РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 451 326** (13) **C2**

(51) МПК
G06F 11/30 (2006.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2009120574/08, 30.11.2007**

(24) Дата начала отсчета срока действия патента:
30.11.2007

Приоритет(ы):
(30) Конвенционный приоритет:
01.12.2006 US 11/566,170

(43) Дата публикации заявки: **10.12.2010** Бюл. № 34

(45) Опубликовано: **20.05.2012** Бюл. № **14**

(56) Список документов, цитированных в отчете о поиске: **US 7124400 B2, 17.10.2006. US 2003/0046676 A1, 06.03.2003. US 6041333, 21.03.2003. US 2005/0010916 A1, 13.01.2005. RU 2192039 C2, 27.10.2002.**

(85) Дата начала рассмотрения заявки PCT на национальной фазе: **29.05.2009**

(86) Заявка PCT:
US 2007/086195 (30.11.2007)

(87) Публикация заявки PCT:
WO 2008/070587 (12.06.2008)

Адрес для переписки:
129090, Москва, ул. Б. Спасская, 25, стр.3, ООО "Юридическая фирма Городисский и Партнеры", пат.пов. Ю.Д.Кузнецову, рег.№ 595

(72) Автор(ы):
ВЕРБОВСКИ Чад (US),
ЛИ Дзухан (US),
ЛЮ Сяоган (US),
РУССЕВ Русси (US),
ВАН И-минь (US)

(73) Патентообладатель(и):
МАЙКРОСОФТ КОРПОРЕЙШН (US)

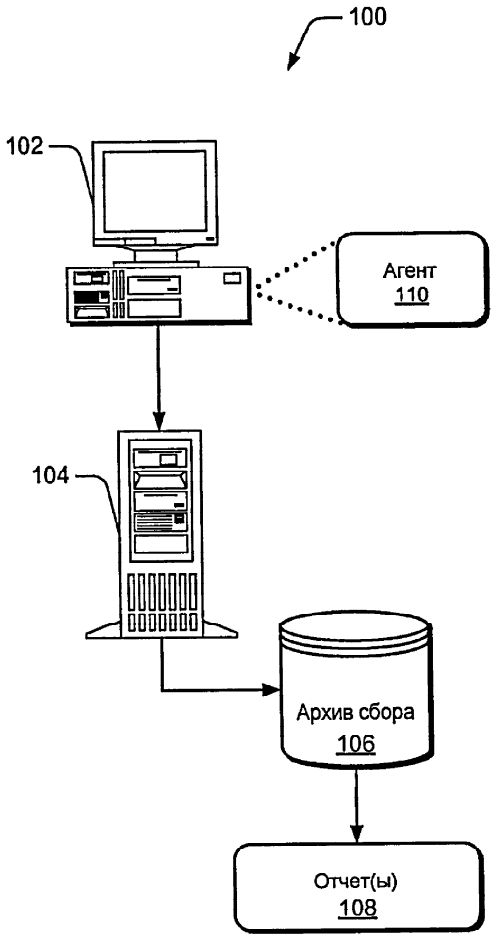
RU 2451326 C2

(54) СИСТЕМНЫЙ АНАЛИЗ И УПРАВЛЕНИЕ

(57) Реферат:

Изобретение относится к средствам поддержания устойчивого состояния системы. Технический результат заключается в эффективном управлении устойчивым состоянием компьютерной системы. Обнаруживают устаревшие файлы. Каталогизируют программы, загруженные на вычислительное устройство, и получают временные моменты последней загрузки файлов, связанных с программами, которые зарегистрированы в вычислительном устройстве. Получают временные моменты последней модификации файлов, связанных с упомянутыми программами, и сравнивают временные моменты последней модификации с временными моментами последней загрузки файлов, связанных с упомянутыми программами, которые зарегистрированы в вычислительном устройстве. Уведомляют о любых несоответствиях, найденных при сравнении, что включает в себя сообщение, что программа не отвечает на последнюю попытку модификации, и попытку повторно выполнить

последнюю попытку модификации программы. 3 н. и 11 з.п. ф-лы, 11 ил.



Фиг.1

RU 2451326 C2

RU 2451326 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 451 326** (13) **C2**

(51) Int. Cl.
G06F 11/30 (2006.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2009120574/08, 30.11.2007**

(24) Effective date for property rights:
30.11.2007

Priority:
(30) Convention priority:
01.12.2006 US 11/566,170

(43) Application published: **10.12.2010** Bull. 34

(45) Date of publication: **20.05.2012 Bull. 14**

(85) Commencement of national phase: **29.05.2009**

(86) PCT application:
US 2007/086195 (30.11.2007)

(87) PCT publication:
WO 2008/070587 (12.06.2008)

Mail address:
129090, Moskva, ul. B. Spasskaja, 25, str.3, OOO "Juridicheskaja firma Gorodisskij i Partnery", pat.pov. Ju.D.Kuznetsovu, reg.№ 595

(72) Inventor(s):
VERBOVSKI Chad (US),
LI Dzukhan (US),
LJu Sjaogan (US),
RUSSEV Russi (US),
VAN I-min' (US)

(73) Proprietor(s):
MAJKROSOFT KORPOREJShN (US)

RU 2451326 C2

(54) SYSTEM ANALYSIS AND CONTROL

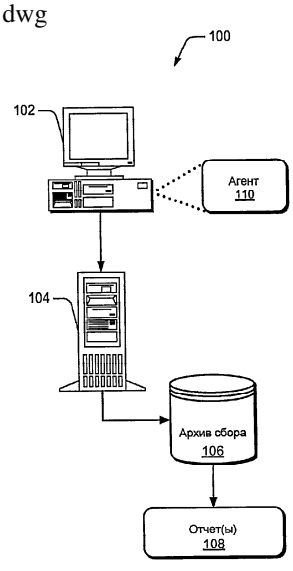
(57) Abstract:

FIELD: information technology.

SUBSTANCE: outdated files are identified. Programs loaded on a computing device are catalogued and time moments for the last loading of files associated with the programs registered in the computing device are determined. Time moments for the last modification of files associated with said programs are obtained and then compared with time moments of the last loading of the files associated with said programs registered in the computing device. Any disparities found during comparison are signalled, which includes a message that the program is not responding to the last modification attempt, and an attempt to repeat the last program modification attempt.

EFFECT: efficient management of the stable state of a computer system.

14 cl, 11 dwg



Фиг.1

RU 2451326 C2

Уровень техники

Главной проблемой при создании надежной и безопасной компьютерной системы является поддержание устойчивого состояния (УС) системы, которое включает в себя все исполняемые файлы, конфигурационные настройки и другие данные, которые регулируют функционирование системы. Неправильные конфигурации и другие проблемы УС входят в число основных причин сбоев и уязвимостей безопасности множества различных систем - от отдельных настольных машин до крупномасштабных Интернет-сервисов. Проблемы УС вместе с проблемами, вызываемыми сбоями в системных элементах, таких как аппаратные компоненты и программная логика, могут оказать пагубное влияние на всю систему.

Цена невыполнения эффективного поддержания УС системы очень высока. Например, проблемы УС могут воспроизводить сами себя после перезагрузки системы или перезапуска приложения. Дополнительно, УС подвергается сдвигу в течение времени исполнения из-за изменений, таких как заплатки и обновления, связанных с приложениями. В настоящее время не существует эффективных способов замыкания цикла изменений, возникающих в системе. В подобном сценарии если не удается идентифицировать известную проблему и если последующая перезагрузка системы/перезапуск проблемы не исправляет проблему УС, то единственной возможностью решения проблемы УС может быть исследование системы вручную, чтобы идентифицировать основную причину УС.

Ручное исследование системы в целях идентификации основной причины УС представляется сложным и дорогостоящим из-за большого количества потенциальных проблем. Например, потенциальный набор состояний, которые могут влиять на испытывающее проблемы приложение, огромен, и список потенциальных основных причин может включать в себя весь набор состояний системы. Сверх того, ситуация может быть потенциально хуже, если также рассматривать каждую возможную комбинацию из этого набора состояний для случая, где нет одной отдельной основной причины УС.

Раскрытие изобретения

Настоящий раздел предоставлен для представления упрощенных концепций настоящего изобретения, которые подробно описаны в разделе "Осуществление изобретения". Раздел "Раскрытие изобретения" не предназначен ни для определения существенных отличительных признаков заявленного изобретения, ни для использования при определении объема заявленного изобретения.

В одном варианте осуществления программы вычислительного устройства каталогизируют и перечисляют, получают временные моменты последней загрузки программ, зарегистрированных в вычислительном устройстве, и выполняют сравнение временных моментов последней модификации файлов, связанных с этими программами, которые зарегистрированы в упомянутом вычислительном устройстве, с временными моментами последней загрузки.

Краткое описание чертежей

Осуществление изобретение приведено со ссылкой на прилагаемые чертежи. На чертежах крайняя левая цифра ссылочного номера обозначает чертеж, на котором данный ссылочный номер используется в первый раз. Для обозначения схожих признаков или компонентов во всех чертежах используются одинаковые номера.

Фиг.1 - иллюстрация примера архитектуры для управления системой;

Фиг.2 - иллюстрация примера сервера сбора;

Фиг.3 - иллюстрация примера графического интерфейса, отображающего

сгенерированное уведомление;

Фиг.4 - иллюстрация примера графического интерфейса, отображающего зависимость исполнения одной программы от исполнения первой программы;

Фиг.5 - иллюстрация примера(ов) способа(ов) для захвата данных, связанных с модификацией УС системы;

Фиг.6 - иллюстрация примера(ов) способа(ов) для классификации отмеченных изменений;

Фиг.7 - иллюстрация примера(ов) способа(ов) для запрета исполнения неавторизованных взаимодействий;

Фиг.8 - иллюстрация примера(ов) способа(ов) для детектирования одной или более точек расширяемости;

Фиг.9 - иллюстрация примера(ов) способа(ов) для детектирования просочившихся элементов;

Фиг.10 - иллюстрация примера(ов) способа(ов) для детектирования общих неправильных конфигураций или устаревших файлов;

Фиг.11 - иллюстрация примера вычислительного окружения.

Осуществление изобретения

Для реализации настоящего изобретения система включает в себя одну или более компьютерных программ или агентов, которые сообщают данные, связанные с происходящими внутри системы модификациями. Эти данные включают в себя информацию, связанную со всеми взаимодействиями с файлами и/или настройками. Подобные типы взаимодействий включают в себя такие действия, как чтение или запись в элементы системного реестра, файлы, а также взаимодействия бинарного модуля, такие как загрузка и т.д. Агенты сообщают собранные данные в серверную службу, которая обрабатывает полученную информацию для, например, генерации веб-отчетов, предупреждений или для интегрирования с другими службами в целях выполнения управления системой. Сверх того, обработка также может быть выполнена на той же машине, где были собраны эти данные. Это включает в себя генерацию отчетов, предупреждений и т.п. В частности, рассматривается УС системы, причем УС включает в себя все исполняемые файлы, конфигурационные настройки и другие данные, которые регулируют функционирование системы. Несмотря на то что описывается устойчивое состояние, следует понимать, что описанные способы и методы применимы также к другим типам состояний.

Сообщаемые данные могут быть использованы в нескольких целях. Например, данные могут быть исследованы, чтобы верифицировать, что инициируемые взаимодействия соответствуют установленной политике или связаны с авторизованным взаимодействием.

Наряду с тем, что аспекты описанных систем и способов для управления системой могут быть реализованы в любых различных вычислительных системах, окружениях и/или конфигурациях, варианты осуществления настоящего изобретения описаны в контексте следующего конкретного примера архитектуры (архитектур) системы.

Иллюстративная система

Фиг.1 представляет собой иллюстрацию примера компьютерной системы 100, в которой может быть собрана и проанализирована информация, связанная с взаимодействиями между одной или более программами. Система 100 включает в себя вычислительное устройство 102, на котором выполняются или инсталлированы одна или более программ, сервер 104 сбора, архив 106 сбора и отчет(ы) 108.

Информация, связанная с взаимодействиями между одной или более программами

и/или файловыми системами и настройками, представляет модификации устойчивого состояния (УС), которое может иметь место в системе 100. Вычислительное устройство 102 может включать в себя любое количество вычислительных устройств 102. Например, в одном варианте осуществления система 100 также может включать в себя корпоративную сеть, включающую в себя тысячи офисных персональных компьютеров, различных серверов и других вычислительных устройств, которые расположены в разных странах и действуют как вычислительные устройства 102. Альтернативно, в еще одном возможном варианте осуществления система 100 может включать в себя домашнюю сеть с ограниченным количеством персональных компьютеров. Вычислительные устройства 102 могут быть соединены друг с другом в различных комбинациях через проводную и/или беспроводную сеть, включающую в себя локальную сеть, глобальную сеть или любую другую сеть, основанную на известных сетевых технологиях.

Вычислительные устройства 102 могут включать в себя агент 110 с функциями контроля, чтобы захватывать информацию, связанную с взаимодействиями между одним или более вычислительными устройствами 102 и/или файловыми системами и настройками. В одном варианте осуществления агент 110 может представлять собой Регистратор Данных Потока (Thread Data Recorder, TDR), способный модифицировать, добавлять и/или удалять машиночитаемые инструкции в функции, чтобы перехватывать потоки, вызывающие эту функцию. В еще одном возможном варианте осуществления функции контроля также включают в себя модифицирование, добавление и/или удаление машиночитаемых инструкций в функции, чтобы принудить поток к выполнению машиночитаемых инструкций в функции, которая позволяет захватывать данные, связанные с этим потоком. В еще одном варианте осуществления данные, связанные с потоком, включают в себя информацию, относящуюся к программе, с которой связан этот поток, одно или более взаимодействий, связанных с этим потоком, а также информацию, относящуюся к пользователю программы, с которой связан этот поток. Несмотря на то в настоящем документе описывается вариант с применением TDR, следует понимать, что перехват необязательно выполнять для всех алгоритмов, и, следовательно, описанные способы и методы могут быть не связаны со сбором данных, основанным на TDR. Сверх того, контроль, основанный на применении Виртуальной Машины (VM), может отличаться от контроля на базе TDR, где код добавляется динамическим образом. В случае VM контроль может быть реализован как функция жестко запрограммированных внутренних компонентов VM, предназначенных для выполнения сбора этого типа.

