走近微软安全技术Shim

转载 cosmoslife 于 2016-09-14 09:46:34 发布 阅读量475

分类专栏： Windows编程

Shim是微软系统中一个小型函数库，用于透明地拦截API调用，修改传递的参数、自身处理操作、或把操作重定向到其他地方。Shim主要用于解决遗留应用程序在新版Windows系统上的兼容性问题，但Shim也可用于其他方面。例如上周微软紧急推出针对“微软Office Powerpoint 0day漏洞（CVE-2014-6352）”的Fix It，其中就采用了Shim技术，用于修复存在安全缺陷的函数。

一、什么是Shim

Shim是微软极少使用的四字母单词之一，也不是某种形式的缩写。它是英语单词Shim的引申含义。Shim是一个工程术语，描述为了让两个物体更好地组装在一起而插入的一块木头或金属。在计算机编程中，shim是一个小型的函数库，用于透明地拦截API调用，修改传递的参数、自身处理操作、或把操作重定向到其他地方。Shim也可以用来在不同的软件平台上运行程序。

二、shim如何工作

Shim架构实现了一种API钩子，而Windows API是通过一组DLL来实现的。Windows系统上的每个应用程序导入这些DLL，并在内存中维护一个存储调用函数地址的表（导入表）。由于Windows函数的地址位于一个表中，Shim直接把导入表中的地址替换为shim DLL中的地址。通常，应用程序没有意识到请求被重定向到一个Shim DLL而不是Windows系统，而Windows系统也没意识到请求并非来自应用程序（因为Shim DLL刚好也位于应用程序的进程中）。

在这个例子中，两个主体分别是应用程序和Windows系统，而shim是能够两者更好协作的附加代码，如下所示：

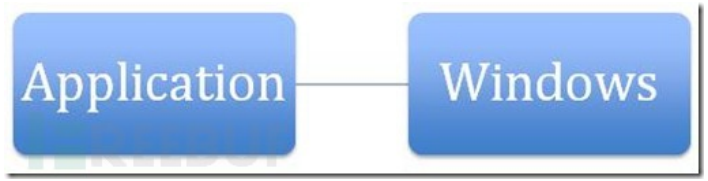


图1 应用shim之前，应用程序直接和Windows系统进行交互

图2应用shim之后，应用程序间接地与Windows系统进行交互

shim代码被注入，并能够修改发向Widows系统的请求、从Windows系统返回的响应或全部,

尤其是，shim利用链接的特性将API调用重定向至替换的代码-Shim。通过导入表（IAT）实现调用外部的二进制文件。因此，调用Windows的函数类似于：

图3 应用程序通过IAT表调用Windows系统

你可以修改IAT表中已解析的Windows函数地址，然后替换为指向shim中替代函数的指针，如图4所示。

图4 应用程序被重定向到shim而不是调用Windows系统

静态链接的DLL重定向发生在应用程序启动的时候。你也可以通过拦截GetProcAddressAPI调用来重定向动态链接的DLL文件。

三、为什么使用Shim

你无需访问源代码就可以修复应用程序，或甚至不需要修改应用程序。你只需承担极少的管理开销（针对Shim数据库），然而你通过这种方式可以修复数量相当可观的应用程序。缺点是支撑不足，因为大部分供应商不支持经Shim修复的应用程序。你不能够应用Shim来修复所有的应用程序。在软件供应商已经倒闭关门、软件已被淘汰而不予支持、或只购买一段时间的授权的情况下，人们可能考虑对应用程序进行Shim修复。

例如，最经常使用的Shim是version-lie（版本欺骗）Shim。为了实现这个Shim，我们拦截应用程序用于判断Windows版本的几个API。正常情况下，请求能够直接发送到Windows系统，并能够给予真实的回复。使用了Shim之后，向应用程序回复一个伪造的Windows版本（例如，Windows XP 而不是 Windows 7）。如果应用程序只能运行在Windows XP上，通过这种方式就能够让应用程序误以为自己运行在正确的操作系统上。（通常用来解决兼容性问题）。

你可以利用shim玩很多花样，例如：

1.**ForceAdminAccess shim**试图欺骗应用程序相信当前用户是本地管理员用户组，即使实际情况并不是如此。（如果你不是一个本地管理员用户，尽管你可能使用了其他的技巧来解决这个问题，如UAC文件和注册表虚拟化，许多应用程序还是启动失败）。Shim如何实现版本检查是相当简单明了。例如shim拦截shell32.dll中IsUserAnAdminAPI调用。shimmed修复后的函数（相对于实际的API，修复后的函数具有极佳的性能）只是简单返回True。

