第35卷 第18期 计 算 机 工 程 2009年9月
Vol.35 No.18 Computer Engineering September 2009
·安全技术· 文章编号：1000—3428(2009)18—0127—03 文献标识码：A 中图分类号：TP309.2

TDS协议分析与漏洞检测

郭丽红，吴海涛
(南京工程学院通信工程学院，南京 211167)

摘 要：对应用于Microsoft SQL Server 2000的未公开协议TDS 8.0进行研究分析，通过批量发送和接收TDS数据包，分析TDS协议的通信规则和各种类型的TDS包结构，总结整个TDS协议的报文结构。在协议分析的基础上，研究Microsoft SQL Server 2000体系结构和存在的安全漏洞，编写漏洞测试程序，验证TDS 8.0协议并找出其中存在的安全问题。

关键词：表格数据流协议；协议分析；漏洞检测

TDS Protocol Analysis and Loophole Detection

GUO Li-hong, WU Hai-tao
(School of Communications Engineering, Nanjing Institute of Technology, Nanjing 211167)

【Abstract】 This paper researches and analyzes the undisclosed protocol TDS 8.0 which used in the Microsoft SQL Sever 2000, receives and sends the TDS data packets, analyzes the communication rule of TDS protocol and all types of TDS package, sums up the entire TDS protocol structural framework and gives a detailed description of every TDS segment meaning. Basing on the protocol analysis, this paper researches the system structure and security loopholes of the Microsoft SQL Sever 2000, compiles the loophole test program to validate TDS 8.0 protocol and finds out the secure problems in the TDS 8.0 protocol.

【Key words】 Tabular Data Stream(TDS) protocol; protocol analysis; loophole detection

随着信息技术的迅速发展，数据库的应用越来越广泛。多数企业、组织以及政府部门的电子数据都保存在各种数据库中，这些数据关系到整个部门的生存与发展，所以数据的完整性和合法存取越来越受到人们的重视[1-2]。以前，人们只能通过几个互相隔离的复杂业务逻辑层来访问的数据库，现在则可通过Web应用环境直接对数据库进行访问，这种方式与以往的方式相比流动性更强，但安全性能降低，数据库更容易受到来自网络的攻击。由于数据都是通过网络传输的，这就可以在传输的过程中被截获或通过非常手段进入数据库。基于上述原因，详细分析数据库通信协议，因此，尽早发现其漏洞并做出相应的防范尤为重要。

1 TDS简介

表格数据流(Tabular Data Stream, TDS)协议的首字母缩写，它是SQL Server 2000客户端和服务器之间使用的语言。TDS协议描述了2个计算机之间如何传输数据的规则，它定义了传输信息的类型和传输的顺序，TDS协议是建立在TCP/IP Net-Library之上的，包含在TCP数据段内，用1433端口进行数据库通信。目前，SQL Server支持3种版本的TDS：(1)TDS8.0，适用于SQL 2000客户端；(2)TDS 7.0，适用于SQL Server 7.0客户端；(3)TDS4.2，适用于SQL Server 4.2、SQL Server 6.0和SQL Server 6.5客户端。完全支持所有SQL Server 2000功能的版本只有TDS8.0，其他版本则保持向后兼容。

2 TDS协议分析

2.1 TDS协议报文格式

TDS协议多数都未被公开，可能是处于商业秘密的考虑或是技术所有权的问题。TDS 8.0协议报文由8 Byte的TDS包头信息和协议单元的内容组成。标准的TDS包都有8 Byte的TDS头，TDS 8.0报头格式见图1[3]。

token	*status*	*length*	*signn*	*packetn*	*windows*

图1 TDS 8.0报头格式

*token*标志(1 Byte)，核心标志字节，用来表示TDS操作请求种类；*status*标志(1 Byte)，用来表示信息状态。值为0时，表示还有后续报；值为1时表示此包为当前TDS会话中的最后一个包；*length*标志(2 Byte)，表示TDS数据包总长度，其中含TDS包头的长度；*signn*标志(2 Byte)，是命名管道信息要用的通道数，此字段通常值为0；*packetn*标志(1 Byte)，表示TDS包在当前TDS操作请求中的序号；*windows*(1 Byte)，表示在确认信息收到以前必须发送的框架数目。

