ASP 源文件保护策略

ISPAI filter and protecting ASP source file

刘玉林　曹二堂　（河北经贸大学 050061）

摘要：本文介绍了利用 ISAPI 筛选器保护 ASP 源文件的方法。

关键词：IIS ISAPI ASP 代码保护

1 引言

ASP 作为一种网络开发的脚本语言，由于编程简单、功能强大，得到了广泛的应用。然而，由于 ASP 脚本是采用明文(plain text)方式来编写的，所以应用开发商辛苦开发出来的 ASP 应用程序，一旦发布到运行环境中去后，就很难确保这些"源代码"不会被流传出去。这样就产生了如何有效地保护开发出来的 ASP 脚本源代码的需求。

对于 ASP 源代码保护的常用方法主要有两种：

(1) 使用微软提供的官方加密程序加密。该程序可以从微软免费下载，安装后生成 screnc.exe 文件，这是一个运行在 DOS PROMAPT 的命令工具。使用该程序加密后，ASP 脚本变成不可阅读的密文。加密后的 ASP 文件在 IIS5.0 下可直接运行，因而使用非常方便。但是，在微软提供加密程序后很快就出现了解密程序，解密程序能完全恢复源代码，使加密形同虚设，不能说不是一种遗憾。

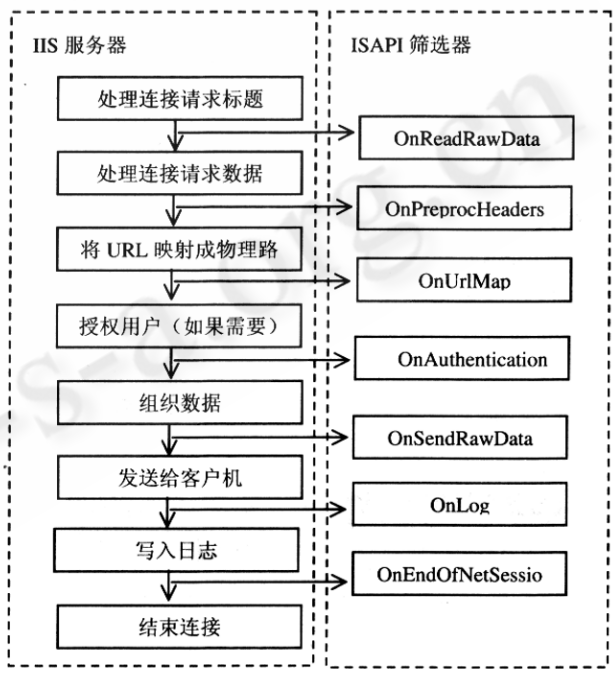
(2) 使用 ASP 组件，即 ASP 文件中只编写尽可能少的源代码，将核心代码脚本部分被封装到一个 COM /DCOM 组件，并在 ASP 脚本中创建该组件，进而调用相应的方法。因此，需要在开发 ASP 脚本应用之前就可按此思路来开发，或者直接用 ASP 脚本快速开发出原型系统后，针对需要保护、加密的重要脚本用 COM /DCOM 组件来重新开发、实现并替换。但该方法给开发调试带来诸多不便，因此一般只对最主要的代码采用这种方式。

鉴于上述两种方法的不足，本文研究了使用 ISAPI 的 IIS 筛选器的方法。该方法的基本思想是，开发过程中仍然使用按源代码方式进行，程序发布前将脚本进行加密处理，程序发布后给 IIS 添加一个筛选器。利用筛选器与 IIS 的紧密结合特性，在 ASP.DLL 读取脚本前还原加密程序。其特点是，脚本的开发与加密分离，加密方式可以自行决定，筛选器只起到还原密文的功能。

2 ISAPI 筛选器的构成

ISAPI 筛选器是微软针对 IIS 提供的专用 API，提供了对 IIS 的纵向扩充功能。ISAPI 筛选器和 HTTP 服务器运行在同一地址空间，并且可以访问可由 HTTP 服务器使用的所有资源。ISAPI 筛选器随 IIS 启动而加载，客户机的每个 HTTP 请求都会导致筛选器执行，并且 IIS 处理 HTTP 请求的各个阶段都会产生一个通知，从而调用筛选器处理相应事件。

下图显示了 IIS 服务器从接收到客户机请求数据到结束连接整个过程中 IIS 的处理步骤，以及筛选器可以获得的通知和处理方式。



使用 MFC 的 CHttpFilter 类创建筛选器来管理 ISAPI 服务器的传入和输出数据时，使用两个入口点成员函数 CHttpFilter::GetFilterVersion 和 CHttpFilter::HttpFilterProc 。

