浅析IE浏览器劫持 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/461/2021_2022__E6_B5_85_E6_9E_90IE_E6_B5_c101_461681.htm 一. 谁误导了浏览器 今天是大年初二，王先生家中来了许多客人，把平时埋头于工作的王先生弄了个手忙脚乱，由于客人带来的几个小孩子嚷嚷着要出去上网，王先生只好把寝室里的电脑让给了这一群孩子玩，好容易到了晚上，客人散尽，王先生想在休息前上网浏览一下新闻，可是当他打开IE的时候，却发现它自动连接到一个莫名其妙的网站去了，而且收藏夹里也多了一些奇怪的网址，王先生担心是系统感染了病毒，赶紧输入一个在线杀毒工具的网址，结果IE打开的却是另一个不知所谓的网站。看着IE地址栏里准确无误的杀毒工具网址和下面那根本扯不上关系的内容，王先生真的手忙脚乱了…… 相信不少用户也遇到过与之类似的奇怪事情，究竟是谁，把我们的浏览器领到了一处陌生的地方呢? 近年来，针对浏览器的攻击手段层出不穷，对浏览器的渗透攻击逐渐成为入侵者攻破用户层层防御的首要目标，在“ 钓鱼 ”(Publishing)危机尚未解除的时候，另一种攻击方式也在同时进行着，这就是“ 浏览器劫持 ”故意误导浏览器行进路线的策划者。“ 浏览器劫持 ”(Browser Hijack)是一种不同于普通病毒木马感染途径的网络攻击手段，它的渗透途径很多，目前最常见的方式有通过BHO、DLL插件、Hook技术、Winsock LSP等载体达到对用户的浏览器进行篡改的目的。这些载体可以直接寄生于浏览器的模块里，成为浏览器的一部分，进而直接操纵浏览器的行为，轻者把用户带到自家门户网站，严重的则会在用户计

算机中收集敏感信息，危及用户隐私安全。“ 浏览器劫持 ”的后果非常严重，用户只有在受到劫持后才会发现异常情况，但是这时候已经太迟了。目前，浏览器劫持已经成为Internet用户最大的威胁之一。 二. BHO，你是助手还是敌人? 为什么“ 浏览器劫持 ”能够如此猖狂呢?放眼众多论坛的求助贴，我们不时可以看到诸如“ 我的IE被主页被改了，我用杀毒工具扫了一遍都没发现病毒，我把主页改回自己的地址，可是一重启它又回来了! ”、“ 我的系统一开机就跳出一个广告，我明明用了最新版的杀毒软件的啊! ”等这类关于IE异常问题的求助，80%的提问者都表示纳闷，他们已经安装了杀毒软件，可是IE仍然被“ 黑 ”了，这又是为什么? 其实这些都是典型的“ 浏览器劫持 ”现象，但是受害者不是已经安装了杀毒软件吗?为什么浏览器依然躲不过这只黑手?许多用户对这个领域都存在一种误区心理:浏览器劫持?我有最新的杀毒软件，我不怕! 于是，当他们遭遇“ 浏览器劫持 ”时，惊讶了。 要知道，杀毒软件自身也只是一种辅助工具，它不可能完全保护系统的安全，更何况，杀毒软件用户必须知道一个事实:“ 浏览器劫持 ”的攻击手段是可以通过被系统认可的“ 合法途径 ”来进行的!杀毒软件只能通过“ 特征码 ”的形式来判断程序是否合法，但这是建立在人为定义以后的，而实施“ 浏览器劫持 ”的程序可以有很多，防不胜防。 为什么说“ 浏览器劫持 ”可以说是合法的呢?因为大部分浏览器劫持的发起者，都是通过一种被称为“ BHO ”(Browser Helper Object,浏览器辅助对象)的技术手段植入系统的。 BHO是微软早在1999年推出的作为浏览器对第三方程序员开放交互接口的业界标准，它是一种可以让程序员使用简单代码进入浏览器

领域的“ 交互接口 ”(INTERACTIVED Interface)。通过BHO接口，第三方程序员可以自己编写代码获取浏览器的一些行为(Action)和事件通知(Event)，如“ 后退 ”、“ 前进 ”、“ 当前页面 ”等，甚至可以获取浏览器的各个组件信息，像菜单、工具栏、坐标等。由于BHO的交互特性，程序员还可以使用代码去控制浏览器的行为，比如常见的修改替换浏览器工具栏、在浏览器界面上添加自己的程序按钮等操作，而这些操作都被视为“ 合法 ”的，这就是一切罪恶根源的开始。
BHO的出现帮助程序员更好的打造个性化浏览器或者为自己的程序实现了方便简洁的交互功能，可以说，如果没有BHO接口的诞生，我们今天就不能用一些工具实现个性化IE的功能了。从某一方面来看，BHO的确是各种缤纷网络互动功能的幕后功臣，但是一切事物都是有两面性的，这个恒古不变的真理同样对BHO有效，于是就有了今天让安全界头痛的“ 浏览器劫持 ”的攻击手段诞生。 看看前面我提到的BHO接口特性，你想到了什么?BHO可以获知和实现浏览器的大部分事件和功能，也就是说，它可以利用少量的代码控制浏览器行为。程序员可以设计出一个BHO按钮以实现用户点击时通知浏览器跳转到某个页面完成交互功能，当然就可以进一步写出控制浏览器跳转到他想让用户去的页面，这就是最初的“ 浏览器劫持 ”的成因:BHO劫持。 在描述BHO劫持之前，我们先要对BHO接口的启动做个简单介绍:符合BHO接口标准的程序代码被写为DLL动态链接库形式在注册表里注册为COM对象，还要在BHO接口的注册表入口处进行组件注册，以后每次IE启动时都会通过这里描述的注册信息调用加载这个DLL文件，而这个DLL文件就因此成为IE的一个模块(BHO

组件)，与IE共享一个运行周期，直到IE被关闭。 IE启动时，会加载任何BHO组件，这些组件直接进入IE领域，而IE则成为它们的父进程和载体，从此IE的每一个事件都会通过IUnknown接口传递到BHO用以提供交互的IObjectWithSite接口里，这是BHO实现与IE交互的入口函数。 BHO接收到IE接口传递来的参数后开始判断IE正在做什么，理论上BHO可以获取IE的大部分事件，然后根据程序员编写的代码，BHO持有对特定事件做出反应的决定权，例如一个可以实现“ 中文网址 ”的BHO，就是通过GetSite方法获取到IE当前打开的站点URL(或通过IURLSearchHook接口来获知)，如果BHO发现获取到的URL和内置的判断条件匹配，该BHO就会启用SetSite方法强制IE跳转到程序员设定的页面去，这个过程就是利用about:blank篡改主页的“ 浏览器劫持 ”方法之一，它的实现原理其实很简单，程序员编写一个恶意BHO组件，当它获取到IE窗口的当前站点为“ about:blank ”时就强制IE内部跳转到指定的广告页面，于是闹出了不久之前沸沸扬扬的“ IE空白页劫持事件 ”。 了解了这种类似恶作剧的作案手段，要解决它就容易了，只要找到并删除这个隐藏在系统里的BHO程序即可。 