2.2 TDS协议具体报文分析

因为TDS协议是依靠TCP协议来携带的，所以首先使用Sniffer的过滤功能，设置捕获TCP协议，然后使Sniffer处于监听状态。在客户端利用SOCKET发包器向服务器发包，数据库在接到客户端发来的命令后作出回应,把它的应答信息发向客户端。

根据TDS包头的token标志大体上可以将TDS数据报文分成如下4类：(1)客户预登录包及其对应的服务器端响应包；(2)客户登录包及其对应的服务器端响应包；(3)客户端的语言命令包；(4)语言命令的服务器响应包。

作者简介：郭丽红(1975－)，女，讲师、硕士，主研方向：数据安全，计算机网络；吴海涛，讲师、硕士

收稿日期：2009-01-10 **E-mail：**guolihong@njit.edu.cn

2.2.1 客户预登录包及其对应的服务器端响应包

下面以一个具体的客户预登录包为例，用 Sniffer 捕获到的 TDS 数据包 16 进制数据，内容如下：

12 01 00 34 00 00 00 00 00 00 15 00 06 01 00 1b
00 01 02 00 1c 00 0c 03 00 28 00 04 ff 08 00 01
55 00 00 00 4d 53 53 51 4c 53 65 72 76 65 72 00
24 08 00 00

具体分析如下：

12 01 00 34 00 00 00 00//包头信息，12 为预登录标志

00 00 15 00 06 //0 字段信息

01 00 1b 00 01 //1 字段信息

02 00 1c 00 0c //2 字段信息

03 00 28 00 04 //3 字段信息

ff //字段信息结束标记

08 00 01 55 00 00 // NETLIB 的版本号

00 //强制加密标志，0 不强制加密，1 强制加密

4d 53 53 51 4c 53 65 72 76 65 72 00//客户端要求使用的实例 MS SQL Server

24 08 00 00 //进程的线程 ID

说明：0~3 字段信息，存储结构相同，每一字段都占 5 Byte，其中，开头字节存放字段号，接着的 2 Byte 存储字段偏移，最后 2 Byte 存储字段长度值。具体分析：这里的 0 字段存储关于 NETLIB 的版本信息的偏移和长度；第 1 字段存储关于用强制加密标记的偏移和长度；第 2 字段存储关于服务器的实例名偏移和长度；第 3 字段存储关于进程的线程 ID 偏移和长度。ff 之后存储的是具体的字段值信息。

对应客户预登录的服务器响应包，与客户端预登录包类似，只是 TDS 包头中的 *token* 值为 4，代表服务器响应包类型，其他的是服务器实例名、进程的线程等的字段名、，偏移值、长度及其具体的值等，与客户端预登录包类似，本文不再赘述。

2.2.2 客户登录包及其对应的服务器端响应包

客户端和服务器之间的会话是从客户端向服务器端发送一个 TDS 登录信息开始的。对于客户登录数据包，具体内容分析如表 1 所示。

表 1 客户登录数据包

内容(16 进制表示)	描述
10 01 00 98 00 00 01 00	TDS 包头 8 Byte，其中 10 是客户登录包标志
90 00 00 00	登录报长度为 144 Byte(不含包头)
01 00 00 71	服务器端 TDS 版本号：8.0
00 00 00 00	表示登录包只有 1 个，无后继包
00 00 00 71	客户端 TDS 版本号：8.0
b1 0b 00 00	客户进程 ID 号：2993
00 00 00 00	连接 ID 号：0
e0	标志 1
03	标志 2
00	SQL 类型标志。MSSQLEM (0)，SQL(1)，MSSQL Server(2)
00	保留标志
20 fe ff ff	时区
04 08 00 00	校对码

后续数据都是以 4 Byte 为一组存储的，其中偏移值占 2 Byte，长度占 2 Byte。分别存储客户端主机名称偏移和长度、登录的用户名称偏移和长度、登录的密码偏移和长度、客户端应用程序名称偏移和长度、服务器端主机名称偏移和长度、预留 4 Byte、库名称偏移和长度、本地名称偏移和长度、数据库名称偏移和长度。紧接着的数据依次是 6 Byte 的客户 MAC 地址：00 0a e6 bb df 92；2 字节的授权部分偏移值：00 00；2 Byte 的授权部分的长度：00 00，2 Byte 的下一位置偏移值：8c 00，2 Byte 的下一位置长度 00 00。最后依次存储的是具体的客户端主机名称、登录的用户名称、登录密码、应用程序名称、服务器端主机名称、库名称。

