维普资讯 http://www.cqvip.com

一种瘦客户系统的安全性研究及改进

柳闻鹃[1,2],黄东军[1]
(1. 中南大学 信息科学与工程学院,湖南 长沙 410075;
2. 株洲工学院,湖南 株洲 412008)

摘 要:瘦客户机计算以网络通信技术为基础,它以其自身的特点得到了广泛的应用。在具体实现中,为了使客户端的计算尽可能简单,有的系统所采用的安全机制不够完善,存在明显的安全漏洞。文中以VNC(虚拟网络计算)为例,研究其提供的安全机制,分析存在的安全缺陷,并在此基础上提出增强其安全性能的措施。最后利用OpenSSL提供的密码算法库,给出了实现方法,提高了其安全性。

关键词:瘦客户机计算;VNC;安全机制
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2006)07-0135-03

A Security Study and Improvement of Thin Client System

LIU Wen-juan[1,2], HUANG Dong-jun[1]
(1. College of Information Science & Engineering, Central South University, Changsha 410075, China;
2. Zhuzhou Institute of Technology, Zhuzhou 412008, China)

Abstract: Thin client computing is based on network communication technology, it has been widely applied according to its characteristics. In order to insure its characteristic, there is obvious weakness in some of thin client system. In this paper, the research work emphasized on the security mechanism and security weakness of VNC. Based on this, one method of increasing security is presented by using cipher function provided by OpenSSL.

Key words: thin client computing; VNC; security mechanism

0 引 言

随着网络通信及操作系统多用户技术的发展,瘦客户机计算开始迅速流行,这主要是因为它具有成本低、集中管理、控制灵活、资源共享等特点。瘦客户机计算采用瘦客户/服务器的计算模式,其应用程序的执行和数据的处理在服务器端进行。而客户机仅执行少量的或者根本不执行应用、数据的处理和存储,从而保证了客户端的硬件配置要求低。瘦客户机计算的具体实现中常用的有:虚拟网络计算(VNC)、微软终端服务、Citrix的Metaframe等。而作为网络软件,瘦客户系统必须要保证网络和系统平台安全性。瘦客户系统,由于它的客户端的硬件配置低,所以不使用计算过于复杂的安全策略,以免影响其性能。有些系统为了达到这一要求,采用的安全机制不够完善,存在明显的安全漏洞。文中以VNC为例,研究其提供的安全机制,分析其存在的安全缺陷,并在此基础上提出增强其安全性能的措施。最后利用OpenSSL提供的密码算法库,给出了实现方法。

收稿日期:2005-10-21
作者简介:柳闻鹃(1967-),女,重庆人,硕士研究生,副教授,研究方向为网络计算和信息安全。

1 VNC安全机制分析

1.1 VNC工作原理

源自AT&T剑桥实验室的虚拟网络计算(Virtual Network Computing,简称VNC)以其真正的瘦客户技术、跨平台特性、开放代码和低带宽需要,得到了广泛应用。其核心功能是:在用户使用VNC客户端连接到运行VNC服务器的机器上时,客户机通过键盘和鼠标动作来执行存放在服务器上的应用程序,服务器桌面的快照被压缩并且通过其通信协议RFB(Remote Frame Buffers)协议发送到客户端[1],客户端与服务器之间的通信是通过架构在TCP/IP上的VNC协议(即RFB协议)来实现的。VNC协议基于Frame Buffer层,使其具有优秀的平台独立性,可以应用于所有的Windows系统(如:MS Windows,X Windows及Macintosh等)。由于该协议在客户端只要支持Frame Buffer的显示即可,所以它对客户端的硬件资源需求很少,是一种真正的瘦客户端协议。

VNC在Linux下的基本工作原理如图1所示。在客户端请求连接时,服务器端在通过对其身份验证后,把X server的桌面环境、输入设备和X资源交给VNC服务器掌控,这样X系统上的应用程序就通过X协议运行在VNC Server桌面会话中。VNC服务器将桌面环境通过VNC协议送给VNC客户端,让VNC客户来操纵VNC服

