(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103473501 B
(45)授权公告日 2016.05.25

(21)申请号 201310370145.6

(22)申请日 2013.08.22

(73)专利权人 北京奇虎科技有限公司
地址 100088 北京市西城区新街口外大街28号D座112室(德胜园区)
专利权人 奇智软件(北京)有限公司

(72)发明人 张晓霖 董杰

(74)专利代理机构 北京市隆安律师事务所 11323
代理人 权鲜枝 齐辉

(51)Int.Cl.
G06F 21/51(2013.01)
H04L 29/06(2006.01)
H04L 29/08(2006.01)

(56)对比文件
CN 103077353 A,2013.05.01,
CN 103077353 A,2013.05.01,
CN 103020521 A,2013.04.03,
US 2006/0059469 A1,2006.03.16,
CN 101808102 A,2010.08.18,

审查员 张琳

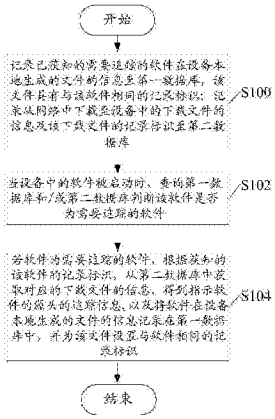
权利要求书3页 说明书15页 附图3页

(54)发明名称

一种基于云安全的恶意软件追踪方法

(57)摘要

本发明公开了一种基于云安全的恶意软件追踪方法。本发明实施例提供的一种追踪软件的方法,包括:记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库,该文件具有与该软件相同的记录标识;以及,记录从网络中下载至设备中的下载文件的信息及该下载文件的记录标识至第二数据库;当设备中的软件被启动时,查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件;若软件为需要追踪的软件,根据获知的该软件的记录标识,从第二数据库中获取对应的下载文件的信息,得到追踪信息;以及,将软件在设备本地生成的文件的信息记录在第一数据库中,并为该文件设置与软件相同的记录标识。



CN 103473501 B

1.一种追踪软件的方法,包括:

记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库,所述文件具有与该软件相同的记录标识;以及,记录从网络中下载至所述设备中的下载文件的信息及该下载文件的记录标识至第二数据库;

当设备中的软件被启动时,查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件;

若所述软件为需要追踪的软件,根据在查询第一数据库和/或第二数据库时获知的该软件的记录标识,从第二数据库中获取对应的下载文件的信息,得到指示所述软件的源头的追踪信息;以及,将所述软件在设备本地生成的文件的信息记录在第一数据库中,并为该文件设置与所述软件相同的记录标识。

2.根据权利要求1所述的方法,其中,所述将所述软件在设备本地生成的文件的信息记录在第一数据库中包括:

提取所述文件的文件路径中的各级文件目录,按照预定算法对提取出的每一级文件目录对应的字符串进行运算,将各级文件目录的运算值组合在一起得到该文件的文件指纹;

将所述文件的信息记录在第一数据库中该文件的文件指纹所指示的位置。

3.根据权利要求2所述的方法,其中,所述查询第一数据库和/或第二数据库判断该软件是否为待追踪软件包括:

判断所述软件的进程链上是否存在至少一个进程的相关文件被记录在所述第一数据库和/或第二数据库中,若是,确认所述软件为需要追踪的软件,若否,确认所述软件不是需要追踪的软件。

4.根据权利要求3所述的方法,其中,所述进程的相关文件包括进程的exe文件,以及,当所述进程是通过快捷方式启动时,所述进程的相关文件包括快捷方式文件;当所述进程为批处理进程时,所述进程的相关文件包括批处理文件;当进程为脚本进程时,所述进程的相关文件包括脚本文件;当所述进程为rundll32或regsvr32进程时,所述进程的相关文件包括相关的动态链接库DLL文件;当所述进程为解压缩进程时,所述相关文件包括解压缩文件。

5.根据权利要求3所述的方法,其中,所述查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件包括:

提取所述软件的进程链上当前进程的当前文件的文件路径中的各级文件目录,按照预定算法对提取出的每一级文件目录对应的字符串进行运算,将各级文件目录的运算值组合在一起得到该当前文件的查询值;

利用所述查询值对第一数据库中的文件指纹进行匹配;

当匹配成功时,确认所述软件为需要追踪的软件;

当匹配失败时,在第二数据库中查询所述当前文件,当查询到所述当前文件时,确认所述软件为需要追踪的软件;否则,确认所述软件不是需要追踪的软件。

6.根据权利要求2所述的方法,其中,所述将所述软件在设备本地生成的文件的信息记录在第一数据库中包括:

监控设备本地软件的进程创建操作,获取被创建进程执行时的命令行参数;

解析所述被创建进程执行的命令行参数,根据所述被创建进执行时的命令行参数判断

所述被创建进程是否为解压缩进程；

如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

7.根据权利要求1所述的方法，其中，所述方法还包括：

当所述软件在设备本地生成的文件包括可执行文件时，监控所述可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；

如果是，则至少将所述可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；

根据所述安全扫描器或云安全服务器返回的结果，确定对所述可执行文件进行放行、拦截或提示。

8.根据权利要求1所述的方法，其中，所述方法还包括：

当所述软件在设备本地生成的文件包括可执行文件时，监控所述可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL文件，并记录在内存中所述可执行文件进程的相关数据结构中；

如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；

根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据所述可执行文件的安全等级决定放行、拦截或提示。

9.根据权利要求1所述的方法，其中，所述方法还包括：

当设备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对该软件进行检测判断；

接收云安全服务器返回的对该软件的追踪信息。

10.一种追踪软件的装置，包括：

记录单元，适于记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库，所述文件具有与该软件相同的记录标识；以及，记录从网络中下载至所述设备中的下载文件的信息及该下载文件的记录标识至第二数据库；

判断单元，适于当设备中的软件被启动时，查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件；

获取单元，适于当所述判断单元判断所述软件为需要追踪的软件时，根据在查询第一数据库和/或第二数据库时获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示所述软件的源头的追踪信息；

所述记录单元，还适于对所述判断单元判断为需要跟踪的软件在设备本地生成文件时，将该文件的信息记录在第一数据库中，并为该文件设置与所述软件相同的记录标识。

11.根据权利要求10所述的装置，其中，所述记录单元，适于提取文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该文件的文件指纹；将该文件的信息记录在第一数据库中该文件的文件指纹所指示的位置。

12.根据权利要求10所述的装置，其中，所述判断单元，进一步适于判断所述软件的进程链上是否存在至少一个进程的相关文件被记录在所述第一数据库和/或第二数据库中，

若是，确认所述软件为需要追踪的软件，若否，确认所述软件不是需要追踪的软件。

13.根据权利要求12所述的装置，其中，

所述判断单元，适于提取所述软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该当前文件的查询值；利用所述查询值对第一数据库中的文件指纹进行匹配；当匹配成功时，确认所述软件为需要追踪的软件；当匹配失败时，在第二数据库中查询所述当前文件，当查询到所述当前文件时，确认所述软件为需要追踪的软件；否则，确认所述软件不是需要追踪的软件。

