Nanchang 330013; 2·Jiangxi Normal University, Nanchang 330027, China)

Abstract: To the password security of the network information system, this paper proposes a new kind of verification scheme based on SHA and OTP, Practice has proved that this scheme has very high security and practicality·
Key words: SHA; OTP; security engineering

(上接第 42 页)

The Effect of Polycarboxlic Series of High Performance Water Reducer On Fly Ash Concrete′Property and its Application

ZHAO Bi-hua[1], TAO Xiao-lin[2]

(1·School of Civil Engineering and Architecture, East China Jiaotong Univ·, Nanchang 330013; 2·Nanchang Construction and Engineering Group Company, Nanchang 330008, China)

Abstract: This paper discusses its action principle and analyses its technical characters when polycarboxlic series of high performance water reducer used in commercial concrete· Experience shows that polycarboxlic series of high performance water reducer owns some advantages such as much reducing water、better keeping slump and better fixing to cement· It has better market value in technology and economy·
Key words: polycarboxlic series of high performance water reducer; fly ash concrete; application