对应于客户登录包的服务器端响应包，主要依靠从服务器端返回的各种标志来识别各类信息。这里特别说明的是，每一组信息都是用 1 Byte 来存储标志信息，用 2 Byte 来存储长度信息，其他长度不定的既是具体的内容信息组成的。这些标志信息代表的意义总结如下：0xe3 表示环境变量的改变，指数据库的改变或包长度的改变等等；0xaa 表示错误信息，会给出错误信息代码；0xab 表示无错误信息；0xee 表示结果集合；0xac 表示输出参数等等。

2.2.3 客户端的 SQL 语言命令包

客户端的 SQL 语言命令包报文格式就是 8 Byte 的 TDS 报头加上后面紧跟着的 SQL 语言命令的 UNICODE 编码，最后面也都是用 4 Byte(0x0d 0x00 0x0a 0x00)既回车换行来表示该语言命令报的结束，其中报头第 1 个字节 *token* 值为 0x01，表示此 TDS 包为客户端 SQL 语言命令报。

2.2.4 SQL 语言命令的服务器响应包

SQL 语言命令分为 3 类：数据查询(select)，数据定义(create, drop, alter)，数据操纵(insert, update, delete)。对应于客户端的 SQL 语言命令请求报，服务器的响应报就只有 2 种类型：一种是对应 select 命令的响应；另一种是对应除 select 以外的其他命令响应。

(1)对应非 select 语言命令的服务器响应包

此服务器响应包由 17 Byte 构成，其中前 8 Byte 是 TDS 包头，定义如表 1 所示，第 1 个字节固定为 04，表示为服务器响应报。后续的 9 个字节分别是 fd(固定值)，表示服务器响应报的结束标志；10 00(固定值)，表示为 SQL 语言命令响应；接着的 2 字节对应不同的 SQL 语言命令，值会有所不同：c1 00 表示 select 命令的返回；c3 00 表示 insert 命令的返回；c4 00 表示 delete 命令的返回；c6 00 表示 create table 命令的返回；c7 00 表示 drop 命令的返回；de 00 表示 create procedure 命令的返回；df 00 表示 drop procedure 命令的返回；00 00 表示登录的返回；dd 00 表示 create trigger 命令的返回；最后的 4 字节存储的是对应 SQL 命令语句返回的行数。

(2)对应 select 语言命令的服务器响应包

对应 select 语言命令的服务器响应包，具体内容分析如表 2 所示。

表 2 对应 SELECT 语言命令的服务器响应包

长度/Byte	描述
8	TDS8.0 包头(第 1 个字节固定为 04，表示为服务器响应报)
1	此值固定为 0x81，表示为 select 命令响应报
长度不定	0x81 标志结构，包括列数、表名、字段名称、字段类型信息及返回的字段内容
9	fd 结束标志及其结构，此内容和非 select 命令的服务器响应结束标志一致

3 TDS 漏洞检测

SQL Server 2000 是微软推出的数据库产品，占领的市场份额仅次于 Oracle 和 DB2，居世界第 3，但是其安全性也一直受到用户的置疑。从 1996 年，Microsoft 公司推出的 SQL Server 6.5 版本到 1998 年推出的 SQL Server 7.0，以及到 2000 年 8 月推出了 SQL Server 2000，在版本和功能不断升级的情况下，安全问题却没有得到很好地改善，保护数据库安全的

常用方法是：及时打补丁，最小的权限，安装防火墙，改变端口，删除不需要的扩展存储过程等。但是，数据库安全问题依然存在，2003 年 1 月，针对 SQL Server 的 Slammer 蠕虫在 Internet 上肆虐，导致网络流量激增，严重影响了世界范围内的计算机和网络系统，因此协议漏洞检测势在必行。