维普资讯 http://www.cqvip.com

务器桌面环境和输入设备,以实现在远程的 VNC viewer 上的显示和控制[2]。

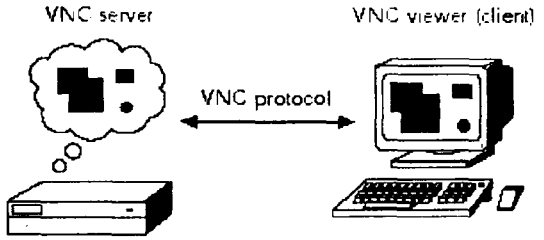

图 1 虚拟网络计算(VNC)的工作原理

1.2 安全机制分析

VNC 提供的安全机制仅限于客户的身份验证,所采用的方法是随机挑战响应(Random Challenge - Response)。在客户端请求连接的初始化阶段,双方协商了所要使用的 RFB 协议版本后,对客户的身份进行验证。这时服务器首先产生一个随机的 16 字节挑战并发送给客户机,客户机使用用户口令作为密钥用 DES 对挑战加密,并将加密后的密文作为响应返回到服务器,服务器用保存在本地的口令副本进行解密。若得到的明文与所发送挑战相符则允许客户与服务器之间建立会话,否则断开连接[3]。

VNC 除了对客户的身份验证,就没有再提供其他的安全机制了。在客户端,从客户输入到传送至服务器期间的数据包,对于每一次按下和释放键盘按键的操作,有两个包,一个告知服务器按下了什么键,而另一个包告知该按键已经被释放。鼠标动作的传输也采用类似的方式。而客户端的显示更新,是通过对服务器端的屏幕以压缩、但未加密的形式发送到客户机的。可见,当与服务器建立了连接之后,VNC 浏览器和 VNC 服务器之间的数据传输就完全是以未加密的明文方式传送的,这样传输的数据就可能被窃听,造成关键数据和秘密的泄漏。因此,要使 VNC 能够提供安全的应用,就必须在 VNC 客户和服务器之间建立安全的网络数据传输通道。

2 改进方案及实现方法

2.1 安全机制改进方案

通过以上分析,VNC 的安全本质上可以通过在 VNC 客户和服务器之间建立一个数据传输的安全通道来解决,它需要解决两方面的问题:对客户的身份确认和确保通道中数据传送的完整性和机密性。这可通过在系统中提供身份验证和会话密钥加密机制来实现。它要求每个会话由两个阶段组成。首先,通信双方运行一个密钥交换协议,建立一个双方共享的鉴别和秘密会话密钥;然后,会话密钥与对称密钥密码函数一道被使用,以保护欲传送数据的完整性和机密性。

据此,给出改善 VNC 安全性能的解决方案:

* 使用公开密钥加密算法 RSA 来进行身份认证和会话密钥的传送。

* 使用会话密钥做为 HMAC 密钥对欲传输的分组净荷求出校验和,以保证数据的完整性。

* 将求出的校验和附加在分组净荷后,两者合起来作为加密算法的明文,使用会话密钥做为 RC4 流对称算法的密钥来加密此明文,以保证数据的秘密性。

为实现上述方案,将 VNC 中的文件~/.vnc/passwd 改为一个存放 RSA 1024 位长的公钥/私钥对的密钥文件,且采用 PEM 格式,以方便 OpenSSL 中库函数的处理。其中私钥用 IDEA 算法加密,用客户口令作为密钥。会话密钥在服务器端随机产生,且用公开密钥来加密会话密钥并传送到客户端,客户端用私钥解码以建立客户和服务器之间的加密通讯。HMAC(Hash - based Message Authentication Checksum,基于散列的消息认证校验和)作为 Internet 标准的消息认证码[4]在此用来确保数据的完整性。将 RC4 流密码用于客户和服务器之间数据传送的加密,这是因为 RC4 算法的加密和解密速度均非常快且算法简单,对于一个瘦客户系统来说是很重要的,RC4 对称流密码作为一种广泛应用于数据通信信道中传输的数据流的加解密算法,虽然对其算法有一些攻击方法,但没有哪一种方法对于攻击足够长密钥(如 128bit)的 RC4 算法有效[4]。求校验和和加密的顺序问题,被称为确定鉴别和加密次序,采用不同的加密算法,其次序的选定影响所建立的通道的安全性,对于使用流密码建立安全通道,必须使用鉴别后加密的方法才是安全的[5]。根据该方案所设计的安全协议如图 2 所示。

