РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 541 120**(13) **C2**

(51) МПК
G06F 21/56 (2013.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: **2013125979/08, 06.06.2013**

(24) Дата начала отсчета срока действия патента:
06.06.2013

Приоритет(ы):
(22) Дата подачи заявки: **06.06.2013**

(43) Дата публикации заявки: **20.12.2014** Бюл. № **35**

(45) Опубликовано: **10.02.2015** Бюл. № **4**

(56) Список документов, цитированных в отчете о поиске: **US 2003/0023865 A1, 30.01.2003. US 2007/0240217 A1, 11.10.2007. US 2013/0097704 A1, 18.04.2013. US 2006/0167860 A1, 27.07.2006. US 7809667 B1, 05.10.2010. RU 2468418 C2, 27.11.2012**

Адрес для переписки:
125212, Москва, Ленинградское ш., 39а, стр. 3, ЗАО Лаборатория Касперского, Управление по интеллектуальной собственности, Надежда Васильевна Кащенко

(72) Автор(ы):
Татаринов Иван Иванович (RU)

(73) Патентообладатель(и):
Закрытое акционерное общество "Лаборатория Касперского" (RU)

RU 2541120 C2

(54) СИСТЕМА И СПОСОБ ОБНАРУЖЕНИЯ ВРЕДОНОСНЫХ ИСПОЛНЯЕМЫХ ФАЙЛОВ НА ОСНОВАНИИ СХОДСТВА РЕСУРСОВ ИСПОЛНЯЕМЫХ ФАЙЛОВ

(57) Реферат:

Изобретение относится к вычислительной технике. Технический результат заключается в повышении эффективности обнаружения вредоносных исполняемых файлов. Система обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов содержит средство обработки ресурсов для определения вида исполняемого файла и, по крайней мере, одного типа в соответствии с определенным видом исполняемого файла, выявления, по крайней мере, одного ресурса определенного типа исполняемого файла при помощи средства обработки ресурсов, преобразования, по крайней мере, одного выявленного ресурса определенного типа в формат для сравнения и его передачи средству сравнения; средство сравнения для подсчета степени сходства, по крайней мере, одного выявленного ресурса определенного типа с ресурсами упомянутого типа из ресурсов известных вредоносных исполняемых файлов из базы данных ресурсов с помощью алгоритмов сравнения для соответствующих типов ресурсов, передачи результата подсчета степени сходства средству проверки; средство проверки для определения того, является ли исполняемый файл вредоносным при помощи правил определения на основании подсчитанной степени сходства, по крайней мере, одного выявленного ресурса определенного типа с ресурсами упомянутого типа из ресурсов известных вредоносных исполняемых файлов. 2 н. и 1 з.п. ф-лы, 7 ил., 3 табл.

RU 2541120 C2

Фиг. 2

RU 2541120 C2

RU 2541120 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 541 120**(13) **C2**

(51) Int. Cl.
G06F 21/56 (2013.01)

(12) ABSTRACT OF INVENTION

(21)(22) Application: **2013125979/08, 06.06.2013**

(24) Effective date for property rights:
06.06.2013

Priority:
(22) Date of filing: **06.06.2013**

(43) Application published: **20.12.2014** Bull. № **35**

(45) Date of publication: **10.02.2015** Bull. № **4**

Mail address:
125212, Moskva, Leningradskoe sh., 39a, str. 3, ZAO Laboratorija Kasperskogo, Upravlenie po intellektual'noj sobstvennosti, Nadezhda Vasil'evna Kashchenko

(72) Inventor(s):
Tatarinov Ivan Ivanovich (RU)

(73) Proprietor(s):
Zakrytoe aktsionernoe obshchestvo "Laboratorija Kasperskogo" (RU)

RU 2541120 C2

(54) SYSTEM AND METHOD FOR DETECTING MALICIOUS EXECUTABLE FILES BASED ON SIMILARITY OF EXECUTABLE FILE RESOURCES

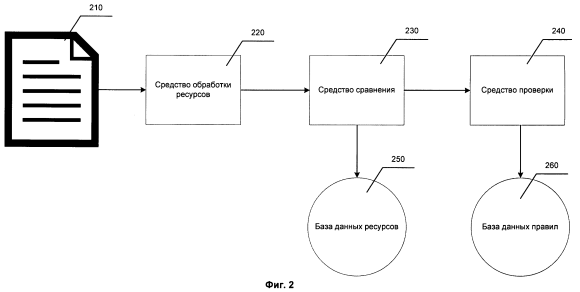
(57) Abstract:

FIELD: physics, computer engineering.

SUBSTANCE: invention relates to computer engineering. A system for detecting malicious executable files based on similarity of executable file resources comprises resource processing means for determining the type of an executable file and at least one type in accordance with the determined type of the executable file, detecting at least one resource of the determined type of executable file using the resource processing means, converting the at least one detected resource of the determined type into a format for comparison and transmission thereof to a comparing means; a comparing means for calculating the degree of similarity of at least one detected resource of the determined type with resources of said type from resources of known malicious executable files from a resource database using comparison algorithms for corresponding types of resources, transmitting the result of calculating the degree of similarity to a verification means; a verification means for determining if an executable file is malicious using determination rules based on the calculated degree of similarity of at least one detected resource of the determined type with resources of said type from resources of known malicious executable files.

EFFECT: high efficiency of detecting malicious executable files.

3 cl, 7 dwg, 3 tbl



Фиг. 2

RU 2541120 C2

Область техники

Изобретение относится к системам и способам обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов.

Уровень техники

Современные вредоносные приложения создаются в основном с целью похищения личных данных и денежных средств пользователей, организации бот-сетей. Для достижения подобных целей создатели вредоносных приложений используют различные технические приемы и приемы социальной инженерии. Вредоносное приложение типа лжеантивирус, например, под видом антивирусной программы обнаруживает несуществующие угрозы и требует заплатить деньги за их устранение. При этом вредоносное приложение внешне схоже с легальной антивирусной программой и использует схожие иконки, шрифты, курсоры и звуки.

Блокеры - тип вредоносных приложений, которые ограничивают доступ к файлам и операционной системе пользователя и требуют выкуп для возобновления работы, например, под угрозой уничтожения данных. Вредоносное приложение этого типа может отображать текст или картинку с надписью о нарушении прав и требованием внести плату в качестве компенсации. При этом тексты или картинки, используемые в различных модификациях блокеров, отличаются незначительно.

При создании вредоносных приложений количество используемых инструментов и методов, которые маскируют и/или модифицируют вредоносный код, постоянно растет. Использование таких методик, как полиморфизм и метаморфизм, позволяет создавать вредоносные приложения, которые могут быть не обнаружены при антивирусной проверке с использованием некоторых известных способов: сигнатурного анализа, поиска по хэш-сумме файла, анализа на основе эвристик. При этом возникают ситуации, например, когда пользователь видит один и тот же интерфейс ранее известного вредоносного приложения, но антивирусная программа не считает его вредоносным.

Существующие методы поиска и обнаружения вредоносных приложений имеют высокую эффективность, но не дают стопроцентного результата. В связи с этим возникает необходимость в их совершенствовании и развитии. Например, степень сходства используемых программами данных может позволить распознать среди новых неизвестных программ вредоносные приложения до проверки кода антивирусной программой.

