

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局



(43) 国际公布日
2016年6月9日 (09.06.2016)

WIPO | PCT



(10) 国际公布号

WO 2016/086767 A1

(51) 国际专利分类号:
G06F 21/50 (2013.01)

(21) 国际申请号: PCT/CN2015/094845

(22) 国际申请日: 2015年11月17日 (17.11.2015)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201410743201.0 2014年12月5日 (05.12.2014) CN

(71) 申请人: 北京奇虎科技有限公司 (BEIJING QIHOO TECHNOLOGY COMPANY LIMITED) [CN/CN]; 中国北京市西城区新街口外大街 28 号 D 座 112 室 (德胜园区), Beijing 100088 (CN)。 奇智软件 (北京) 有限公司 (QIZHI SOFTWARE (BEIJING) COMPANY LIMITED) [CN/CN]; 中国北京市朝阳区酒仙桥路 6 号院 2 号楼 B 座 2 层、3 层 301-306 室, Beijing 100015 (CN)。

(72) 发明人: 党壮 (DANG, Zhuang); 中国北京市朝阳区酒仙桥路 6 号院 2 号楼, Beijing 100015 (CN)。 吴亮 (WU, Liang); 中国北京市朝阳区酒仙桥路 6 号院 2

号楼, Beijing 100015 (CN)。 王天平 (WANG, Tianping); 中国北京市朝阳区酒仙桥路 6 号院 2 号楼, Beijing 100015 (CN)。 梁志辉 (LIANG, Zhihui); 中国北京市朝阳区酒仙桥路 6 号院 2 号楼, Beijing 100015 (CN)。

(74) 代理人: 北京润泽恒知识产权代理有限公司 (BEIJING RISEHIGH INTELLECTUAL PROPERTY LAW FIRM); 中国北京市海淀区中关村南大街 31 号神舟大厦 702, Beijing 100081 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA,

[见续页]

(54) Title: METHOD, BROWSER CLIENT, AND DEVICE FOR ACHIEVING BROWSER SECURITY

(54) 发明名称: 实现浏览器安全的方法、浏览器客户端和装置

在浏览器进行安装时, 通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务

110

在浏览器中内置一安全组件, 浏览器启动后通过该安全组件调用所述系统服务, 拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改

120

图 1 / FIG. 1

110 When browser is being installed, system service that launches with operating system is installed, by means of browser installation package, on operating system on which browser is installed

120 Browser has built-in security component; after launching, browser invokes system service by means of security component; first process independent of browser process is intercepted from modifying browser installation files and/or browser data

(57) Abstract: Disclosed are a method and browser client for achieving browser security, relating to the technical field of browsers. The method comprises: when a browser is being installed, a system service that launches with the operating system is installed, by means of the browser installation package, on the operating system on which the browser is installed; the browser has a built-in security component; after launching, the browser invokes said system service by means of the security component; a first process independent of the browser process is intercepted from modifying browser installation files and/or browser data. In the method for achieving browser security of the present invention, a system service related to security is written to the logic of the browser, such that a security function becomes a function of the browser itself; the security component built in to the browser invokes said system service to protect the security of the browser itself, thus resolving the problem of a browser being unable to monitor and protect its security by its own means.

(57) 摘要:

[见续页]

WO 2016/086767 A1



RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG,

CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第 21 条(3))。

本发明公开了一种实现浏览器安全的方法、浏览器客户端，涉及浏览器技术领域。所述方法包括：在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。本发明的实现浏览器安全的方法，将与安全相关的系统服务写入浏览器的逻辑中，使安全功能成为浏览器本身的一个功能，通过浏览器内置的安全组件调用所述系统服务保护浏览器本身的安全，由此解决了浏览器无法通过自身对自己的安全性进行监控和保护的问题。

实现浏览器安全的方法、浏览器客户端和装置

技术领域

本发明涉及浏览器技术领域，具体涉及一种实现浏览器安全的客户端和一种带有浏览器客户端的装置。

5 背景技术

浏览器是指可以显示网页服务器或者文件系统的 HTML(超文本 HyperText Mark-up Language)文件内容，并让用户与这些文件交互。网页浏览器主要通过 HTTP 协议与网页服务器交互并获取网页，URL(统一资源定位符,Uniform Resource Locator)指定，文件格式通常

10 但是传统的浏览器对于自身的安全很少能进行监控和处理，第三方的杀毒软件对浏览器的进行安全保护，由于需要与其他软件进行将很多浏览器接口开放给第三方程序，而很多不安全的程序也可以访问，导致浏览器的信息和操作很容易被劫持，使用户在使用浏览器时的不安全性，其浏览器安全保护的自主性、灵活性差。

15 发明内容

鉴于上述问题，提出了本发明以便提供一种克服上述问题或者解决上述问题的浏览器客户端和相应的实现浏览器安全的方法。

依据本发明的一个方面，提供了一种实现浏览器安全的方法，

20 在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统启动而启动的系统服务；

在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，

拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

25 依据本发明的另外一个方面，提供了一种浏览器客户端，包括：

安装组件，其配置为在浏览器进行安装时，通过浏览器安装包在操作系统中安装一随操作系统启动而启动的系统服务；

安全组件，其配置为在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

30 依据本发明的另外一个方面，提供了一种带有浏览器客户端的装置，包括：处理器，以及加载有多条可执行指令的存储器，所述多条指令表示步骤的方法：

在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统启动而启动的系统服务；

在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

根据本发明的又一个方面，提供了一种计算机程序，其包括计算机可读代码，
5 当所述计算机可读代码在终端设备上运行时，导致所述终端设备执行上述的任一个实现浏览器安全的方法。

根据本发明的再一个方面，提供了一种计算机可读介质，其中存储了执行上述的任一个实现浏览器安全的方法的计算机程序。

10 根据本发明的实现浏览器安全的方法，可在传统浏览器的功能上，将与安全相关的系统服务写入浏览器的逻辑中，使安全功能成为浏览器本身的一个功能，通过浏览器内置的安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改，由此解决了浏览器无法通过自身对自己的安全性进行监控和保护的问题，取得了可以由浏览器自身对浏览器的安全
15 进行保护的有益效果。

上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，而可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂，以下特举本发明的具体实施方式。

附图说明

20 通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。
在附图中：

25 图 1 示出了根据本发明一个实施例的一种实现浏览器安全的方法的流程示意图；

图 2 示出了根据本发明一个实施例的一种实现浏览器安全的方法的流程示意图；

图 3 示出了根据本发明一个实施例的一种实现浏览器安全的方法的流程示意图；

30 图 4 示出了根据本发明一个实施例的一种浏览器客户端的结构示意图；

图 5 示出了根据本发明一个实施例的一种浏览器客户端的结构示意图；

图 6 示出了根据本发明一个实施例的一种浏览器客户端的结构示意图；

图 7 示出了根据本发明一个实施例的一种带有浏览器客户端的装置的结构示意图；

35 图 8 示出了用于执行根据本发明的方法的终端设备的框图；

图 9 示出了用于保持或者携带实现根据本发明的方法的程序代码的存储单元。

具体实施方式

下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

实施例一

参照图 1，其示出了本发明一种实现浏览器安全的方法的流程示意图，具体可以包括：

步骤 110，在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

在本发明实施例中，将浏览器中会内置一个安全组件，对应安全组件会设置一个系统服务，提供安全保护需要的系统权限，该系统服务在浏览器安装包中跟随浏览器的安装逻辑一起安装，该系统服务只与浏览器的安全组件进行内部交互，其不需要适于浏览器针对外部应用程序的接口，可以在浏览器内部进行安全保护。

那么，在本发明实施例中，在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务。在安装所述系统服务时，还可由浏览器安装包由所述系统服务控制安装对应的虚拟的设备级驱动程序，虚拟的设备级驱动程序属于内核级程序，其具有操作系统的最高权限，所述系统服务可以在需要时调用所述虚拟的设备级驱动程序去执行拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改的操作。

在系统服务安装时会在系统文件中生成一个 dll 文件，并将该 dll 的相关参数写入操作系统服务的注册表中。同时，会将虚拟的设备级驱动程序的 sys 文件安装至操作系统，并将 sys 文件的相关参数写入注册表中。操作系统启动后，会启动系统服务的 exe 文件，等待浏览器的安全组件的通知。

步骤 120，在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

本发明在浏览器传统的功能组件架构之上，还内置了一个安全组件，浏览器启动后通过该安全组件调用所述启动后的系统服务，以拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改，即对浏览器的相关数据进行保护。

优选地，所述通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

子步骤 131，通过该安全组件调用所述系统服务，控制虚拟的设备级驱动程

序拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

对于安全保护来说，很多的操作需要系统的高级权限，比如对于 windows 系统，其至少分为内核级权限和用户级权限，对于浏览器这个用户层的程序来说，其属于用户级权限，其操作受到很多限制，比如修改拦截其他进程的操作、修改某些注册表的操作在用户级权限的定义中均被认为是不允许的，其无法实现对浏览器的安全保护操作。那么本发明则可以通过系统服务去控制虚拟的设备级驱动，得到内核级权限，而内核级权限是最高权限，可以进行任意操作。因此，可以实现浏览器对自身的安全保护操作。