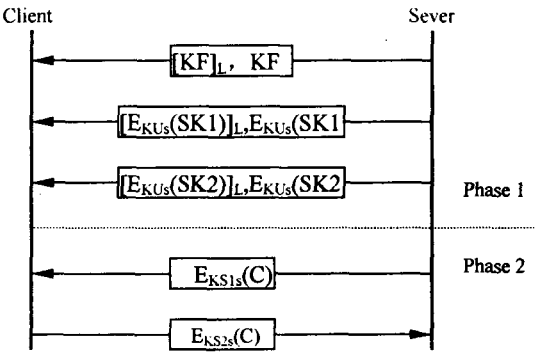

图 2 安全协议

该安全协议是在系统初始化握手阶段,双方协商了所要使用的 RFB 协议版本后开始的。它分为两个阶段,在阶段 1 使用公开密钥加密算法,客户从服务器获得私钥并建立共享会话密钥。

(1)服务器将密钥文件的长度$[KF]_L$及密钥文件 KF 发给客户,客户使用用户口令作为密钥,解码密钥文件中使用 IDEA 加密的 RSA 私钥,即:$D_P(KR_{KF})$。

(2)服务器将已用 RSA 公钥加密后的会话密钥 1 的长度$[E_{KUs}(SK1)]_L$及会话密钥 1 $E_{KUs}(SK1)$发给客户,客户读出会话密钥 1 并用 RSA 私钥解码,即:$D_{KR}[E_{KUs}(SK1)]$。

(3)服务器将已加密的会话密钥 2 的长度$[E_{KUs}(SK2)]_L$及用公钥加密的会话密钥 2 $E_{KUs}(SK2)$发给客户,客户读出会话密钥 2 并用私钥解码,即:$D_{KR}[E_{KUs}$

维普资讯 http://www.cqvip.com

(SK2)]。

阶段 2 使用在阶段 1 建立起的共享的会话密钥 1 和会话密钥 2 作为加密密钥使用 RC4 算法,采用随机挑战响应机制对客户的身份进行验证。

(4)服务器将用会话密钥 1 加密的挑战 E1 = (E_{KS1s}(C)),客户用会话密钥 1 解码,即:D_{SK1}(E1)。

(5)客户用会话密钥 2 加密的挑战 E2 = (E_{KS2s}(C))发给服务器,服务器用会话密钥 2 解码挑战 E2,即:D_{SK2}(E2)。

若 D_{SK2}(E2) = C 则为合法用户,否则客户不能连接到服务器上。一旦客户登录成功后,所有客户和服务器之间要交换的数据块 x,均先用 HMAC 求校验和;尔后,将求出的校验和 HMAC(x)附加在 x 后,再用 RC4 将其加密即 E(x,HMAC(x))。密码文件在连接时由服务器送到客户端,比用户之间的相互拷贝要安全。这里挑战和会话密钥均为 16 字节的随机数。

2.2 实现方法

OpenSSL 是 SSL 协议的一个开源产品的具体实现。SSL 协议,即安全套接字层(Secure Socket Layer)协议,为网络应用层的通信提供了身份认证、敏感数据加密和保证数据完整性的数字签名服务,其主要目的是在介于 TCP/IP 和应用层之间,为网络环境中两个通信应用进程之间提供一个安全通道。OpenSSL 采用 C 语言开发,具备跨平台的性能,可移植性好,是进行网络安全开发的优秀软件包。整个软件包按功能可分成 3 个部分:密码算法库、SSL 协议库以及应用程序[6]。其中密码算法库存放在 crypto 目录下,是 OpenSSL 中最重要的部分,它包括密码算法(Cipher)、消息摘要算法(Digest)、I/O 系统以及数据库等。在 Linux 下编译后生成的库文件名为 libcrypto.a。