Существуют различные способы поиска сходства вредоносных приложений. Такие подходы, несомненно, увеличивают количество обнаруживаемых вредоносных приложений среди новых. Ряд патентных публикаций описывает подобные подходы. Так, например, в публикации WO 2012110501 A1 описывается способ сравнения атрибутов, метаданных и другой информации для обнаружения похожих объектов (вредоносных приложений). В патенте US 8261344 B2 описано использование библиотеки факторов, по которой производится сравнение и обнаружение вредоносных приложений. Факторы могут включать в себя как код программ, характерные функции, свойства, хэш-суммы частей программ, образцы вредоносных приложений, так и другую информацию, которая способствует идентификации вредоносного приложения. Вышеописанные публикации не рассматривают поиск сходства и обнаружение вредоносных приложений, при котором происходит сравнение ресурсов исполняемых файлов.

Настоящее изобретение позволяет эффективно решить задачу обнаружения вредоносных исполняемых файлов на основании степени сходства ресурсов.

Раскрытие изобретения

Изобретение предназначено для проверки и обнаружения вредоносных приложений на основании сходства ресурсов исполняемых файлов.

Технический результат настоящего изобретения заключается в повышении эффективности обнаружения вредоносных исполняемых файлов. Указанный технический результат достигается за счет поиска сходства ресурсов исполняемого файла с известными ресурсами вредоносных исполняемых файлов. Поиск сходства осуществляется по типам ресурсов, которые содержит исполняемый файл.

Система обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов, которая содержит:

средство обработки ресурсов, предназначенное для извлечения ресурсов анализируемого исполняемого файла и их передачи средству сравнения; средство сравнения, предназначенное для поиска сходства ресурсов анализируемого исполняемого файла с известными ресурсами вредоносных исполняемых файлов из базы данных ресурсов, определения и передачи результата поиска сходства средству проверки; базу данных ресурсов, предназначенную для хранения известных ресурсов вредоносных исполняемых файлов; средство проверки, предназначенное для определения того, является ли анализируемый исполняемый файл вредоносным на основании результата поиска сходства ресурсов анализируемого исполняемого файла с известными ресурсами вредоносных исполняемых файлов при помощи правил определения, хранимых в базе данных правил; базу данных правил, предназначенную для хранения правил определения того, является ли файл вредоносным на основании результата поиска сходства ресурсов анализируемого исполняемого файла с известными ресурсами вредоносных исполняемых файлов.

В частном случае реализации системы средство обработки ресурсов перед извлечением ресурсов исполняемого файла определяет вид анализируемого исполняемого файла.

В другом частном случае реализации системы средство обработки ресурсов исполняемого файла по виду анализируемого исполняемого файла определяет типы извлекаемых ресурсов исполняемого файла.

Еще в одном частном случае реализации системы в базе данных ресурсов хранятся ресурсы исполняемого файла, которые были найдены в исполняемых файлах, не содержащих вредоносный код.

В другом частном случае реализации системы средство сравнения производит поиск сходства ресурсов анализируемого исполняемого файла с известными ресурсами вредоносных исполняемых файлов путем подсчета степени сходства при сравнении ресурса исполняемого файла, с известными ресурсами вредоносных исполняемых файлов, которые хранятся в базе данных ресурсов по алгоритмам сравнения для соответствующих типов ресурсов.

Еще в одном частном случае реализации системы при подсчете степени сходства ресурса исполняемого файла с найденным ресурсом исполняемого файла из базы данных ресурсов средство сравнения применяет алгоритмы сравнения для соответствующих типов сравниваемых ресурсов исполняемого файла.

В другом частном случае реализации системы средство проверки производит антивирусную проверку анализируемого исполняемого файла, который определен как вредоносный.

Еще в одном частном случае реализации системы средство проверки производит антивирусную проверку, по крайней мере, одним из способов антивирусной проверки.

Способ обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов, в котором: извлекают, по крайней мере, один ресурс

анализируемого исполняемого файла при помощи средства обработки ресурсов; при помощи средства сравнения производят поиск сходства, по крайней мере, одного извлеченного ресурса анализируемого исполняемого файла с ранее известными ресурсами вредоносных исполняемых файлов; при помощи средства сравнения определяют, является ли анализируемый исполняемый файл вредоносным на основании сходства, по крайней мере, одного из ресурсов исполняемого файла.

В частном случае реализации способа перед извлечением ресурсов исполняемого файла определяют вид анализируемого исполняемого файла.

В другом частном случае реализации способа по виду анализируемого исполняемого файла определяют типы извлекаемых ресурсов исполняемого файла.

Еще в одном частном случае реализации способа подсчитывают степень сходства, по крайней мере, одного ресурса исполняемого файла с найденными ресурсами исполняемого файла из базы данных ресурсов по алгоритмам сравнения.

В другом частном случае реализации способа при подсчете степени сходства ресурса исполняемого файла с найденным ресурсом исполняемого файла из базы данных ресурсов применяют алгоритм сравнения, соответствующий типу сравниваемых ресурсов исполняемого файла.

Еще в одном частном случае реализации способа анализируемый исполняемый файл считается схожим с файлом, содержащим известные ресурсы исполняемого файла из базы данных ресурсов, когда степень сходства, по крайней мере, одного из ресурсов анализируемого исполняемого файла с известным ресурсом вредоносного исполняемого файла из базы данных ресурсов превышает заданный порог.

В другом частном случае реализации способа анализируемый исполняемый файл считается несхожим с файлом, содержащим ресурсы исполняемого файла из базы данных ресурсов, когда степень сходства хотя бы одного из ресурсов анализируемого исполняемого файла с известным ресурсом вредоносного исполняемого файла из базы данных ресурсов не превышает заданный порог.

Еще в одном частном случае реализации способа хранят ресурсы исполняемого файла, которые были найдены в исполняемых файлах, не содержащих вредоносный код.

В другом частном случае реализации способа производят антивирусную проверку анализируемого исполняемого файла, который определен как вредоносный.

Еще в одном частном случае реализации способа производят антивирусную проверку, по крайней мере, одним из способов антивирусной проверки.

Краткое описание чертежей

Дополнительные цели, признаки и преимущества настоящего изобретения будут очевидными из прочтения последующего описания осуществления изобретения со ссылкой на прилагаемые чертежи, на которых:

Фиг.1 отображает пример структуры исполняемого файла.

Фиг.2 показывает структурную схему системы обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов.

Фиг.3 иллюстрирует результат работы алгоритма сравнения графических ресурсов путем анализа Y-гистограмм.

Фиг.4А и Фиг.4Б иллюстрируют методику сравнения элементов диалоговых окон.

Фиг.5 показывает схему способа работы системы обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов.

Фиг.6 представляет пример компьютерной системы общего назначения, персональный компьютер или сервер.

Хотя изобретение может иметь различные модификации и альтернативные формы, характерные признаки, показанные в качестве примера на чертежах, будут описаны подробно. Следует понимать, однако, что цель описания заключается не в ограничении изобретения конкретным его воплощением. Наоборот, целью описания является охват всех изменений, модификаций, входящих в рамки данного изобретения, как это определено приложенной формуле.

Описание вариантов осуществления изобретения

Объекты и признаки настоящего изобретения, способы для достижения этих объектов и признаков станут очевидными посредством отсылки к примерным вариантам осуществления. Однако настоящее изобретение не ограничивается примерными вариантами осуществления, раскрытыми ниже, оно может воплощаться в различных видах. Сущность, приведенная в описании, является ничем иным, как конкретными деталями, необходимыми для помощи специалисту в области техники в исчерпывающем понимании изобретения, и настоящее изобретение определяется в объеме приложенной формулы.