14.根据权利要求10所述的装置，其中，所述记录单元，适于监控设备本地软件的进程创建操作，获取被创建进程执行时的命令行参数；解析所述被创建进程执行的命令行参数，根据所述被创建进执行时的命令行参数判断所述被创建进程是否为解压缩进程；如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

15.根据权利要求10所述的装置，其中，所述装置还包括主动防御单元，适于当所述软件在设备本地生成的文件包括可执行文件时，监控所述可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；如果是，则至少将所述可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；根据所述安全扫描器或云安全服务器返回的结果，确定对所述可执行文件进行放行、拦截或提示。

16.根据权利要求10所述的装置，其中，所述装置还包括主动防御单元，适于当所述软件在设备本地生成的文件包括可执行文件时，监控所述可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL文件，并记录在内存中所述可执行文件进程的相关数据结构中；如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据所述可执行文件的安全等级决定放行、拦截或提示。

17.根据权利要求10所述的装置，其中，所述获取单元，还适于当设备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对该软件进行检测判断；接收云安全服务器返回的对该软件的追踪信息。

18.一种基于云安全的恶意软件追踪方法，包括：

记录已获知的需要追踪的软件在设备本地生成的文件的信息至云安全服务端的第一数据库，所述文件具有与该软件相同的记录标识；以及，记录从网络中下载至所述设备中的下载文件的信息及该下载文件的记录标识至云安全服务端的第二数据库；

当设备中的软件被启动时，将该软件的信息上报至云安全服务端，以使云安全服务端根据第一数据库和/或第二数据库确认该软件为需要追踪的软件后，根据获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示所述软件的源头的追踪信息；以及，将所述软件在设备本地生成的文件的信息记录在第一数据库中，并为该文件设置与所述软件相同的记录标识；

接收所述云安全服务端返回的所述软件的追踪信息。

一种基于云安全的恶意软件追踪方法

技术领域

[0001] 本发明涉及计算机软件领域，特别涉及一种基于云安全的恶意软件追踪方法。

背景技术

[0002] 随着计算机技术在社会生活中各个领域的广泛运用，恶意程序也如同其附属品一样接踵而来。由于这些恶意程序所具有的感染性、复制性及破坏性，其已成为困扰计算机使用的一个重大问题。

[0003] 恶意程序是一个概括性的术语，指任何故意创建用来执行未经授权并通常是有害行为的软件程序。计算机病毒、后门程序、键盘记录器、密码盗取者、Word和Excel宏病毒、引导区病毒、脚本病毒(如batch，windows shell，java等)、木马、犯罪软件、间谍软件和广告软件等等，都是一些可以称之为恶意程序的例子。以木马为例，木马能够盗取网银密码、盗取网游装备、泄露隐私照片等等。

[0004] 可以看出，恶意程序对计算机设备以及用户造成的危害是巨大的，因此如何对恶意程序进行查杀就显得更为重要。传统的查杀方式是特征库匹配，但是随着恶意程序爆发式的增长，又由于特征库的生成与更新相对于病毒的产生通常滞后，导致传统特征库匹配的查杀方式越来愈力不从心。于是出现了主动防御技术，主动防御是基于程序行为自主分析判断的实时防护技术，不以病毒的特征码作为判断病毒的依据，而是从最原始的病毒定义出发，直接将程序的行为作为判断病毒的依据，解决了传统安全软件无法防御未知恶意软件的弊端，从技术上实现了恶意程序的主动防御。

[0005] 然而，一些恶意程序可以通过生成新的可执行文件或创建快捷方式等，诱导用户运行该恶意程序派生出的文件，由于现有方案无法对这些派生出的文件进行准确定位，从而导致一些恶意程序能够绕过主动防御拦截，降低了主动防御的有效性。

发明内容

[0006] 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至少部分地解决上述问题的一种基于云安全的恶意软件追踪方法。

[0007] 依据本发明的一个方面，本发明实施例提供了一种追踪软件的方法，包括：

[0008] 记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库，该文件具有与该软件相同的记录标识；以及，记录从网络中下载至设备中的下载文件的信息及该下载文件的记录标识至第二数据库；

[0009] 当设备中的软件被启动时，查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件；

[0010] 若软件为需要追踪的软件，根据在查询第一数据库和/或第二数据库时获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示软件的源头的追踪信息；以及，将软件在设备本地生成的文件的信息记录在第一数据库中，并为该文件设置与软件相同的记录标识。

[0011] 其中，上述将软件在设备本地生成的文件的信息记录在第一数据库中包括：

[0012] 提取文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该文件的文件指纹；将文件的信息记录在第一数据库中该文件的文件指纹所指示的位置。

[0013] 其中，上述查询第一数据库和/或第二数据库判断该软件是否为待追踪软件包括：判断软件的进程链上是否存在至少一个进程的相关文件被记录在第一数据库和/或第二数据库中，若是，确认软件为需要追踪的软件，若否，确认软件不是需要追踪的软件。

[0014] 其中，上述进程的相关文件包括进程的exe文件，以及，当进程是通过快捷方式启动时，进程的相关文件包括快捷方式文件；当进程为批处理进程时，进程的相关文件包括批处理文件；当进程为脚本进程时，进程的相关文件包括脚本文件；当进程为rundll32或regsvr32进程时，进程的相关文件包括相关的动态链接库DLL文件；当为解压缩进程时，该进程的相关文件包括解压缩文件。

[0015] 其中，上述查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件包括：

[0016] 提取软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该当前文件的查询值；利用查询值对第一数据库中的文件指纹进行匹配；当匹配成功时，确认软件为需要追踪的软件；当匹配失败时，在第二数据库中查询当前文件，当查询到当前文件时，确认软件为需要追踪的软件；否则，确认软件不是需要追踪的软件。

[0017] 其中，上述将软件在设备本地生成的文件的信息记录在第一数据库中包括：

[0018] 监控设备本地软件的进程创建操作，获取被创建进程执行时的命令行参数；解析被创建进程执行的命令行参数，根据被创建进执行时的命令行参数判断被创建进程是否为解压缩进程；如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

[0019] 其中，上述方法还包括：当软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；如果是，则至少将可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；根据安全扫描器或云安全服务器返回的结果，确定对可执行文件进行放行、拦截或提示。

[0020] 其中，上述方法还包括：当软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL文件，并记录在内存中可执行文件进程的相关数据结构中；如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据可执行文件的安全等级决定放行、拦截或提示。

[0021] 其中，上述方法还包括：当设备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对该软件进行检测判断；接收云安全服务器返回的对该软件的追踪信息。

[0022] 根据本发明的另一方面，本发明实施例提供了一种追踪软件的装置，包括：

[0023] 记录单元，适于记录已获知的需要追踪的软件在设备本地生成的文件的信息至第

一数据库，该文件具有与该软件相同的记录标识；以及，记录从网络中下载至设备中的下载文件的信息及该下载文件的记录标识至第二数据库；

[0024] 判断单元，适于当设备中的软件被启动时，查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件；