Оснащенные функции могут включать в себя функции, которые могут быть вызваны программой/процессом. В одном варианте осуществления оснащенные функции могут включать в себя функции низкого уровня, такие как драйверы файловой системы, функции системного реестра, функции, создающие новые процессы и/или службы и т.п.

Данные, захватываемые из потоков посредством регистратора данных потока, могут быть сохранены и/или обработаны, чтобы регулировать поведение системы 100 и чтобы исследовать режим или устойчивое состояние системы 100. Типы данных, которые могут быть захвачены из потоков посредством регистратора данных потока, а также порядок работы регистратора данных потока подробно описаны в заявке на патент США "Перехват и анализ Потока" Вербовки и др., с серийным номером _____, поданной _____, которая включена в настоящий документ посредством ссылки.

Сервер 104 сбора отвечает за сбор информации, связанной с модификацией, которая происходит в системе 100. В одном варианте осуществления агент 110 сохраняет информацию, связанную с взаимодействиями, в сервере 104 сбора в виде сжатых регистрационных журналов. В еще одном варианте осуществления информация, связанная с взаимодействиями, может быть, сверх того, загружена в архив 106 сбора. Анализ информации, собранной в сервере 104 сбора или архиве 106 сбора, используется для генерации отчета(ов) 108. Отчет(ы) 108, сгенерированный в результате анализа информации, собранной в сервере 104 сбора или архиве 106 сбора, предоставляет способность понимания взаимодействий или модификаций, которые могут происходить в одном или более вычислительных устройствах 102. В еще одном варианте осуществления отчет(ы) 108 может генерироваться через графический интерфейс. В еще одном варианте осуществления графический интерфейс может быть реализован посредством браузера для извлечения и отображения ранее созданных и/или кэшированных отчетов, а также для выполнения программируемого доступа к информации, хранимой в сервере 104 сбора или архиве 106 сбора. Сервер 104 сбора и архив 106 сбора могут находиться в одном устройстве или быть частью одного устройства, которое выполняет роль сервера 104 сбора или архива 106 сбора.

Как упоминалось выше, информация, собранная агентом 110 и сохраненная в сервере 104 сбора или архиве 106 сбора, может быть проанализирована, чтобы предоставить способность понимания того, как функционирует система 100. Выполняемый анализ может включать в себя детектирование аномалий, управление изменениями, управление аномальной системной активностью, идентификацию уязвимостей безопасности, идентификацию неавторизованных приложений, выполнение аудита соответствия и т.п.

Фиг.2 представляет собой иллюстрацию примера сервера 106 сбора, сконфигурированного так, чтобы сохранять, обрабатывать и/или анализировать получаемые от агента 110 данные. Сервер 106 сбора включает в себя один или более процессоров 202 и память 204. Процессор 202 включают в себя, например, микропроцессоры, микрокомпьютеры, микроконтроллеры, цифровые процессоры сигналов, центральные процессорные блоки, конечные автоматы, логические схемы и/или любые другие устройства, которые манипулируют сигналами на основе операционных инструкций. Кроме того, процессоры 202 сконфигурированы так, чтобы выбирать и исполнять машиночитаемые инструкции, хранимые в памяти 204.

Память 204 может представлять собой любой известный машиночитаемый носитель, например энергозависимую память (например, ОЗУ) и/или энергонезависимую память (например, ПЗУ, флэш-память и т.п.). Память 204 также может включать в себя программу(ы) 206 и данные 208. Программа(ы) 206 может выполнять, среди прочего, связанные с запросами процессы на данных, связанных с взаимодействиями между программами, выполняемыми на одном или более вычислительных устройствах 102, и файловой системой и/или настройками. Программа(ы) 206, сверх того, включает в себя, например, модуль 210 опроса, модуль 212 уведомления, операционную систему 214 и другие приложения 216. Операционная система 214 предоставляет рабочее окружение для функционирования одного или более модулей в программе(ах) 206.

Модуль 210 опроса выполняет основанные на запросах операции над информацией, собранной агентом 110, такой как информация, включенная в хранилище 218 регистрационных журналов. Собранная агентом 110 информация также может быть извлечена из архива 106 сбора. Запрос(ы) 220 включает в себя множество запросов,

таких как предопределенные запросы. Подобные предопределенные запросы могут относиться к состояниям, связанным с одной или более политиками, такими как политика безопасности, которая может быть предписана для системы 100. В таком случае какой-либо один или все анализы, которые могут выполняться модулем 210 опроса, могут соответствовать подобной предопределенной политике или предопределенным запросам.

Модуль 210 опроса может ограничивать запрос(ы) 220 одним или более атрибутами. Подобные атрибуты могут включать в себя имя файла, тип приложения, время выполнения и т.п. При работе на основе ограниченного запроса модуль 210 опроса сканирует всю информацию, которая хранится в хранилище 218 регистрационных журналов и/или архиве 106 сбора, на предмет наличия значений, которые указывают присутствие атрибута. Например, если нужно выполнить в архиве 106 сбора поиск данных, связанных с определенным приложением, таким как текстовой процессор, то модуль 210 опроса выполняет поиск элементов или событий, связанных с взаимодействиями, которые были инициированы и затронуты текстовым редактором.

Запрос(ы) 220 может включать в себя запросы, вводимые или программируемые одним или более лицами или объектами, как, например, системным администратором. Например, запрос(ы) 220 может включать в себя инструкции для детектирования взаимодействий, связанных с заданным идентификатором пользователя. Сверх того, запрос(ы) 220 может включать в себя инструкции для детектирования всех взаимодействий, связанных с приложением, которое выполняется на одном или более вычислительных устройствах 102.

Возвращаясь к серверу 104 сбора, анализ информации, связанной с взаимодействиями одного или более вычислительных устройств 102 с файловыми системами и/или настройками, должен быть выполнен таким образом, чтобы определить рабочее состояние и/или устойчивое состояние системы 100. Модуль 210 опроса может быть использован для выполнения анализа информации, собранной агентом 110 и сохраненной в хранилище 218 регистрационного журнала и/или архиве 106 сбора. Модуль 210 опроса может реализовывать это путем выполнения поиска в хранилище 218 регистрационных журналов и/или архиве 106 сбора, используя один или более запросов, определяемых запросом(ами) 220. Генерируемые путем выполнения запроса(ов) 220 результаты указывают взаимодействия между одним или более вычислительными устройствами 102 и файловыми системами и/или настройками.

Модуль 210 опроса может предписывать модулю 212 уведомления выдавать уведомление для результатов, сгенерированных путем выполнения запроса(ов) 220. Уведомление, генерируемое модулем 212 уведомления, может быть сохранено в уведомлении(ях) 222 в данных 208. Уведомления, выдаваемые модулем 212 уведомления, могут быть сохранены во внешней базе данных, такой как внешнее устройство хранения. Модуль 212 уведомления также может получить от модуля 210 опроса команду передачи уведомлений, сгенерированных в результате выполнения запроса(ов) 220.

Модуль 210 опроса может выполнять поиск в хранилище 218 регистрационных журналов и/или архиве 106 сбора, чтобы детектировать отклонения в информации, связанной с взаимодействиями между одним или более вычислительными устройствами 102, относительно запроса(ов) 220, выполняемого модулем 210 опроса. В подобном случае детектирование таких отклонений в отношении определенных взаимодействий также может быть сообщено модулем 212 уведомления, и

соответствующее уведомление(ия) 222 может быть передано некоторым лицам, таким как системный администратор, или вычислительным системам в целях сохранения уведомлений в качестве справки для будущего.

Модуль 212 уведомления также способен предоставлять контекстную информацию, связанную с уведомлением(ями) 222. Контекстная информация может дополнительно описывать настройку, которая может быть связана с соответствующим взаимодействием. Контекстная информация может снабжать комментариями соответствующее уведомление(ия) 222 на одном или более уровнях. Например, один уровень предоставляет статистическую информацию, относящуюся к количеству машин, которые могут содержать установленную на них программу, к общей версии файлов и т.д. Еще один уровень комментариев указывает сравнение хеш-значений инсталлированных файлов с фондом данных, в котором индексируются атрибуты, например имя программы, информация версии и т.п. Еще один уровень комментариев может предоставлять комментарии или какую-либо вспомогательную информацию, относящуюся к известным проблемам, поставщикам и т.п. Таким образом, могут быть реализованы дополнительные уровни комментариев, которые описывают дополнительные атрибуты в отношении связанного уведомления(ий) 222. Уведомление(я) 222 также может быть отображено через графический интерфейс, позволяющий некоторым лицам, например системному администратору, просматривать уведомление(я) 222 и предпринимать необходимые действия.

Модуль 210 опроса может быть использован для детектирования отмеченных изменений, которые имеют место в системе 100 из-за взаимодействия между программами, выполняемыми на одном или более вычислительных устройствах 102, и файловыми системами и/или настройками. Отмеченные изменения включают в себя изменения или модификации УС системы, которые могут возникнуть в результате неожиданного исполнения программы, операционной системы, программ, которые используются для выполнения специфичных задач, такие как бухгалтерские и иные программы. Подобные отмеченные изменения УС системы авторизуются и управляются, чтобы предотвращать нежелательные ситуации, такие как снижение производительности системы, проблемы безопасности и т.п. Также следует отметить, что не все изменения, которые возникают в УС, являются отмеченными изменениями.

Отмеченные изменения могут быть снабжены идентифицирующим комментарием и классифицированы на основании присвоенного комментария. Комментирование отмеченных изменений может быть выполнено путем спецификации правила классификации модулем 210 опроса. На основании параметров, указанных в правиле классификации, подходящие параметры связываются с определенными атрибутами, относящимися к отмеченному изменению. Например, модуль 210 опроса ассоциирует каждое совпадение подстроки, содержащейся в правиле классификации, с именем или типом модификации, содержащимся в каждом отмеченном изменении. Классификации для отмеченных изменений могут быть назначены на основе величины приоритета. Например, в подобном случае совпадения с подстроками классификации с более высоким приоритетом имеют старшинство относительно подстрок классификации с более низким приоритетом. Тогда подстроки классификации с более высоким приоритетом определяются как подходящая классификация для соответствующего отмеченного изменения.

Отмеченные изменения могут быть классифицированы путем маркирования изменений согласно одной или более из следующих классификаций:

• Проблема: Указывает известную проблему или результаты существования или

удаления текущего УС.

• Инсталляция: Указывает изменение УС в результате инсталляции или модернизации.

• Настройка: Указывает изменения, выполненные относительно конфигурационных настроек или конфигурации УС.

• Содержимое: Указывает веб-страницы, изображения, текстовые и пользовательские данные.

• Изменение Управления: Указывает изменения инсталляции, заплатки или конфигурации, которые были выполнены в программах, ответственных за управление системой.

• Неавторизованный: Указывает инсталляцию неавторизованных или неразрешенных приложений или изменения конфигурации, которые включают в себя запрещенные величины.

• Пользовательская Активность: Указывает изменения УС в результате пользовательских действий по регистрации или выполнению приложений.

• Шум: Указывает временное или кэшированное УС.

• Неизвестно: Указывает неклассифицированное УС.

Для более подробной классификации отмеченных изменений и обеспечения возможности отличать их от других изменений могут быть предоставлены дополнительные комментарии.

Модуль 210 опроса также может быть использован для определения статуса авторизации выполняемой в системе 100 программы (авторизована/неавторизована). Это может быть основано на требовании, что в системе 100 должны выполняться только авторизованные процессы или программы. Модуль 210 опроса определяет статус выполняемой в системе 100 программы как авторизованный или неавторизованный путем сравнения атрибутов, указанных в запросе(ах) 220, и атрибутов, которые определяют конкретное изменение или модификацию УС системы. Например, модуль 210 опроса выполняет запрос(ы) 220, который указывает тип приложения как "неавторизованная программа". Результаты, полученные при выполнении запроса(ов) 220, содержат информацию, относящуюся к изменениям УС, которые имели место под действием выполнения указанного типа приложения. При получении этих результатов модуль 210 опроса маркирует их как изменения, вызванные выполнением или действием неавторизованной программы.

Модуль 210 опроса может сравнивать атрибуты, указанные в предварительно сформированном списке согласованных и/или несогласованных программ, с атрибутами, которые описывают конкретное изменение или модификацию в УС системы. Данный список в этом случае может содержать конкретное число согласованных или несогласованных программ. Выполняемые в системе 100 программы, которые схожи с программами, идентифицированными в предварительно сформированном списке как несогласованные, маркируются как неавторизованные программы.

Согласованные и/или несогласованные программы, указанные в предварительно сформированном списке, также могут содержать дополнительную информацию, такую как метка, описывающая природу и/или различные характеристики программ. Примеры подобной дополнительной информации включают в себя такие метки, как "согласованный", "тип", "категория", "информация продукта", "информация производителя" и "описание продукта". Например, программы, снабженные меткой "согласованный", рассматриваются как авторизованные программы для выполнения

на одном или более вычислительных устройствах 102 в системе 100, а метка "категория" указывает целевое использование программы.

Изменения или модификации, выполняемые программой в первый раз, по умолчанию являются несогласованными и маркируются как "несогласованные". Например, при детектировании изменений или модификаций, выполняемых программой в первый раз, модуль 210 опроса маркирует эту программу и ее соответствующие взаимодействия как "неавторизованные". Подобные программы, которые были маркированы как "неавторизованные", могут быть переданы модулем 212 уведомления для анализа, например, системным администратором в целях выполнения диагностики или принятия решения по согласованию. Если получается согласование, то согласованная программа, сверх того, ассоциируется с подходящей меткой, описывающей программу, и она может быть добавлена в предварительно сформированный список согласованных и/или несогласованных программ.