Перечисленные в уровне техники примеры вредоносных приложений обычно запускаются из вредоносных исполняемых файлов. Существует множество видов исполняемых файлов. Виды исполняемых файлов представлены в Таблице 1.

Таблица 1.

.acr	Сценарий ACRobot			CGI
.actm	Макрос AutoCAD		.cmd	Пакетный файл Windows
.ahk	Сценарий AutoHotkey		.cof	Исполняемый файл MPLAB COFF
.air	Установочный пакет Adobe AIR		.corn	Приложение MS-DOS
.apk	Пакет приложения Android		.csh	Сценарий C Shell
.app	Приложение Mac OS X		.cyw	Файл Rbot.CYW Worm
.app	Приложение FoxPro		.dek	Пакетный файл Eavesdropper
.app	Приложение Symbian OS		.did	Скомпилированная программа EdLog
.asb	Макрос Visual Basic (Alphacam)		.dmc	Сценарий Medical Manager
.awk	Сценарий AWK		.ds	Источник данных TWAIN
.bat	Пакетный файл MS-DOS		.dxl	Сценарий Rational DOORS
.bin	Исполняемый файл Unix		.ebm	Основной макрос EXTRA!
.cgi	Web-страница, Сценарий			
.ecf	Файл компонента SageCRM		.es	Файл сценария SageCRM
.elf	Исполняемый файл Playstation		.esh	Расширенный пакетный файл DOS
.elf	Исполняемый файл игры Nintendo Wii			

Фиг.1 отображает пример структуры исполняемого файла. Структура исполняемого файла 110 состоит из нескольких частей. Часть, содержащая секции, обычно называется «Object table». Секция в исполняемом файле представляет либо код, либо некоторые данные (глобальные переменные, таблицы импорта и экспорта, ресурсы, таблицу соответствий). Каждая секция имеет набор атрибутов, задающий ее свойства. Атрибуты секции определяют, доступна ли секция для чтения и записи, содержит ли она исполняемый код, должна ли она оставаться в памяти после загрузки исполняемого файла, могут ли различные процессы использовать один экземпляр этой секции и т.д. В одной из секций, которая часто имеет название «.rsrc» 120, располагаются ресурсы 130 по типам: иконки, текстовые инструкции, диалоговые окна, шрифты и другие.

Ресурсы необходимы для корректной работы исполняемого файла. Помимо этого, тип ресурсов RT_RCDATA может содержать в себе произвольные данные.

В общем случае структурная схема системы обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов имеет вид, показанный на Фиг.2. Система обнаружения вредоносных исполняемых файлов на основании сходства ресурсов состоит из средства обработки ресурсов 220, средства сравнения 230, средства проверки 240, базы данных ресурсов 250, базы данных правил 260. Средство обработки ресурсов 220 предназначено для определения вида анализируемого исполняемого файла 210, определения типов извлекаемых ресурсов по виду анализируемого исполняемого файла 210, извлечения 9 ресурсов анализируемого исполняемого файла 210, передачи ресурсов средству сравнения 230. Средство сравнения 230 предназначено для поиска сходства ресурсов анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов путем подсчета степени сходства при сравнении ресурса анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов, которые хранятся в базе данных ресурсов 250, по алгоритмам сравнения для соответствующих типов ресурсов, определения и передачи результата поиска сходства ресурсов анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов средству проверки 240. Сравнение осуществляется по всем типам ресурсов поочередно. Результат поиска сходства ресурсов анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов содержит степени сходства всех ресурсов анализируемого исполняемого файла 210. Средство проверки 240 предназначено для определения того, является ли анализируемый исполняемый файл 210 вредоносным на основании результата поиска сходства ресурсов при помощи правил, хранимых в базе данных правил 260, а также для антивирусной проверки анализируемых исполняемых файлов, которые были определены как вредоносные. При антивирусной проверке используются сигнатурный анализ, эвристический анализ, а также другие способы проверок. База данных ресурсов 250 предназначена для хранения известных ресурсов вредоносных исполняемых файлов. База данных правил 260 предназначена для хранения правил определения того, является ли анализируемый исполняемый файл вредоносным на основании результата поиска сходства ресурсов анализируемого исполняемого файла с известными ресурсами вредоносных исполняемых файлов.

В качестве базы данных ресурсов 250 и базы данных правил 260 могут использоваться различные виды баз данных, а именно: иерархические (IMS, TDMS, System 2000), сетевые (Cerebrum, Cronospro, DBVist), реляционные (DB2, Informix, Microsoft SQL Server), объектно-ориентированные (Jasmine, Versant, POET), объектно-реляционные (Oracle Database, PostgreSQL, FirstSQL/J, функциональные и т.д. Пример возможной базы данных ресурсов представлен в Таблице 2.

Таблица 2.

ID	Ресурс	Хэш	Гистограмма	Битность	Параметр N	Имя исполняемого файла
1	1.bmp	Значение параметра 1	Значение параметра 2	Значение параметра 3	Значение параметра N	exapmiei.exe

Помимо известных ресурсов вредоносных исполняемых файлов в базе данных ресурсов 250 могут храниться известные ресурсы исполняемых файлов, не содержащих вредоносный код. В случае если анализируемый исполняемый файл одновременно содержит известные ресурсы вредоносных исполняемых файлов и известные ресурсы исполняемых файлов, не содержащих вредоносный код, может быть вынесен ошибочный

вердикт, что является ложным срабатыванием. В этом случае следует проводить более детальную антивирусную проверку анализируемого исполняемого файла, в ходе которой необходимо использовать сигнатурный анализ, анализ при помощи эвристик, поведенческий анализ и т.д.

Правила в базе данных правил 260 могу быть изначально заданы антивирусной программой и изменяться после обновления антивирусных баз. Все правила задают порог степени сходства для конкретных типов ресурсов, превышение которого является признаком наличия вредоносного кода в исполняемом файле. Правило может содержать порог как для степени схожести конкретного типа ресурсов исполняемого файла, так и для множества ресурсов исполняемого файла из результата поиска сходства ресурсов анализируемого исполняемого файла с известными ресурсами вредоносных исполняемых файлов. В одном случае имеет значение лишь схожесть по одному конкретному типу ресурсов, например иконки. В другом случае даже стопроцентная степень схожести одного типа ресурсов может не иметь особого значения. Пример базы данных правил 260 представлен в Таблице 3.

Таблица 3.

ID	Формулировка правила	Вердикт
1.	Степень схожести любого ресурса с ресурсом из базы данных ресурсов >80%.	Вредоносное приложение
2.	Степень схожести иконки с ресурсом из базы данных ресурсов >50%; ресурса, содержащего аудиозапись >70%.	Вредоносное приложение
3.	Степень схожести всех ресурсов исполняемого файла с ресурсами из базы данных ресурсов >50%.	Вредоносное приложение

Иконки и курсоры являются графическими типами ресурсов исполняемого файла, которые после извлечения из исполняемого файла можно конвертировать в файл формата.bmp.Шрифты могут быть конвертированы в файлы.ttf. В RT_RCDATA могут храниться файлы и ресурсы любых типов и форматов, например:.jpg, .wav, .txt. Каждый тип ресурсов может иметь свои алгоритмы сравнения. Например, известным алгоритмом сравнения всех типов ресурсов считается сравнение хэш-сумм файлов. В случае, когда есть возможность конвертировать извлеченный ресурс в файл определенного формата, все алгоритмы сравнения файлов одного формата применимы к сравнению типа ресурсов соответственно.

