kefire：日志是系统记录用户操作情况的地方，任何一个学习网络安全的人都应该明确日志在系统中的重要作用，但是安全方面该如何保护日志？入侵者又该如何方便、实用地删除日志？这对矛盾并不是人人都能化解的，下面就给大家带来一篇介绍如何通过编程实现日志删除的文章，目的是想让大家在学到编程知识的同时，也能明白删除日志的大体流程并对应地去体会保护日志被删除的方法！

难度等级：初级

前置知识：C语言基础

VC也玩清除日志

文／小华健

清除日志是每次入侵后都必须要做的事，以免被别人发现入侵痕迹，网络上虽然有很多非常流行的清除日志的工具，但遗憾的是几乎都没有实际的效用。考虑到朋友们的需要，今天我们就自己来打造一份功能完备的清楚日志工具！

我们知道，如果想要清除日志，那么首先应该停止服务，然后用GetSystemDirectory()来获取系统目录，再删除日志文件，最后用一个函数来重启服务。我们比较常见的是W3SVC服务，下面就以它为例来说明如何通过编程来删除日志。

预备函数

首先，OpenSCManager函数是用来打开指定计算机上的service control manager database。其函数原型：

```
SC_HANDLE OpenSCManager(
LPCTSTR lpMachineName,\\ 指定计算机名，若为空则指定为本机，
LPCTSTR lpDatabaseName,\\ 指定要打开的service control manager database名，默认为空，
DWORD dwDesiredAccess\\ 指定操作的权限，
)
```

其中参数dwDesiredAccess，可以为下面取值之一：

```
SC_MANAGER_ALL_ACCESS //所有权限，
SC_MANAGER_CONNECT //允许连接到service control manager database，
SC_MANAGER_CREATE_SERVICE //允许创建服务对象并把它加入database，
SC_MANAGER_ENUMERATE_SERVICE //允许枚举database中的Service，
SC_MANAGER_LOCK //允许锁住database，
SC_MANAGER_QUERY_LOCK_STATUS //允许查询database的封锁信息，
```

再有，OpenService函数能打开指定的Service，函数调用成功则返回打开的Service句柄，失败则返回NULL。其函数原型如下：

```
SC_HANDLE OpenService(
SC_HANDLE hSCManager,//指向service control manager database的句柄，由OpenSCManager返回，
LPCTSTR lpServiceName,//为Service的名字，
DWORD dwDesiredAccess//访问权限，
)
```

Service程序没有专门的停止函数，而是用ControlService函数来控制Service的暂停、继续、停止等操作。其函数原型如下：

```
BOOL ControlService(
SC_HANDLE hService,
DWORD dwControl,LPSERVICE,
STATUS lpServiceStatus//一个指向SERVICE_STATUS的指针，
)
```

参数dwControl指定发出的控制命令，可以为以下几个值：

```
SERVICE_CONTROL_STOP //停止Service；
SERVICE_CONTROL_PAUSE //暂停Service；
SERVICE_CONTROL_CONTINUE  // 继续Service；
SERVICE_CONTROL_INTERROGATE //查询Service的状态；
SERVICE_CONTROL_SHUTDOWN  //让ControlService调用失效；
```

编写过程

首先要停止服务，具体程序如下：

```
void StopServices(LPCTSTR lpServiceName)
{
    SC_HANDLE sc=OpenSCManager(NULL,NULL,SC_MANAGER_ALL_ACCESS);
    if(sc)
    {
        SC_HANDLE sh=OpenService(sc,lpServiceName,SERVICE_STOP);
        if(sh)
        {
            BOOL bControl;
            SERVICE_STATUS ServiceStatus;
            bControl=ControlService(sh,SERVICE_CONTROL_STOP,&ServiceStatus);
            if(bControl)
            {
            printf("success to stop the service\"%s\"\n",lpServiceName);
            }
            else
            {
            printf("failed to stop the service\"%s\"\n",lpServiceName);
            }
        }CloseServiceHandle(sh);
    }
    CloseServiceHandle(sc);
    return;
}
```

然后再删除已经记录的日志文件，删除文件的函数如下：

```
void DelFiles(LPCTSTR lpFileName)
{
    BOOL dDel=DeleteFile(lpFileName);
    TCHAR tcSystemDirectory[1024];
    GetSystemDirectory(tcSystemDirectory,1024)
    if(dDel)
    {
        printf("delete file \"%s\" success\n",lpFileName);
    }
    else
    {
        DWORD i=GetLastError();
        printf("delete file \"%s\" failed\n",lpFileName);
    }
}
```

在删除日志成功以后，还需要重新启动服务，我们可以用StartService函数来启动指定的Service。其函数原型如下：

```
BOOL StartService(
SC_HANDLE hService,\\ 指向Service的句柄，由OpenService返回，
DWORD dwNumServiceArgs,\\ 为启动服务所需的参数的个数，
LPCTSTR *lpServiceArgVectors \\为启动服务所需的参数，
)
```

其中，参数lpServiceStatus是一个指向SERVICE_STATUS的指针。SERVICE_STATUS是一个比较重要的结构，它包含了Service的各种信息，如当前状态、可接受何种控制命令等等。

```
void StartServices(LPCTSTR lpServiceName)
{
    SC_HANDLE sc=OpenSCManager(NULL,NULL,SC_MANAGER_ALL_ACCESS);
    if(sc)
    {
        SC_HANDLE sh=OpenService(sc,lpServiceName,SERVICE_START);
        if(sh)
        {
        BOOL bControl;
        bControl=StartService(sh,1,
```