[0025] 获取单元，适于当判断单元判断软件为需要追踪的软件时，根据在查询第一数据库和/或第二数据库时获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示软件的源头的追踪信息；

[0026] 记录单元，还适于对判断单元判断为需要跟踪的软件在设备本地生成文件时，将该文件的信息记录在第一数据库中，并为该文件设置与软件相同的记录标识。

[0027] 其中，记录单元，适于提取文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该文件的文件指纹；将该文件的信息记录在第一数据库中该新文件的文件指纹所指示的位置。

[0028] 其中，判断单元，进一步适于判断软件的进程链上是否存在至少一个进程的相关文件被记录在第一数据库和/或第二数据库中，若是，确认软件为需要追踪的软件，若否，确认软件不是需要追踪的软件。

[0029] 其中，判断单元，适于提取软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该当前文件的查询值；利用查询值对第一数据库中的文件指纹进行匹配；当匹配成功时，确认软件为需要追踪的软件；当匹配失败时，在第二数据库中查询当前文件，当查询到当前文件时，确认软件为需要追踪的软件；否则，确认软件不是需要追踪的软件。

[0030] 其中，记录单元，适于监控设备本地软件的进程创建操作，获取被创建进程执行时的命令行参数；解析被创建进程执行的命令行参数，根据被创建进执行时的命令行参数判断被创建进程是否为解压缩进程；如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

[0031] 其中，上述装置还包括主动防御单元，适于当软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；如果是，则至少将可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；根据安全扫描器或云安全服务器返回的结果，确定对可执行文件进行放行、拦截或提示。

[0032] 其中，上述装置还包括主动防御单元，适于当软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL文件，并记录在内存中可执行文件进程的相关数据结构中；如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据可执行文件的安全等级决定放行、拦截或提示。

[0033] 其中，获取单元，还适于当设备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对该软件进行检测判断；接收云安全服

务器返回的对该软件的追踪信息。

[0034] 根据本发明的又一方面，本发明实施例提供了一种基于云安全的恶意软件追踪方法，包括：

[0035] 记录已获知的需要追踪的软件在设备本地生成的文件的信息至云安全服务端的第一数据库，该文件具有与该软件相同的记录标识；以及，记录从网络中下载至设备中的下载文件的信息及该下载文件的记录标识至云安全服务端的第二数据库；

[0036] 当设备中的软件被启动时，将该软件的信息上报至云安全服务端，以使云安全服务端根据第一数据库和/或第二数据库确认该软件为需要追踪的软件后，根据获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示软件的源头的追踪信息；以及，将软件在设备本地生成的文件的信息记录在第一数据库中，并为该文件设置与软件相同的记录标识；

[0037] 接收云安全服务端返回的软件的追踪信息。

[0038] 本发明实施例通过对设备本地需要监控的软件的每个文件的信息进行记录以及对设备从外部网络下载到本地的每个下载文件的信息进行记录的技术手段，能够对需要追踪的软件生成或派生出的所有文件的信息进行记录，形成一条该软件的文件链信息；以及，基于这种文件链通过查询第一数据库和/或第二数据库确认需要追踪的文件并从第二数据库中获取到追踪信息技术手段，能够获取到派生文件所属软件的源头信息，准确对派生文件进行定位，从而能够利用该源头信息对软件执行主动防御拦截，提高了主动防御的有效性。

[0039] 并且，本发明实施例缩小了拦截进程加载DLL的范围，不必拦截所有进程，仅对进程文件来源于下载文件或压缩包文件的进程拦截加载DLL，这样既能拦截白利用木马，由于减少了拦截的范围，因此可以减少对计算机性能造成大的影响。

[0040] 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

[0041] 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：

[0042] 图1示出了根据本发明一个实施例的追踪软件的方法流程示意图；

[0043] 图2示出了根据本发明一个实施例的追踪信息的查询方法流程示意图；以及

[0044] 图3示出了根据本发明又一个实施例的追踪软件的装置的结构示意图。

具体实施方式

[0045] 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。

[0046] 本发明一个实施例提供了一种追踪软件的方法，参见图1，该方法包括如下步骤：

[0047] S100：记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库，该文件具有与该软件相同的记录标识；以及，记录从网络中下载至设备中的下载文件的信息及该下载文件的记录标识至第二数据库。

[0048] 本实施例中建立了两个数据库，第一数据库和第二数据库，本实施例不对这两个数据库的具体名称进行限定，例如，第一数据库也可以称之为文件链数据库，第二数据库也可以称之为网盾数据库。

[0049] 上述第一数据库中记录的一个文件的信息包括生成该文件的文件、压缩包或进程的名称，该文件的文件路径，该文件的文件名称，但不局限于此，可以将文件生成过程中获取到其他信息也都记录在第一数据库中，如文件的生成时间、文件的版本信息以及文件的描述信息等。

[0050] 当一个文件的进程或者父进程、父父进程(包括系统进程和第三方进程)等只要有一个进程生成的文件是可以在第二数据库或第一数据库中查找到的，就将该文件的信息记录在第一数据库中。这种方式下，利用记录的信息可以获知一个文件上游的各文件(生成该文件的原始文件、生成该原始文件的文件等)的信息以及该文件下游(该文件派生出的派生文件、派生文件又派生出的文件等)的各文件，从而形成了一条文件链，利用该文件链可以追踪到文件的源头信息。

[0051] 需要说明的是，本实施例中第一数据库记录的文件主要包括两种类型：一种类型是设备在本地从无到有新生成的文件，该新文件中的数据在创建时间点之前不存在于设备中，另一种类型是设备中原有的文件发生了变化而新创建出来的文件，该新文件中的数据在创建时间点之前已存在于设备中，如当修改了原有文件的名称、移动了原有文件的存储位置时，则原有文件由于这些变化会生成出新文件，将这些新文件的信息也记录在第一数据库中，从而保证了建立的文件链的完整性。

[0052] 上述第二数据库可以基于设备中的网盾工具实现，网盾为一种能够对设备的下载操作进行监控的工具，通过网盾可以获知设备是否从网络中下载了新的文件，当网盾监测到设备下载了新的文件时，提取该新的下载文件的信息记录在第二数据库中。第二数据库中记录的下载文件的信息包括下载工具类型、下载URL(Uniform Resource Locator，统一资源定位符)及网页URL等，除了此处列举的之外，如果还有其他可以在下载过程中或下载完成时能获的信息，也可以记录下来。可以理解，当不采用网盾工具时，本实施例也可以在设备设置一个用于监控设备的下载操作的监控功能，利用该监控功能替代于网盾工具。

[0053] 其中，下载工具类型，一般指文件是通过什么途径下载的，比如即时通讯工具、邮件客户端等，例如，Outlook/Foxmail等邮件客户端、WEB浏览器、IE/Chrome等专用下载工具以及迅雷/电驴等下载工具。下载URL，一般指该下载文件自身的下载链接。网页URL，一般指下载URL所在的web网页的URL。