Например, популярным алгоритмом сравнения, графических файлов.bmp является анализ Y-гистограмм. Фиг.3 отображает результат работы алгоритма сравнения графических файлов путем анализа Y-гистограмм. Файл 300 является иконкой размером 32×32 пикселей и битностью 32 бита, MD5: 87241a4f92f1efee41938d925f3ba303. Производится сравнение Y-гистограммы файла 300, с Y-гистограммами других известных файлов по следующим критериям: размер - 32×32 пикселя, битность - 32. В результате сравнения найдено 5 записей (из 1000): 310, 320, 330, 340, 350, удовлетворяющих критериям поиска. Анализ гистограмм дает результат более 95% сходства на файлы 310, 320, 340, со следующими MD5: b14ale29d8a630c365a05349e8fccd 9a, bc221dea2e39fdl02261b2e65aaba41c, e3c763646e2a60658a21d72f8alfb9e7. Таким образом, файлы 300, 310, 320, 340, схожи внешне, но имеют различные MD5.

Помимо этого, существуют алгоритм сравнения содержимого аудио файлов (http://www.ionio.gr/~karydis/myjapers/KNPM2004%20-%20Evaluation%20oP/o20Similaritv%20Searching%20Methods%20for%20Music %20Data%20in%20Peer-to-Peer%20Networks.pdf), текстовых файлов (http://ucrel.lancs.ac.uk/publications/CL2003/papers/piao.pdf).

Другим типом ресурсов для сравнения могут быть диалоговые окна. Диалоговое

окно - окно графического пользовательского интерфейса, предназначенное для вывода информации и/или получения ответа от пользователя. Таким образом, диалоговое окно осуществляет двустороннее взаимодействие компьютер-пользователь («диалог»). Структура диалоговых окон напоминает структуру окон приложений и подчиняется общим правилам. В верхней части окна располагается строка заголовка, под ней все пространство занимает рабочая область. Все окно заключено в рамку. Изменение размеров диалоговых окон не допускается. Рабочая область в диалоговых окнах содержит элементы управления. Настройка в диалоговых окнах производится путем взаимодействия с элементами управления. Элементы управления служат для ввода данных (текста или числового значения), выбора одного или нескольких вариантов из числа заданных, выполнения вспомогательных операций, ответа на заданные пользователю вопросы и др.

Создание диалоговых окон в приложениях происходит как напрямую, через создание окна и элементов (через Windows API), так и с использованием шаблонов. Шаблон представляет собой данные о параметрах и элементах, используемых при отображении диалогового окна. В исходном коде шаблон представлен в виде текстовой информации, в приложении после компиляции - в виде сжатой информации. Фиг.4А и Фиг.4Б иллюстрируют методику сравнения элементов диалоговых окон. Например, шаблон диалогового окна 410 во время исполнения изображен на рисунке 405.

Существует несколько алгоритмов сравнения схожести двух диалоговых окон по шаблонам.

1. Сравнение «элемент к элементу»:

В данном случае шаблоны сравниваются элемент к элементу. Элементы исследуемого шаблона должны быть абсолютно идентичным элементам шаблона, с которым происходит сравнение. В этом случае можно сделать вывод о том, что диалоговые окна схожи.

2. Сравнение по наличию элементов:

Данная проверка основана на том факте, что перестановка элементов никак не влияет на функционал диалоговых окон. Например, шаблоны диалоговых окон 420 и 430 являются идентичными. Алгоритм сравнения в данном случае будет выглядеть следующим образом:

а) Приведение шаблонов к универсальному виду. Элементы следуют друг за другом не в том порядке, в котором их расположил программист, а по некоторому правилу (например, по возрастанию идентификатора (ID) элемента); свойства внутри элементов указаны не беспорядочно, а по некоторому правилу (например, по возрастанию идентификатора (ID) свойства).

б) Сравнение полученных универсальных шаблонов элемент к элементу.

Например, вышеупомянутый шаблон 430 после выполнения алгоритма 2 преобразуется в шаблон 440. Стоит отметить, что алгоритмы 1 и 2 сравнивают полностью идентичные диалоговые окна, без каких-либо различий при отображении, но алгоритм 2 является более универсальным, хотя и более медленным, поскольку требует анализа шаблона диалогового окна.

Способ, речь о котором пойдет ниже, схож с алгоритмом сравнения 2, с тем отличием, что при создании универсального шаблона игнорируются некоторые параметры. Основная причина игнорирования состоит в том, что даже при минимальном изменении шаблона диалогового окна (смещение элемента, изменение текста и т.п.) алгоритм сравнения 2 неэффективен. Алгоритм сравнения 3 идентичен алгоритму 2, за исключением правил, по которым строится универсальный шаблон.

3. Сравнение с измененными элементами:

3.1 Игнорирование ID элементов

Рассмотрим следующий элемент:

IDD_DIALOG_UPDATE_DB DIALOGEX 0, 0, 340, 93

IDD_DIALOG__UPDATE_DB - идентификатор диалогового окна, который может отличаться от программы к программе. На функционал диалогового окна 450 этот элемент не влияет и может быть исключен из сравнения.

3.2 Игнорирование элементов по умолчанию

Например,

DEFPUSHBUTTON "Обновить", 7,65,75,21

станет

PUSHBUTTON "Обновить", 7,65,75,21

3.3 Игнорирование некоторых несущественных параметров элементов, которые никак не влияют на внешний вид диалогового окна

Например, флаг WS_TABSTOP показывает, что этот элемент можно активировать путем нажатия клавиши Tab. На внешний вид этот флаг диалогового окна никак не влияет.

3.4 Игнорирование размеров и положений элементов при сохранении их связей

Также из шаблона можно исключить размеры и координаты элементов. Но для того чтобы шаблон после выполнения алгоритма 3 не превратился в набор элементов, необходимо оставить связи между элементами. Например, все элементы, принадлежащие области «GROUPBOX», должны иметь координаты, входящие в эту область. Положение и размер области «GROUPBOX» заданы набором значений параметров {xgroupbox, ygroupbox, xgroupbox+wgroupbox, ygroupbox+hgroupbox}. Таким образом, указанные элементы являются зависимыми элементами.

Фиг.5 показывает способ обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов. На этапе 510 антивирусная программа инициирует процесс проверки, в ходе которого исполняемый файл 210, содержащий ресурсы, передается системе обнаружения вредоносных исполняемых файлов на основании степени сходства ресурсов. На этапе 520 средство обработки ресурсов 220 производит обработку анализируемого исполняемого файла 210, которая заключается в определении вида анализируемого исполняемого файла 210, определении типов извлекаемых ресурсов по виду анализируемого исполняемого файла 210, извлечении ресурсов анализируемого исполняемого файла 210 и передаче их средству сравнения 230. На этапе 530 средство сравнения 230 производит поиск сходства ресурса анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов путем подсчета степени сходства при сравнении ресурса анализируемого исполняемого файла 210, полученного от средства обработки ресурсов 220, с известными ресурсами вредоносных исполняемых файлов, которые хранятся в базе данных ресурсов 250 по алгоритмам сравнения для соответствующих типов ресурсов. На этапе 540 средство сравнения 230 проверяет наличие следующего ресурса для поиска. В случае если не по всем ресурсам произведен поиск, средство сравнения 230 принимает очередной ресурс. В случае если поиск произведен по всем ресурсам, на этапе 550 средство сравнения 230 определяет результат поиска сходства ресурсов анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов и передает его средству проверки 240. На этапе 560 средство проверки 240 на основе результата поиска сходства ресурсов анализируемого исполняемого файла 210 с известными ресурсами вредоносных исполняемых файлов

по правилам из базы данных правил 260 определяет, является ли файл вредоносным. В случае если анализируемый исполняемый файл определен как вредоносный, средство проверки 240 дополнительно производит его антивирусную проверку с использованием сигнатурного анализа, эвристического анализа и т.д.