Модуль 210 опроса также может детектировать Точки Расширяемости (ТР). Точки расширяемости представляют собой взаимодействия, которые указывают динамическую загрузку и выполнение инструкций, связанных с программой или операционной системой, выполняемой на одной или более вычислительных машинах 102. Например, когда запускается первая программа, такая как текстовой процессор, приложение для работы с электронными таблицами и т.п., выполняемая на одном или более вычислительных устройствах 102, она может также инициировать выполнение инструкций, связанных с другими программами, такими как дополнительные программы-надстройки, которые предоставляют дополнительные функции для первой программы. Таким образом, выполнение первой программы может сгенерировать различные взаимодействия, включающие в себя взаимодействия между первой программой и файловой системой, и взаимодействия между другими программами, предоставляющими дополнительные функциональные возможности при выполнении первой программы, и файловой системой. Подобная информация может предоставлять возможность понимания функционирования системы, на которой инсталлирована первая программа, а также понимать влияние, которое оказывают на систему подобные инсталляции.

Информация, связанная с различными взаимодействиями, которые генерируются в результате выполнения первой программы, может быть перехвачена и скопирована, например, агентом 110. Информация события, связанная с различными взаимодействиями, может быть сохранена в виде сжатых регистрационных журналов в хранилище 218 регистрационных журналов и/или архиве 106 сбора. Наряду с тем, что информация события сохраняется в сжатом виде, следует понимать, что применение сжатой формы необязательно. Тем не менее, применение сжатой формы обеспечивает лучшую масштабируемость системы благодаря меньшему объему требуемого пространства. Сохраненная информация события может быть исследована объектами, такими как системный оператор, или модулем 210 опроса, чтобы детектировать взаимодействия, связанные с первой программой и другими программами в файловой системе. Таким образом, могут быть детектированы другие программы, если они связаны с выполнением первой программы.

Модуль 210 опроса также может быть использован для детектирования непосредственных точек расширяемости для первой программы. Например, модуль 210 опроса может детектировать непосредственные точки расширяемости путем изолирования взаимодействий, которые (1) относятся к различным программам,

загруженным для выполнения в системную память до выполнения первой программы, и которые (2) ссылаются на первую программу или связаны с выполнением первой программы.

В одном примере осуществления модуль 210 опроса может идентифицировать потенциальные непосредственные точки расширяемости для первой программы путем опроса хранилища 218 регистрационных журналов и/или архива 106 сбора на предмет взаимодействий, относящихся к различным программам, загруженным для выполнения в системную память до выполнения первой программы. Например, модуль 210 опроса может выполнить запрос взаимодействий, относящихся к различным программам, загруженным для выполнения в системную память в заданном временном интервале, таком как одна секунда, до выполнения первой программы. Модуль 210 опроса может идентифицировать из потенциальных точек расширяемости непосредственные точки расширяемости для первой программы путем опроса потенциальных точек расширяемости на предмет взаимодействий, которые ссылаются на первую программу или которые связаны с выполнением первой программы. Непосредственные точки расширяемости могут храниться в других данных 224.

Модуль 210 опроса также может быть использован для детектирования косвенных точек расширяемости. Например, в случае вышеописанного примера первой программы модуль 210 опроса может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет взаимодействий, которые ссылаются на непосредственные точки расширяемости или которые связаны с ними. Подобные взаимодействия могут быть обозначены термином "косвенные точки расширяемости". Косвенные точки расширяемости могут храниться в других данных 224.

Модуль 210 опроса также может быть использован для детектирования присутствия вредоносных программных приложений путем мониторинга непосредственных точек расширяемости. Вредоносные программные приложения могут включать в себя "шпионские программы", "троянские кони", "черви" и т.п., которые при нормальных обстоятельствах не будут связаны с программой. Например, модуль 210 опроса может сравнить точки расширяемости для программы, выполняемой на одном или более вычислительных устройствах 102, с контрольными точками расширяемости для той же программы, которые были обнаружены, когда эта программа выполнялась на вычислительных устройствах 102 при отсутствии вредоносного программного обеспечения. Различия между упомянутыми точками расширяемости и контрольными точками расширяемости могут быть исследованы объектами, такими как модуль 210 опроса и/или системный администратор, чтобы определить, указывают ли эти различия на присутствие вредоносного программного обеспечения, выполняемого в связи с этой программой. Вредоносное программное обеспечение, обнаруженное с помощью точек доступа, может быть удалено из затронутого вычислительного устройства 102 модулем 210 опроса, системным оператором и т.п.

В еще одном варианте осуществления модуль 210 опроса может быть использован для генерации уведомления(ий) 222 модулем 212 уведомления в ответ на детектирование точек расширяемости. В еще одном варианте осуществления генерируемое уведомление(я) 222 может быть, сверх того, просмотрено или извлечено через графический интерфейс, что облегчает исследование уведомления(ий) 222 некоторым лицом, таким как системный администратор, в целях дальнейшего анализа или выполнения запрашиваемого диагностирования.

Фиг.3 представляет собой иллюстрацию примера графического интерфейса 300, на которой показано сгенерированное уведомление(я) 222 в одном из возможных вариантов осуществления. В проиллюстрированном варианте осуществления на графическом интерфейсе 300 отображены загрузки, которые были выполнены первой программой (например, веб-браузером) при его исполнении. В сегментах 302 и 304 графического интерфейса показан список программ, которые были загружены первой программой (в конкретном примере на Фиг.3 это программы "MSN Search Toolbar" и "Winamp Media Player") в течение ее исполнения. Можно увидеть, что загрузка программ, отображенных в сегментах 302, 304, также приводит к созданию программных файлов, соответствующих программам, отличающимся от первой программы, и программам, проиллюстрированным в сегментах 302, 304. Таким образом, визуальное представление в форме графического интерфейса 300 предоставляет список программ, которые неумышленно инсталлируются на одном или более вычислительных устройствах 102 системы 100 в течение выполнения, загрузки и/или инсталляции первой программы.

Сегмент 306 также может указывать влияние или модификации УС системы 100, образуемые в результате инсталляции или выполнения программ, отличных от первой программы. Дальнейшее выполнение программы или программ, отличных от первой программы, может зависеть от выполнения первой программы. Например, проиллюстрированная программа "MSN Search Toolbar" может активироваться при выполнении программных файлов первой программы. Детектирование этого эффекта может быть осуществлено посредством детектирования точек расширяемости, соответствующих первой программе. Путем мониторинга точек расширяемости, связанных с первой программой, могут быть детектированы случаи выполнения других программ, зависящие от выполнения первой программы, и при необходимости могут быть предприняты корректирующие действия.

Как показано на Фиг.4, случаи выполнения других программ, которые зависят от выполнения первой программы, также могут быть отображены через другой графический интерфейс 400. На Фиг.4 проиллюстрировано выполнение первой программы, показанной в виде сегмента 402, например "iexplorer.exe", которое вызывает выполнение программы Winamp, показанной в виде сегмента 404, которая в свою очередь выполняет "emusic.exe", показанную в виде сегмента 406. Из данной иллюстрации видно, что с помощью графических средств можно осуществить подробное детектирование точек расширяемости, связанных с первой программой.

Модуль 210 опроса может быть использован для детектирования УС просачивания. Просочившиеся файлы включают в себя файлы или настройки системного реестра, которые остаются в системе, такой как система 100, после того как программа, которая создала эти файлы или настройки системного реестра, деинсталлируется. Они также могут включать в себя файлы и настройки, которые могли быть созданы в результате инсталляции, например временные файлы, но которые не удалось удалить после завершения процесса инсталляции. Сверх того, еще один класс УС может рассматриваться как УС просачивания, такой как УС, которые генерируются в течение времени выполнения программы (то есть после инсталляции). Примерами этого типа являются состояния, которые могут быть сгенерированы при первом использовании или при инсталляции расширений программы, которые отдельно инсталлируются после исходной инсталляции.

Для детектирования просочившихся файлов модуль 210 опроса каталогизирует изменения инсталляционных файлов и настроек, связанные с каждой загруженной в

систему 100 программой. Эти изменения могут быть отслежены через использование программ, а также начальную инсталляцию. Позже, если эта программа деинсталлируется, то может быть вызван соответствующий каталог инсталляционных файлов и конфигурационных настроек или настроек системного реестра для этой программы и система 100 может быть проверена, чтобы убедиться в том, что все инсталляционные файлы и настройки системного реестра были удалены или сброшены на исходные значения. Для детектирования просочившихся файлов на вычислительном устройстве 102 модуль 210 опроса каталогизирует инсталляционные файлы путем выполнения сканирования среди одного или более вычислительных устройств 102, чтобы детектировать все программы, такие как приложения, инсталлированные на вычислительных устройствах 102.

Модуль 210 опроса также может получить список всех программ, зарегистрированных в базе данных инсталляторов операционной системы одного или более вычислительных устройств 102. Примеры базы данных инсталляторов включают в себя компоненты, которые производят наполненный список программ, инсталлированных на рассматриваемом вычислительном устройстве.

Модуль 210 опроса выполняет опрос хранилища 218 регистрационных журналов и/или архива 106 сбора для выявления информации конфигураций или настроек системного реестра и для перечисления списка программ, зарегистрированных в операционных системах вычислительных устройств 102. Далее, модуль 210 опроса может сканировать хранилище 218 регистрационных журналов и/или архив 106 сбора, чтобы перечислить файлы и элементы системного реестра, которые могут быть обобщены на все УС всех программ, установленных на вычислительном устройстве 102. Чтобы перечислить файлы и элементы системного реестра, модуль 210 опроса может опросить все файлы и элементы системного реестра, соответствующие одному или более атрибутам, например идентификаторам программ, инсталлированных на вычислительном устройстве 102.

Если файл или настройка вычислительного устройства не входит в число файлов или элементов системного реестра, соответствующих идентификаторам программ, инсталлированных на вычислительном устройстве 102, то модуль 210 опроса может сделать вывод, что этот файл или настройка представляет собой просочившийся файл. Просочившиеся файлы могут быть удалены модулем 210 опроса или различными другими программами, в том числе операционной системой, системным оператором и т.п.

Детектированные просочившиеся файлы могут быть отображены через графический интерфейс, позволяющий некоторым лицам, например системному администратору, просматривать просочившиеся файлы и предпринимать необходимые действия. Сверх того, отображенные просочившиеся файлы (УС) и связанная информация могут быть сохранены во внешнем фонде хранения, например внешней базе данных, как справка для будущего применения. Список УС просачивания может быть использован для автоматического удаления системами состояния просачивания, когда удаляется главное приложение. Упомянутый список УС просачивания также может быть использован для ассоциации каждого УС в системе с ответственным приложением.

Модуль 210 опроса может детектировать устаревшие процессы, образуемые из-за измененных файлов, настроек или устаревших модулей, включая общие ошибки конфигурации, старые версии программного обеспечения и т.п. Устаревшие процессы возникают, когда, например, при обновлении программного обеспечения не удается

перезапустить затронутые процессы после замены находящихся на диске исполняемых файлов, программных файлов или настроек. В результате вычислительное устройство, на котором был обнаружен устаревший процесс, проигнорирует обновление и продолжит выполнение на основе старых исполняемых файлов, программных файлов или настроек.

Для детектирования устаревших процессов модуль 210 опроса может запросить информацию, связанную с взаимодействиями программ, которые хранятся в хранилище 218 регистрационных журналов и/или архиве 106 сбора. Модуль 210 опроса опрашивает хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет времени последней загрузки программ, инсталлированных на одном или более вычислительных устройствах 102. Модуль 210 опроса также может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет времени последней загрузки файлов или настроек системного реестра, связанных с инсталлированным программным обеспечением. В одном примере осуществления модуль 210 опроса опрашивает хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет времени последней загрузки файлов или настроек системного реестра, относящихся к динамически подключаемым библиотекам (Dynamic Link Library, DLL), связанным с программным обеспечением и инсталлированным вместе с ним. Модуль 210 опроса также может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет времени или даты последней модификации программного обеспечения, инсталлированного на вычислительном устройстве. Подобные модификации включают в себя, например, доступы, выполненные к одному или более файлам или программным настройкам, связанным с последней известной версией инсталлированного программного обеспечения.

В случае если время последней загрузки программного обеспечения оказывается позже времени или даты последней известной модификации программного обеспечения, могут возникнуть несовместимости, возникающие в результате того, что программное обеспечение не использует последнее загруженное обновление. Если модулем 210 опроса детектируются подобные несовместимости, то они могут быть отмечены и исправлены некоторым лицом, таким как системный администратор.

Детектированные устаревшие файлы могут быть отображены через графический интерфейс, позволяющий лицам, например системному администратору, просматривать устаревшие файлы и предпринимать необходимые действия. Отображенные устаревшие файлы и связанная информация могут быть сохранены во внешнем фонде хранения, например внешней базе данных, как справка для будущего применения.

Модуль 210 опроса может детектировать присутствие известных неуместных программ, включая программные приложения, такие как "вредоносные программы", "шпионские программы", "троянские кони", "вирусы" и т.п. Для этой цели модуль 210 опроса может опросить и выполнить поиск в хранилище 218 регистрационных журналов и/или архиве 106 сбора на предмет программ, загруженных для выполнения в память одного или более вычислительных устройств 102. Далее, программы, загруженные для выполнения в память, могут быть сравнены посредством, например, модуля 210 опроса со списком известных неуместных программ.

Например, модуль 210 опроса может детектировать присутствие программ, загруженных для выполнения в память вычислительного устройства 102 на основании идентификаторов, таких как идентификаторы программы, связанные с программами.

Далее, модуль 210 опроса может сравнить идентификаторы программ, загруженных для выполнения в память вычислительного устройства 102, со списком идентификаторов, таких как идентификаторы известных неуместных программ. Если идентификатор программы, загруженной для выполнения в память, совпадает с идентификатором известной неуместной программы, то модуль 210 опроса может выполнить удаление этой программы из вычислительного устройства 102. В одном возможном варианте осуществления список идентификаторов известных неуместных программ может быть составлен, по меньшей мере, частично, системным администратором.

Неуместная программа, детектированная модулем 210 опроса, может быть отображена посредством графического интерфейса, что позволяет лицам, таким как системный администратор, просматривать эту неуместную программу и предпринимать необходимые действия для ее удаления. Отображенная неуместная программа и связанная информация могут быть сохранены во внешнем фонде хранения, например внешней базе данных, для использования в будущем в качестве справки для детектирования таких или схожих неуместных программ.