优选地，所述浏览器数据包括浏览器访问的网页数据。

10 在本发明实施例中，用户在使用浏览器访问网页过程中可从服务器获得的网页数据，那么本发明可对网页数据进行保护。

进一步的，所述拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

子步骤 132，针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。

15 在用户使用浏览器发起网络访问请求，获取网页数据进行解析、渲染的过程中，本发明实施例的安全组件则针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。在扫描时，可以根据预先在云端服务器搜集和统计得到的 url (Uniform Resource Locator: 统一资源定位符) 网址库，扫描所述网页的 url 是否为安全的 url，比如是否为诈骗的 url、钓鱼 url 等，如果 20 不安全，则可获取所述 url 对应的网页内容，提示用户关闭所述网页，当用户选择继续访问所述网页时才继续获取所述 url 对应的网页内容；也可以扫描所述网页内容中是否存在不安全的链接，比如分析所述网页内容中的广告部分内容的 url，根据前述 url 网址库判断所述广告 url 是否安全，如果不安全，则可暂停渲染所述广告内容部分或者将广告内容部分替换为安全内容，还可提示用户提示用户关闭所述网页，当用户选择继续访问所述网页时才继续访问所述 url。

也可以对网页中调用的的 js 文件进行判断，判断该 js 文件是否安全，如果不安全，则禁止所述 js 文件的调用。

当然对于不安全的 url 的处理可以将其放入沙箱中运行，即优选的，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

30 子步骤 133，判断当前打开的网页是否安全，如果不安全，则将所述网页对于的网页进程放入沙箱运行。

沙箱是一种按照安全策略限制程序行为的执行环境。由于每个网页数据的处理均需要在一个网页进程中执行，那么当判断该网页数据不安全时，可以将处理该网页数据的网页进程放入沙箱中运行，限制其运行权限。避免网页中的木马、 35 恶意脚本的执行而影响到本地系统的安全。

优选地，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数

据的修改包括：

子步骤 134，拦截独立于浏览器进程的第一进程对浏览器数据中的安全信息的获取；所述安全信息包括网址、下载文件、电话号码、公众号、即时聊天号其中至少一个。

5 在浏览器的使用过程中，其存在大量的数据，比如访问网页时的 url，收藏夹中收藏的 url，由浏览器的下载器所下载的文件，用户在网页中输入的电话号码、登录微博等社交网站的公众号、在网页中输入的即时聊天号、银行账户信息等个性化信息，以及 cookie 中记录的用户登录各个网站的登录账号和密码等登录信息。本发明对浏览器本身记录的上述个性化信息均可进行保护，拦截独立于浏览器进程的第一进程对所述个性化信息的获取。本发明可以监控在浏览器指定位置读取浏览器 cookie 信息的进程，或者监控读取收藏夹的 url 的进程是否为浏览器进程，如果不是，则可认为其是独立于浏览器进程的第一进程，那么拦截其获取动作。或者判断当前网页是否需要输入账号信息（公众号、银行账户、邮箱账户、即时通讯账户等信息）的网页，如果是，则判断是否有获取所述账号信息的进程是否为浏览器进程，如果不是，则可认为其是独立于浏览器进程的第一进程，那么拦截其获取动作。

10

15

优选地，还包括：

子步骤 135，调用系统服务获取浏览器的安全的更新文件，以进行更新。

在本发明实施例中，对于浏览器的更新，为了防止有被篡改了内容的更新文件，比如加入了木马的更新文件，在浏览器获取更新文件时获取到上述更新文件，本发明则调用系统服务获取浏览器的更新文件，因为系统服务本身具备较高的安全性，其获取更新文件时，其更新文件不容易被替换，同时也可检测所述更新文件是否是安全的更新文件，那么进行更新时，能对浏览器进行安全更新。

20

优选地，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

25 子步骤 136，拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置参数的修改。

在本发明实施例中，浏览器本身在操作系统中会进行相关的配置，比如设置为操作系统的默认浏览器，比如存储浏览器功能配置参数。

30 本发明则可拦截独立于浏览器进程的第一进程对上述这些操作系统中与浏览器相关的配置参数的修改。

进一步的，优选地，所述拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置信息的修改包括：

子步骤 137，拦截独立于浏览器的第二进程将当前操作系统中 HTTP 协议的关联处理程序从当前浏览器修改为其他处理程序的操作；

35 本发明可以拦截修改当前操作系统默认浏览器的操作，保证当前浏览器设置为操作系统的默认浏览器。在拦截过程中，可以拦截对注册表中

HKEY_CLASSES_ROOT\http\shell\open\command 子键的默认根值项和注册表中 KEY_CLASSES_ROOT\http\shell\open\ddeexec\Application 子键的默认根值项的修改操作。比如当有独立于浏览器的第二进程调用 RegSetValueEx()函数，修改上述注册表项时，则对该进程的调用进行拦截，不让其调用。

5 其中 RegSetValueEx()为注册表修改函数，其函数原型为：

```
RegSetValueEx(  
    HKEY hKey,//打开当前句柄，也可以是注册表五个根键之一  
    LPCTSTR lpValueName,//字符串类型指针，指向设置键值的值项名称  
    LPDWORD lpReserved,//保留置，通常为 0  
10    DWORD dwType,//要设置键值项数值的类型  
    const BYTE *lpData,//指向设置的数值所在的缓冲区指针，如果不设置可设  
    为 NULL  
    DWORD cbData);//指定 lpData 数据的缓冲区的长度，以字节为单位。  
    和/或，子步骤 138，拦截独立于浏览器的第二进程对当前浏览器功能的配置  
15 信息的修改。
```

另外，也可以对浏览器功能配置信息的修改，比如配置的浏览器的首页页信息，配置的是否打开广告过滤插件，配置的工具栏显示内容，配置的快捷键的功能等等功能配置信息。以浏览器首页为例，可以拦截修改注册表中浏览器首页键值的函数，独立于浏览器的第二进程可以先查找注册表中浏览器首页键值，比如 20 通过当 ADVAPI32！RegQueryValueExW 或者 SHDOCVW!URLSubRegQueryW 查询注册表首页键值，然后调用 RegSetValueEx()函数修改所述键值，本发明则可直接对浏览器的第二进程对上述函数的调用进行拦截。

优选地，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：
子步骤 138，拦截独立于浏览器进程的第一进程对当前浏览器记录的用户个
25 性化数据的修改。

在本发明实施例中，浏览器本地会记录很多用户的个性化数据，比如用户收藏的网页，在收藏是一般是以网页名字的形式显示 URL。那么可能有独立于浏览器进程的第一进程在网页名字的基础之下把内部的 URL 修改了，那么用户点在收藏夹中击该网页名字时，访问的并不是其收藏时的网页，而是修改后的网页，其 30 存在安全风险。另外，浏览器本地也可能存储 cookie 信息，而 cookie 中可能记录了用户访问的各种信息，比如访问了哪些网站、登录了哪些账户和密码等，那么独立于浏览器的进程如果获取到上述数据，则用户的个性化数据则泄密了。

那么，本发明则可拦截独立于浏览器进程的第一进程对上述浏览器记录的用户个性化数据的修改以及获取。

35 优选地，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：
子步骤 139，对于浏览器访问的网址，利用云杀毒引擎对所述网址进行安全

判定；如果所述网址不安全，则进行拦截。

在本发明实施例中，为了降低浏览器的规模，即避免浏览器对自己进行安全保护时，其文件特别大，本发明则结合了云杀毒引擎的方式，即安全组件获取到所述url后，通过所述安全模组调用云杀毒引擎，在云杀毒引擎中对所述url的安全性进行判断，然后云杀毒引擎将判断结果返回给安全模组，安全模组则分析所述判断结果，如果所述URL不安全，则通过系统服务拦截所述url的加载，进一步的，还可将所述url对应的网页进程放入沙箱中运行。当然，也可以提示用户的该url的不安全状态。

优选地，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

子步骤 140，对于通过浏览器下载的文件，利用云杀毒引擎对所述文件进行安全检测。

在用户使用浏览器的过程中，也可能通过浏览器的下载器以下载文件，本发明实施例也可对浏览器下载文件的过程进行安全保护。比如对浏览器触发的下载链接，通过云杀毒引擎判断所述下载链接是否安全，如果不安全，则通知安全组件，提示用户是否需要继续下载，安全组件同时通过系统服务对所述下载过程进行拦截。对于浏览器下载器中下载完成的文件，安全组件则可以通过系统服务获取所述文件的特征信息，系将所述特征信息上传至云杀毒引擎以判断所述文件是否安全，并将云杀毒引擎的判断结果返回浏览器的安全组件，然后安全组件则可在下载器中该文件的相应位置提示所述文件是否安全。

优选地，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

子步骤 141，当确定浏览器打开的网页为网购页面时，检测当前的网购环境是否安全；

在本发明实施例中对于浏览器打开的网页，还可判断该网页是否为网购页面，当确定浏览器打开的网页为网购页面时，检测系统环境是否安全。

具体的，本实施例中，根据预先收集的购物类网站集合，分析每个购物类网站的域名，提取每个购物类网站的网购特征词，得到网购特征词集合；例如，淘宝网的域名为 www.taobao.com，将 taobao 设为淘宝网的网购特征词添加到网购特征词集合；当用户通过终端浏览器当前登陆的网站域名为 paimai.taobao.com 时，由于该域名中包括的关键词 taobao 与网购特征词集合中包括的网购特征词 taobao 相匹配，则可以确定用户当前登陆的网站为购物类网站，同理，本实施例也可以根据预先收集的支付类网站集合，分析每个支付类网站的域名，提取每个支付类网站的支付特征词，得到支付特征词集合；例如，招商银行的域名为 www.cmbchina.com，将 cmbchina 设为招商银行网站的支付特征词，并添加到预设支付特征词集合中；当用户当前登陆的网站域名为 ccclub.cmbchina.com 时，由于该域名中包括的关键词 cmbchina 与支付特征词集合中包括的支付特征词 cmbchina 相匹配，则可以确定用户当前登陆的网站为支付类网站，其对应的网页