[0054] 本实施例采用一种应用层和驱动层配合的追踪机制，由应用层监测是否在本地生成了新文件和/或下载了新的文件，是则由应用层通知驱动层对本地的新文件和/或新的下载文件的信息进行记录。

[0055] 一些恶意软件通过把CMD之类的文件，或者bat的文件，或者快捷方式打包在一个压缩包里，或者传递其中的单个文件(pif)，图标，可能是应用程序的文件，或者VBS(一种脚

本文件)，建立一个文件夹并放置一个文件夹配置文件(desktop.ini)，使用计划任务，或者，使用模拟鼠标点击等。甚至网购木马等会传送一个压缩包，后续解压缩到用户电脑的卓面上，如果用户双击压缩包使用关联解压软件打开该压缩包，然后在解压软件中双击伪装的木马程序，或者对压缩包进行解压缩，然后双击解压后的木马程序，则会启动压缩包中的恶意程序。然而，压缩包软件(如Winrar)在国内是一款装机必备软件，占据90%以上的压缩市场份额，可见木马通过压缩包传播的广泛程度。

[0056] 鉴于恶意软件通过压缩包传播带来的危害，本实施例采用应用层在执行监测时，主要监测两个源头：网盾下载的文件，以及解压缩文件。应用层也可以对解压缩文件之外的其他本地文件进行监测，但由于解压缩文件是木马等恶意软件传播的一个主要途径，所以本方案对解压缩文件进行重点监测。

[0057] 则上述将该解压缩文件的信息以及该压缩包的文件标识记录至第一数据库包括：监控设备本地软件的进程创建操作，获取被创建进程执行时的命令行参数；解析被创建进程执行的命令行参数，根据被创建进执行时的命令行参数判断被创建进程是否为解压缩进程；如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

[0058] 命令行参数包含的内容较多，例如一般包括被创建进程的进程路径、压缩包的存放路径、更具体的进程参数等。如，某一解压缩命令行参数为"C:\Program Files\AAA\AAAzip\AAAzip.exe"-s"C:\Test\test.zip""C:\Test\test，其中，"C:\Program Files\AAA\AAAzip\AAAzip.exe"即为被创建进程的进程路径，"-s"是用来表明该进程是解压缩进程、而非压缩进程的一种参数信息，"C:\Test\test.zip"是压缩包的存放路径参数信息。当然，有些情况下，命令行参数中可能也没有进程路径。但是，通过监控进程创建操作，捕获进程创建的相关函数，也是可以获得被创建进程的进程路径的。

[0059] 可选的，可以解析被创建进程的进程路径和命令行参数，然后根据进程路径以及命令行参数，判断被创建进程是否为解压缩进程。例如，根据被创建进程的进程路径，判断被创建进程的进程文件是否为压缩或解压缩软件。通常进程路径中会有进程文件的相关信息，如“WinRAR.exe”或“AAAZip.exe”等文件名，另外还有进程文件内部的版本信息中包含特定的文件内部名称、文件描述等参数信息，进而，即可根据参数信息判断出该进程文件是否属于压缩或解压缩软件。如果是，则进一步根据命令行参数判断被创建进程是否为解压缩进程。如果进程是解压缩进程，那么在命令行参数中一般会有可以体现出是解压缩、而非压缩的参数，例如前文给出的命令行参数示例中的“-S”参数，因此可以根据这类参数判断出本进程是解压缩进程，而不是压缩进程。

[0060] 进一步的，本实施例为每个下载文件设置一个记录标识，将该记录标识和下载文件的信息一起记录在第二数据库中，而当在第一数据库中记录需要追踪的软件生成的文件的信息时，会查询该文件在第二数据库中对应的记录标识，将该记录标识记录在第一数据库中存储该文件的数据中，并将该记录标识“继承”该需要追踪的软件后续通过创建或修改生成的文件，即该需要追踪的软件创建或修改的文件都具有该相同的记录标识。利用该记录标识能够从第二数据库中获取到需要追踪的软件的源头信息，从而实现对文件的来源追踪。

[0061] 进一步的，本实施例中第一数据库可以采用注册表的形式实现。注册表进行分层记录，具有树形的结构，通过注册表的记录方式，可以有效保证系统和驱动的性能。

[0062] 本实施例根据文件的文件目录记录该文件到相应的位置。例如，提取文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该文件的文件指纹。即对文件的文件路径中的每一级文件目录分别按照预定算法进行运算，将各运算值组合在一起得到该文件的文件指纹；将文件的信息记录在第一数据库中该文件的文件指纹所指示的位置。

[0063] 上述预定算法可以为MD5(Message-Digest Algorithm5，信息-摘要算法)，或SHA1，或CRC(Cyclic Redundancy Check，循环冗余校验)等等。

[0064] 例如，文件路径对应的是C:盘下对应的某个目录，如"C:\Program Files\XXX\XXXzip\XXXzip.exe"，则第一级文件目录为C:，第二级文件目录C:\Program Files，第三级文件目录为C:\Program Files\XXX，第四级文件目录为C:\Program Files\XXX\XXXzip，第五级文件目录为C:\Program Files\XXX\XXXzip\XXXzip.exe，将该文件路径的各级文件目录对应的字符串分别进行MD5运算得到的MD5值组合在一起得到该文件的文件指纹。利用上述方式生成的文件指纹，建立了一种树形多级的数据结构，相比于仅利用一级索引查询数据的存储结构，本实施例的注册表形式的第一数据库大大减少了查询时的数据处理量，提高了查询效率。

[0065] 可选的，本实施例对所使用的数据库的数量不进行限定，例如，本实施例不局限于建立第一数据库和第二数据库两个数据库，该第一数据库和第二数据库可以由一个数据库实现。

[0066] 另外，当第一数据库中记录的文件被删除，第一数据库会将被删除文件的信息从记录中删除，以及，对于第二数据库中记录的下载文件，当该下载文件的大小超过预定文件大小和/或超过预定存储时间时，从第二数据库中将该下载文件的信息删除。

[0067] S102：当设备中的软件被启动时，查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件。

[0068] 本实施例采用在软件每次被启动时，开始对该软件的追踪操作。首先需要通过查询数据库判断被启动的软件是否为需要追踪的软件。在执行查询操作时，判断软件的进程链上是否存在至少一个进程的相关文件被记录在第一数据库和/或第二数据库中，若是，确认所述软件为需要追踪的软件，若否，确认所述软件不是需要追踪的软件。对于不需要追踪的软件，不执行对该软件的追踪，也不需将该软件的信息记录在第一数据库中。

[0069] 对于上述两个监测源头的软件在运行过程中新生成的文件，本实施例可以对该软件的进程链上的任一个或多个进程(包括系统进程和第三方进程)生成的文件进行查询和追踪，这些文件即包括来自网络的下载文件或本地存储的独立文件，或者由下载文件或独立文件生成的派生文件等。