2.**WrpMitigation shim**欺骗应用程序的安装程序相信可以写入被WRP保护的文件。如果你试图一个被WRP保护的文件，shim首先创建一个新的临时文件，标记为文件句柄关闭后立即删除，然后返回临时文件的句柄冒充为实际被保护的文件。

3.**CorrectFilePaths shim**可以把文件从一个位置重定向到另一个位置。因此，你如果有一个程序试图写入c:\myprogramdir（不能利用UAC文件和注册表虚拟化来自动解决），你可以把运行时修改的文件重定向到每个用户的位置。这样就允许你作为标准用户运行，同时又不放松ACL。

Freebuf科普

> WRP（Windows资源保护）对系统稳定性有重要影响的文件进行隐秘拷贝，
> 但是存储的位置变成了%Windir%\WinSxS\Backup，依靠Access Control List（访问控制列表，ACL）为系统提供实时保护，
> 在WRP的管理下允许对被保护的资源进行写入只授权给了TrustedInstaller，即使是系统管理员也没有权限。

注意：因为Shim代码运行在用户模式的程序进程中，所以你不能使用shim来修复内核模式的代码。例如，你不能利用shim来解决设备驱动或其他内核模式代码的兼容性问题。（例如，一些反病毒软件、防火墙以及反间谍软件代码运行内核模式下）

四、何时使用Shim

1.从已破产的供应商获得的应用程序：既然供应商已经倒闭，技术支持自然无从说起。然而，因为源码不能获得，shim是解决兼容性问题的唯一选择。

2.内部开发的应用程序：虽然大部分用户更倾向于让自己开发的应用程序本身可以解决兼容性问题，但有些场景中时间上并不允许这么做。团队也许不能在新版Windows部署计划之前解决所有的兼容性问题，因此他们可能选择利用shim修复应用程序，同时shim不能修复的部分就需要修改应用程序的源代码。

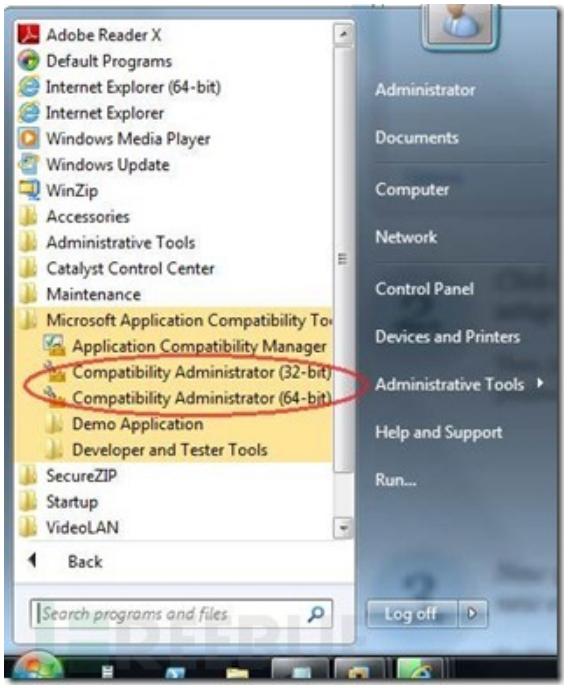
3.供应商将要发布一个兼容性的版本，但当前的技术支持不够：当现有的应用程序既不是关键业务也不是很重要，一些用户使用Shim作为临时解决方案。理论上讲，用户可以等到兼容性的版本发布，但会阻碍整个部署计划。在兼容版本可用之前，先为用户提供一个Shim修复过的且能正常运转的应用程序不失为两全之计。