也可以理解为网页页面。

那么本发明则可以监控网购环境是否安全。比如判断本地系统环境是否安全，比如有独立于浏览器的进程获取所述网页中的信息，如果有则可将系统环境调整为针对网购页面的安全的系统环境。还比如检测网购页面本身的安全，比如根据5 所述网购页面所在网站的 IP 地址，若所述 IP 地址包括在 IP 地址黑名单中，则确定所述网站是危险网页，网购页面也是危险页面。又比如根据所述网购页面的统一资源定位符 URL，计算所述 URL 的哈希值，若计算的所述哈希值包括在哈希值黑名单中，则确定所述网购页面是危险页；举例来说，在实际应用中，根据黑名单网站列表中包括的危险网页的 URL，检测每个危险网页的 URL 的 refer 链地址，10 计算每个危险网页的 refer 链地址的哈希值，得到哈希值黑名单；因此，当前访问的网站是网购页面时，获取该网购页面的 URL 的 refer 链地址，计算该网购页面的 refer 链地址的哈希值，若该网购页面的 refer 链地址的哈希值在上述哈希值黑名单中，则确定该网购页面的危险概率较大。

进一步的，还包括：

15 子步骤 142，针对当前的网购页面，在所述安全的系统环境下生成保护单号。

在本发明实施例中，对于用户使用浏览器进行网购时，本发明可针对用户的网购行为进行额外赔偿保护，当用户在本发明的安全浏览器的架构下网购时被骗之后，本发明可以对其在当前的安全的系统环境下生成保护单号，记录用户的网购行为，并传输至服务器，如果用户被骗，则可以向服务器申请赔偿，服务器接收到所述申请后，则根据所述保护单号判断是否符合赔偿条件以进行赔偿。
20

优选地，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

子步骤 144，拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器。

在浏览器使用过程中，可能有其他程序向浏览器注入一些动态链接库，以控制浏览器执行自己需要的逻辑，比如网络请求被重定向到不安全网页、收藏夹里自动反复添加不安全网站、IE 选项卡中出现不能更改或被隐藏的项目、获取在网页中的登录名和密码等，因此，这些程序注入的动态链接库对于用户的浏览器来说并不安全。而本发明则可通过安全组件调用所述系统服务拦截上述独立于浏览器的第一进程向浏览器注入劫持浏览器的代码。
30

在本发明实施例中，所述对浏览器安装文件和/或浏览器数据的修改中的修改可以理解为对其进行篡改或者获取。

所述拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

子步骤 145，利用所述系统服务，通过调用一虚拟的设备级驱动程序拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。
35

在本发明实施例中，前述提及的拦截过程，可以通过所述系统服务调用虚拟

的设备级驱动进行拦截，其通过内核级权限进行拦截，保证拦截的成功率。

本发明实施例中，对于浏览器架构也进行了全新的改动，在浏览器的传统架构之上，即在浏览器传统的组件：用户界面组件，包括地址栏、后退/前进按钮、书签目录等，也就是除了用来显示你所请求页面的主窗口之外的其他部分；浏览器引擎组件，用来查询及操作渲染引擎的接口；渲染引擎组件，用来显示请求的内容，例如，如果请求内容为 html，它负责解析 html 及 css，并将解析后的结果显示出来；网络组件，用来完成网络调用，例如 http 请求；UI 后端组件，用来绘制类似组合选择框及对话框等基本组件；JS 解释器组件，用来解释执行 JS 代码；数据存储组件，浏览器需要在硬盘中保存类似 cookie 的各种数据；等组件的架构之上，添加了安全组件，并相应该安全组件设置了系统服务，该安全组件将上述组件工作过程中产生的数据通过系统服务进行保护，提高了浏览器安全保护的自主性、灵活性，不用依赖于第三方的杀毒软件。

实施例二

15 参照图 2，其示出了本发明一种实现浏览器安全的方法的流程示意图，具体可以包括：

步骤 210，在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

20 步骤 220，在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器，具体包括：

步骤 S222，复制当前浏览器的源分层服务提供商链表，获得第一分层服务提供商链表；

25 在实际应用中，其他应用程序可按照正常方式向浏览器注入 LSP (Layered Service Provider，分层服务提供商)节点，即向浏览器注入 LSP 的 DLL(Dynamic Link Library，动态链接库)，注入后会将 LSP 的 DLL 写入注册表中(比如写入注册表

30 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters 的相应位置中)，并将相关配置信息写入浏览器的源 LSP 链表的配置信息中，该配置信息中记录了上述 dll 的注册表位置等信息。那么，按照传统的方式，浏览器启动后，向外发送请求之前，会根据浏览器源 LSP 链表的配置信息加载源 LSP 链表，即加载 LSP 链表中各节点的 dll，然后浏览器的网络请求会从源 LSP 链表中的第一个 LSP 节点开始，向下逐个通过 LSP 节点进行传输，直到传输到其他协议层，比如 TCP/IP 协议层。

35 但是本发明在浏览器的第一个网络请求发出之前，会首先对源 LSP 链表进行转换。首先即复制一份源 LSP 链表，比如将源 LSP 链表中的有序的 dll 文件

复制一份，该复制版本作为第一 LSP 链表以备后续处理。

比如源 LSP 链表为：A.dll—>B.dll—>C.dll—>D.dll, 复制得到的第一 LSP 链表为 A.dll—>B.dll—>C.dll—>D.dll。当然，本发明实施例中可以通过浏览器的源 LSP 链表的配置信息，查找注册表中记录的各源节点的路径，然后通过所述路
5 径将源 LSP 链表的各个源节点进行复制。

步骤 S224，将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，得到转换后的第二分层服务提供商链表；所述虚拟分层服务提供商节点实现各分层服务提供商接口并返回空值；

将前述复制得到的第一 LSP 链表，可逐个判断所述第一 LSP 链表中的各个节
10 点是否为不允许访问的源节点。其中，对源节点的判断可以通过节点的名称进行判断，比如一个 LSP 节点的名称为 mswsock.dll，可以通过白名单或者说黑名单进行判断。比如将允许访问的源节点的名称写入白名单，那么第一 LSP 链表中的各节点不在白名单中时，即不允许访问，或者可以理解为不允许加载该 LSP 节点的 dll。在本发明实施例中，可以只将系统初始情况下默认的 LSP 节点名称写入白名单，当然还可以将其他安全的应用程序注入的 LSP 节点名称写入白名单，该白名单可以通过服务器进行更新。同理，也可以构建 LSP 节点的黑名单。
15

对于不允许访问的源节点，本发明实施例则将其转换为虚拟节点，即 fake.dll，该虚拟的 LSP 节点可以实现 LSP 的所有接口，那么该虚拟节点的上一个节点传输的网络请求可以正常访问该虚拟节点，该虚拟节点对网络请求的不进行
20 处理，即返回空值 NULL，然后继续将网络请求向下传输。因此该虚拟节点不会产生网络请求发送的异常，导致不能上网等情况。那么在将不允许访问的源节点替换为前述虚拟节点后，即得到第二 LSP 链表。

优选地，将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，包括：

25 子步骤 S2241，通过所述源分层服务提供商链表的配置信息，获得源分层服务提供商链表的各源节点的身份信息；

由于第一 LSP 链表中的各个节点与源 LSP 链表的各节点完全一致，那么即可通过读取浏览器中源 LSP 链表的配置信息，获得第一分层服务提供商链表的各源节点的身份信息。在源 LSP 链表的配置信息中，一般存储了源节点的身份信息，
30 比如对于每个节点记录的注册表项及记录的名称、顺序等信息，那么本发明实施例可以通过配置信息确定各个节点身份信息，比如其名称。比如上述例子中，可以获得各第一 LSP 链表中各个节点的身份信息按序为 A、B、C、D。

子步骤 S2242，将所述各源节点的身份信息与预置的身份信息名单进行匹配，根据匹配结果确定不允许访问的源节点；

35 在本发明实施例中，可以构建身份信息白名单或者身份信息黑名单，对所述各源节点的身份信息进行匹配。比如白名单中设置[A、D]，那么将 A、B、C、D

分别与上述白名单进行匹配后，确定名称 B、C 的源节点不允许访问。

子步骤 S2243，将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路径，得到第二分层服务提供商链表；

在本发明实施例中，可以预先设置虚拟节点，如 fake.dll，存储与指定路径。

而源节点如果要使用，则需要通过注册表中对应的注册表项所记录的源节点路径去加载所述源节点，那么可将不允许访问的源节点在注册表中对应注册表项的路径替换为虚拟节点的路径。

在本发明实施例中，可以针对所有不允许访问的源节点设置一个虚拟节点，将将不允许访问的源节点在注册表中对应注册表项的路径替换为该虚拟节点的路径，比如都替换为 fake.dll 的路径。当然，也可以根据确定的不允许访问的源节点个数，以初始设置的虚拟节点为蓝本，复制相应个数的虚拟节点，并将各个虚拟节点的文件名修改为不一样，比如前述例子有 B、C 两个节点，那么可以复制得到两个虚拟节点 fake1.dll、fake2.dll，各自有一个路径，那么 B.dll 的注册表路径修改为 fake1.dll 的路径，C.dll 的注册表路径修改为 fake2.dll 的路径。

如此，得到第二 LSP 链表，该链表的中允许加载的源节点保留，不允许加载的源节点即转换为了虚拟节点。

优选地，所述将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路径包括：

子步骤 A2242，所述系统服务接收安全组件向系统服务发送的注册表路径设置请求，并根据所述注册表路径设置请求创建 I/O 请求包下发至所述虚拟的设备级驱动程序；

在本发明实施例中，系统服务会随系统启动而启动，并一直维持运行，监听是否收到浏览器发送的请求，如果接收到浏览器发送的注册表路径设置请求，则会根据所述注册表路径设置请求创建 I/O 请求包（I/O Request Packet，IRP）下发至所述虚拟的设备级驱动。因为 windows 操作系统从应用层向底层驱动传送指令是通过 I/O 请求包传输的。系统服务调用本发明实施例中虚拟的设备级驱动，则标需要以所述设备级驱动为目构建 IRP，然后将所述 IRP 下发至所述设备级驱动中。所述 IRP 包括控制所述设备级驱动将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路径的指令，比如包括了不允许访问节点的注册表项信息，对应该不允许访问节点的虚拟节点的路径等信息。