[0070] 可以同时在第一数据库和第二数据库中执行查询操作，也可以先在第一数据库进行查询，然后再在第二数据库进行查询，或反之。

[0071] 优选的，考虑到解压缩文件为恶意软件的主要传播途径，本实施例采用先在第一数据库进行查询，然后再在第二数据库进行查询的方式，以提高查询效率。参见图2，一种示例性的追踪信息的查询方法可以包括如下：

[0072] S200：提取软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算

值组合在一起得到该当前文件的查询值。即对待追踪文件的文件路径中的每一级文件目录分别按照预定算法进行运算，将各运算值组合在一起得到该待当前文件的查询值。将得到的当前文件的查询值作为索引，在注册表的树形结构中逐级进行匹配。上述当前进程为包括多个进程的进程链上当前正在查询的任一进程，当前文件为包括多个文件的当前进程的当前正在查询的任一文件。当在第一数据库和/或第二数据库中查询到进程链上的任一进程的任一相关文件时，即可结束查询操作。

[0073] S202：利用查询值对第一数据库中的文件指纹进行匹配。

[0074] S204：判断是否能够匹配成功，若否，执行步骤S208；若是，执行步骤S206。

[0075] S206：确认软件为需要追踪的软件。这种场景下，当前文件通常为下载文件在运行中派生出的文件，例如压缩包文件在解压缩过程中生成的解压缩文件。

[0076] S208：在第二数据库中查询上述当前文件，判断是否有对应的下载文件，若是，进入步骤S206，与步骤S206不同的时，本场景下的当前文件通常就是设备从外部下载到的下载文件；若否，进入步骤S210。

[0077] 在第二数据库中查询文件时，可以利用该文件的文件路径和/或文件的MD5值（只要是可以用来唯一标识文件的信息即可，比如还可以是文件的SHA1等信息摘要）进行查询。

[0078] S210：确认软件不是需要追踪的软件。

[0079] 需要说明的是，上述进程的相关文件包括进程的exe文件，以及，当进程是通过快捷方式启动时，上述进程的相关文件包括快捷方式文件；当进程为批处理进程时，上述进程的相关文件包括批处理文件；当进程为脚本进程时，上述进程的相关文件包括脚本文件；当进程为rundll32或regsvr32进程时，上述进程的相关文件包括相关的动态链接库DLL文件；当为解压缩进程时，上述进程的相关文件包括解压缩文件。

[0080] 另外，在初始时，第一数据库中的信息为空，第二数据库中记录了设备从外部下载到的文件的信息，则此时在判断设备中任意启动的软件是否为需要追踪的软件时，先在第二数据库中进行查询，当该软件记录在第二数据库中时，记录需要追踪的软件。

[0081] S103：若软件为需要追踪的软件，根据在查询第一数据库和/或第二数据库时获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示软件的源头的追踪信息。

[0082] 以及，将软件在设备本地生成的文件的信息记录在第一数据库中，并为该文件设置与软件相同的记录标识。本实施例中在执行对软件的追踪操作时，会同时将该软件生成的文件在继承与该软件相同的记录标识后记录在第一数据库中，以用于后续软件的追踪操作。

[0083] 由上所述，本发明实施例通过对设备本地需要监控的软件的每个文件的信息进行记录以及对设备从外部网络下载到本地的每个下载文件的信息进行记录的技术手段，能够对需要追踪的软件生成或派生出的所有文件的信息进行记录，形成一条该软件的文件链信息；以及，基于这种文件链通过查询第一数据库和/或第二数据库确认需要追踪的文件并从第二数据库中获取到追踪信息技术手段，能够获取到派生文件所属软件的源头信息，准确对派生文件进行定位，从而能够利用该源头信息对软件执行主动防御拦截，提高了主动防御的有效性。

[0084] 并且，本发明实施例缩小了拦截进程加载DLL的范围，不必拦截所有进程，仅对进

程文件来源于下载文件或压缩包文件的进程拦截加载DLL，这样既能拦截白利用木马，由于减少了拦截的范围，因此可以减少对计算机性能造成大的影响。

[0085] 本发明又一个实施例对基于追踪信息的主动防御拦截方法进行说明。

[0086] 软件运行时生成的信息包括但不限于文件，可能还有与文件相关其他信息，比如文件大小、文件格式、行为记录等，这些信息都可以记录到第一数据库。文件可能是可执行文件，也可能是非可执行文件。对于非可执行文件，如文本文件、图片等，通常是安全文件，本方案在执行主动防御拦截时，可以先过滤掉这些非可执行文件，只关心所记录的可执行文件。可执行文件包括但不限于exe类文件、脚本文件、批处理(bat)文件、msi文件以及链接(link)文件等。

[0087] 并且，为了适应恶意程序的更新速度，快速地识别和查杀恶意程序，目前的安全防护软件越来越多地使用云安全技术对恶意程序进行拦截。所谓云安全技术，就是把客户端的可疑文件的特征传给云安全中心的服务器，由云安全中心对其安全做出判定，然后客户端安全软件根据云安全中心传回的信息对木马进行报告和处理。云结构就是一个大型的客户端/服务器(C/S)架构，本发明通过在客户端设备侧获取软件的追踪信息，追踪到软件的源头，将追踪信息上报至云安全服务器，利用云安全服务器对追踪信息进行归纳和分析，从而有助于对软件或程序进行黑白的分类判别。

[0088] 另外，本实施例还提供一种从云安全服务端查询追踪信息的方案，当设备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对待追踪文件进行检测判断；设备侧接收云安全服务器返回的对该需要追踪的软件的追踪信息。

[0089] 进一步的，本实施例还包括：当需要追踪的软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；如果是，则至少将可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；根据安全扫描器或云安全服务器返回的结果，确定对可执行文件进行放行、拦截或提示。

[0090] 上述安全扫描器位于服务器侧，适于根据预置的扫描规则，例如针对已知的恶意程序或安全程序行为分析出的判断条件，其中包括是基于软件源头作为依据的判断条件，进而对客户端侧设备发送的可执行文件所关联的下载来源以及其他信息进行检查，并告知客户端侧设备检查结果。

[0091] 上述可疑操作包括但不限于：写入注册表进行自动加载；修改注册表；修改系统文件；修改指定的应用文件；执行进程间注入；结束进程；修改浏览器中网页内容；以及记录键盘操作。可疑操作还可以包括：调用shell程序，修改程序文件或写程序文件；调用ftp或tftp，创建ftp，或tftp服务；创建大量相同线程，修改和创建用户账号；危险网络操作；向系统注册表添加启动项；修改系统启动文件；向其他进程注入线程；堆栈溢出；拦截系统API调用等等。或者是一系列行为的组合。可疑操作还可以包括：删除注册表启动项或服务、终止电脑安全程序工具的进程、弱口令破解局域网其他电脑的管理员帐号并复制传播、修改注册表键值导致不能查看隐藏文件和系统文件、尝试破坏硬盘分区下的文件、删除用户的系统备份文件等等。