Неидентифицированная программа на одном или более вычислительных устройствах 102, которая не имеет идентификатора, может быть детектирована модулем 210 опроса и сообщена системному администратору для проверки и подтверждения того, является ли данная неидентифицированная программа неуместной программой. Системный администратор может исследовать природу неидентифицированной программы путем изучения списка неидентифицированных программ в форме отчета. Упомянутое изучение, выполняемое системным администратором, может включать в себя исследование цели неидентифицированной программы, зависимость неидентифицированной программы от других программ и определение того, является ли неидентифицированная программа неуместной. Дополнительно, системный администратор может исследовать прошлый опыт, связанный с программами, у которых характеристики схожи с характеристиками неидентифицированной программы, чтобы определить, является ли неидентифицированная программа неуместной.

Если системный администратор определяет, что неидентифицированная программа является неуместной, то системный администратор может реализовать удаление неидентифицированной программы из вычислительных устройств 102. Например, системный администратор может удалить саму неидентифицированную программу либо дать команду элементам вычислительного устройства 102 для удаления неидентифицированной программы.

Дополнительно, на основании сгенерированного отчета или одного или более уведомлений 222 системный администратор может назначить неидентифицированной программе идентификатор, такой как идентификатор программы, и включить этот идентификатор в список неуместных программ. Таким образом, если неидентифицированная программа повторно появляется на вычислительном устройстве 102, то она может быть быстро идентифицирована как неуместная программа на основании соответствующего идентификатора. Более того, удаление неидентифицированной программы может быть реализовано элементами вычислительного устройства 102, агентом 110 и т.п.

Неидентифицированные программы и их связанные процессы, которые были детектированы модулем 210 опроса, могут быть отображены через графический интерфейс, что позволяет лицам, таким как системный администратор, просматривать

эти неидентифицированные программы и предпринимать необходимые действия для их удаления. Отображенная неидентифицированная программа и связанная информация могут быть сохранены во внешнем фонде хранения, например внешней базе данных, для использования в будущем в качестве справки для детектирования таких или схожих неуместных программ. Сверх того, нежелательные изменения также могут быть идентифицированы и/или отслежены.

Модуль 210 опроса может блокировать копирование файлов на сетевые или съемные диски путем запрета записи в эти локации, причем запрет реализуется программой, которая выполняется на одном или более вычислительных устройствах 102. Модуль 210 опроса также может исследовать предыдущие подобные запреты записи, выполненные в целях аудита для предотвращения подобных записей в будущем.

Примерные способы

Примеры способов для перехвата и анализа потока описаны со ссылкой на Фиг.1-4. Эти примеры способов могут быть описаны в общем контексте выполняемых компьютером инструкций. В общем, выполняемые компьютером инструкции могут включать в себя рутинные процедуры, программы, объекты, компоненты, структуры данных, процедуры, модули, функции и т.п., которые выполняют конкретные функции или осуществляют конкретные абстрактные типы данных. Настоящие способы могут быть применены в распределенных вычислительных окружениях, где задачи выполняются посредством удаленных устройств обработки, которые объединены через сеть связи. В распределенном вычислительном окружении выполняемые компьютером инструкции могут быть расположены как в носителе хранения локального компьютера, так и в носителе хранения удаленного компьютера, включая запоминающие устройства.

Фиг.5 представляет собой иллюстрацию примера способа 500 для захвата и сбора информации, которая связана с взаимодействиями между программами, выполняемыми на одном или более вычислительных устройствах 102, и/или файловыми системами и настройками. Порядок, в котором описан данный способ, не должен истолковываться как ограничение, и для реализации данного способа или альтернативного способа может использоваться любое количество описанных блоков, которые могут быть комбинированы в любом порядке. Дополнительно, в рамках сущности и объема объекта настоящего изобретения из способа могут быть исключены отдельные блоки. Сверх того, данный способ может быть реализован в любом подходящем аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении или в их комбинациях.

В блоке 502 информация или данные, связанные с выполняемой или исполняемой в системе программой, перехватываются. В одном варианте осуществления упомянутая информация собирается, когда программа/процесс вызывает оснащенную функцию, включающую в себя модифицированный код функции. Например, машиночитаемые инструкции могут быть модифицированы таким образом, чтобы подавать команду одной или более функции для захвата данных, связанных с взаимодействиями между программой, выполняемой на одном или более вычислительных устройствах 102, и/или файловыми системами и настройками. В одном варианте реализации Виртуальная Машина применяет логику сбора данных, когда она перехватывает выполнение исходного кода. Согласно этому способу не требуется модифицировать исходный код. Аналогично это может быть реализовано непосредственно в аппаратной части процессора.

Агент, такой как агент 110, может оснащать одну или более функций в системе 100. Одна или более функций могут быть оснащены путем модификации машиночитаемых инструкций, связанных с упомянутой одной или более функциями.

Агент 110, такой как регистратор данных потока, может перехватывать потоки, вызывающие модифицированные функции. Программы, с которыми связаны эти потоки, могут выполняться на одном из нескольких операционных уровней, таких как программный уровень, уровень межплатформного программного обеспечения, уровень операционной системы и т.п. Файловая система, с которой программа может пытаться связаться, может включать в себя файлы (такие, как файлы данных, исполняемые файлы), информацию настроек (такую, как конфигурационные настройки или настройки системного реестра) и т.п.

В блоке 504 различная информация или данные, связанные с выполнением программ, исполняемых на одном или более вычислительных устройствах 102, собираются или копируются в ячейку памяти. Информация, связанная с взаимодействиями программ с файловой системой и/или настройками, включая взаимодействия, которые инициируются модифицированными функциями, копируется и передается в ячейку памяти. Например, агент 110 может скопировать все или выбранные данные, связанные с этими взаимодействиями, и сохранить эти данные в ячейке памяти, такой как сервер 104 сбора. Данные, связанные с взаимодействиями, могут включать в себя информацию, относящуюся к взаимодействиям, которые инициируются оснащенной функцией.

В блоке 506 данные, сохраненные в ячейке памяти, сжимаются. В одном варианте реализации сжатые данные могут быть сохранены в другой ячейке памяти. Например, сжатые данные могут быть сохранены в хранилище 218 регистрационных журналов в сервере 104 сбора и/или архиве 106 сбора.

В блоке 508 сжатые данные периодически выгружаются для анализа. Сжатые данные могут быть выгружены в сервер 106 сбора и/или ячейке памяти, которая выполняет роль сервера 106 сбора. Периодичность выгрузки сжатых данных в целях анализа может меняться. В одной реализации сжатые данные выгружаются через заданные интервалы времени. В еще одной реализации сжатые данные могут быть выгружены, когда объем сжатых данных превышает предопределенное пороговое значение.

Фиг.6 представляет собой иллюстрацию примера способа 600 для классификации отмеченных изменений. Отмеченные изменения включают в себя изменения или модификации, которые могут возникнуть в результате неожиданного исполнения программы, операционной системы, программ, которые используются для выполнения специфичных задач, такие как бухгалтерские и иные программы. Порядок, в котором описан данный способ, не должен истолковываться как ограничение, и для реализации данного способа или альтернативного способа может использоваться любое количество описанных блоков, которые могут быть комбинированы в любом порядке. Дополнительно, в рамках сущности и объема объекта настоящего изобретения из способа могут быть исключены отдельные блоки. Сверх того, данный способ может быть реализован в любом подходящем аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении или в их комбинациях.

В блоке 602 указывается правило классификации, характеризуемое различными значениями параметров. Например, модулем 210 опроса может быть указано правило классификации вместе со значениями параметров, которые описывают правило

классификации.

В блоке 604 параметры, описывающие правило классификации, ассоциируются с одним или более атрибутами, которые описывают природу и характеристики отмеченного изменения. Модуль 210 опроса ассоциирует параметры, описывающие правило классификации, с атрибутами, характеризующими отмеченное изменение. В результате ассоциирования одного или более значений параметров с атрибутами, характеризующими рассматриваемое отмеченное изменение, предоставляется набор вероятных классификаций. Например, модуль 210 опроса ассоциирует каждое совпадение подстроки, содержащейся в правиле классификации, с именем УС, содержащимся в каждом отмеченном изменении.

В блоке 606 одной или более вероятным классификациям назначается значение приоритета. Модуль 210 опроса может назначить значение приоритета одной или более вероятным классификациям. Например, конкретному отмеченному изменению, происходящему в течение более долгого периода времени, назначается более высокое значение приоритета.

В блоке 608 вероятная классификация с наивысшим значением приоритета назначается рассматриваемому отмеченному изменению. В одной реализации модуль 210 опроса определяет наивысшее значение приоритета, назначенное вероятным классификациям, и назначает эту классификацию рассматриваемому отмеченному изменению.

Фиг.7 представляет собой иллюстрацию примера способа 700 для запрета выполнения неавторизованных взаимодействий (согласно определению системного оператора), например, на одном или более вычислительных устройствах 102. Примеры неавторизованных взаимодействий включают в себя действия чтения и/или записи, выполняемые в файловой системе объектом или программой, которая не авторизована на выполнение подобного действия.

Порядок, в котором описан данный способ, не должен истолковываться как ограничение, и для реализации данного способа или альтернативного способа может использоваться любое количество описанных блоков, которые могут быть комбинированы в любом порядке. Дополнительно, в рамках сущности и объема объекта настоящего изобретения из способа могут быть исключены отдельные блоки. Сверх того, данный способ может быть реализован в любом подходящем аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении или в их комбинациях.

В блоке 702 принимается информация, связанная с программой, которая выполняется в системе. Эта информация ассоциируется с взаимодействиями между программой и файловой системой и/или конфигурационными настройками. Модуль 210 опроса может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет информации, относящейся к взаимодействиям, которые выполняются программой на одном или более вычислительных устройствах 102. Информация, полученная в результате выполнения запроса, характеризуется одним или более атрибутами.

В блоке 704 атрибуты программы, выполняемой в системе, сравниваются с атрибутами множества согласованных и несогласованных программ/процессов, входящих в предопределенный список. Например, модуль 210 опроса сравнивает атрибуты программы, например тип программы, с атрибутами программ, входящими в предопределенный список.

В блоке 706 определяется, соответствуют ли упомянутые атрибуты атрибутам

несогласованных программ/процессов или взаимодействий. Например, если атрибут программы, выполняемой в системе 100, соответствует атрибуту, связанному с несогласованными взаимодействиями (то есть ветвь "да" из блока 706), то дальнейшее выполнение взаимодействий, связанных с этой программой, блокируется (то есть блок 708). Альтернативно, если атрибут программы, выполняемой в системе 100, не соответствует атрибуту, связанному с несогласованными взаимодействиями (то есть ветвь "нет" из блока 706), то дальнейшее выполнение взаимодействий, связанных с этой программой, разрешается (то есть блок 710).

Фиг.8 представляет собой иллюстрацию примера способа 800 для детектирования одной или более точек расширяемости программы, инсталлированной на одном или более вычислительных устройствах 102. Точки расширяемости включают в себя взаимодействия, которые управляют динамической загрузкой и выполнением компьютерного приложения. Порядок, в котором описан данный способ, не должен истолковываться как ограничение, и для реализации данного способа или альтернативного способа может использоваться любое количество описанных блоков, которые могут быть комбинированы в любом порядке. Дополнительно, в рамках сущности и объема объекта настоящего изобретения из способа могут быть исключены отдельные блоки. Сверх того, данный способ может быть реализован в любом подходящем аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении или в их комбинациях.

В блоке 802 проверяются предыдущие взаимодействия (то есть взаимодействия, относящиеся к различным программам, загруженным для выполнения в системную память до выполнения первой программы). Например, модуль 210 опроса может идентифицировать потенциальные непосредственные точки расширяемости для первой программы путем опроса хранилища 218 регистрационных журналов и/или архива 106 сбора на предмет взаимодействий, относящихся к различным программам, загруженным для выполнения в память одного или более вычислительных устройств 102 до выполнения первой программы. Модуль 210 опроса может выполнить запрос взаимодействий, относящихся к различным программам, загруженным для выполнения в системную память в заданном временном интервале, таком как две секунды до выполнения первой программы.

В блоке 804 выполняется проверка, чтобы обнаружить предыдущие взаимодействия, которые ссылаются на имя файла первой программы, загруженной для выполнения в системную память вычислительного устройства. Например, модуль 210 опроса может запросить взаимодействия, связанные с различными другими программами, которые ссылаются на первую программу или которые связаны с выполнением первой программы на вычислительных устройствах 102. Модуль 210 опроса может запросить взаимодействия, включающие в себя различные атрибуты, такие как имя файла первой программы, идентификатор первой программы и т.п.

В блоке 806 предыдущие взаимодействия, ссылающиеся на имя файла первой программы, помечаются как непосредственные точки расширяемости. Например, модуль 210 опроса может идентифицировать точки расширяемости для первой программы путем запроса всех предыдущих взаимодействий, которые ссылаются на первую программу или которые связаны с выполнением первой программы.

Фиг.9 представляет собой иллюстрацию примера способа 900 для детектирования просочившихся элементов, которые могли быть оставлены в результате деинсталляции программы с одного или более вычислительных устройств 102.

Порядок, в котором описан данный способ, не должен истолковываться как ограничение, и для реализации данного способа или альтернативного способа может использоваться любое количество описанных блоков, которые могут быть комбинированы в любом порядке. Дополнительно, в рамках сущности и объема объекта настоящего изобретения из способа могут быть исключены отдельные блоки. Сверх того, данный способ может быть реализован в любом подходящем аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении или в их комбинациях.

В блоке 902 изменения инсталляционных файлов и настроек, связанные с каждой программой, которая загружается в вычислительное устройство и/или систему, каталогизируются и перечисляются. Перечисление включает в себя создание списка программ, зарегистрированных в операционной системе вычислительного устройства.

Например, система 100 может быть отсканирована, чтобы детектировать все программы, которые инсталлированы на вычислительных устройствах в системе, а также все инсталляционные файлы операционной системы, связанные с программами на вычислительных устройствах. Все программы, инсталлированные на вычислительных устройствах, и/или все инсталляционные файлы операционной системы, связанные с программами, могут быть перечислены путем включения их в список.