Фиг.6 представляет пример компьютерной системы общего назначения, персональный компьютер или сервер 20, содержащий центральный процессор 21, системную память 22 и системную шину 23, которая содержит разные системные компоненты, в том числе память, связанную с центральным процессором 21. Системная шина 23 реализована, как любая известная из уровня техники шинная структура, содержащая в свою очередь память шины или контроллер памяти шины, периферийную шину и локальную шину, которая способна взаимодействовать с любой другой шинной архитектурой. Системная память содержит постоянное запоминающее устройство (ПЗУ) 24, память с произвольным доступом (ОЗУ) 25. Основная система ввода/вывода (BIOS) 26, содержит основные процедуры, которые обеспечивают передачу информации между элементами персонального компьютера 20, например, в момент загрузки операционной системы с использованием ПЗУ 24.

Персональный компьютер 20 в свою очередь содержит жесткий диск 27 для чтения и записи данных, привод магнитных дисков 28 для чтения и записи на сменные магнитные диски 29 и оптический привод 30 для чтения и записи на сменные оптические диски 31, такие как CD-ROM, DVD-ROM и иные оптические носители информации. Жесткий диск 27, привод магнитных дисков 28, оптический привод 30 соединены с системной шиной 23 через интерфейс жесткого диска 32, интерфейс магнитных дисков 33 и интерфейс оптического привода 34 соответственно. Приводы и соответствующие компьютерные носители информации представляют собой энергонезависимые средства хранения компьютерных инструкций, структур данных, программных модулей и прочих данных персонального компьютера 20.

Настоящее описание раскрывает реализацию системы, которая использует жесткий диск 27, сменный магнитный диск 29 и сменный оптический диск 31, но следует понимать, что возможно применение иных типов компьютерных носителей информации 56, которые способны хранить данные в доступной для чтения компьютером форме (твердотельные накопители, флеш карты памяти, цифровые диски, память с произвольным доступом (ОЗУ) и т.п.), которые подключены к системной шине 23 через контроллер 55.

Компьютер 20 имеет файловую систему 36, где хранится записанная операционная система 35, а также дополнительные программные приложения 37, другие программные модули 38 и данные программ 39. Пользователь имеет возможность вводить команды и информацию в персональный компьютер 20 посредством устройств ввода (клавиатуры 40, манипулятора «мышь» 42). Могут использоваться другие устройства ввода (не отображены): микрофон, джойстик, игровая консоль, сканнер и т.п. Подобные устройства ввода по своему обычаю подключают к компьютерной системе 20 через последовательный порт 46, который в свою очередь подсоединен к системной шине, но могут быть подключены иным способом, например, при помощи параллельного порта, игрового порта или универсальной последовательной шины (USB). Монитор 47 или иной тип устройства отображения также подсоединен к системной шине 23 через интерфейс, такой как видеоадаптер 48. В дополнение к монитору 47, персональный компьютер может быть оснащен другими периферийными устройствами вывода (не отображены), например, колонками, принтером и т.п.

Персональный компьютер 20 способен работать в сетевом окружении, при этом

используется сетевое соединение с другим или несколькими удаленными компьютерами 49. Удаленный компьютер (или компьютеры) 49 являются такими же персональными компьютерами или серверами, которые имеют большинство или все упомянутые элементы, отмеченные ранее при описании существа персонального компьютера 20, представленного на Фиг.5. В вычислительной сети могут присутствовать также и другие устройства, например маршрутизаторы, сетевые станции, пиринговые устройства или иные сетевые узлы.

Сетевые соединения могут образовывать локальную вычислительную сеть (LAN) 50 и глобальную вычислительную сеть (WAN). Такие сети применяются в корпоративных компьютерных сетях, внутренних сетях компаний и, как правило, имеют доступ к сети Интернет. В LAN- или WAN-сетях персональный компьютер 20 подключен к локальной сети 50 через сетевой адаптер или сетевой интерфейс 51. При использовании сетей персональный компьютер 20 может использовать модем 54 или иные средства обеспечения связи с глобальной вычислительной сетью, такой как Интернет. Модем 54, который является внутренним или внешним устройством, подключен к системной шине 23 посредством последовательного порта 46. Следует уточнить, что сетевые соединения являются лишь примерными и не обязаны отображать точную конфигурацию сети, т.е. в действительности существуют иные способы установления соединения техническими средствами связи одного компьютера с другим.

В заключение следует отметить, что приведенные в описании сведения являются только примерами, которые не ограничивают объем настоящего изобретения, определенного формулой.

Формула изобретения

1. Система обнаружения вредоносных исполняемых файлов на основании сходства ресурсов исполняемых файлов, которая содержит:

а) средство обработки ресурсов, предназначенное для:

- определения вида исполняемого файла,

- определения, по крайней мере, одного типа в соответствии с определенным видом исполняемого файла,

- выявления, по крайней мере, одного ресурса определенного типа исполняемого файла при помощи средства обработки ресурсов,

- преобразования, по крайней мере, одного выявленного ресурса определенного типа в формат для сравнения и его передачи средству сравнения;

б) средство сравнения, предназначенное для подсчета степени сходства, по крайней мере, одного выявленного ресурса определенного типа с ресурсами упомянутого типа из ресурсов известных вредоносных исполняемых файлов из базы данных ресурсов с помощью алгоритмов сравнения для соответствующих типов ресурсов, передачи результата подсчета степени сходства средству проверки;

в) базу данных ресурсов, предназначенную для хранения известных ресурсов вредоносных исполняемых файлов;

г) средство проверки, предназначенное для определения того, является ли исполняемый файл вредоносным при помощи правил определения, хранимых в базе данных правил, на основании подсчитанной степени сходства, по крайней мере, одного выявленного ресурса определенного типа с ресурсами упомянутого типа из ресурсов известных вредоносных исполняемых файлов;

д) базу данных правил, предназначенную для хранения правил определения.

2. Способ обнаружения вредоносных исполняемых файлов на основании сходства

ресурсов исполняемых файлов, в котором:

а) определяют вид исполняемого файла при помощи средства обработки ресурсов;

б) определяют, по крайней мере, один тип ресурсов исполняемого файла в соответствии с определенным видом исполняемого файла при помощи средства обработки ресурсов;