文中使用 OpenSSL 密码算法函数库实现以上方案。在 redhat linux9.0 机系统中,该函数库被安装在/usr/lib 目录中,包含这些函数的头文件在/usr/include/openssl 目录中。为实现上述方案须改写 VNC 原来的个别程序。OpenSSL 提供的 EVP 系列函数封装了加密库里所有的对称加密算法函数,其定义包含在"evp.h"里。使用统一封装 EVP 函数,具有简单且可移植性较高的特点。限于篇幅所限,以下仅给出求客户键盘输入缓冲区 buf 中数据的 HMAC 校验和及使用 RC4 加密所用的主要函数:

```
HMAC_Init(&ctx, sk1, n, EVP_md5());
/* 使用 sk1 和 MD5 算法初始化 HMAC 对应的结构 ctx */
HMAC_Update(&ctx, buf, readlen);
/* 将存放在 buf 中,长度为 readlen 数据传给结构 ctx */
HMAC_Final(&ctx, result, &dlen);
/* 使用 ctx 计算其 MAC,结果存放在 result */
HMAC_cleanup(&ctx);
/* 清除 ctx 并释放其占用的资源 */
EVP_BytesToKey (EVP_rc4(), EVP_md5(), NULL, sk2, m, 1, key, iv);
/* 使用 sk2 产生 rc4 加密所需的 key 和 iv */
EVP_CipherInit(&ectx, EVP_rc4(), key, iv, operation);
/* 初始化加密算法 rc4 对应的结构 ctx */
EVP_CipherUpdate(&ectx, ebuf, &ebuflen, buf, strlen(buf));
/* 将 buf 中明文和其 MAC 使用 ectx 规定的算法结构加密并存放在 ebuf 中 */
EVP_CipherFinal(&ectx, ebuf, &ebuflen);
/* 结束加密动作 */
```

ctx 和 ectx 分别为 HMAC 和 RC4 算法所使用的两个基本结构,该结构主要定义了与该结构相关的另外一个算法结构 EVP_CIPHER(它主要指出应该采用什么算法进行数据处理)、需进行处理的数据和处理后数据等。sk1 为会话密钥 1,作为 HMAC 的密钥,n 为 sk1 的长度,sk2 为会话密钥 2,作为 RC4 算法的密钥,m 为 sk2 的长度。iv,key 和 result 的长度在头文件 evp.h 中分别由符号常量 EVP_MAX_IV_LENGTH, EVP_MAX_KEY_LENGTH 和 EVP_MAX_MD_SIZE 定义,其大小均为 16 字节。根据流密码的特点,不能对两个明文采用相同的 key 和 iv。在每传送 4096 个数据包后,服务器会重新生成新的会话密钥。在 BytesToKey 函数中 iv 和 key 是根据所给 sk1,通过 hash 函数产生的,且 hash 次数可在函数中指定,次数越大,安全强度越大,但速度会降低,在此指定为 1。

3 结束语

随着网络经济在全球范围内的飞速发展,瘦客户机计算模式以其低运营费用、强支持率和低风险的特点,将成为未来网络计算的一种发展方向,而 VNC 这一典型的瘦客户软件,作为网络软件,它并没有提供足够的安全保证。它在使用中存在着安全隐患,严重威胁着网络和系统的安全。为使之能够应用于企业应用服务系统,必须增强 VNC 的安全性。文中提出了一种解决方案,并利用 OpenSSL 提供的加密函数库实现了该方案。

参考文献:

[1] Richardson T. The RFB Protocol[EB/OL]. http://www.uk.research.att.com/vnc,2004-05.

[2] Richardson T, Stafford-Fraser Q, Wood K R, et al. Virtual Network Computing[J/OL]. IEEE Internet Computing, 1998,2(1). http://www.uk.research.att.com/pub/docs/2005.

[3] NISCC. Securing VNC[EB/OL]. http://www.uk.research.att.com/vnc,2003-06.

[4] Schneier B. 应用密码学—协议、算法 C 源程序[M]. 吴世忠,祝世雄,张文政,等译. 北京:机械工业出版社,2000.

[5] Krawczyk H. The Order of Encryption and Authentication for Protecting Communications[R/OL]. http://eprint.iacr.org/2001/2001-09.

[6] Chandra P, Messier M, Viega J. Network Security with OpenSSL[M]. Calif, USA: O'Reilly, 2002.