Модуль 210 опроса выполняет в системе 100 сканирование, посредством которого детектируются все программы, которые инсталлированы на одном или более вычислительных устройствах 102, чтобы каталогизировать и перечислить все инсталляционные файлы операционной системы на вычислительных устройствах 102. Например, модуль 210 опроса может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет всех программ, зарегистрированных в операционной системе одного или более вычислительных устройств 102. Обнаруженные программы могут быть каталогизированы и перечислены посредством различных устройств, таких как модуль 210 опроса, агент 110 и т.п. Более того, все инсталляционные файлы операционной системы на вычислительных устройствах 102, связанные с обнаруженными программами, могут быть каталогизированы и перечислены различными устройствами, такими как модуль 210 опроса, агент 110 и т.п.

В блоке 904 перечисляются присутствующие на вычислительном устройстве и/или системе устойчивые состояния, включающие в себя файлы и настройки системного реестра, связанные с деинсталлированными программами. Это может включать в себя сканирование памяти вычислительного устройства и/или системы на предмет файлов и настроек системного реестра всех программ, которые были инсталлированы на вычислительном устройстве и/или системе, включая файлы и настройки системного реестра для программ, которые были деинсталлированы. Например, модуль 210 опроса может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора, чтобы получить все файлы и настройки системного реестра, соответствующие идентификаторам, таким как идентификаторы программ, для всех программ, которые были инсталлированы на вычислительных устройствах 102.

В блоке 906 файлы и настройки системного реестра, связанные с зарегистрированными в операционной системе программами, сравниваются с файлами и настройками системного реестра программ, которые были инсталлированы на вычислительных устройствах 102 и/или системе 100. Например, модуль 210 опроса может сравнить идентификаторы, такие как идентификаторы программ перечисленных файлов и настроек системного реестра, связанных с

зарегистрированными в операционной системе программами, с идентификаторами файлов или настроек всех программ, которые были инсталлированы на вычислительном устройстве 102.

В блоке 908 файлы и настройки системного реестра, связанные с программами, присутствующие в обоих списках, могут быть исключены из рассмотрения. Остальные файлы и настройки системного реестра, которые представляют файлы и настройки системного реестра, соответствующие программам, которые были деинсталлированы с вычислительного устройства 102 и/или системы, могут быть помечены как просочившиеся файлы, и они могут быть удалены с вычислительного устройства 102 и/или системы 100. Например, модуль 210 опроса может коррелировать идентификаторы, такие как идентификаторы программ, перечисленных файлов и настроек системного реестра, связанных с зарегистрированными в операционной системе программами, с идентификаторами файлов и настроек системного реестра, связанных с программами, которые были инсталлированы на вычислительном устройстве 102. Файлы и настройки системного реестра, связанные с некоррелированными программами, могут быть обозначены модулем 210 опроса как просочившиеся файлы, и они могут быть удалены с вычислительных устройств 102 посредством таких элементов, как модуль 210 опроса, агент 110 и т.п.

Фиг.10 представляет собой иллюстрацию примера способа 1000 для детектирования устаревших файлов, включая обычные неправильные конфигурации, старые версии программного обеспечения и т.п., которые инсталлированы на одном или более вычислительных устройствах 102. Устаревшие файлы возникают, когда, например, при обновлении программного обеспечения не удается перезапустить затронутые процессы после замены находящихся на диске исполняемых файлов. В результате вычислительные устройства 102, на которых был обнаружен устаревший файл, будут игнорировать обновление и продолжат выполнять программу со старого файла. Порядок, в котором описан данный способ, не должен истолковываться как ограничение, и для реализации данного способа или альтернативного способа может использоваться любое количество описанных блоков, которые могут быть комбинированы в любом порядке. Дополнительно, в рамках сущности и объема объекта настоящего изобретения из способа могут быть исключены отдельные блоки. Сверх того, данный способ может быть реализован в любом подходящем аппаратном обеспечении, программном обеспечении, встроенном программном обеспечении или в их комбинациях.

На этапе 1002 программы, загруженные в вычислительное устройство и/или систему, каталогизируются и перечисляются. В одной реализации перечисление включает в себя создание списка программ, зарегистрированных в операционной системе вычислительного устройства. Например, система может быть отсканирована, чтобы детектировать все программы, которые инсталлированы на вычислительных устройствах в системе. Все программы, инсталлированные на вычислительных устройствах, могут быть перечислены путем включения их в список.

В одной возможной реализации модуль 210 опроса выполняет в системе 100 сканирование, посредством которого детектируются все программы, которые инсталлированы на одном или более вычислительных устройствах 102, чтобы каталогизировать и перечислить все программы, зарегистрированные в операционной системе вычислительных устройств 102. Например, модуль 210 опроса может опросить хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет всех программ, зарегистрированных в операционной системе одного или более

вычислительных устройств 102 в системе 100. Обнаруженные программы могут быть каталогизированы и перечислены посредством различных устройств и/или объектов, таких как модуль 210 опроса, агент 110 и т.п.

В блоке 1004 получают временные моменты последней загрузки всех программ, зарегистрированных в вычислительном устройстве и/или системе, а также временные моменты последней загрузки для файлов, связанных с программами, зарегистрированными в вычислительном устройстве и/или системе. Например, модуль 210 опроса опрашивает хранилище 218 регистрационных журналов и/или архив 106 сбора на предмет временных моментов последней загрузки программ, зарегистрированных в вычислительных устройствах 102, и/или временные моменты последней загрузки файлов, таких как файлы DLL, инсталлированных вместе с программами, зарегистрированными в вычислительных устройствах 102 в системе 100.

В блоке 1006 получают временные моменты последней модификации файлов и настроек, связанных с программами, которые зарегистрированы в вычислительном устройстве и/или системе, и сравнивают эти временные моменты с упомянутыми временными моментами последней загрузки программ. Например, модуль 210 опроса может опросить хранилище 218 и/или архив 106 сбора на предмет времени или даты последней модификации программы, зарегистрированной в операционной системе вычислительных устройств 102. Модуль 210 опроса может сравнить время или дату последней модификации с временем последней загрузки данной программы.

В блоке 1008 отмечаются все несоответствия, обнаруженные во время упомянутого сравнения. Например, в случае если временной момент последней загрузки программы, зарегистрированной в вычислительном устройстве и/или системе, оказывается позже временного момента или даты последней известной модификации программы, то есть вероятность того, что программа не ответила на последнюю модификацию. В подобном случае может быть сгенерирован отчет об ошибке, который направляется объектам, таким как пользователь или системный администратор, причем в отчете отражается, что эта программа не отреагировала на последнюю модификацию. Альтернативно, может быть предпринята повторная попытка выполнения последней неудавшейся модификации программы.

В одной иллюстративной реализации модуль 210 опроса может запросить как временной момент последней загрузки, так и временной момент последней модификации программы, зарегистрированной в операционной системе вычислительных устройств 102 в системе 100. Модуль 210 опроса может сравнить временной момент последней загрузки и временной момент последней модификации, и если временной момент последней загрузки программы позже временного момента последней модификации программы, то модуль 210 опроса может издать отчет об ошибках и направить его объектам, таким как пользователь или системный администратор, причем в отчете отражается, что данная программа не отвечает на последнюю неуспешную модификацию. В еще одной реализации модуль 210 опроса также может выполнить попытку повторения последней неуспешной модификации упомянутой программы.

Примерная вычислительная среда

Фиг.11 представляет собой иллюстрацию примера вычислительной среды 1100, которая может быть использована для реализации описанных здесь способов и которая может представлять, в целом или частично, описанные здесь элементы. Компьютерная среда 1100 является лишь одним примером подходящей вычислительной среды, и она не предназначена для определения каких-либо

ограничений объема использования или функциональных возможностей компьютерных и сетевых архитектур. Кроме того, компьютерная среда 1100 не должна интерпретироваться как имеющая какую-либо зависимость или требование, относящиеся к какому-либо компоненту или комбинациям компонентов, проиллюстрированных в примере компьютерной среды 1100.

Компьютерная среда 1100 включает в себя вычислительное устройство общего назначения в форме компьютера 1102. Компьютер 1102 может представлять собой, например, настольный компьютер, карманный компьютер, ноутбук или портативный компьютер, серверный компьютер, игровую консоль и т.п. Компоненты компьютера 1102 могут включать в себя, но не ограничиваются перечисленным, один или более процессоров или процессорных блоков 1104, системную память 1106 и системную шину 1108, которая соединяет различные компоненты системы, включая соединение процессора 1104 с системной памятью 1106.

Системная шина 1108 представляет один или более типов структур шины, включающих в себя шину памяти или контроллер памяти, периферийную шину, порт ускоренной графики и процессорную или локальную шину, использующих любую архитектуру из ряда архитектур шин. В качестве примера, подобные архитектуры могут включать в себя шину стандарта Industry Standard Architecture (ISA), шину стандарта Micro Channel Architecture (MCA), шину стандарта Enhanced ISA (EISA), локальную шину стандарта Video Electronics Standards Association (VESA) и шину стандарта Peripheral Component Interconnect (PCI), также известную как шина расширения.

Компьютер 1102, как правило, включает в себя ряд машиночитаемых носителей. Такие машиночитаемые носители могут быть любым доступным носителем, к которому компьютер 1102 может выполнить доступ, и они включают в себя как энергозависимые, так и энергонезависимые носители, съемные и несъемные носители.

Системная память 1106 включает в себя машиночитаемый носитель в форме энергозависимой памяти, такой как Оперативное Запоминающее Устройство (ОЗУ) 1110, и/или энергонезависимой памяти, такой как Постоянное Запоминающее Устройство (ПЗУ) 1112. Базовая система 1114 ввода/вывода (BIOS), содержащая базовые рутинные процедуры, которые помогают передавать информацию между элементами в компьютере 1102, как, например, во время загрузки, как правило, хранится в ПЗУ 1112. ОЗУ 1110, как правило, содержит данные и/или программные модули, которые непосредственно доступны и/или задействованы процессорным блоком 1104.

Компьютер 1102 может также включать в себя другой съемный/несъемный энергозависимый/энергонезависимый компьютерный носитель хранения. В качестве примера на Фиг.11 проиллюстрирован привод 1116 жесткого диска для считывания и записи на несъемный энергонезависимый магнитный носитель (не показан), привод 1118 магнитного диска для считывания и записи на съемный энергонезависимый магнитный диск 1120 (например, "дискету") и привод 1122 оптического диска для считывания и/или записи на съемный энергонезависимый оптический диск 1124, такой как CD-ROM, DVD-ROM или другой оптический носитель. Привод 1116 жесткого диска, привод 1118 магнитного диска и привод 1122 оптического диска соединены с системной шиной 1108 посредством одного или более интерфейсов 1126 носителей данных. Альтернативно, привод 1116 жесткого диска, привод 1118 магнитного диска и привод 1122 оптического диска могут быть соединены с системной шиной 1108 посредством одного или более интерфейсов (не

показаны).

Приводы дисков и их связанные машиночитаемые носители предоставляют энергонезависимое хранилище машиночитаемых инструкций, структур данных, программных модулей, а также других данных для компьютера 1102. Несмотря на то что данный пример иллюстрирует жесткий диск 1116, съемный магнитный диск 1120 и съемный оптический диск 1124, очевидно, что для реализации примера вычислительной системы и среды могут также использоваться другие типы машиночитаемых носителей, которые могут хранить данные, которые доступны компьютеру, такие как магнитные кассеты и другие магнитные запоминающие устройства, карты флэш-памяти, CD-ROM, цифровые универсальные диски (DVD) или другие оптические носители, Оперативные Запоминающие Устройства (ОЗУ), Постоянные Запоминающие Устройства (ПЗУ), Электрически Стираемое Программируемое Постоянное Запоминающее Устройство (ЭСППЗУ) и т.п.

На жестком диске 1116, магнитном диске 1120, оптическом диске 1124, ПЗУ 1112 и/или ОЗУ 1110 может храниться любое количество программных модулей, которые, например, включают в себя операционную систему 1127, одну или более прикладных программ 1128, другие программные модули 1130 и программные данные 1132. Операционная система 1127, одна или более прикладных программ 1128, другие программные модули 1130 или программные данные 1132 (или некоторые их комбинации) могут реализовывать все или часть резидентных компонентов, которые поддерживают распределенную файловую систему.

Пользователь может вводить команды и информацию в компьютер 1102 через устройства ввода, такие как клавиатура 1134 и указательное устройство 1136 (например, "мышь"). Другие устройства ввода 1138 (не показаны конкретно) могут включать в себя микрофон, джойстик, игровой планшет, спутниковую антенну, последовательный порт, сканер и/или т.п. Эти и другие устройства ввода соединены с процессорным блоком 1504 через интерфейсы 1140 ввода/вывода, которые соединены с системной шиной 1108, но они также могут быть соединены посредством другого интерфейса и структур шины, такой как параллельный порт, игровой порт или универсальная последовательная шина (USB).

Монитор 1142 или другой тип устройства отображения также может быть соединен с системной шиной 1108 посредством интерфейса, такого как видеоадаптер 1144. Дополнительно к монитору 1142 другие периферийные устройства вывода могут включать в себя такие компоненты как громкоговорители (не показаны) и принтер 1146, которые могут быть соединены с компьютером 1102 через интерфейсы 1140 ввода/вывода.

Компьютер 1102 может работать в сетевой среде, используя логические соединения с одним или более удаленными компьютерами, такими как удаленное вычислительное устройство 1148. Удаленное вычислительное устройство 1148, например, может представлять собой персональный компьютер, портативный компьютер, сервер, маршрутизатор, сетевой компьютер, одноранговое устройство или иной обычный сетевой узел и т.п. Удаленное вычислительное устройство 1148 проиллюстрировано как портативный компьютер, который может включать в себя многие или все элементы и функциональные особенности, описанные относительно компьютера 1102.

Логические соединения между компьютером 1102 и удаленным компьютером 1148 изображены как локальная сеть 1150 и глобальная сеть 1152. Подобные сетевые среды типичны для офисов, компьютерных сетей масштаба предприятия, интранета и Интернета.

При использовании в сетевой среде локальной сети компьютер 1102 соединяется с локальной сетью 1150 через сетевой интерфейс или адаптер 1154. При использовании в сетевой среде глобальной сети компьютер 1102, как правило, включает в себя модем 1156 или иное средство для установления соединения через глобальную сеть 1152. Модем 1156, который может быть внутренним или внешним относительно компьютера 1102, может быть соединен с системной шиной 1108 через интерфейсы 1140 ввода/вывода или другие подходящие механизмы. Следует понимать, что проиллюстрированные сетевые соединения являются лишь примерами и для установления, по меньшей мере, одной линии связи между компьютерами 1102 и 1148 могут использоваться другие средства.