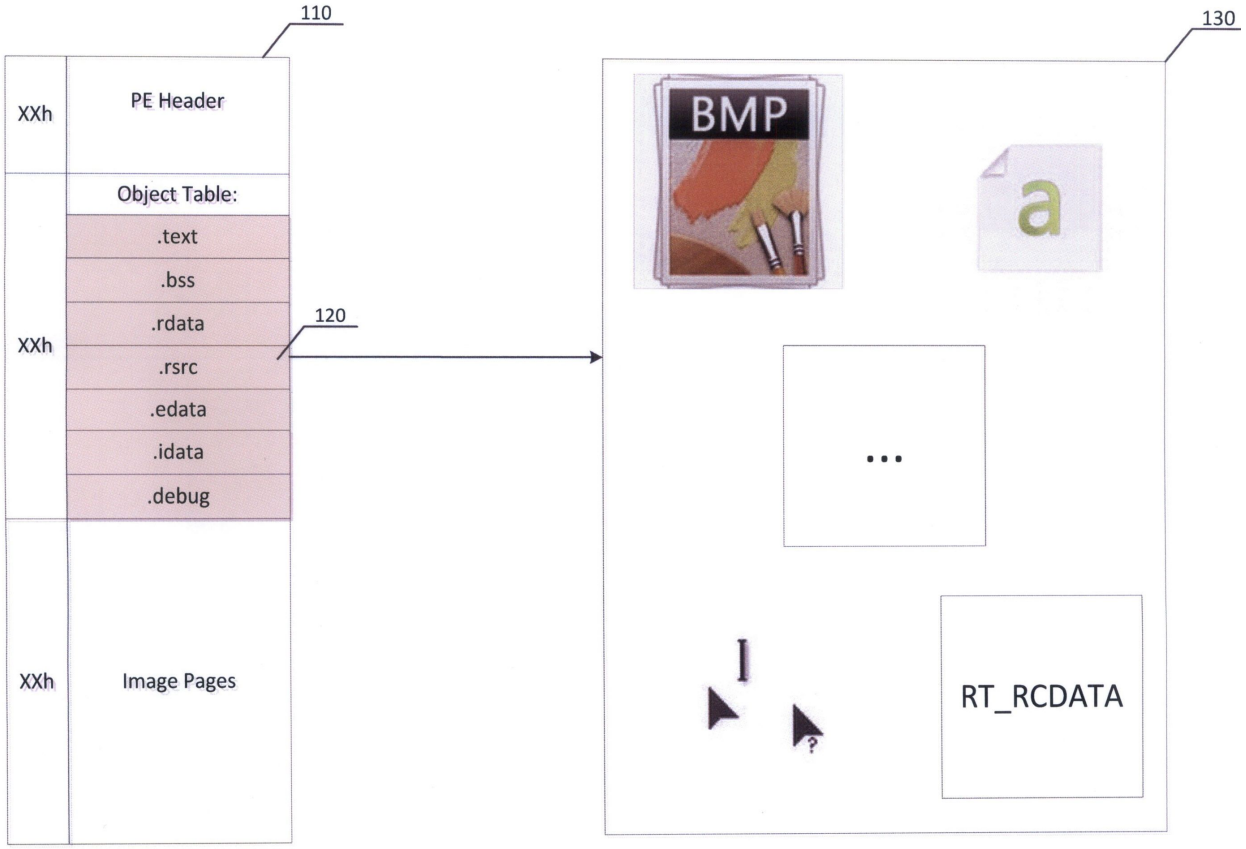
в) выявляют, по крайней мере, один ресурс определенного типа исполняемого файла при помощи средства обработки ресурсов;

г) преобразовывают, по крайней мере, один выявленный ресурс определенного типа в формат для сравнения при помощи средства обработки ресурсов;

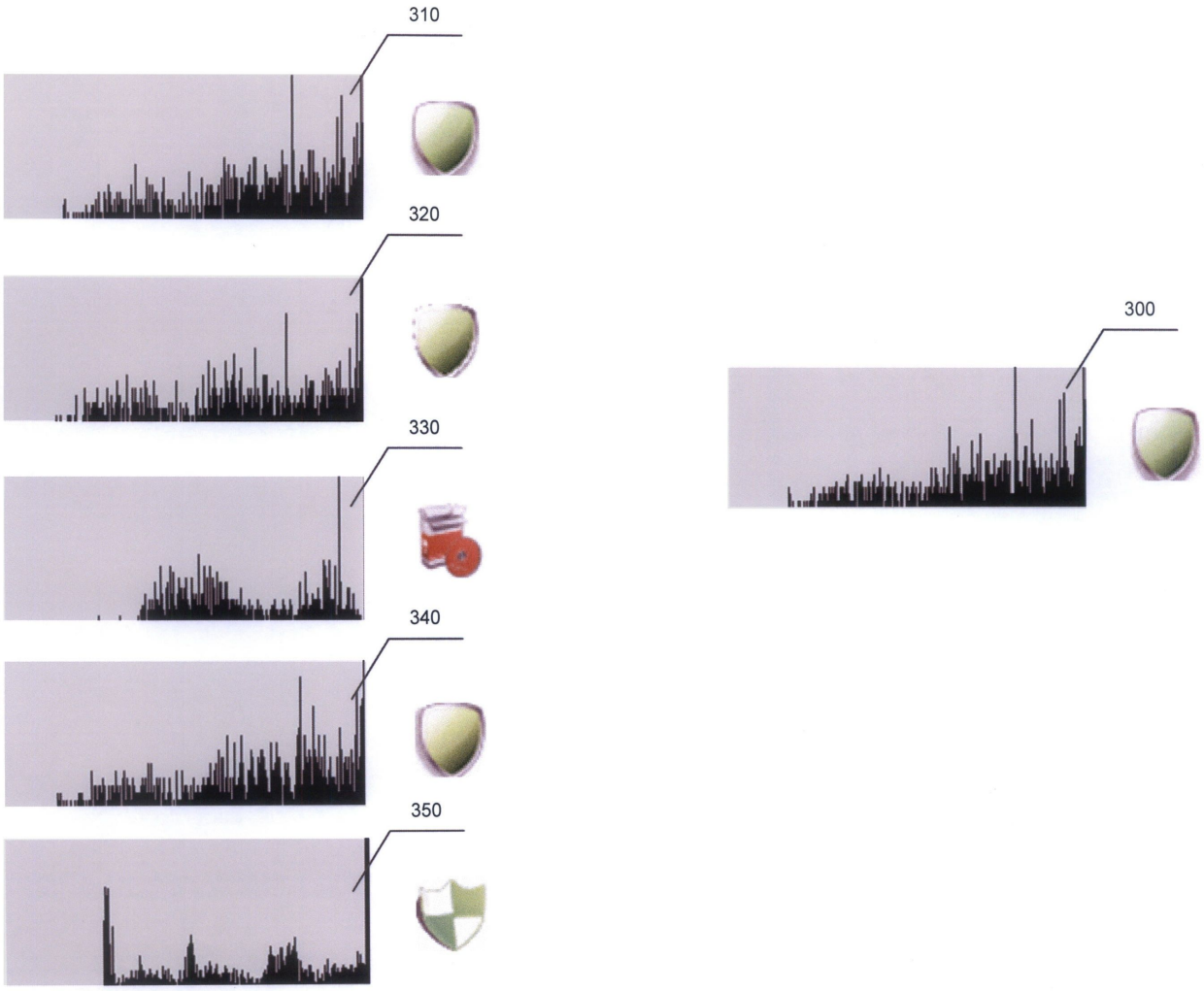
д) при помощи средства сравнения производят подсчет степени сходства, по крайней мере, одного выявленного ресурса определенного типа с ресурсами упомянутого типа из ресурсов известных вредоносных исполняемых файлов с помощью алгоритмов сравнения для соответствующих типов ресурсов;

е) при помощи средства проверки определяют, является ли исполняемый файл вредоносным при помощи правил определения на основании подсчитанной степени сходства, по крайней мере, одного выявленного ресурса определенного типа с ресурсами упомянутого типа из ресурсов известных вредоносных исполняемых файлов.