[0092] 进一步的，本实施例还包括：当需要追踪的软件在设备本地生成的文件包括可执

行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL(Dynamic Link Library，动态链接库)文件，并记录在内存中可执行文件进程的相关数据结构中；如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据可执行文件的安全等级决定放行、拦截或提示。

[0093] 本实施例在内存中保存有系统中所有进程的信息及进程关系，进程加载的DLL就记录在内存中该可执行文件进程的相关数据结构中；如果驱动程序监控到该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查，并根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据该可执行文件进程的安全等级决定放行、拦截或提示。

[0094] 其中，服务器端预先保存有文件特征值和安全等级信息的对应关系，服务器端确定的安全等级可以自定义，例如包括安全、危险、未知等级别，也可以采用一级、二级、三级等方式来进行区分，只要能够体现出各模块是否安全状态即可。或者，所述安全级别信息包括：安全等级、未知等级、可疑等级、高度可疑等级和恶意等级，其中，恶意等级为最高等级，安全等级为最低等级。例如，可以设置等级为10-20时为安全等级，等级为30-40时为未知等级，等级为50-60时为可疑等级和高度可疑等级，等级大于70时为恶意等级。或者，危险等级可以包括：可疑，未知，白，黑等四种情况。第一等级：10和20，其是白文件；第二等级：30和40，其是灰文件；第三等级：50和60，其是可疑文件；第四等级：70，其是病毒。只有第一等级，10和20，是可以信任的。

[0095] 此外，还可以根据进程所属进程链中相关文件的等级综合确定该进程文件的等级，例如当前进程为进程D，通过该进程D的创建关系追溯到对应进程D的进程链为A→B→C→D，在该进程链中查找：首先查找到进程D的第一级父进程为进程C，并且获取到所述进程C的文件等级为可疑等级；然后查找到进程D的第二级父进程为进程B，并且获取到所述进程B的文件等级为未知等级；最后查找到进程D的第三级父进程为进程A，并且获取到所述进程A的文件等级为安全等级。进而，可以根据待监控的进程D相关的多个进程文件的不同等级，综合判断进程D的进程文件的等级，进而告知客户端拦截、提示或放行。

[0096] 例如，有些恶意程序使用DLL劫持技术将木马DLL与可信任的白程序打包在一起，进而当白程序被执行时，木马DLL就会被加载，从而实现利用白程序突破主动防御的目的，也就是所谓的白利用木马。可见，DLL文件是比较危险的文件，可以重点监控。本实施例中通过驱动程序监控到解压缩进程生成的可执行文件是否加载了DLL文件，如果加载，则一方面记录相关的DLL文件，可选的，记录该进程加载的所有的DLL文件，监控路径不限于当前目录，因为很多木马或将DLL文件放在系统目录下。另一方面会加强对该可执行文件的执行行为进行监控，如果驱动程序监控到该可执行文件进程执行的操作包括可疑操作，具体的可疑操作前面已经有已描述，此处不再赘述。

[0097] 然后，通过安全扫描器对该可执行文件进程加载的DLL文件进行检查。例如，可执行文件进程被主动防御中的某个规则拦截到，例如命中了RD(Registry Defend，注册表防护)、FD(File，文件防护)或AD(application Defend，进程防护)的一个规则，则通过安全扫描器对所有或主要的已被加载的DLL进行检查，如果某个DLL是木马就提示用户并终止执行；如果DLL的危险等级高于存在进程可执行文件的危险等级，则修改该进程的危险等级为

DLL的高危险等级，并呈现危险提示。这样可以比较好的防御未知白利用木马，同时不会影响正常程序的执行效率。

[0098] 可选的，可以将该可执行文件进程加载的所有DLL文件都进行检查，也可以进一步优化，只检查部分DLL。例如，可以对于系统级别的DLL放过，这些DLL往往还是比较安全的，所以不去检查这些DLL文件，对主动防御的性能影响不大，而且降低了监控量，提高了主动防御的执行效率。

[0099] 本发明实施例通过文件追踪机制配合服务器侧(或云端)规则，能够提高主动防御拦截对白利用木马的拦截能力，由此解决了现有主动防御效果不好的技术问题，取得了对恶意程序进行更有效主动防御的有益效果。而通过本发明实施例的方案，使得无论是通过几层解压出的文件，还是不容易打开的文件，都可以知道其下载来源，进而可以通过这些下载来源判断是否安全，从而达到从源头上进行主动防御的目的，提高了主动防御的有效性。

[0100] 并且，本发明实施例缩小了拦截进程加载DLL的范围，不必拦截所有进程，仅对进程文件来源于下载文件或压缩包文件的进程拦截加载DLL，这样既能拦截白利用木马，由于减少了拦截的范围，因此可以减少对计算机性能造成大的影响。

[0101] 本发明又一个实施例还提供了一种追踪软件的装置，参见图3，包括记录单元300、判断单元302、获取单元304和主动防御单元306。下面分别对这些单元进行说明。

[0102] 记录单元300，适于记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库，所述文件具有与该软件相同的记录标识；以及，记录从网络中下载至所述设备中的下载文件的信息及该下载文件的记录标识至第二数据库。其中，记录单元300适于提取文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该文件的文件指纹；将文件的信息记录在第一数据库中该文件的文件指纹所指示的位置。针对解压缩文件进行监控和记录的场景，记录单元300适于监控设备本地软件的进程创建操作，获取被创建进程执行时的命令行参数；解析被创建进程执行的命令行参数，根据被创建进执行时的命令行参数判断被创建进程是否为解压缩进程；如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

[0103] 判断单元302适于当设备中的软件被启动时，查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件。判断单元302进一步适于判断软件的进程链上是否存在至少一个进程的相关文件被记录在第一数据库和/或第二数据库中，若是，确认软件为需要追踪的软件，若否，确认软件不是需要追踪的软件。一种方式下，判断单元302适于提取软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该当前文件的查询值；利用查询值对第一数据库中的文件指纹进行匹配；当匹配成功时，确认软件为需要追踪的软件；当匹配失败时，在第二数据库中查询当前文件，当查询到该当前文件时，确认软件为需要追踪的软件；否则，确认软件不是需要追踪的软件。

[0104] 获取单元304，适于当判断单元302判断软件为需要追踪的软件时，根据在查询第一数据库和/或第二数据库时获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示软件的源头的追踪信息。

[0105] 另外，本装置还可以从云安全服务器获取到追踪信息，则获取单元304还适于当设

备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对软件进行检测判断；接收云安全服务器返回的对该软件的追踪信息。

[0106] 主动防御单元306，适于当需要追踪的软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；如果是，则至少将可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；根据安全扫描器或云安全服务器返回的结果，确定对可执行文件进行放行、拦截或提示。

[0107] 可选的，主动防御单元306还适于当需要追踪的软件在设备本地生成的文件包括可执行文件时，监控可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL文件，并记录在内存中可执行文件进程的相关数据结构中；如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据可执行文件的安全等级决定放行、拦截或提示。