В сетевой среде, такой как вычислительная среда 1100, программные модули, изображенные относительно компьютера 1102, или их части могут храниться в удаленном запоминающем устройстве. В качестве примера, удаленные прикладные программы 1158 могут храниться в запоминающем устройстве удаленного компьютера 1148. Для целей иллюстрации прикладные программы и другие исполняемые программные компоненты, такие как операционная система, проиллюстрированы здесь как дискретные блоки, однако следует понимать, что такие программы или компоненты в различное время могут находиться в различных запоминающих компонентах вычислительного устройства 1102, и они исполняются процессором(ами) данных компьютера.

Различные модули и способы могут быть описаны здесь в общем контексте выполняемых компьютером инструкций, таких как программные модули, которые выполняются одним или более компьютерами или другими устройствами. В общем, программные модули включают в себя рутинные процедуры, программы, объекты, компоненты, структуры данных и т.п., которые выполняют конкретные задачи или осуществляют конкретные абстрактные типы данных. Как правило, функциональность программных модулей может быть комбинирована или распределена согласно требованиям в различных вариантах осуществления.

Реализация этих модулей и методик может быть сохранена или передана через машиночитаемый носитель некоторой формы. Машиночитаемый носитель может быть любым доступным носителем, к которому может быть выполнен доступ компьютером. В качестве примера, но не ограничиваясь перечисленным, машиночитаемые носители могут содержать "компьютерные носители хранения" и "средства связи".

"Компьютерный носитель хранения" включает в себя энергозависимый и энергонезависимый, съемный и несъемный носитель, реализованный посредством какого-либо способа или технологии для хранения информации, такой как машиночитаемые инструкции, структуры данных, программные модули или другие данные. Компьютерные носители хранения включают в себя, но не ограничиваются перечисленным, ОЗУ, ПЗУ, ЭСППЗУ, флэш-память или другой тип памяти, диски CD-ROM, цифровые универсальные диски (DVD) или иные оптические дисковые носители, магнитные кассеты, магнитные ленты, магнитные дисковые носители или другие магнитные устройства хранения или любой другой носитель, который может быть использован для хранения желаемой информации и к которому может быть выполнен доступ компьютером.

Альтернативно, части данной структуры могут быть реализованы в аппаратном обеспечении или в комбинации аппаратного обеспечения, программного обеспечения и/или встроенного программного обеспечения. Например, одна или более

специализированных интегральных схем или программируемых логических устройств могут быть спроектированы и запрограммированы таким образом, чтобы реализовывать одну или более частей данной структуры.

Заключение

Несмотря на то что варианты осуществления системного управления и анализа были описаны на языке, присущем структурным особенностям и/или способам, следует понимать, что объект формулы изобретения не ограничен конкретными особенностями или способами, описанными выше. Скорее, эти конкретные особенности и способы раскрыты в качестве примеров реализаций системного управления и анализа.

Формула изобретения

1. Способ управления устойчивым состоянием файловой системы, реализуемый на вычислительном устройстве процессором, конфигурированным для исполнения инструкций, которые, при исполнении процессором, побуждают вычислительное устройство выполнять действия, содержащие:

обнаружение устаревших файлов;

каталогизацию программ, загруженных на вычислительное устройство;

получение временных моментов последней загрузки файлов, связанных с программами, которые зарегистрированы в вычислительном устройстве;

получение временных моментов последней модификации файлов, связанных с упомянутыми программами, которые зарегистрированы в вычислительном устройстве; и

сравнение временных моментов последней модификации с временными моментами последней загрузки файлов, связанных с упомянутыми программами, которые зарегистрированы в вычислительном устройстве; и

уведомление о любых несоответствиях, найденных при сравнении, что включает в себя сообщение, что программа не отвечает на последнюю попытку модификации и попытку повторно выполнить последнюю попытку модификации программы.

2. Способ по п.1, в котором каталогизация содержит создание списка программ, зарегистрированных в операционной системе вычислительного устройства.

3. Способ по п.1, в котором каталогизация содержит перечисление программ путем включения упомянутых программ в список.

4. Способ по п.1, в котором каталогизация включает в себя сканирование, чтобы обнаружить все программы, которые зарегистрированы в вычислительном устройстве.

5. Способ по п.1, в котором упомянутое получение включает в себя опрашивание файла регистрационного журнала.

6. Способ по п.1, в котором упомянутое сравнение включает в себя опрашивание файла регистрационного журнала.

7. Вычислительное устройство, содержащее:

процессорный блок;

компонент хранения регистрационного журнала, конфигурированный для хранения:

временных моментов последней загрузки файлов, связанных с программами, зарегистрированными в вычислительном устройстве; и временных моментов последней модификации файлов, связанных с упомянутыми программами, зарегистрированными в вычислительном устройстве; и

компонент опрашивания регистрационного журнала, конфигурированный, чтобы:

каталогизировать программы, зарегистрированные в вычислительном устройстве; и

сравнивать временные моменты последней модификации с временными моментами последней загрузки, чтобы обнаружить устаревшие файлы в вычислительном устройстве, причем присутствие устаревших файлов приводит к игнорированию модернизации в программах, а игнорирование модернизации продолжит исполнение программы из старых файлов;

сообщать об устаревших файлах таким объектам, как пользователь или системный администратор, причем сообщение включает в себя программу, которая не отвечает на последнюю попытку модификации и попытку повторно выполнить последнюю попытку модификации программы.

8. Вычислительное устройство по п.7, дополнительно содержащее компоненты сбора архива, конфигурированные для хранения временных моментов последней загрузки и временных моментов последней модификации файлов, связанных с программами, зарегистрированными в вычислительном устройстве.

9. Вычислительное устройство по п.7, в котором временные моменты последней загрузки и временные моменты последней модификации включают в себя даты временных моментов последней загрузки и временных моментов последней модификации.

10. Вычислительное устройство по п.7, в котором компонент опрашивания регистрационного журнала дополнительно конфигурирован, чтобы сканировать все программы, зарегистрированные в вычислительном устройстве.

11. Считываемый компьютером носитель хранения, содержащий считываемые компьютером инструкции, которые, при исполнении компьютером, реализуют способ, содержащий:

каталогизацию и перечисление файлов и настроек, связанных с программами, зарегистрированными в вычислительном устройстве;

получение временных моментов и дат последней загрузки файлов и настроек, связанных с программами, которые зарегистрированы в вычислительном устройстве;

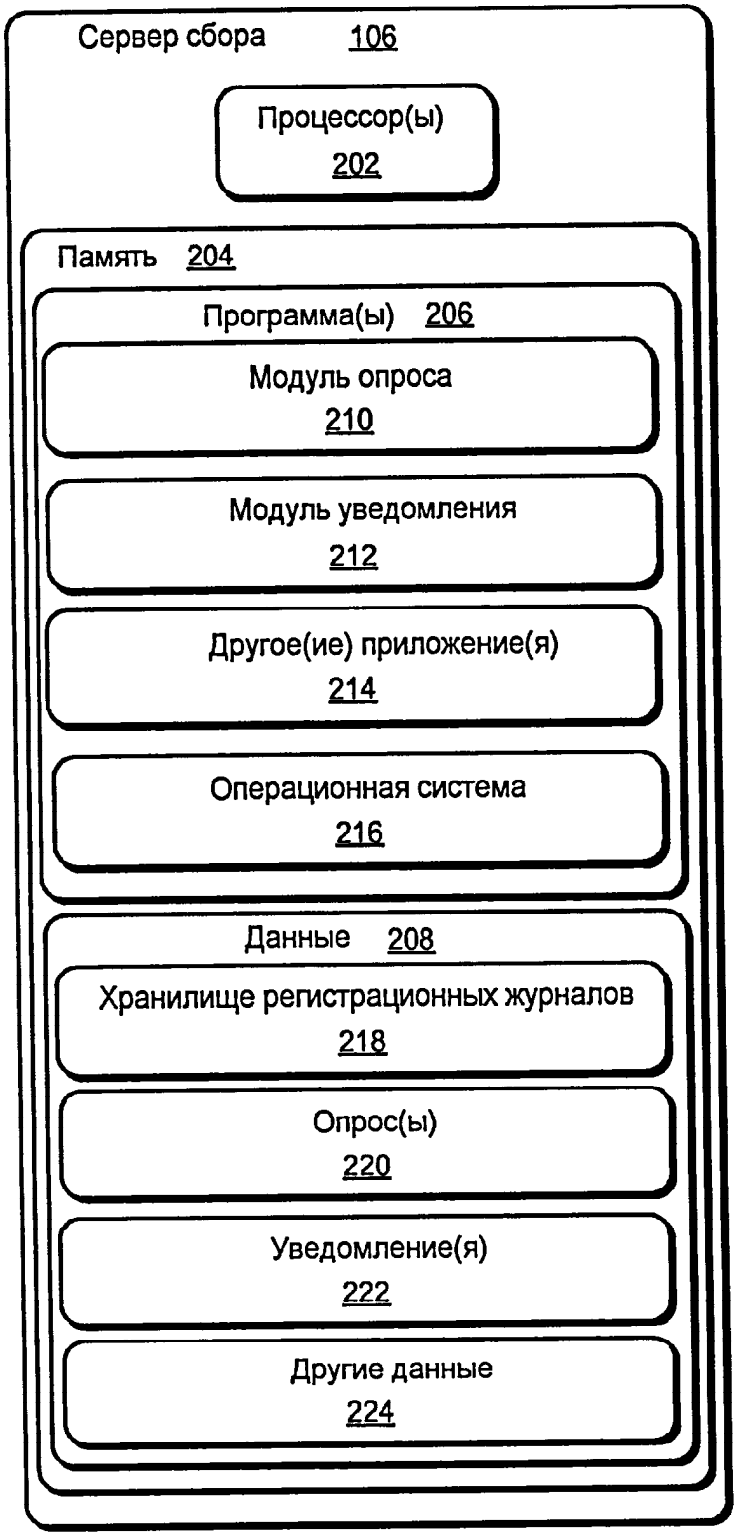
получение временных моментов и дат последней модификации файлов и настроек, связанных с упомянутыми программами, которые зарегистрированы в вычислительном устройстве; и

сравнение временных моментов и дат последней модификации с временными моментами и датами последней загрузки файлов и настроек, связанных с программами, которые зарегистрированы в вычислительном устройстве, причем сравнение дополнительно включает в себя уведомление о несоответствиях в программах, зарегистрированных в вычислительном устройстве, которое формирует сообщение, которое включает в себя программу, которая не отвечает на последнюю попытку модификации.

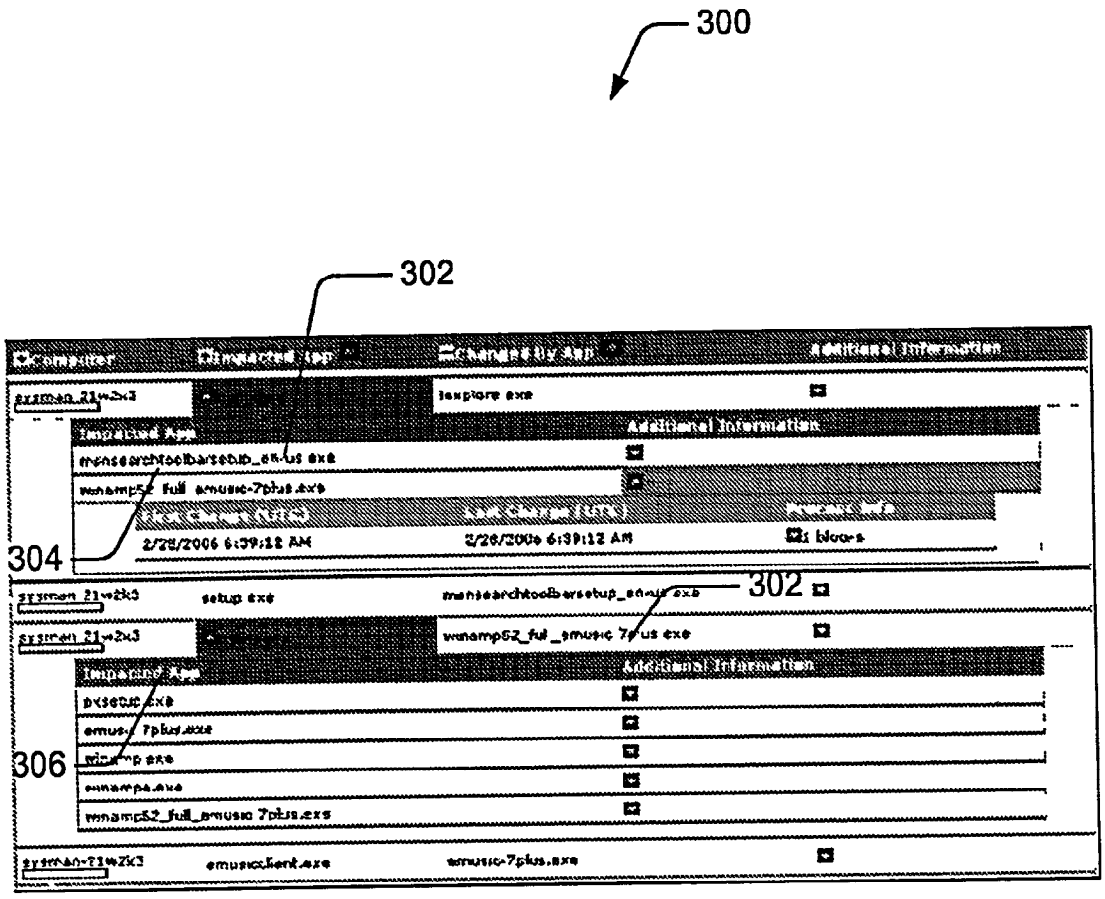
12. Считываемый компьютером носитель хранения по п.11, причем сравнение дополнительно включает в себя сравнение файлов и настроек, связанных с неинсталлированными программами, зарегистрированными в вычислительном устройстве, чтобы определить просочившиеся файлы.