子步骤 A2243，所述虚拟的设备级驱动程序接收到所述 I/O 请求包后，调用注册表修改函数将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路径。

所述虚拟的设备级确定程序接收到所述系统服务下发的 I/O 请求包后，解析所述 I/O 请求包中的指令，得到不允许访问节点的注册表项信息，以及对应该不允许访问节点的虚拟节点的路径信息，那么可以调用注册表修改函数，将该不允

许访问的源节点在注册表中的路径转换为虚拟节点的路径。

步骤 S226，将当前浏览器的网络请求通过所述第二分层服务提供商链表传输。

那么，对于浏览器的网络请求，即可控制其通过所述第二 LSP 链表进行传
5 输。

优选地，所述将当前浏览器的网络请求通过所述第二分层服务提供商链表传
输包括：

子步骤 S2261，通过所述源分层服务提供商链表的配置信息，从注册表查找
第二分层服务提供商链表各节点的动态链接库并进行加载。

10 由于本发明实施例没有修改浏览器的源分层服务提供商链表的配置信息，只是修改了与配置信息对应的节点路径以及节点内容，浏览器根据原 LSP 链表的配置信息去获取相应的 dll 时，对于替换了路径的源节点配置信息，其会从其注册表项中记录的路径加载虚拟节点，最终即加载了第二 LSP 链表，并未加载不允许访问的真实的源节点的 dll。

15 在本发明实施例中，可以安全组件可以通过系统服务，去调用虚拟的设备级驱动将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点。其中，可以通过注册表修改函数 RegSetValueEx() 函数将该不允许访问的源节点在注册表中的路径转换为虚拟节点的路径。

20 在浏览器的网络请求向外传输过程中，其需要先通过 LSP 链表的处理，才能向下传输至通信协议层（比如 TCP/IP 层），然后再传输至外部，那么传统技术可以向 LSP 链表中注入自定义的 LSP 节点，以对浏览器的网络请求进行劫持和处理，可能产生安全风险等问题。而本发明实施例中，无论其他应用程序如何注入 LSP 节点，本发明实施例中，在浏览器发送第一个网络请求之前，将系统中包括 25 应用程序注入的 LSP 节点的源 LSP 链表进行替换为第二 LSP 链表，其中将不需要访问的源节点替换为虚拟节点，完全不用理会会有多少个应用程序注入了多少个 LSP 节点，也可保证浏览器下发的网络请求通过安全的 LSP 链表进行传输，提高了浏览器的安全性。

实施例三

参照图 3，其示出了本发明一种实现浏览器安全的方法的流程示意图，具体可
30 以包括：

步骤 310，在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

步骤 320，在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器，具体包括：

35 步骤 S321，加载用于拦截窗口消息的窗口消息钩子函数；

在本发明实施例中加载 CBT 钩子函数 WH_CBT，该 WH_CBT 钩子函数当 windows 窗口激活、创建、释放（关闭）、最小化、最大化或改变窗口时的窗口消息都可通过该 WH_CBT 进行拦截。本发明则可以加载上述 CBT 钩子函数。

优选地，所述加载用于拦截窗口消息的窗口消息钩子函数，包括：

5 子步骤 S3211，调用动态链接库加载函数加载所述窗口消息钩子函数所在的动态链接库，以加载所述窗口消息钩子函数。

WH_CBT 需要通过 SetWindowsHookEx 函数进行安装，其函数原型为：

SetWindowsHookEx(

```
10    int idHook,
    HOOKPROC lpfn,
    HINSTANCE hMod,
    DWORD dwThreadId;
```

其中，int idHook = WH_CBT；

15 HOOKPROC lpfn 为/钩子过程的指针，也即拦截到指定系统消息后的预处理过程，须定义在 DLL 中；

HINSTANCE hMod，应用程序实例的句柄，可以为 CBT 钩子所在 DLL；

HINSTANCE hMod，该参数被设置为 0，表示此钩子为监视系统所有线程的全局钩子。

由于上述安装逻辑需要以 dll 的方式实现，而对于 dll 的加载和运行，则可通
20 过动态链接库加载函数 LoadLibrary 加载 CBT 钩子所在的 dll，并把 CBT 钩子的执行逻辑也一并加载。LoadLibrary 函数原型如下：

LoadLibraryA(

```
    _in LPCSTR lpLibFileName
);
```

25 其中 lpLibFileName 为 dll 的名称。

那么通过上述方式，把 CBT 钩子函数所在的 dll 进行加载，从而即加载了 CBT 钩子函数以及其钩取到窗口消息后的处理逻辑。

优选地，所述加载用于拦截窗口消息的窗口消息钩子函数，包括：

30 子步骤 A322，所述系统服务接收安全组件向系统服务发送的加载请求，根据所述加载请求创建 I/O 请求包下发至所述虚拟的设备级驱动程序；

在本发明实施例中，系统服务会随系统启动而启动，并一直维持运行，监听是否收到浏览器发送的请求，如果接收到浏览器发送的加载请求，则会根据所述加载请求创建 I/O 请求包（I/O Request Packet，IRP）下发至所述虚拟的设备级驱动。因为 windows 操作系统从应用层向底层驱动传送指令是通过 I/O 请求包传输的。系统服务调用本发明实施例中虚拟的设备级驱动，则标需要以所述设备级驱动为目构建 IRP，然后将所述 IRP 下发至所述设备级驱动中。所述 IRP 包括控制

所述设备级驱动加载 CBT 钩子函数的信息，比如 CBT 钩子函数所在 dll 的路径。

子步骤 A323，所述虚拟的设备级驱动程序接收到所述 I/O 请求包后，调用动态链接库加载函数加载用于拦截窗口消息的窗口消息钩子函数。

所述虚拟的设备级确定程序接收到所述系统服务下发的 I/O 请求包后，解析 5 所述 I/O 请求包中的指令，得到 CBT 钩子函数所在 dll 的信息，那么可以调用动态链接库加载函数，加载所述窗口消息钩子函数所在的动态链接库，以加载所述窗口消息钩子函数。通过上述方式，即加载 CBT 钩子函数。

步骤 S322，针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截；

10 在本发明实施例中，当有应用程序向浏览器注入不安全的动态链接库时，其是通过窗口消息控制操作系统向浏览器注入，那么本发明可以在其发送窗口消息时即可通过 CBT 钩子函数对其进行拦截。

步骤 S323，判断所述窗口消息是否为劫持浏览器的窗口消息；如果所述窗口消息是劫持浏览器的窗口消息，转入步骤 S324；

15 在本发明实施例中，可以根据拦截的窗口消息的窗口句柄判断其是否为劫持浏览器的窗口消息。

优选地，所述判断所述窗口消息是否为劫持浏览器的窗口消息，包括：

子步骤 S3231，将所述窗口消息所属的窗口句柄名与预置的窗口句柄名单 20 进行匹配；如果所述窗口句柄匹配上，则确定所述窗口消息为劫持浏览器的窗口消息。

在本发明实施例中，对于浏览器之外的其他应用程序，如果要向浏览器注入劫持浏览器的 dll，其需要通过窗口消息启动相应的窗口等操作，在该窗口之下发送执行 dll 注入过程，windows 系统则对接收到窗口消息进行处理，比如执行 dll 安装过程，将该 dll 写入浏览器指定位置，将 dll 的相关参数写入与浏览器相关的 25 注册表项中。而每个窗口均有窗口句柄，那么本发明可以预先对向浏览器注入不符合安全要求的 dll 的应用程序启动的窗口句柄进行统计，生成窗口句柄黑名单。那么本发明对于拦截到的窗口消息，可以直接通过所述窗口消息获取其所属的窗口句柄，将其与黑名单中的窗口句柄进行匹配，如果匹配上，则确定所述窗口消息为劫持浏览器的窗口消息，即可以通过窗口句柄的匹配结果确定所述窗口消息是否为劫持浏览器的窗口消息。

当然，本发明预置的窗口句柄名单，可以不断根据对应用程序的分析进行更新，其可以通过云服务器更新到客户端中。

优选地，还包括：

子步骤 S3232，获取所述窗口句柄所属应用程序的验证签名；

子步骤 S3233，对所述验证签名进行验证，如果所述验证失败，则确定所述窗口消息为劫持浏览器才窗口消息。

如果验证成功，则放行所述窗口消息。

在本发明实施例中，对于窗口消息，在判断其窗口句柄在预置的窗口句柄名单之内后，还可以获取所述窗口句柄所属应用程序的验证签名，比如第三方安全平台的验证签名，然后对该数字签名与预先记录的验证签名进行匹配，如果匹配上，则说明该窗口句柄的应用程序安装的dll安全，可以允许其进行安装，如果验证失败，则可认为该窗口句柄的应用程序安装的dll不安全，拒绝其进行安装。当然，所述验证签名也可以通过云端服务器进行更新。

子步骤 S3231、子步骤 S3232、子步骤 S3233 的组合可对窗口消息进行多重判断，使对窗口消息的拦截范围可以灵活的进行配置，允许安全的应用程序向浏览器注入 dll，不允许不安全的应用程序向浏览器注入 dll，也保护了浏览器的安全性。

优选地，所述针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截包括：

子步骤 S3234，针对操作系统中的创建窗口的窗口消息，通过所述窗口消息钩子函数进行拦截。

在本发明实施例中，可以理解，当一个应用程序要向浏览器注入 dll 时，其需要执行安装过程，而安装过程在 windows 系统首先需要创建一个安装用的窗口，本发明实施例则可只拦截创建窗口的窗口消息，本发明实施例则可只拦截创建窗口的窗口消息，可判断其是否为向浏览器注入不安全的 dll 的应用程序的窗口消息。