[0108] 本发明装置实施例中各单元的具体工作方式可以参见本发明的方法实施例中的相关内容，在此不再赘述。

[0109] 本发明实施例缩小了拦截进程加载DLL的范围，不必拦截所有进程，仅对进程文件来源于下载文件或压缩包文件的进程拦截加载DLL，这样既能拦截白利用木马，由于减少了拦截的范围，因此可以减少对计算机性能造成大的影响。

[0110] 本发明又一个实施例提供了一种基于云安全的恶意软件追踪方法，采用文件追踪功能配合云端规则的处理方式，大大增强了对突发木马的防御效果，该方法包括如下步骤：

[0111] S400：记录已获知的需要追踪的软件在设备本地生成的文件的信息至云安全服务端的第一数据库，所述文件具有与该软件相同的记录标识；以及，记录从网络中下载至所述设备中的下载文件的信息及该下载文件的记录标识至云安全服务端的第二数据库。

[0112] 本步骤中对设备本地生成新文件的检测操作以及对设备中下载了新文件的感知操作，可以由云安全服务器端执行，也可以由设备本地的监测功能执行。

[0113] S402：当设备中的软件被启动时，将该软件的信息上报至云安全服务端，以使云安全服务端根据第一数据库和/或第二数据库确认该软件为需要追踪的软件后，根据获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示所述软件的源头的追踪信息；以及，将所述软件在设备本地生成的文件的信息记录在第一数据库中，并为该文件设置与所述软件相同的记录标识。

[0114] 根据云端规则在云安全服务端的第一数据库和/或第二数据库中对待跟踪文件进行查询，例如，该云端规则可以指示同时在第一数据库和第二数据库中执行查询操作，也可以指示先在第一数据库进行查询，然后再在第二数据库进行查询，或反之。具体方式可以参见本发明其他实施例中的相关内容。

[0115] S404：接收所述云安全服务端返回的所述软件的追踪信息。

[0116] 本发明实施例通过对设备本地需要监控的软件的每个文件的信息进行记录以及对设备从外部网络下载到本地的每个下载文件的信息进行记录的技术手段，能够对需要追踪的软件生成或派生出的所有文件的信息进行记录，形成一条该软件的文件链信息；以及，基于这种文件链通过查询第一数据库和/或第二数据库确认需要追踪的文件并从第二数据

库中获取到追踪信息技术手段，能够获取到派生文件所属软件的源头信息，准确对派生文件进行定位，从而能够利用该源头信息对软件执行主动防御拦截，提高了主动防御的有效性。

[0117] 根据本发明实施例所述的装置，其中，所述记录单元，适于提取文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该文件的文件指纹；将该文件的信息记录在第一数据库中该新文件的文件指纹所指示的位置。

[0118] 根据本发明实施例所述的装置，其中，所述判断单元，进一步适于判断所述软件的进程链上是否存在至少一个进程的相关文件被记录在所述第一数据库和/或第二数据库中，若是，确认所述软件为需要追踪的软件，若否，确认所述软件不是需要追踪的软件。

[0119] 根据本发明实施例所述的装置，其中，

[0120] 所述判断单元，适于提取所述软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该当前文件的查询值；利用所述查询值对第一数据库中的文件指纹进行匹配；当匹配成功时，确认所述软件为需要追踪的软件；当匹配失败时，在第二数据库中查询所述当前文件，当查询到所述当前文件时，确认所述软件为需要追踪的软件；否则，确认所述软件不是需要追踪的软件。

[0121] 根据本发明实施例所述的装置，其中，所述记录单元，适于监控设备本地软件的进程创建操作，获取被创建进程执行时的命令行参数；解析所述被创建进程执行的命令行参数，根据所述被创建进执行时的命令行参数判断所述被创建进程是否为解压缩进程；如果是，则通知驱动程序记录该解压缩进程生成的信息至第一数据库。

[0122] 根据本发明实施例所述的装置，其中，所述装置还包括主动防御单元，适于当所述软件在设备本地生成的文件包括可执行文件时，监控所述可执行文件的进程创建操作，若监控到进程创建操作，则判断该可执行文件的进程在运行过程中是否执行了可疑操作；如果是，则至少将所述可执行文件的追踪信息发送到安全扫描器或云安全服务器，供安全扫描器或云安全服务器进行检测判断；根据所述安全扫描器或云安全服务器返回的结果，确定对所述可执行文件进行放行、拦截或提示。

[0123] 根据本发明实施例所述的装置，其中，所述装置还包括主动防御单元，适于当所述软件在设备本地生成的文件包括可执行文件时，监控所述可执行文件的进程创建操作，若监控到进程创建操作，则通知驱动程序监控该可执行文件进程加载的DLL文件，并记录在内存中所述可执行文件进程的相关数据结构中；如果该可执行文件进程执行的操作包括可疑操作，则通过安全扫描器对该可执行文件进程加载的DLL进行检查；根据检查结果，确定是否修改该可执行文件进程的安全等级，以及根据所述可执行文件的安全等级决定放行、拦截或提示。

[0124] 根据本发明实施例所述的装置，其中，所述获取单元，还适于当设备中的软件被启动时，将该软件的信息上传至云安全服务器，以由云安全服务器利用保存的文件的信息对该软件进行检测判断；接收云安全服务器返回的对该软件的追踪信息。

[0125] 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。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。

[0126] 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。

[0127] 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。

[0128] 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。

[0129] 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。

[0130] 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追踪软件的装置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。

[0131] 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造成对权利要求的限制。单词“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。