五、创建一个应用程序兼容性的Shim

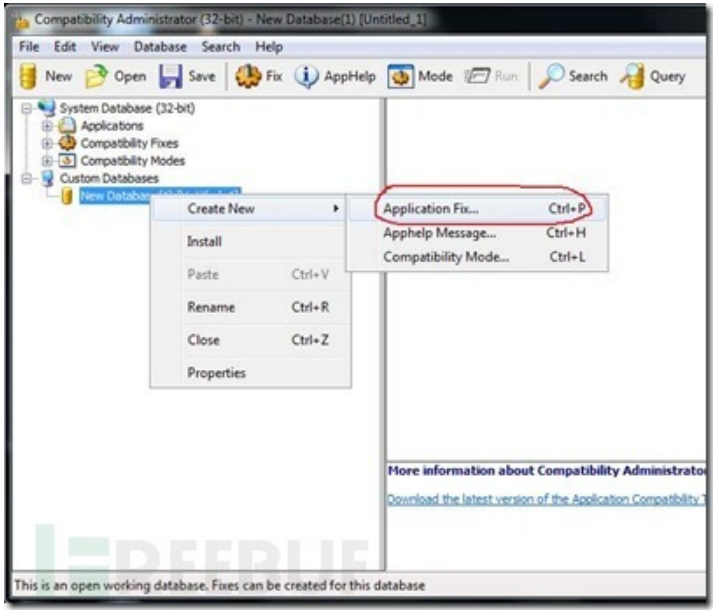
如果你试图在Windows 7运行专为2000或XP创建的应用程序，并出现了问题，你可能总是需要在你的机器上开启兼容模式。然而，如果创建了Shim，你也可在其他机器上运行这个程序，而不需要每次手动开启兼容模式。Shim是体积小且只需运行一次，常常和机器上的特定应用程序联系在一起。