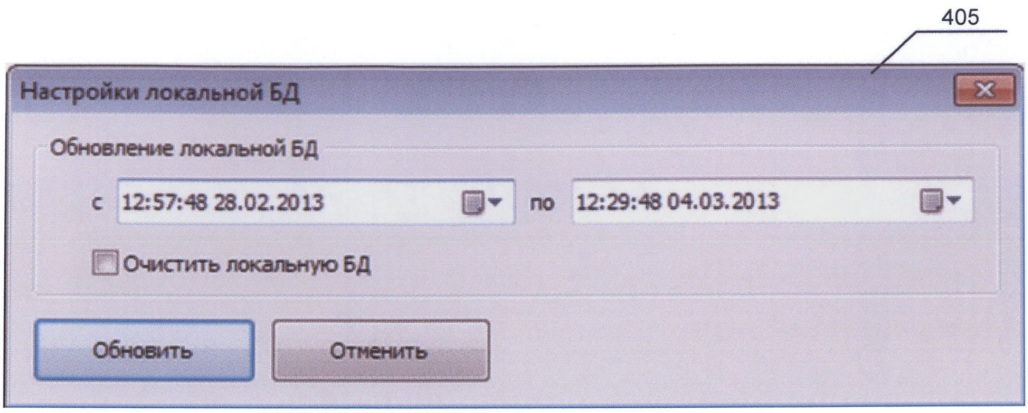
3. Способ по п. 2, в котором правило определения содержит заданный порог для, по крайней мере, одного определенного типа ресурсов, превышение которого является признаком наличия вредоносного кода в исполняемом файле.


110
130
XXh
PE Header
Object Table:
.text
.bss
.rdata
.rsrc
.edata
.idata
.debug
120
XXh
XXh
Image Pages
BMP
a
...
RT_RCDATA

Фиг. 1



Фиг. 3




```
IDD_DIALOG__UPDATE_DB DIALOGEX 0, 0, 340, 93
STYLE DS_SETFONT | DS_MODALFRAME | DS_FIXEDSYS | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Настройки локальной БД"
FONT 8, "MS Shell Dlg", 400, 0, 0x1
BEGIN
    DEFPUSHBUTTON    "Обновить",IDOK,7,65,75,21
    PUSHBUTTON       "Отменить",IDCANCEL,87,65,75,21
    CONTROL          "",IDC_DATETIMEPICKER__PREVIOUS_UPDATE_DATE,
                     "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,35,21,135,15
    LTEXT            "с",IDC_STATIC,27,24,8,8
    CONTROL          "",IDC_DATETIMEPICKER__NEXT_UPDATE_DATE,
                     "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,189,21,135,15
    LTEXT            "по",IDC_STATIC,176,24,9,8
    GROUPBOX         "Обновление локальной БД",IDC_STATIC,7,7,326,52
    CONTROL          "Очистить локальную БД",IDC_CHECK__CLEAN_DB,"Button",BS_AUTOCHECKBOX |
WS_TABSTOP,27,43,100,10
END
```

410

```
IDD_DIALOG__UPDATE_DB DIALOGEX 0, 0, 340, 93
STYLE DS_SETFONT | DS_MODALFRAME | DS_FIXEDSYS | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Настройки локальной БД"
FONT 8, "MS Shell Dlg", 400, 0, 0x1
BEGIN
    DEFPUSHBUTTON    "Обновить",IDOK,7,65,75,21
    PUSHBUTTON       "Отменить",IDCANCEL,87,65,75,21
    CONTROL          "",IDC_DATETIMEPICKER__PREVIOUS_UPDATE_DATE,
                     "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,35,21,135,15
    LTEXT            "с",IDC_STATIC,27,24,8,8
    CONTROL          "",IDC_DATETIMEPICKER__NEXT_UPDATE_DATE,
                     "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,189,21,135,15
    LTEXT            "по",IDC_STATIC,176,24,9,8
    GROUPBOX         "Обновление локальной БД",IDC_STATIC,7,7,326,52
    CONTROL          "Очистить локальную БД",IDC_CHECK__CLEAN_DB,"Button",BS_AUTOCHECKBOX |
WS_TABSTOP,27,43,100,10
END
```

420

Фиг. 4А

```
IDD_DIALOG__UPDATE_DB DIALOGEX 0, 0, 340, 93
STYLE DS_SETFONT | DS_MODALFRAME | DS_FIXEDSYS | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Настройки локальной БД"
FONT 8, "MS Shell Dlg", 400, 0, 0x1
BEGIN
    GROUPBOX        "Обновление локальной БД",IDC_STATIC,7,7,326,52
    CONTROL         "",IDC_DATETIMEPICKER__PREVIOUS_UPDATE_DATE,
                    "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,35,21,135,15
    PUSHBUTTON      "Отменить",IDCANCEL,87,65,75,21
    CONTROL         "",IDC_DATETIMEPICKER__NEXT_UPDATE_DATE,
                    "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,189,21,135,15
    LTEXT           "с",IDC_STATIC,27,24,8,8
    LTEXT           "по",IDC_STATIC,176,24,9,8
    CONTROL         "Очистить локальную БД",IDC_CHECK__CLEAN_DB,"Button",BS_AUTOCHECKBOX |
WS_TABSTOP,27,43,100,10
    DEFPUSHBUTTON   "Обновить",IDOK,7,65,75,21
END
```

430

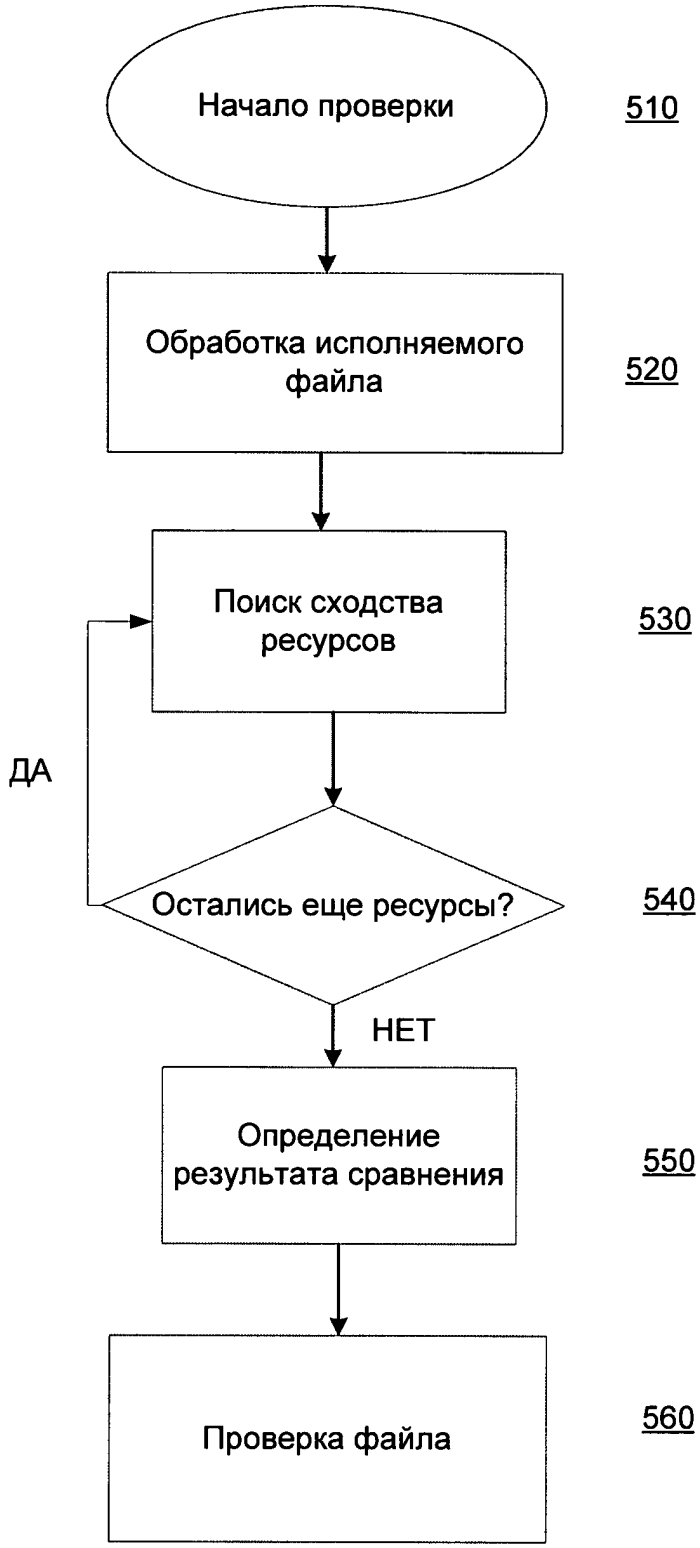
```
IDD_DIALOG__UPDATE_DB DIALOGEX 0, 0, 340, 93
STYLE DS_SETFONT | DS_MODALFRAME | DS_FIXEDSYS | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Настройки локальной БД"
FONT 8, "MS Shell Dlg", 400, 0, 0x1
BEGIN
    GROUPBOX        "Обновление локальной БД",IDC_STATIC,7,7,326,52
    CONTROL         "",IDC_DATETIMEPICKER__PREVIOUS_UPDATE_DATE,
                    "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,35,21,135,15
    CONTROL         "",IDC_DATETIMEPICKER__NEXT_UPDATE_DATE,
                    "SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,189,21,135,15
    CONTROL         "Очистить локальную БД",IDC_CHECK__CLEAN_DB,"Button",BS_AUTOCHECKBOX |
WS_TABSTOP,27,43,100,10
    LTEXT           "с",IDC_STATIC,27,24,8,8
    LTEXT           "по",IDC_STATIC,176,24,9,8
    DEFPUSHBUTTON   "Обновить",IDOK,7,65,75,21
    PUSHBUTTON      "Отменить",IDCANCEL,87,65,75,21
END
```