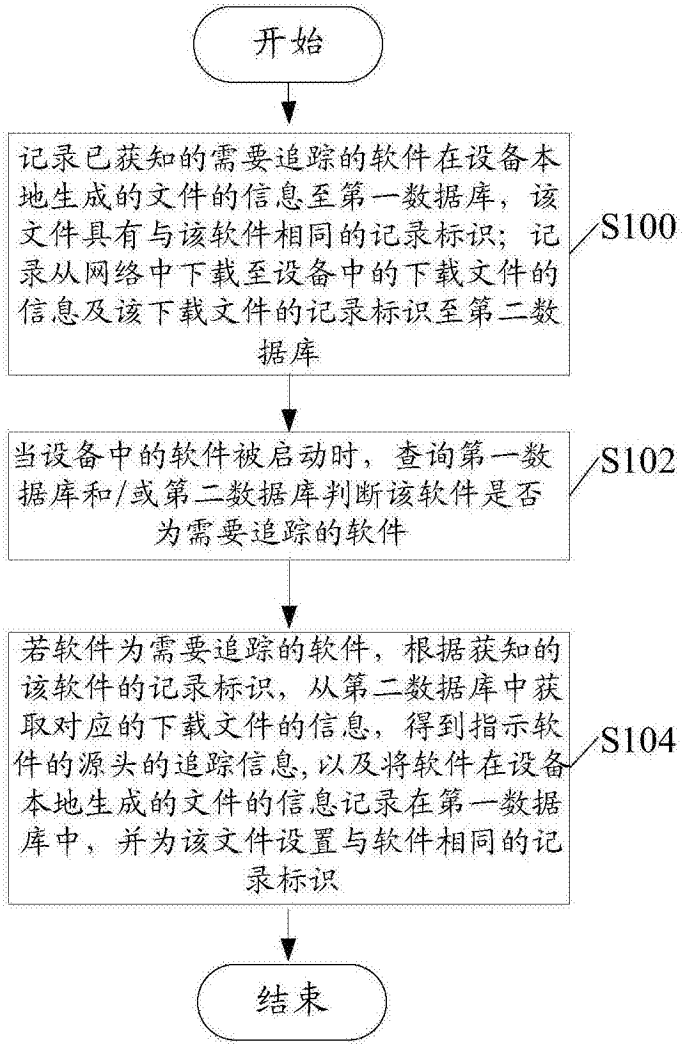

开始
记录已获知的需要追踪的软件在设备本地生成的文件的信息至第一数据库，该文件具有与该软件相同的记录标识；记录从网络中下载至设备中的下载文件的信息及该下载文件的记录标识至第二数据库
S100
当设备中的软件被启动时，查询第一数据库和/或第二数据库判断该软件是否为需要追踪的软件
S102
若软件为需要追踪的软件，根据获知的该软件的记录标识，从第二数据库中获取对应的下载文件的信息，得到指示软件的源头的追踪信息，以及将软件在设备本地生成的文件的信息记录在第一数据库中，并为该文件设置与软件相同的记录标识
S104
结束

图1

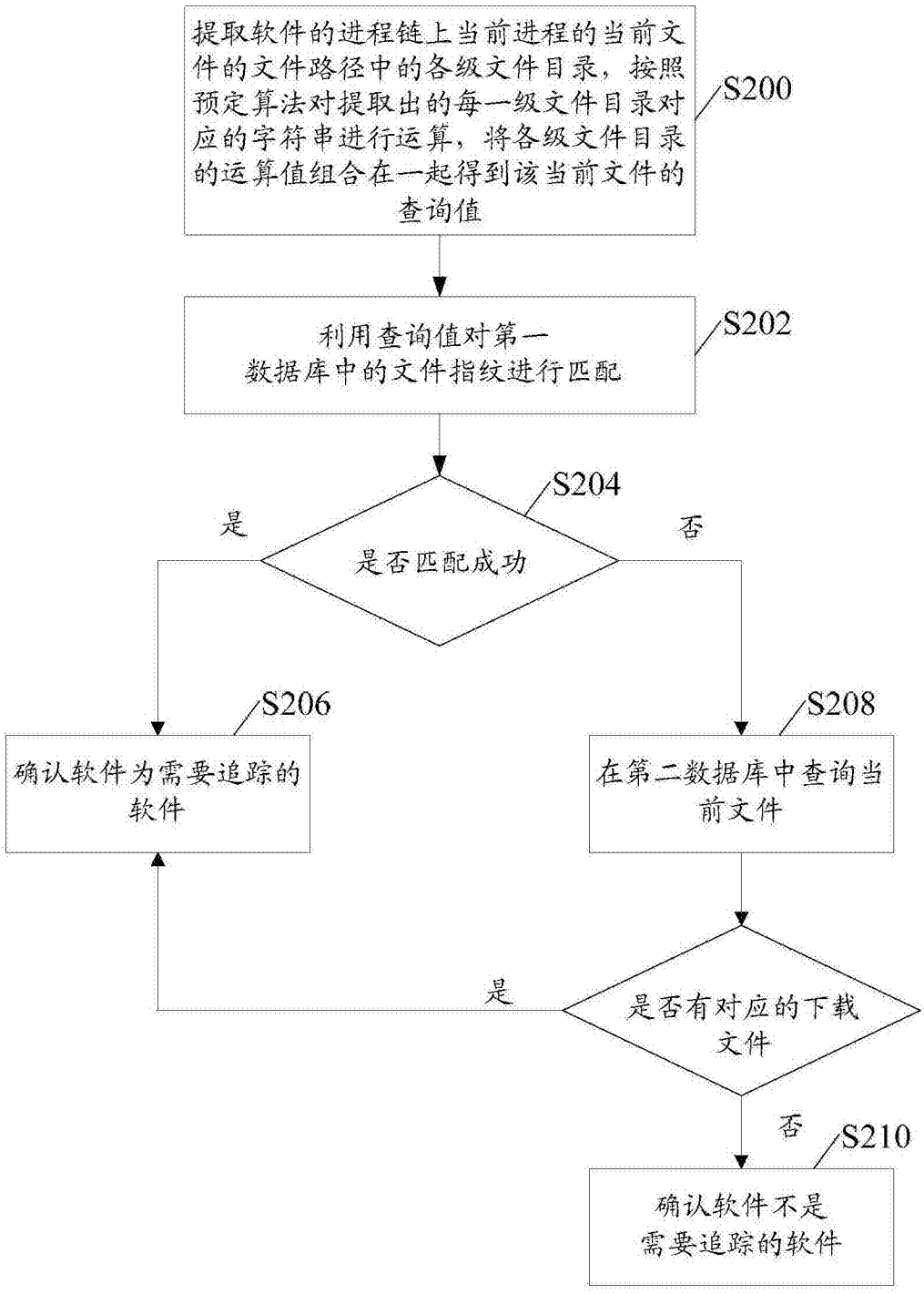

提取软件的进程链上当前进程的当前文件的文件路径中的各级文件目录，按照预定算法对提取出的每一级文件目录对应的字符串进行运算，将各级文件目录的运算值组合在一起得到该当前文件的查询值
S200
利用查询值对第一数据库中的文件指纹进行匹配
S202
是否匹配成功
S204
是
否
确认软件为需要追踪的软件
S206
在第二数据库中查询当前文件
S208
是否有对应的下载文件
是
否
确认软件不是需要追踪的软件
S210

图2

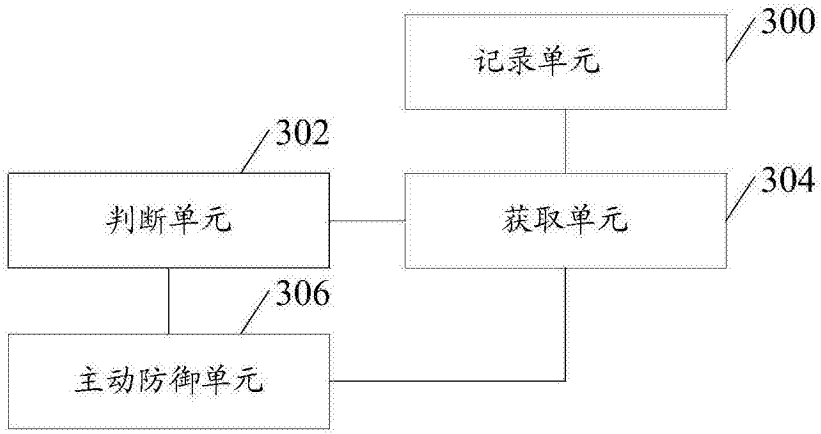


图3