优选地，所述针对操作系统中的创建窗口的窗口消息，通过所述窗口消息钩子函数进行拦截包括：

子步骤 321，针对操作系统中的创建窗口的 WM_CREATE 消息，通过所述窗口消息钩子函数进行拦截。

WM_CREATE 是 windows 中一个窗口消息，当一个应用程序通过 CreateWindowEx 函数或者 CreateWindow 函数请求创建窗口时发送此消息。那么应用程序创建向浏览器注入 dll 的安装窗口时，也会发送 WM_CREATE 消息。那么本发明即可通过 CBT 钩子就可拦截到应用程序创建的所述安装窗口的 WM_CREATE 消息。

本实施例则可只拦截创建窗口的窗口消息，当其为预先记录的要向浏览器注入不安全 dll 的应用程序发送的创建窗口的消息，则可停止对应窗口的创建，从而避免应用程序将不安全的 dll 注入浏览器。并且由于只拦截创建窗口的窗口消息，不拦截其他类型的窗口消息，降低了拦截的范围，避免占用过多的系统资源。

步骤 S324，停止所述窗口消息的传输。

如果所述窗口消息不是劫持浏览器的窗口消息，则放行所述窗口消息。

那么对于确定 CBT 钩子拦截的窗口消息为劫持浏览器的窗口消息后，即可停

止该消息的后续传输过程，不让其进行后续处理。比如将所述窗口消息删除。

当然，确定所述窗口消息为劫持浏览器的窗口消息后，还可生成弹出框，提示用户有应用程序向浏览器注入不安全的 dll，等待用户选择是否运行该窗口消息继续传输，如果用户选择继续传输，则放弃拦截，如果用户选择不继续传输，则
5 可停止所述窗口消息的传输。

本发明实施例可针对想将 dll 注入浏览器的应用程序，在其创建窗口、或者在其所在窗口之下发送窗口消息时即对其进行拦截，即在应用程序执行具体的 dll 注入过程之前就进行拦截，然后对窗口消息进行判断，当根据窗口消息判断其为劫持浏览器的消息时，则停止窗口消息的传输，不让其进行后续操作，可以直接防
10 止所述应用程序对浏览器注入不安全的 dll，从而保护了浏览器的安全性。

实施例四

参照图 4，其示出了本发明一种浏览器客户端的结构示意图，具体可以包括：

安装组件 410，其配置为在浏览器进行安装时，通过浏览器安装包在浏览器
15 所在操作系统中安装一随操作系统启动而启动的系统服务；

安全组件 420，其配置为在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

优选地，所述浏览器数据包括浏览器访问的网页数据，

20 进一步的，所述安全组件包括：

网页安全模组，其配置为针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。

优选地，所述安全组件包括：

25 安全信息拦截模组，其配置为拦截独立于浏览器进程的第一进程对浏览器数据中的安全信息的获取；所述安全信息包括网址、下载文件、电话号码、公众号、即时聊天号其中至少一个。

优选地，还包括：

安全更新模组，其配置为调用系统服务获取浏览器的安全的更新文件，以进行更新。

30 优选地，所述安全组件包括：

配置保护模组，其配置为拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置参数的修改。

优选地，所述配置保护模组包括：

35 默认浏览器保护模组，其配置为拦截独立于浏览器的第二进程将当前操作系统中 HTTP 协议的关联处理程序从当前浏览器修改为其他处理程序的操作；

和/或，浏览器功能配置保护模组，其配置为拦截独立于浏览器的第二进程对

当前浏览器功能的配置信息的修改。

优选地，所述安全组件包括：

个性化数据保护模组，其配置为拦截独立于浏览器进程的第一进程对当前浏览器记录的用户个性化数据的修改。

5 优选地，所述安全组件包括：

网址云保护模组，其配置为对于浏览器访问的网址，利用云杀毒引擎对所述网址进行安全判定；如果所述网址不安全，则进行拦截。

优选地，所述安全组件包括：

10 下载文件保护模组，其配置为对于通过浏览器下载的文件，利用云杀毒引擎对所述文件进行安全检测。

优选地，所述安全组件包括：

网购保护模组，其配置为当确定浏览器打开的网页为网购页面时，检测当前的网购环境是否安全；

进一步的，还包括：

15 保护单号生成模组，其配置为针对当前的网购页面，在所述安全的系统环境生成保护单号。

优选地，所述安全组件包括：

沙箱运行模组，其配置为判断当前打开的网页是否安全，如果不安全，则将所述网页对应的网页进程放入沙箱运行。

20 优选地，所述安全组件包括：

注入拦截模组，其配置为拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器。

优选地，所述安全组件包括：

25 第一安全模组，其配置为利用所述系统服务，通过调用一虚拟的设备级驱动程序拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

实施例五

参照图5，其示出了本发明一种浏览器客户端的结构示意图，具体可以包括：

30 安装组件510，其配置为在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

安全组件520，其配置为在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改，包括：

35 注入拦截模组521，其配置为拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器，其中包括：

链表复制模组5211，其配置为复制当前浏览器的源分层服务提供商链

表，获得第一分层服务提供商链表；

链表转换模组 5212，其配置为将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，得到转换后的第二分层服务提供商链表；所述虚拟分层服务提供商节点实现各分层服务提供商接口并返回空值；

5 请求控制模组 5213，其配置为将当前浏览器的网络请求通过所述第二分层服务提供商链表传输。

优选地，所述链表转换模组 5212 包括：

节点身份获取模组，其配置为通过所述源分层服务提供商链表的配置信息，获得源分层服务提供商链表的各源节点的身份信息；

10 节点身份确定模组，其配置为将所述各源节点的身份信息与预置的身份信息名单进行匹配，根据匹配结果确定不允许访问的源节点；

节点转换模组，其配置为将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路径。

优选地，所述链表转换模组 5212 包括：

15 请求接收模组，其配置为所述第一操作系统服务接收安全模组发送的注册表路径设置请求，并根据所述注册表路径设置请求创建 I/O 请求包下发至所述虚拟的设备级驱动程序；

第二转换模组，其配置为所述虚拟的设备级驱动程序接收到所述 I/O 请求包后，调用注册表修改函数将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路径。

实施例六

参照图 6，其示出了本发明一种浏览器客户端的结构示意图，具体可以包括：

安装组件 610，其配置为在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

25 安全组件 620，其配置为在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改，包括：

注入拦截模组 621，其配置为拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器，具体包括：

30 钩子加载模组 6212，其配置为加载用于拦截窗口消息的窗口消息钩子函数；

窗口信息拦截模组 6213，其配置为针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截；

窗口信息判断模组 6214，其配置为判断所述窗口消息是否为劫持浏览器的窗口消息；

35 窗口信息处理模组 6215，其配置为如果所述窗口消息是劫持浏览器的窗口消息，则停止所述窗口消息的传输。

优选地，所述钩子加载模组包括：

第一钩子加载模组，其配置为调用动态链接库加载函数加载所述窗口消息钩子函数所在的动态链接库，以加载所述窗口消息钩子函数。

优选地，所述窗口信息判断模组包括：

5 句柄匹配模组，其配置为将所述窗口消息所属的窗口句柄名与预置的窗口句柄名单进行匹配；如果所述窗口句柄匹配上，则确定所述窗口消息为劫持浏览器的窗口消息。

优选地，还包括：

签名获取模组，其配置为获取所述窗口句柄所属应用程序的验证签名；

10 签名验证模组，其配置为对所述验证签名进行验证，如果所述验证失败，则确定所述窗口消息为劫持浏览器才窗口消息。

优选地，所述窗口信息拦截模组包括：

创建拦截模组，其配置为针对操作系统中的创建窗口的窗口消息，通过所述窗口消息钩子函数进行拦截。

15 优选地，所述钩子加载模组 6212 包括：

请求接收模组，其配置为第一操作系统服务接收安全组件发送的加载请求，并根据所述加载请求创建 I/O 请求包下发至所述虚拟的设备级驱动程序；

驱动加载模组，其配置为所述虚拟的设备级驱动程序接收到所述 I/O 请求包后，调用动态链接库加载函数加载用于拦截窗口消息的窗口消息钩子函数。

20 实施例七

参照图 7，其示出了本发明一种带有浏览器客户端的装置的结构示意图，所述装置 700 具体可以包括：

处理器 710，以及加载有多条可执行指令的存储器 720，所述多条指令包括执行以下步骤的方法：

25 在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

30 所述浏览器数据包括浏览器访问的网页数据，

进一步的，所述拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。

35 优选地，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

拦截独立于浏览器进程的第一进程对浏览器数据中的安全信息的获取；所述

安全信息包括网址、下载文件、电话号码、公众号、即时聊天号其中至少一个。

优选地，还包括：调用系统服务获取浏览器的安全的更新文件，以进行更新。

优选地，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置参数的
5 修改。

优选地，拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置信息的修改包括：

拦截独立于浏览器的第二进程将当前操作系统中 HTTP 协议的关联处理程序从当前浏览器修改为其他处理程序的操作；

10 和/或，拦截独立于浏览器的第二进程对当前浏览器功能的配置信息的修改。

优选地，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器。

15 优选地，所述拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器包括：

复制当前浏览器的源分层服务提供商链表，获得第一分层服务提供商链表；

将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，得到转换后的第二分层服务提供商链表；所述虚拟分层服务提供商节点实现各分层服务提供商接口并返回空值；