ACT是应用程序兼容性工具包（Application Compatibility Toolkit），可以从这里下载：

一旦我们从“开始”菜单->”微软应用程序兼容性工具包”->”兼容性管理工具”（Compatibility AdministratorTool）启动。

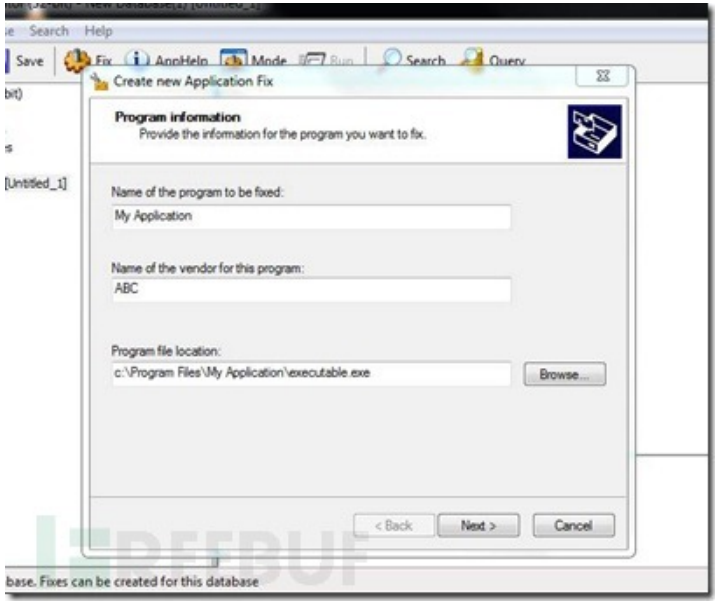


在“New Database”点击右键：



选择“Application Fix”，然后从下面的对话框中选择需要修复的应用程序：

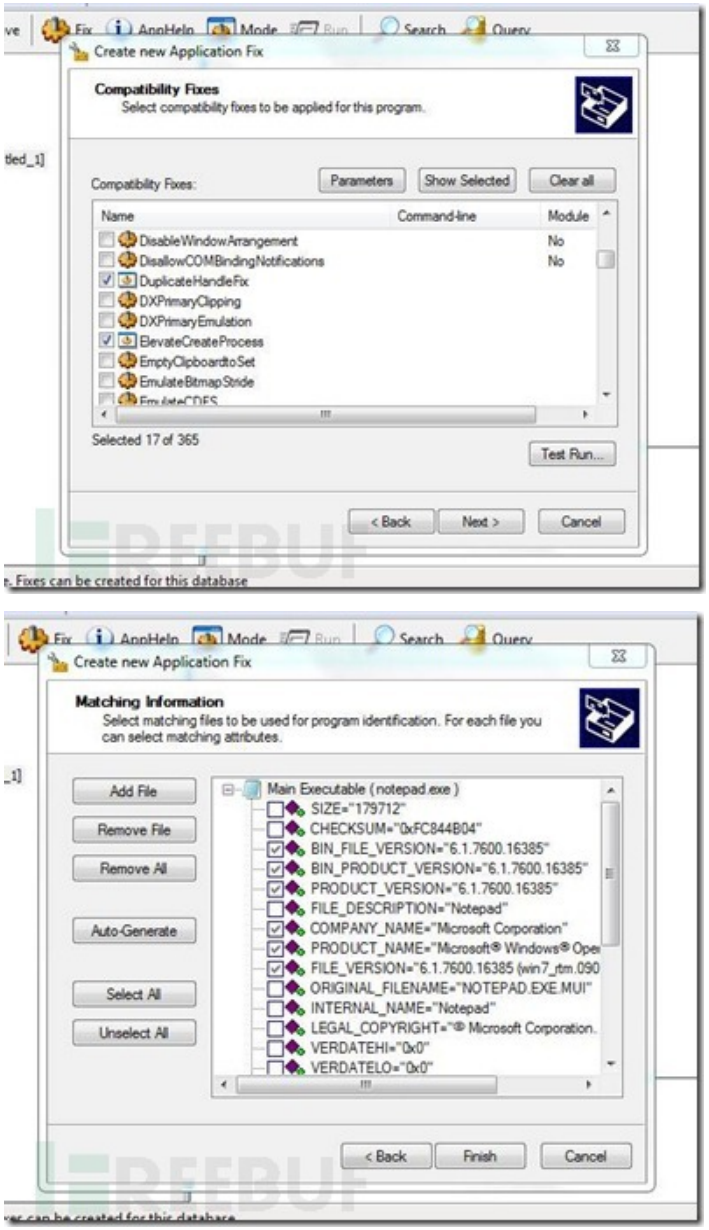
```
①输入需要修复的程序
②输入供应商名称
③浏览可执行程序的位置
```



当你按下“Next”按钮，你接下来将看到默认的兼容模式列表。如果仅存在版本兼容性问题，你可以选择应用程序原先运行的操作系统版本。

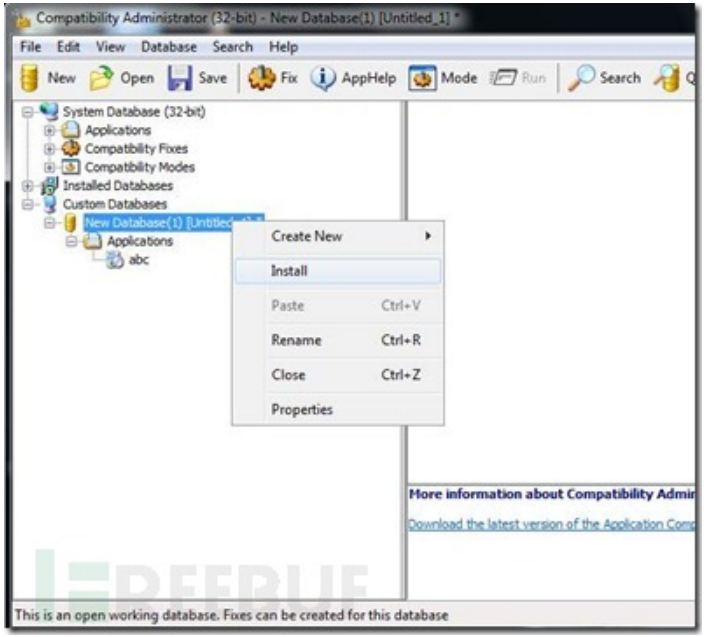
此处我也已经判断出Windows 2000兼容模式适用于这个程序，然后在列表中上下滚动找到“Windows 2000”

接下来的窗口（当需要选择多个Shim进行组合时）。如下图所示，你有许多Shim可供选择。选择修复应用程序所需的全部shim。



单击“Finish”，会显示应用程序和所选修复的全面概要。

现在你只需保存这个shim数据库文件（包含了创建的shim信息的一个小型数据库），然后安装这个shim数据库文件。你可以通过右键单击来安装或通过命令行dbinst.exe <database.sdb>



注意：sdbinst.exe已经默认位于“c:\windows\system32”

一旦应用程序兼容性数据库被安装，我们就可以从预先指定的位置运行程序。现在这个程序运行在你所指定的兼容模式。

Freebuf作者

本文只是简单介绍Shim技术，属科普类的文章。若有朋友想深入了解shim技术，请查看《Windows® Internals》作者Alex Ionescu在2014年会议

[参考信息来源technet，内容有所删减，尽量保留了原文本意。译自Rabbit_Run，喜欢文章请点赞鼓励。转载请注明来自FreeBuf.COM]