13. Считываемый компьютером носитель хранения по п.12, причем просочившиеся файлы удаляются из вычислительного устройства.

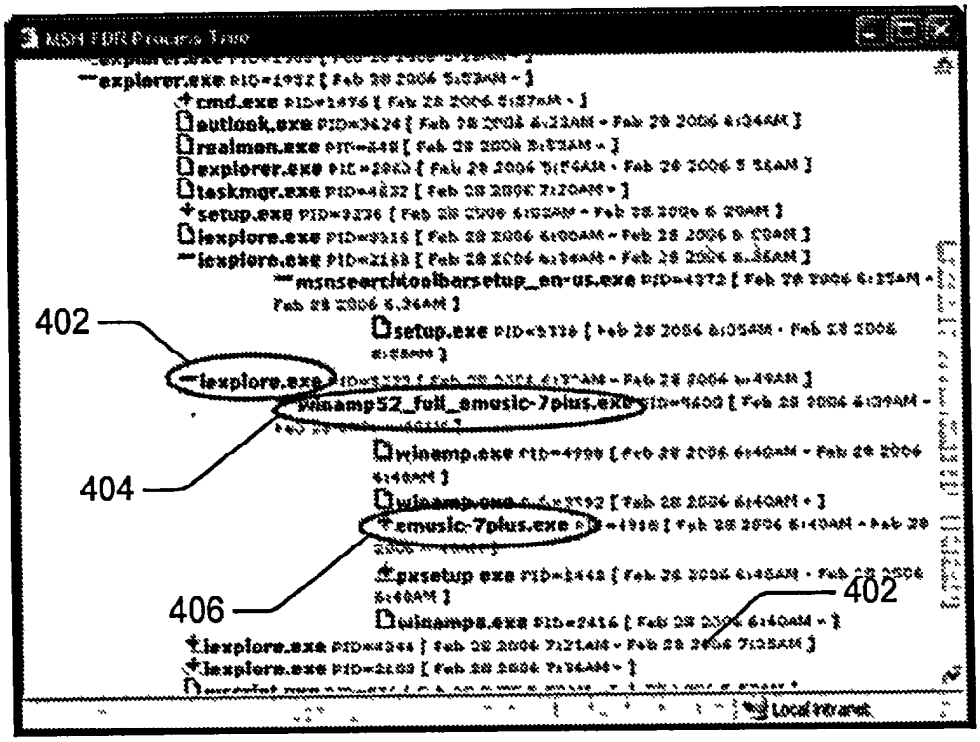
14. Считываемый компьютером носитель хранения по п.11, причем сравнение включает в себя опрашивание файла регистрационного журнала.


Сервер сбора 106
Процессор(ы) 202
Память 204
Программа(ы) 206
Модуль опроса 210
Модуль уведомления 212
Другое(ие) приложение(я) 214
Операционная система 216
Данные 208
Хранилище регистрационных журналов 218
Опрос(ы) 220
Уведомление(я) 222
Другие данные 224

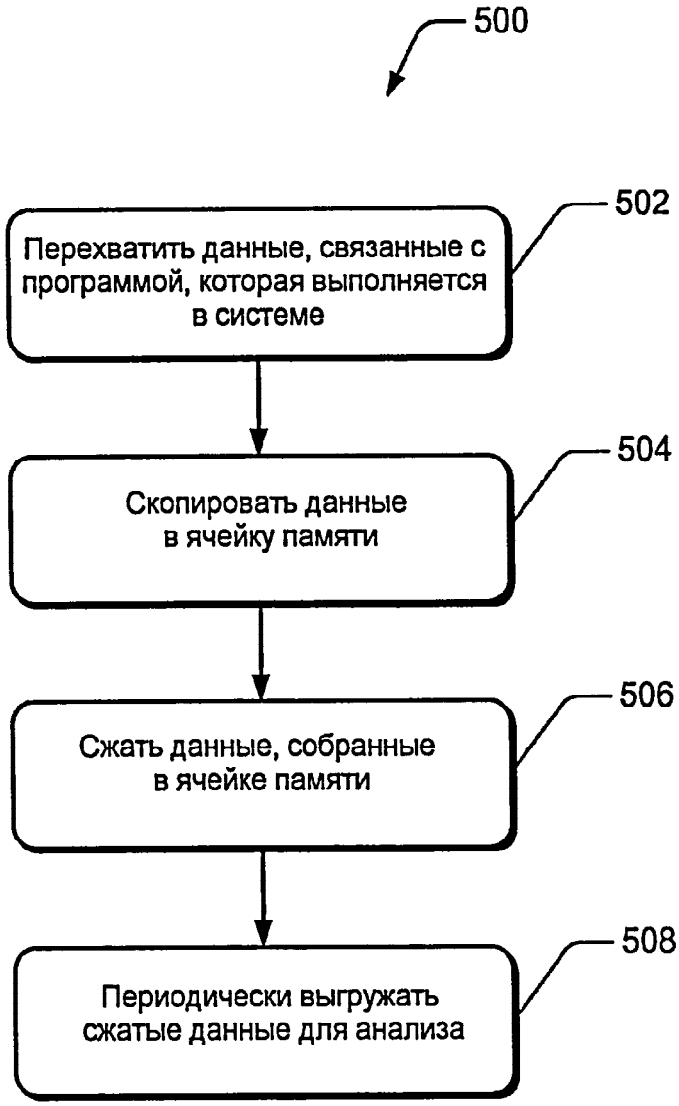
Фиг.2



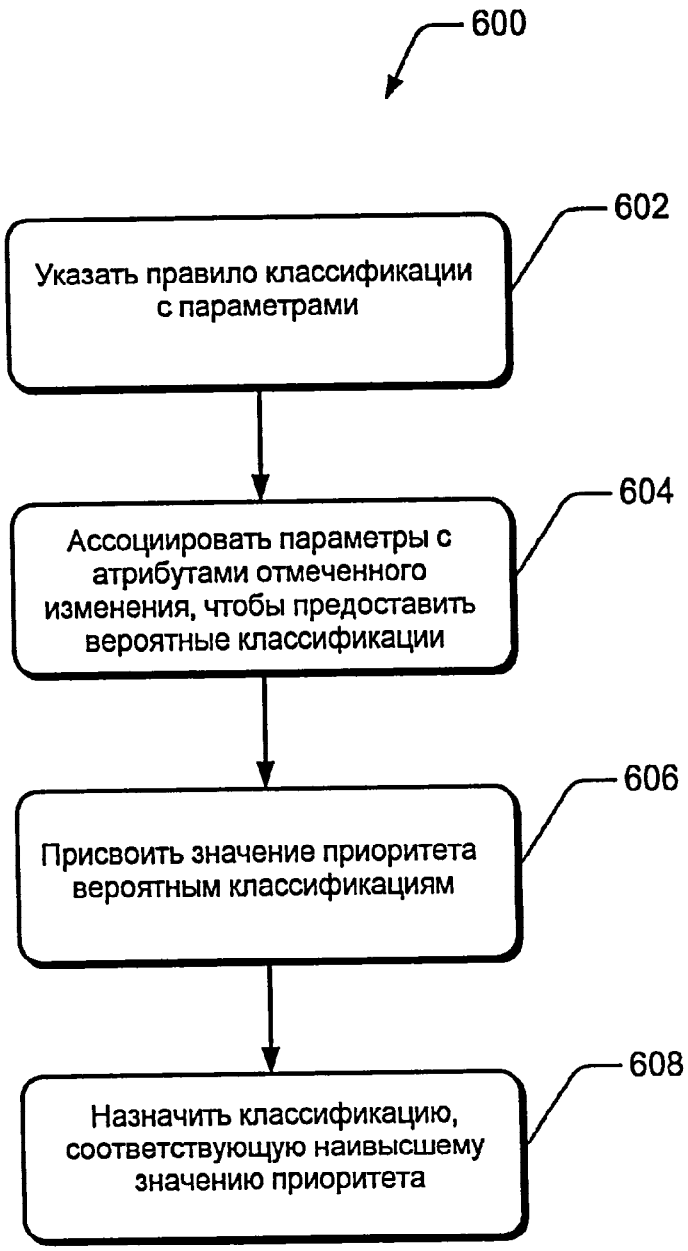
Фиг.3


400
402
404
406
402
explorer.exe
cmd.exe
outlook.exe
realmon.exe
explorer.exe
taskmgr.exe
setup.exe
iexplore.exe
iexplore.exe
msnsearchtoolbarsetup_en-us.exe
setup.exe
iexplore.exe
winamp52_full_emusic-7plus.exe
winamp.exe
emusic-7plus.exe
pxsetup.exe
winamps.exe
iexplore.exe
iexplore.exe
Local intranet

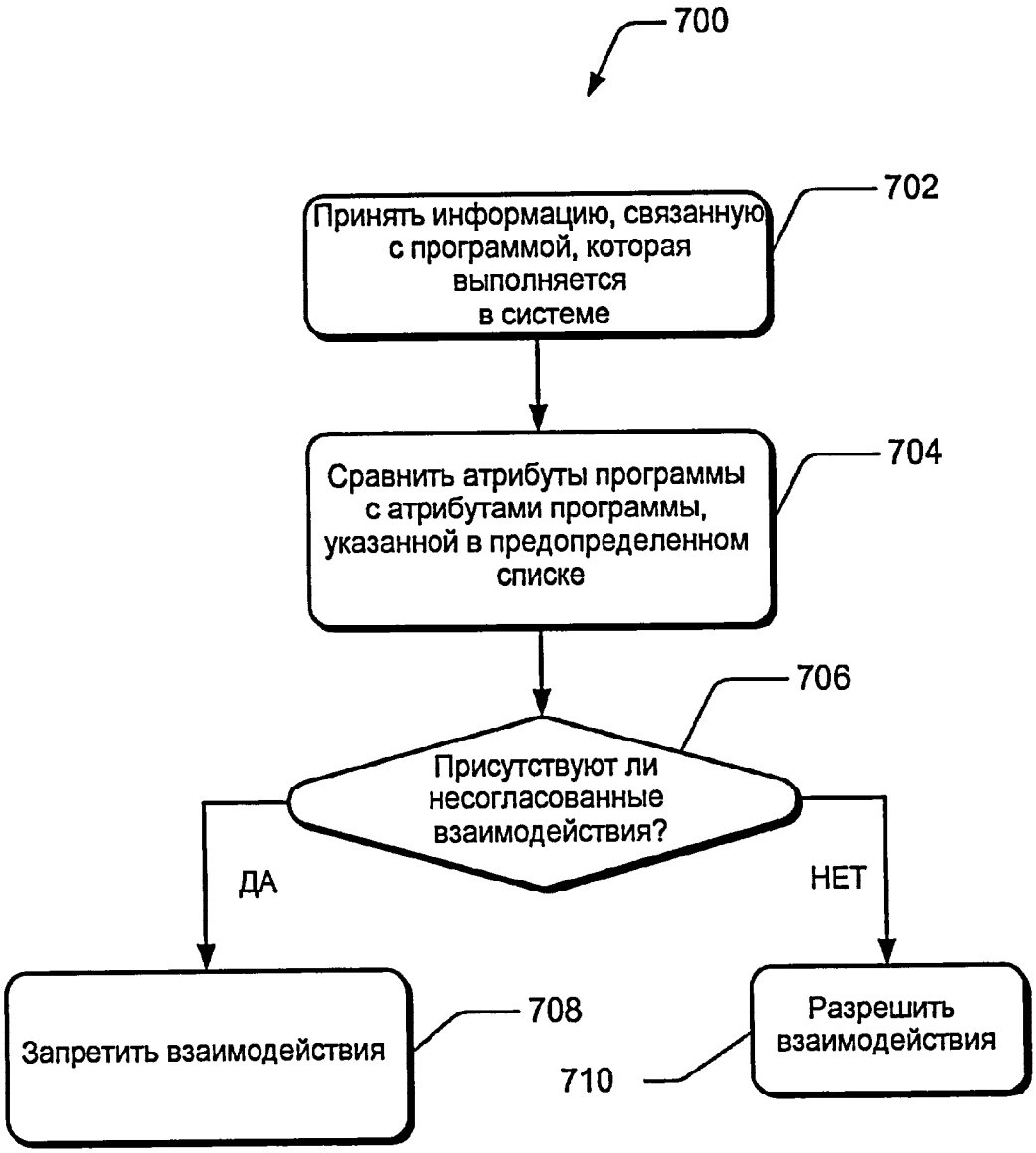
Фиг.4


500
502
Перехватить данные, связанные с программой, которая выполняется в системе
504
Скопировать данные в ячейку памяти
506
Сжать данные, собранные в ячейке памяти
508
Периодически выгружать сжатые данные для анализа

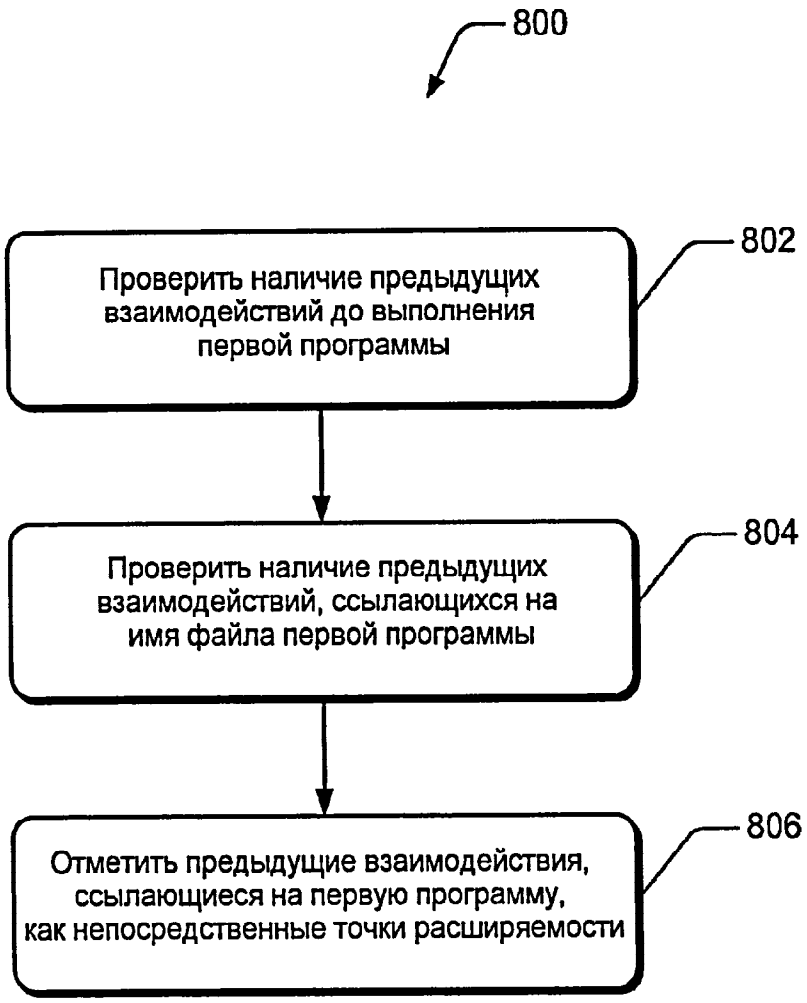
Фиг.5


600
602
Указать правило классификации с параметрами
604
Ассоциировать параметры с атрибутами отмеченного изменения, чтобы предоставить вероятные классификации
606
Присвоить значение приоритета вероятным классификациям
608
Назначить классификацию, соответствующую наивысшему значению приоритета