20 将当前浏览器的网络请求通过所述第二分层服务提供商链表传输。

优选地，所述拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器包括：

加载用于拦截窗口消息的窗口消息钩子函数；

针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截；

25 判断所述窗口消息是否为劫持浏览器的窗口消息；

如果所述窗口消息是劫持浏览器的窗口消息，则停止所述窗口消息的传输。

当然，所述多条指令还包括执行前述各个步骤的方法。

在此提供的算法和显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述，构造这类系统所要求的结构是显而易见的。此外，本发明也不针对任何特定编程语言。

30 应当明白，可以利用各种编程语言实现在此描述的本发明的内容，并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

在此处所提供的说明书中，说明了大量具体细节。然而，能够理解，本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中，并未详细示出公知的方法、结构和技术，以便不模糊对本说明书的理解。

类似地，应当理解，为了精简本公开并帮助理解各个发明方面中的一个或多

个，在上面对本发明的示例性实施例的描述中，本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而，并不应将该公开的方法解释成反映如下意图：即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说，如下面的权利要求书所反映的那样，发明方面在于 5 少于前面公开的单个实施例的所有特征。因此，遵循具体实施方式的权利要求书由此明确地并入该具体实施方式，其中每个权利要求本身都作为本发明的单独实施例。

本领域那些技术人员可以理解，可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例 10 中的模块或单元或组件组合成一个模块或单元或组件，以及此外可以把它们分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外，可以采用任何组合对本说明书（包括伴随的权利要求、摘要和附图）中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述，本说明书（包括伴随的权利要求、摘要和附图）中 15 公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

此外，本领域的技术人员能够理解，尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征，但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如，在下面的权利要求书中，所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

20 本发明的各个部件实施例可以以硬件实现，或者以在一个或者多个处理器上运行的软件模块实现，或者以它们的组合实现。本领域的技术人员应当理解，可以在实践中使用微处理器或者数字信号处理器（DSP）来实现根据本发明实施例的实现浏览器安全设备中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序（例如，计算机程序和计算机程序产品）。这样的实现本发明的程序可以存储在计算机可读介质上，或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到，或者在载体信号上提供，或者以任何其他形式提供。

25 例如，图8示出了可以实现根据本发明的一种实现浏览器安全的终端设备。该终端设备传统上包括处理器810和以存储器820形式的计算机程序产品或者计算机可读介质。存储器820可以是诸如闪存、EEPROM（电可擦除可编程只读存储器）、EPROM、硬盘或者ROM之类的电子存储器。存储器820具有用于执行上述方法中的任何方法步骤的程序代码831的存储空间830。例如，用于程序代码的存储空间830可以包括分别用于实现上面的方法中的各种步骤的各个程序代码831。这些程序代码可以从一个或者多个计算机程序产品中读出或者写入到这 30 一个或者多个计算机程序产品中。这些计算机程序产品包括诸如硬盘，致密盘

(CD)、存储卡或者软盘之类的程序代码载体。这样的计算机程序产品通常为如参考图9所述的便携式或者固定存储单元。该存储单元可以具有与图8的终端设备中的存储器820类似布置的存储段、存储空间等。程序代码可以例如以适当形式进行压缩。通常，存储单元包括计算机可读代码831'，即可以由例如诸如810 5 之类的处理器读取的代码，这些代码当由终端设备运行时，导致该终端设备执行上面所描述的方法中的各个步骤。

应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制，并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中，不应将位于括号之间的任何参考符号构成对权利要求的限制。单10 词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中，这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

15 此外，还应当注意，本说明书中使用的语言主要是为了可读性和教导的目的而选择的，而不是为了解释或者限定本发明的主题而选择的。因此，在不偏离所附权利要求书的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。对于本发明的范围，对本发明所做的公开是说明性的，而非限制性的，本发明的范围由所附权利要求书限定。

1、一种实现浏览器安全的方法，包括：

在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，
5 系统服务，

拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

2、如权利要求 1 所述的方法，其特征在于，所述浏览器数据包括浏览器访问的网页数据，
10 进一步的，所述拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。

3、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：
15

拦截独立于浏览器进程的第一进程对浏览器数据中的安全信息的获取；所述安全信息包括网址、下载文件、电话号码、公众号、即时聊天号其中至少一个。

4、如权利要求 1 所述的方法，其特征在于，还包括：
20 调用系统服务获取浏览器的安全的更新文件，以进行更新。

5、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置参数的修改。

25 6、如权利要求 5 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置信息的修改包括：

拦截独立于浏览器的第二进程将当前操作系统中 HTTP 协议的关联处理程序从当前浏览器修改为其他处理程序的操作；

和/或，拦截独立于浏览器的第二进程对当前浏览器功能的配置信息的修改。

7、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

5 拦截独立于浏览器进程的第一进程对当前浏览器记录的用户个性化数据的修改。

8、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

对于浏览器访问的网址，利用云杀毒引擎对所述网址进行安全判定；如果所述网址不安全，则进行拦截。

9、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

对于通过浏览器下载的文件，利用云杀毒引擎对所述文件进行安全检测。

15 10、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

当确定浏览器打开的网页为网购页面时，检测当前的网购环境是否安全；；

进一步的，还包括：针对当前的网购页面，在所述安全的系统环境生成
20 保护单号。

11、如权利要求 2 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

判断当前打开的网页是否安全，如果不安全，则将所述网页对应的网页进程放入沙箱运行。

25 12、如权利要求 1 所述的方法，其特征在于，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器。

13、如权利要求 12 所述的方法，其特征在于，所述拦截独立于浏览器

的第一进程向浏览器注入代码以劫持浏览器包括:

 复制当前浏览器的源分层服务提供商链表，获得第一分层服务提供商链表；

 将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，得到转换后的第二分层服务提供商链表；所述虚拟分层服务提供商节点实现各分层服务提供商接口并返回空值；

 将当前浏览器的网络请求通过所述第二分层服务提供商链表传输。

14、如权利要求 13 所述的方法，其特征在于，将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，包括：

10 通过所述源分层服务提供商链表的配置信息，获得源分层服务提供商链表的各源节点的身份信息；

 将所述各源节点的身份信息与预置的身份信息名单进行匹配，根据匹配结果确定不允许访问的源节点；

15 将所述不允许访问的源节点在注册表中的路径转换为虚拟节点的路
径。

15、如权利要求 12 所述的方法，其特征在于，所述拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器包括：

 加载用于拦截窗口消息的窗口消息钩子函数；

 针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截；

20 判断所述窗口消息是否为劫持浏览器的窗口消息；

 如果所述窗口消息是劫持浏览器的窗口消息，则停止所述窗口消息的
传输。

16、如权利要求 15 所述的方法，其特征在于，所述加载用于拦截窗口
消息的窗口消息钩子函数，包括：

25 调用动态链接库加载函数加载所述窗口消息钩子函数所在的动态链接
库，以加载所述窗口消息钩子函数。

17、如权利要求 16 所述的方法，其特征在于，所述判断所述窗口消息
是否为劫持浏览器的窗口消息，包括：

将所述窗口消息所属的窗口句柄名与预置的窗口句柄名单进行匹配；如果所述窗口句柄匹配上，则确定所述窗口消息为劫持浏览器的窗口消息。

18、如权利要求 16 所述的方法，其特征在于，还包括：

5 获取所述窗口句柄所属应用程序的验证签名；

对所述验证签名进行验证，如果所述验证失败，则确定所述窗口消息为劫持浏览器才窗口消息。

19、如权利要求 18 所述的方法，其特征在于，所述针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截包括：

10 针对操作系统中的创建窗口的窗口消息，通过所述窗口消息钩子函数进行拦截。

20、如权利要求 1 所述的方法，其特征在于，所述拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

15 利用所述系统服务，通过调用一虚拟的设备级驱动程序拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

21、一种浏览器客户端，包括：

安装组件，其配置为在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

20 安全组件，其配置为在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

22、如权利要求 21 所述的浏览器客户端，其特征在于，所述浏览器数据包括浏览器访问的网页数据，

进一步的，所述安全组件包括：

25 网页安全模组，其配置为针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。

23、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件包括：

安全信息拦截模组，其配置为拦截独立于浏览器进程的第一进程对浏览器数据中的安全信息的获取；所述安全信息包括网址、下载文件、电话号码、公众号、即时聊天号其中至少一个。

24、如权利要求 21 所述的浏览器客户端，其特征在于，还包括：

5 安全更新模组，其配置为调用系统服务获取浏览器的安全的更新文件，以进行更新。

25、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件包括：

10 配置保护模组，其配置为拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置参数的修改。

26、如权利要求 25 所述的浏览器客户端，其特征在于，所述配置保护模组包括：

15 默认浏览器保护模组，其配置为拦截独立于浏览器的第二进程将当前操作系统中 HTTP 协议的关联处理程序从当前浏览器修改为其他处理程序的操作；

和/或，浏览器功能配置保护模组，其配置为拦截独立于浏览器的第二进程对当前浏览器功能的配置信息的修改。

27、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件包括：

20 个性化数据保护模组，其配置为拦截独立于浏览器进程的第一进程对当前浏览器记录的用户个性化数据的修改。

28、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件包括：

25 网址云保护模组，其配置为对于浏览器访问的网址，利用云杀毒引擎对所述网址进行安全判定；如果所述网址不安全，则进行拦截。

29、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件包括：

下载文件保护模组，其配置为对于通过浏览器下载的文件，利用云杀毒

引擎对所述文件进行安全检测。

30、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件包括：

网购保护模组，其配置为当确定浏览器打开的网页为网购页面时，检测
5 当前的网购环境是否安全；；

进一步的，还包括：

保护单号生成模组，其配置为针对当前的网购页面，在所述安全的系统
环境生成保护单号。

31、如权利要求 22 所述的浏览器客户端，其特征在于，所述安全组件
10 包括：

沙箱运行模组，其配置为判断当前打开的网页是否安全，如果不安全，
则将所述网页对应的网页进程放入沙箱运行。

32、如权利要求 22 所述的浏览器客户端，其特征在于，所述安全组件
包括：

15 注入拦截模组，其配置为拦截独立于浏览器的第一进程向浏览器注入代
码以劫持浏览器。

33、如权利要求 32 所述的方法，其特征在于，所述注入拦截模组包
括：

20 链表复制模组，其配置为复制当前浏览器的源分层服务提供商链表，获
得第一分层服务提供商链表；

链表转换模组，其配置为将所述第一分层服务提供商链表中不允许访问
的源节点转换为虚拟节点，得到转换后的第二分层服务提供商链表；所述
虚拟分层服务提供商节点实现各分层服务提供商接口并返回空值；