3.1 SQL Server 2000 密码明文传输漏洞[4]

此漏洞的发现是采用协议分析法。因为 SQL Server 2000 客户端和服务器之间使用的 TDS 协议是依靠 TCP 协议来携带的，所以 SQL Server 2000 的连接过程是先进行 TCP 连接的三次握手，同服务器建立连接过后，然后再进行 TDS 协议的数据交流。在 TDS8.0 中，表示登录数据报标志字节为 0x10，通过大量实验发现：用户名完全是明文的，而密码还不是，但是存在一定的规律。通过不断改变密码来登录 SQL Server 并获取 TDS 包，对捕获到的大量 TDS 数据包进行分析之后，能够得到密码转换规则。通过编写 C 语言程序就可以破译一个在命令行上传递的密码口令。具体流程如图 2 所示。

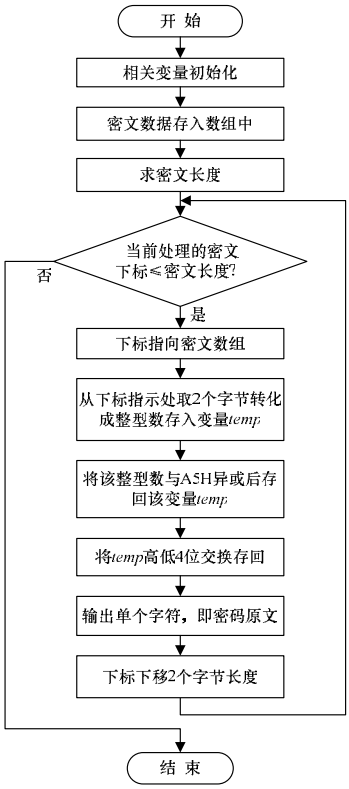

图 2　密码明文传输漏洞测试程序流程

解决方案：微软对 SQL Server 的传输还是提供加密办法的，使用 SSL 来加密即可解决密码明文传输漏洞。

3.2 缓冲区溢出漏洞

此漏洞的发现是采用比较传统的黑盒测试方法，其核心就是把 TDS 协议包分类，然后制定出对各类协议包的测试策略，最后按照策略逐步进行测试。

在实验中首先构造数据库包：TDS 包头+40 个字节的 f 作为 TDS 负载+TDS 包尾，然后不断加长 TDS 负载长度，构造不同长度的测试包测试 SQL Server 2000。当 TDS 包头+560 Byte 的'f'+TDS 包尾，作为测试包时发现，SQL Server 2000 在进行用户验证时产生缓冲溢出，利用这一个漏洞攻击者提供精心构造的请求可以得到系统的权限以执行任意代码，并且可以读取数据库内容或提供不正规登录请求可导致破坏内存。这个漏洞产生溢出的主要原因是：如果源字符串长度超出 532 Byte 后，目标地址后的环境变量就会被覆盖，从而导致溢出，具体测试流程如图 3 所示。

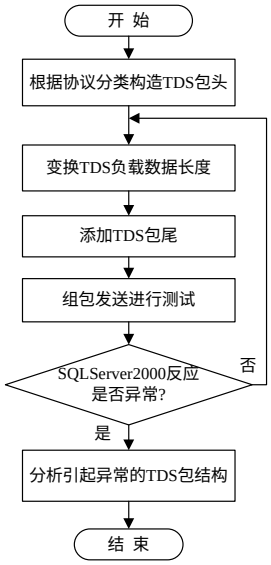

图 3　缓冲区溢出漏洞测试程序流程

解决方案：用防火墙保护 SQL 服务器的 1433 端口或安装微软补丁。事实上，在经历了 Slammer 蠕虫后，目前基本所有的系统都补上了该漏洞。

4 结束语

本文对数据库 TDS 8.0 通信协议进行研究、测试、分析，同时通过对 SQL Server 2000 中存在的安全漏洞进行深入分析、测试，验证并找出了 TDS 8.0 协议中存在的安全问题，这对目前数据库产品存在的安全漏洞防范具有重要意义。

参考文献

[1] 冯登国. 计算机通信网络安全[M]. 北京: 清华大学出版社, 2001.

[2] 徐　婷, 杨欣荣. 数据库安全技术的理论研究[J]. 科技情报开发与经济, 2007, 17(4): 222-223.

[3] 雒　群. 数据库通信协议分析与安全检测[D]. 吉林: 长春理工大学, 2003.

[4] Stallings W. 密码编码学与网络安全[M]. 刘玉珍, 译. 北京: 电子工业出版社, 2001.

编辑　金胡考