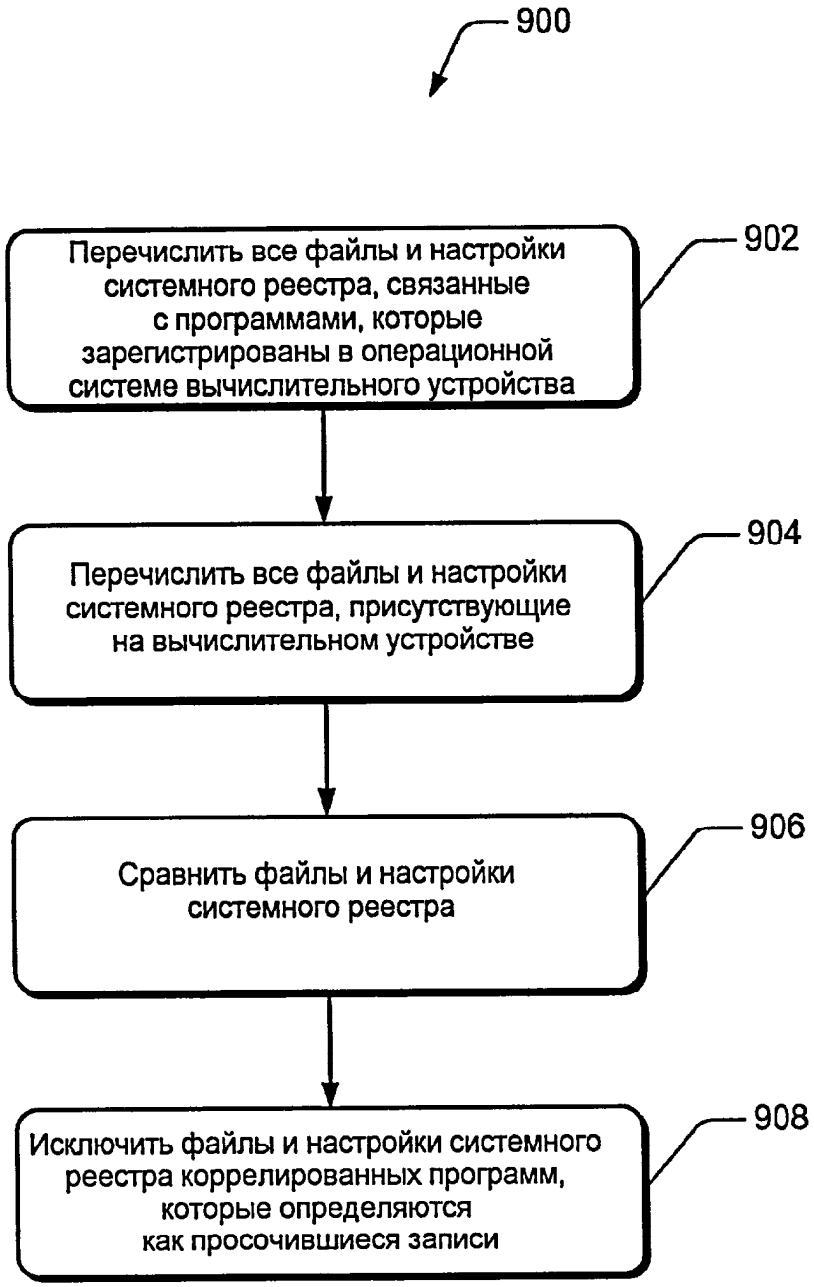
Фиг.6


700
Принять информацию, связанную с программой, которая выполняется в системе
702
Сравнить атрибуты программы с атрибутами программы, указанной в предопределенном списке
704
Присутствуют ли несогласованные взаимодействия?
706
ДА
НЕТ
Запретить взаимодействия
708
Разрешить взаимодействия
710

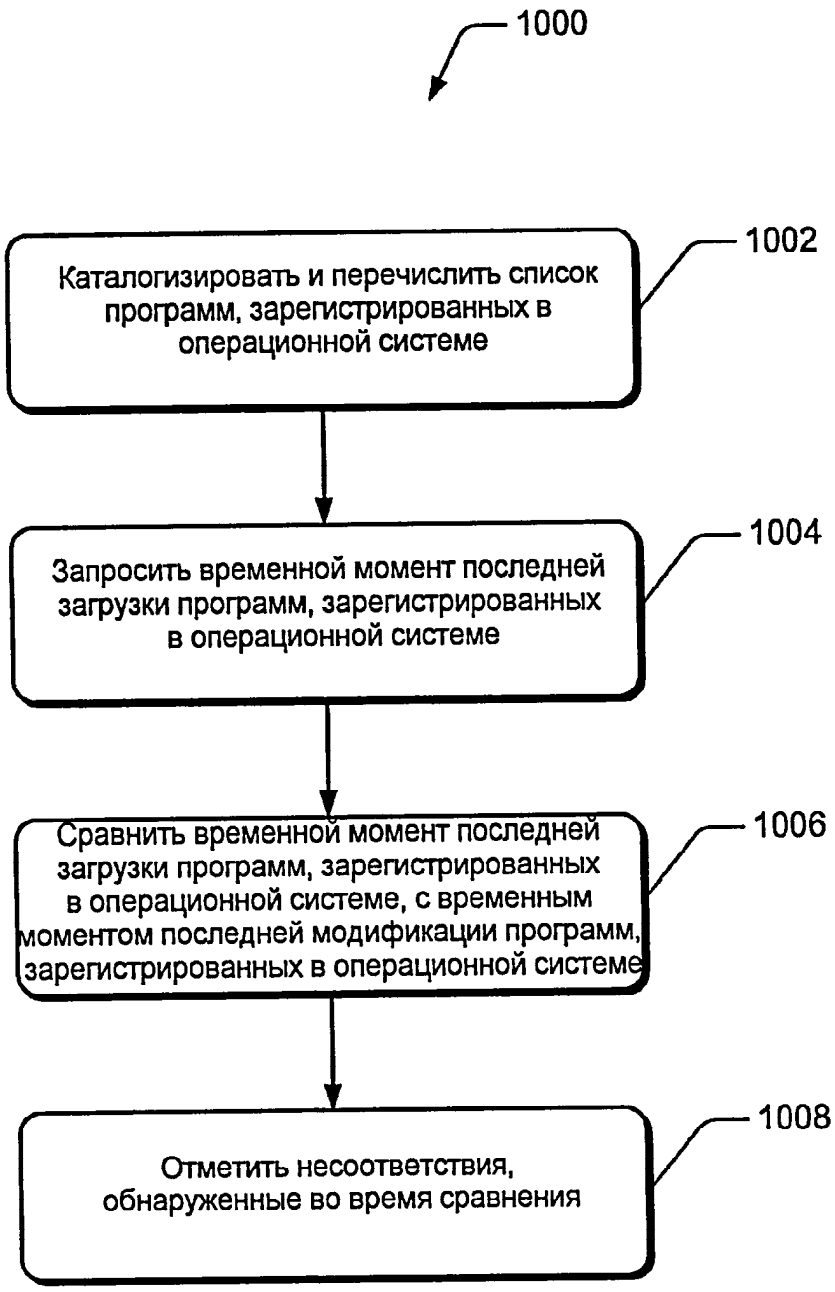
Фиг.7


800
Проверить наличие предыдущих взаимодействий до выполнения первой программы
802
Проверить наличие предыдущих взаимодействий, ссылающихся на имя файла первой программы
804
Отметить предыдущие взаимодействия, ссылающиеся на первую программу, как непосредственные точки расширяемости
806

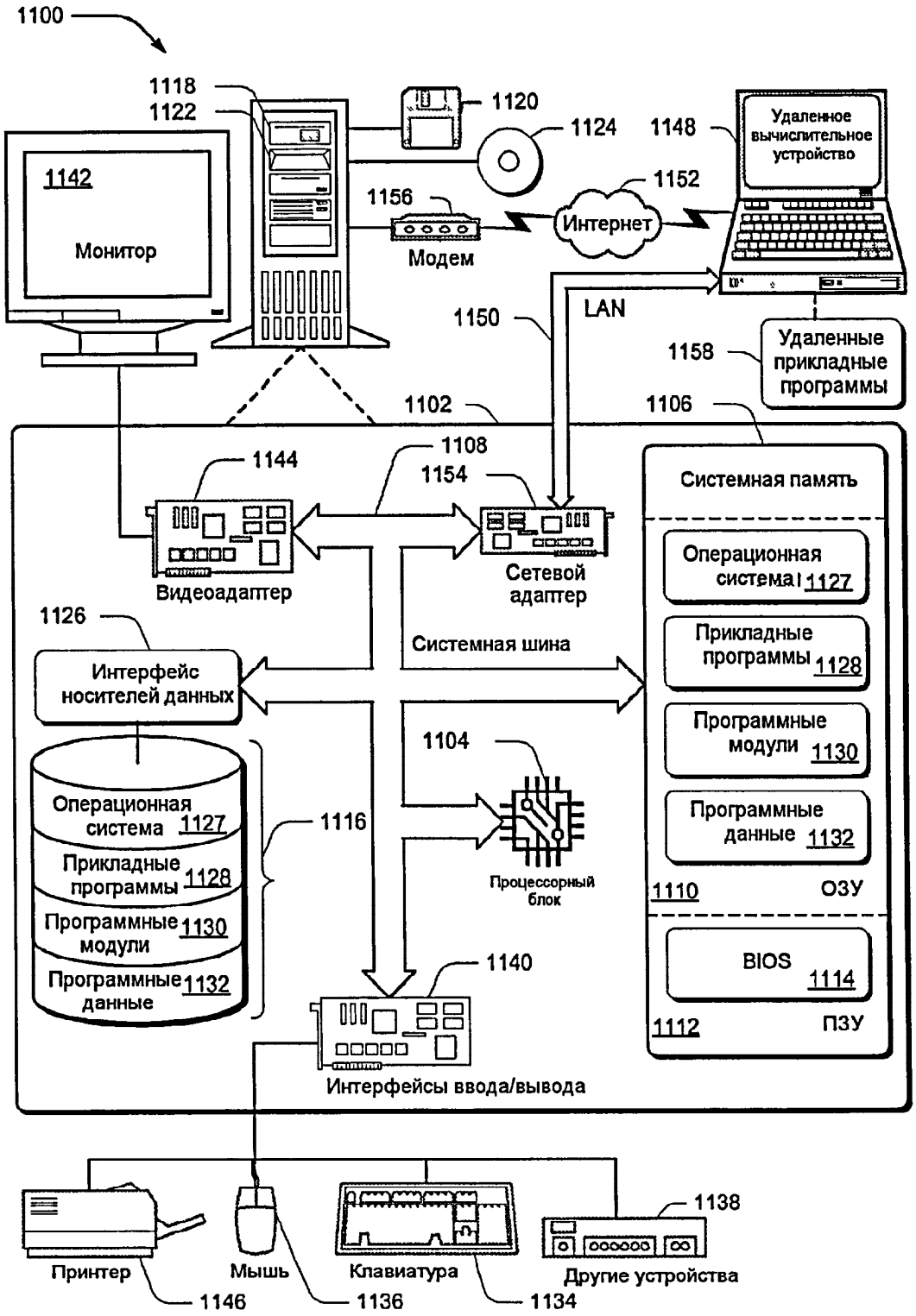
Фиг.8


900
Перечислить все файлы и настройки системного реестра, связанные с программами, которые зарегистрированы в операционной системе вычислительного устройства
902
Перечислить все файлы и настройки системного реестра, присутствующие на вычислительном устройстве
904
Сравнить файлы и настройки системного реестра
906
Исключить файлы и настройки системного реестра коррелированных программ, которые определяются как просочившиеся записи
908

Фиг.9


1000
Каталогизировать и перечислить список программ, зарегистрированных в операционной системе
1002
Запросить временной момент последней загрузки программ, зарегистрированных в операционной системе
1004
Сравнить временной момент последней загрузки программ, зарегистрированных в операционной системе, с временным моментом последней модификации программ, зарегистрированных в операционной системе
1006
Отметить несоответствия, обнаруженные во время сравнения
1008

Фиг.10


1100
1118
1122
1120
1124
1148
Удаленное вычислительное устройство
1142
Монитор
1156
Модем
Интернет
1152
1150
LAN
1158
Удаленные прикладные программы
1102
1106
1108
1144
1154
Системная память
Операционная система 1127
Прикладные программы 1128
Программные модули 1130
Программные данные 1132
1110
ОЗУ
BIOS 1114
1112
ПЗУ
Видеоадаптер
Сетевой адаптер
Системная шина
1126
Интерфейс носителей данных
1104
Процессорный блок
1116
Операционная система 1127
Прикладные программы 1128
Программные модули 1130
Программные данные 1132
1140
Интерфейсы ввода/вывода
Принтер
Мышь
Клавиатура
1138
Другие устройства
1146
1136
1134

Фиг.11