25 请求控制模组，其配置为将当前浏览器的网络请求通过所述第二分层服
务提供商链表传输。

34、如权利要求 33 所述的浏览器客户端，其特征在于，所述链表转换
模组包括：

节点身份获取模组，其配置为通过所述源分层服务提供商链表的配置信

息，获得源分层服务提供商链表的各源节点的身份信息；

节点身份确定模组，其配置为将所述各源节点的身份信息与预置的身份信息名单进行匹配，根据匹配结果确定不允许访问的源节点；

节点转换模组，其配置为将所述不允许访问的源节点在注册表中的路径
5 转换为虚拟节点的路径。

35、如权利要求 32 所述的浏览器客户端，其特征在于，所述注入拦截
模组包括：

钩子加载模组，其配置为加载用于拦截窗口消息的窗口消息钩子函数；

10 窗口信息拦截模组，其配置为针对操作系统中的窗口消息，通过所述窗
口消息钩子函数进行拦截；

窗口信息判断模组，其配置为判断所述窗口消息是否为劫持浏览器的窗
口消息；

窗口信息处理模组，其配置为如果所述窗口消息是劫持浏览器的窗口消
息，则停止所述窗口消息的传输。

15 36、如权利要求 35 所述的浏览器客户端，其特征在于，所述钩子加载
模组包括：

第一钩子加载模组，其配置为调用动态链接库加载函数加载所述窗口消
息钩子函数所在的动态链接库，以加载所述窗口消息钩子函数。

20 37、如权利要求 36 所述的浏览器客户端，其特征在于，所述窗口信息
判断模组包括：

句柄匹配模组，其配置为将所述窗口消息所属的窗口句柄名与预置的窗
口句柄名单进行匹配；如果所述窗口句柄匹配上，则确定所述窗口消息为
劫持浏览器的窗口消息。

38、如权利要求 36 所述的浏览器客户端，其特征在于，还包括：

25 签名获取模组，其配置为获取所述窗口句柄所属应用程序的验证签名；

签名验证模组，其配置为对所述验证签名进行验证，如果所述验证失
败，则确定所述窗口消息为劫持浏览器才窗口消息。

39、如权利要求 38 所述的浏览器客户端，其特征在于，所述窗口信息

拦截模组包括:

创建拦截模组，其配置为针对操作系统中的创建窗口的窗口消息，通过所述窗口消息钩子函数进行拦截。

40、如权利要求 21 所述的浏览器客户端，其特征在于，所述安全组件
5 包括：

第一安全模组，其配置为利用所述系统服务，通过调用一虚拟的设备级驱动程序拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

41、一种带有浏览器客户端的装置，包括：

10 处理器，以及加载有多条可执行指令的存储器，所述多条指令包括执行以下步骤的方法：

在浏览器进行安装时，通过浏览器安装包在浏览器所在操作系统中安装一随操作系统启动而启动的系统服务；

15 在浏览器中内置一安全组件，浏览器启动后通过该安全组件调用所述系统服务，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改。

42、如权利要求 41 所述的装置，其特征在于，所述浏览器数据包括浏览器访问的网页数据，

20 进一步的，所述拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

针对浏览器访的网页数据，通过所述安全组件调用系统服务，对所述网页数据进行安全扫描。

43、如权利要求 41 所述的装置，其特征在于，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

25 拦截独立于浏览器进程的第一进程对浏览器数据中的安全信息的获取；所述安全信息包括网址、下载文件、电话号码、公众号、即时聊天号其中至少一个。

44、如权利要求 41 所述的装置，其特征在于，还包括：

调用系统服务获取浏览器的安全的更新文件，以进行更新。

45、如权利要求 41 所述的装置，其特征在于，拦截独立于浏览器进程的第一进程对浏览器数据的修改包括：

拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置参数的修改。

46、如权利要求 45 所述的装置，其特征在于，拦截独立于浏览器进程的第一进程对操作系统中与浏览器相关的配置信息的修改包括：

拦截独立于浏览器的第二进程将当前操作系统中 HTTP 协议的关联处理程序从当前浏览器修改为其他处理程序的操作；

10 和/或，拦截独立于浏览器的第二进程对当前浏览器功能的配置信息的修改。

47、如权利要求 41 所述的装置，其特征在于，拦截独立于浏览器进程的第一进程对浏览器安装文件和/或浏览器数据的修改包括：

拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器。

15 48、如权利要求 47 所述的装置，其特征在于，所述拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器包括：

复制当前浏览器的源分层服务提供商链表，获得第一分层服务提供商链表；

20 将所述第一分层服务提供商链表中不允许访问的源节点转换为虚拟节点，得到转换后的第二分层服务提供商链表；所述虚拟分层服务提供商节点实现各分层服务提供商接口并返回空值；