2.1 GetFilterVersion

当 IIS 启动时，它读取该值并加载筛选器。然后它调用 CHttpFilter::GetFilterVersion 交换版本信息、确定请求的事件及指定传递请求事件的优先级。本文用到两个确

定请求的事件，即 SF_NOTIFY_URL_MAP 和 SF_NOTIFY_SEND_RAW_DATA，用于将逻辑 URL 映射到物理路径时获得加密文件和将原始数据从服务器发送到客户端之前通知筛选器以删除解密文件。

2.2 HttpFilterProc

当事件发生时，服务器通过调用筛选器的 HttpFilterProc 入口点通知筛选器。当 CHttpFilter::HttpFilterProc 被调用时，接收到的通知将确定将要调用哪一个 CHttpFilter 成员函数。通过重写 HttpFilterProc 成员函数时，可以使筛选器以特定的方式处理数据。本文涉及两个成员函数。

OnUrlMap 在服务器将逻辑 URL 映射到物理路径时通知筛选器执行。

OnSendRawData 在将原始数据从服务器发送到客户端之前通知筛选器执行。

3 ASP 源代码保护的实现过程

ASP 的开发调试完成后可以通过自行设计的加密方法进行加密，通过加密变为密文文件。客户机访问密文文件时，首先通过 IIS 筛选器解密还原为 ASP 源代码，然后再将 ASP 源代码交给 ASP 解释器解释执行，该过程在服务器将逻辑 URL 映射到物理路径时完成，即重写 HttpFilterProc 的成员函数 OnUrlMap。处理方法如下：

筛选器得到通知后，判断是否 ASP 请求，决定是否对文件处理

```
CString isasp,ext;
isasp=pMapInfo->pszPhysicalPath; //获得请求的密文文件物理位置
ext=isasp.Right(3);ext.MakeUpper(); //取得文件扩展名
if(!(ext=="ASP")) return SF_STATUS_REQ_NEXT_NOTIFICATION;
```

//如果不是 ASP 文件，不做解密处理，否则打开获得请求的 asp 密文文件进行解密处理。

以下程序用于解密被加密的文件

```
CFile File_i;
if(! File_i.Open(pstrPhysPath,CFile::modeRead)) //打开请求的 ASP 文件
{ return SF_STATUS_REQ_FINISHED; }
num=File_i.GetLength(); //获得文件长度
num=File_i.Read(str,num); //读文件内容于缓存 str
File_i.Close();
```

对密文文件解密

```
decrypt(str) ; //调用解密函数，对缓存的加密密文进行解密变换
```

将解密后的源码保存在一临时文件中，文件名随机生成，可以存放在系统临时目录下。

```
CFile File_o;
if(! File_o.Open(pt,CFile::modeReadWrite | CFile::modeCreate)) //创建临时文件
{ return SF_STATUS_REQ_FINISHED; }
File_o.Write( str, num ); //将解密后的内容写入文件
File_o.Close();
```

修改映射的物理路径为解密后的临时文件

```
int n1,n2;
n1=pt.GetLength(); //pt－解密文件路径
n2=pMapInfo->cbPathBuff;
for(int i=0;i<n1;i++) //修改 URL 映射的物理路径，使其指向解密文件
{ pstrPhysPath[i]=pt[i]; }
for(int j=n1;j<n2;j++)
{ pstrPhysPath[j]='\0'; }
```

由于映射的物理路径指向解密文件，IIS 会将解密后的文件教给 ASP.DLL 处理程序解释执行。为了不在磁盘留下解密文件，临时文件应及时删除，删除文件操作可通过重写 HttpFilterProc 的成员函数 OnSendRawData 来完成，该函数在系统将处理完数据的数据发送至客户机前被调用，因此，临时文件可以及时被删除。

4 结论

由于 ASP 程序的开发与加密分离，ASP 源文件的保护措施不会影响 ASP 程序的开发。系统开发完成后，可以使用加密程序对其进行一次性加密，生成密文文件，解密是在 IIS 筛选器中进行的，因此也不需要在加密文件时对源文件做任何特殊处理，该方法简单易行，更适合于对现有系统的加密保护，该方法也适合于对所有其他 WEB 文件的加密处理和安全保护。

参考文献

1 微软公司 MSDN Library 。

2 (美)K.Clements 等，ISAPI 实用技术指南，清华大学出版社，1998.7。

3 潘爱民、王国印 译，Visual C++技术内幕，清华大学出版社，1999.1。