440

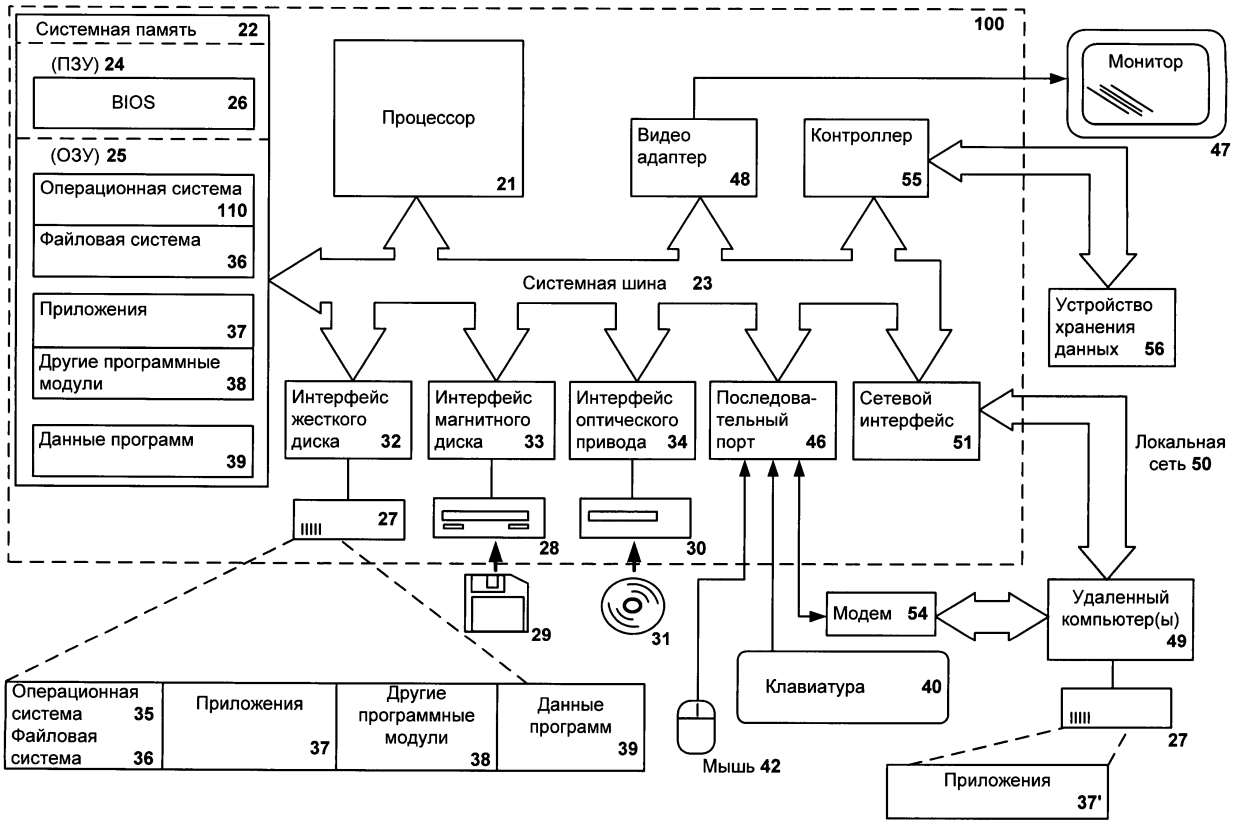
```
DIALOGEX 0, 0, 340, 93
STYLE DS_SETFONT | DS_MODALFRAME | DS_FIXEDSYS | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Настройки локальной БД"
FONT 8, "MS Shell Dlg", 400, 0, 0x1
BEGIN
    GROUPBOX        "Обновление локальной БД",7,7,326,52
    CONTROL         "","SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,35,21,135,15
    CONTROL         "","SysDateTimePick32",DTS_RIGHTALIGN | WS_TABSTOP,189,21,135,15
    CONTROL         "Очистить локальную БД",IDC_CHECK__CLEAN_DB,"Button",BS_AUTOCHECKBOX |
WS_TABSTOP,27,43,100,10
    LTEXT           "с",27,24,8,8
    LTEXT           "по",176,24,9,8
    DEFPUSHBUTTON   "Обновить",7,65,75,21
    PUSHBUTTON      "Отменить",87,65,75,21
END
```

450

Фиг. 4Б


Начало проверки
510
Обработка исполняемого файла
520
Поиск сходства ресурсов
530
ДА
Остались еще ресурсы?
540
НЕТ
Определение результата сравнения
550
Проверка файла
560

Фиг. 5


100
Системная память 22
(ПЗУ) 24
BIOS 26
(ОЗУ) 25
Операционная система 110
Файловая система 36
Приложения 37
Другие программные модули 38
Данные программ 39
Процессор 21
Видео адаптер 48
Контроллер 55
Монитор 47
Системная шина 23
Устройство хранения данных 56
Интерфейс жесткого диска 32
Интерфейс магнитного диска 33
Интерфейс оптического привода 34
Последова-тельный порт 46
Сетевой интерфейс 51
Локальная сеть 50
27
28
30
29
31
Модем 54
Удаленный компьютер(ы) 49
Клавиатура 40
Операционная система 35
Файловая система 36
Приложения 37
Другие программные модули 38
Данные программ 39
Мышь 42
27
Приложения 37'

Фиг. 6