将当前浏览器的网络请求通过所述第二分层服务提供商链表传输。

49、如权利要求 47 所述的装置，其特征在于，所述拦截独立于浏览器的第一进程向浏览器注入代码以劫持浏览器包括：

25 加载用于拦截窗口消息的窗口消息钩子函数；

针对操作系统中的窗口消息，通过所述窗口消息钩子函数进行拦截；

判断所述窗口消息是否为劫持浏览器的窗口消息；

如果所述窗口消息是劫持浏览器的窗口消息，则停止所述窗口消息的

传输。

50、一种计算机程序，包括计算机可读代码，当所述计算机可读代码在终端设备上运行时，导致所述终端设备执行根据权利要求 1-20 中的任一个所述的浏览器防注入方法。

5 51、一种计算机可读介质，其中存储了如权利要求 50 所述的计算机程序。

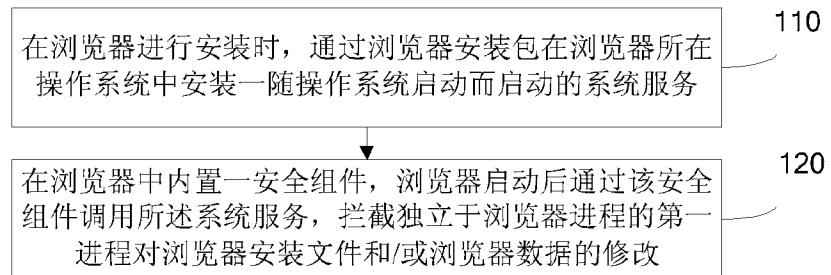


图 1

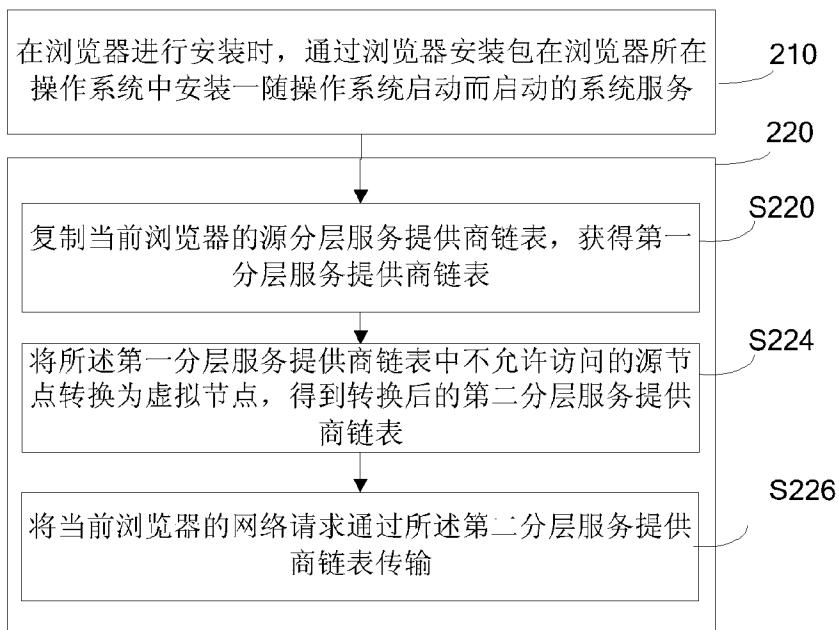


图 2

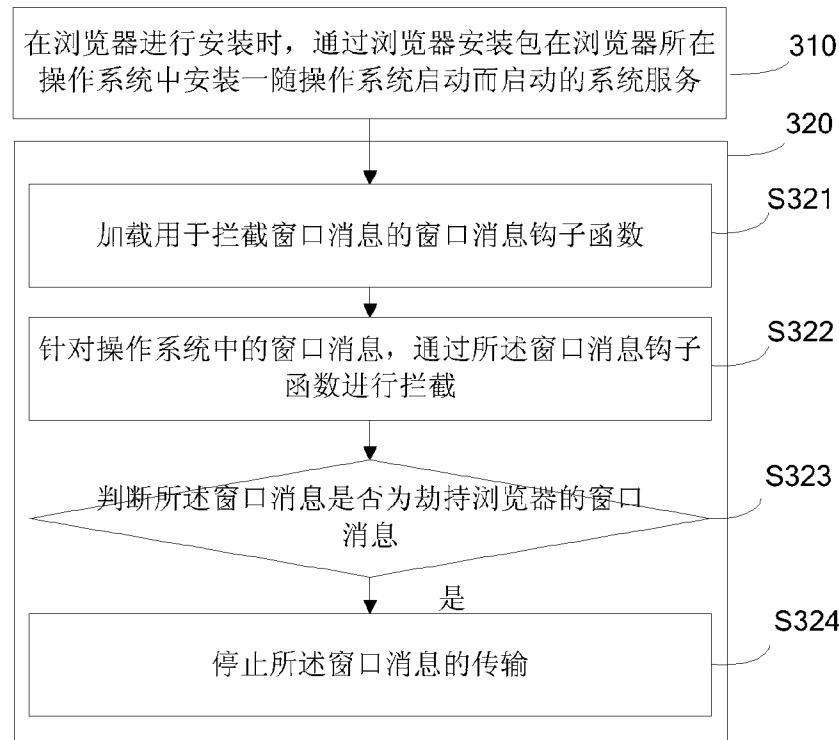


图 3



图 4



图 5



图 6

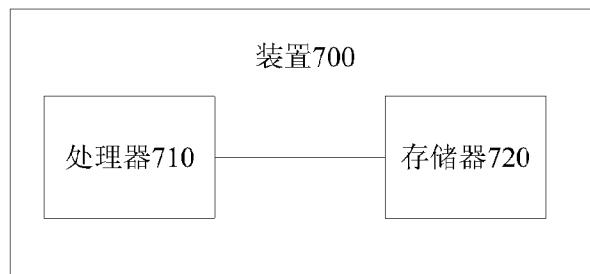


图 7

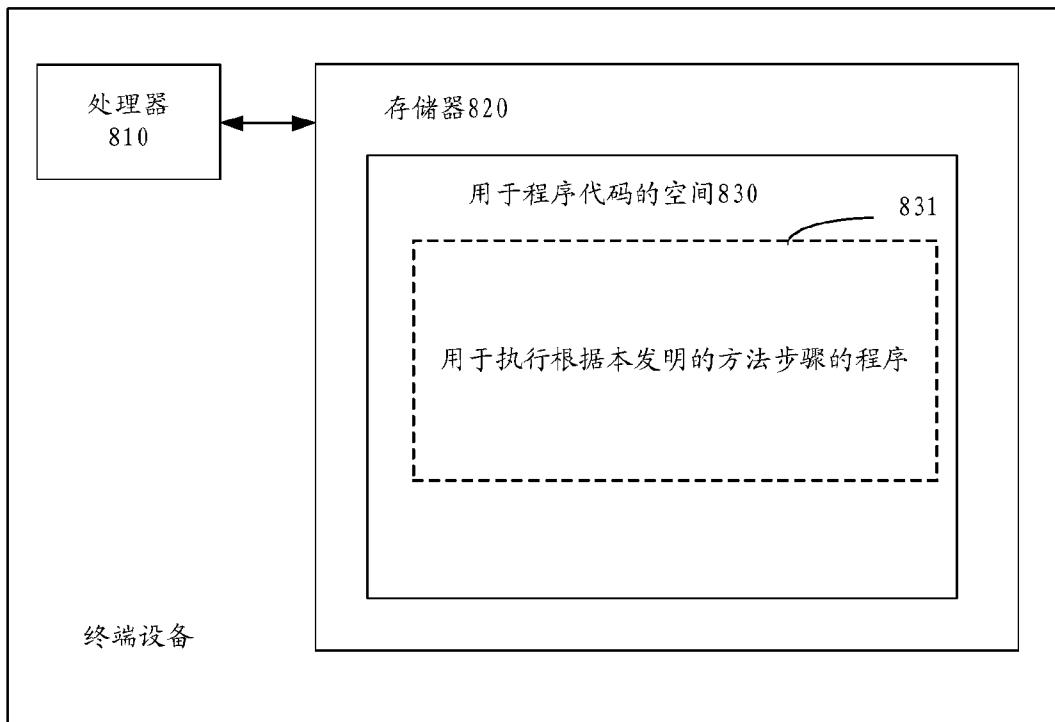


图 8

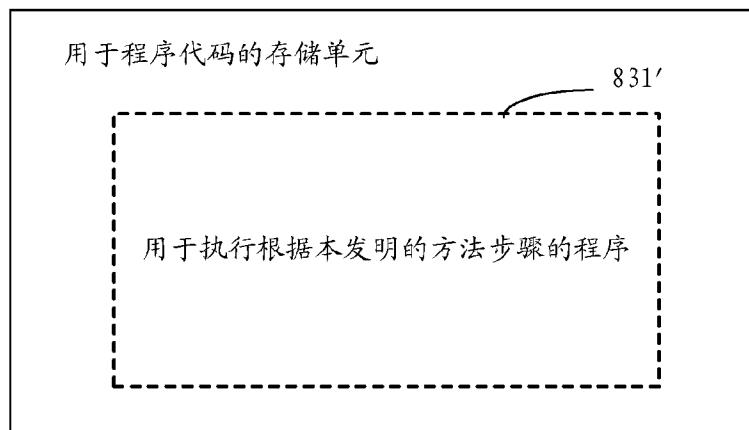


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2015/094845

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/50 (2013.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F 21/-

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT; CNABS; VEN; CNKI: browser, safety, secure, protect, prevent, intercept, embed, inject, built in, invoke, call

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 104536981 A (BEIJING QIHOO SCI & TECHNOLOGY CO LTD) 22 April 2015 (22.04.2015) claims 1-10, description paragraphs [150]-[0536]	1-51
X	CN 103823873 A (BEIJING QIHOO SCI & TECHNOLOGY CO LTD) 28 May 2014 (28.05.2015) description paragraphs [0055]-[0076], [0088]-[0092] and [0108]	1-12,15-32,35-47,49-51
X	CN 103823873 A (BEIJING QIHOO SCI & TECHNOLOGY CO LTD) 28 May 2014 (28.05.2015) description paragraphs [0055]-[0076], [0088]-[0092] and [0108]	13-14,33-34,48
A	CN 103116723 A (BEIJING QIHOO TECHNOLOGY CO LTD) 22 May 2013 (22.05.2013) the whole document	1-51
A	US 5974549 A (SOLITON LTD) 26 October 1999 (26.10.1999) the whole document	1-51
A	CN 103218561 A (BEIJING JINSHAN NETWORK TECHNOLOGY CO et al) 24 June 2013 (24.07.2013) the whole document	1-51

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:

- “A” document defining the general state of the art which is not considered to be of particular relevance
- “E” earlier application or patent but published on or after the international filing date
- “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- “O” document referring to an oral disclosure, use, exhibition or other means
- “P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search
16 December 2015

Date of mailing of the international search report
18 January 2016

Name and mailing address of the ISA
State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao
Haidian District, Beijing 100088, China
Facsimile No. (86-10) 62019451

Authorized officer
HAO, Xiaoli
Telephone No. (86-10) 62089281

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2015/094845

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 104536981 A	22 April 2015	None	
CN 103823873 A	28 May 2014	None	
CN 103116723 A	22 May 2013	WO 2014121713 A1	14 August 2014
US 5974549 A	26 October 1999	None	
CN 103218561 A	24 July 2013	None	

国际检索报告

国际申请号

PCT/CN2015/094845

A. 主题的分类

G06F 21/50 (2013. 01) i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F21/-

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNTXT;CNABS;VEN;CNKI:浏览器, 安全, 保护, 防护, 拦截, 阻截, 阻止, 启动, 调用, 内置; browser, safety, secure, protect, prevent, intercept, embed, inject, built in, invoke, call

C. 相关文件

类型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 104536981 A (北京奇虎科技有限公司等) 2015年 4月 22日 (2015 - 04 - 22) 权利要求1-10、说明书第[0150]-[0536]段	1-51
X	CN 103823873 A (北京奇虎科技有限公司等) 2014年 5月 28日 (2014 - 05 - 28) 说明书第[0055]-[0076], [0088]-[0092]以及[0108]段	1-12, 15-32, 35-47, 49-51
A	CN 103823873 A (北京奇虎科技有限公司等) 2014年 5月 28日 (2014 - 05 - 28) 说明书第[0055]-[0076], [0088]-[0092]以及[01108]段	13-14, 33-34, 48
A	CN 103116723 A (北京奇虎科技有限公司等) 2013年 5月 22日 (2013 - 05 - 22) 全文	1-51
A	US 5974549 A (SOLITON LTD) 1999年 10月 26日 (1999 - 10 - 26) 全文	1-51
A	CN 103218561 A (珠海市君天电子科技有限公司等) 2013年 7月 24日 (2013 - 07 - 24) 全文	1-51

 其余文件在C栏的续页中列出。 见同族专利附件。

* 引用文件的具体类型:

- “A” 认为不特别相关的表示了现有技术一般状态的文件
- “E” 在国际申请日的当天或之后公布的在先申请或专利
- “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)
- “O” 涉及口头公开、使用、展览或其他方式公开的文件
- “P” 公布日先于国际申请日但迟于所要求的优先权日的文件

- “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件
- “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性
- “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性
- “&” 同族专利的文件

国际检索实际完成的日期

2015年 12月 16日

国际检索报告邮寄日期

2016年 1月 18日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

受权官员

郝晓丽

传真号 (86-10) 62019451

电话号码 (86-10) 62089281

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2015/094845

检索报告引用的专利文件		公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	104536981	A 2015年 4月 22日	无	
CN	103823873	A 2014年 5月 28日	无	
CN	103116723	A 2013年 5月 22日	WO 2014121713 A1 2014年 8月 14日	
US	5974549	A 1999年 10月 26日	无	
CN	103218561	A 2013年 7月 24日	无	

表 PCT/ISA/210 (同族专利附件) (2009年7月)