Guardant®

Система защиты от компьютерного пиратства

Эффективная
защита приложений

Урок 1.1: знакомство
с автоматической защитой

Содержание

Общее описание автоматической защиты

В составе комплекта разработчика Guardant поставляется набор утилит автоматической защиты, предназначенных для обработки готовых исполняемых файлов, в результате которой приложение привязывается к электронному ключу и получает защиту от анализа и несанкционированной отладки.

Главное преимущество метода – в малом времени установки защиты. На сам процесс требуется всего несколько секунд. Стоит отметить, что каким бы надежным он ни был, как и в случае любого другого шаблонного метода защиты, наибольшая эффективность его использования достигается лишь в совокупности с уникальными защитными механизмами, реализуемыми при помощи Guardant API.

Функционально утилиты автозащиты можно разделить следующим образом:

- Мастер автозащиты
- Утилиты защиты Native-приложений (в т.ч. профайлер Native)
- Утилиты защиты приложений .NET (в т.ч. профайлер .NET)

В рамках данного урока будет рассмотрен процесс защиты Win32 приложения без использования профайлера.

Используемые термины и обозначения

Вакцина – программный код, внедряемый в защищаемое приложение (внутренняя вакцина), а также исполняемый файл, поставляемый вместе с ним (внешняя вакцина), реализующие необходимый функционал для исполнения защищенного участка кода приложения в соответствии с заданными параметрами лицензирования.

Лицензия – сочетание параметров защиты приложений и данных, записываемых в ключ, определяющих возможности использования программного продукта конечным пользователем.

Лицензирование ПО – совокупность мероприятий, определяющих условия предоставления прав конечным пользователям на использование программного обеспечения путем наложения на него лицензионных ограничений. Целью лицензирования является извлечение материальной выгоды от продажи (распространения) и поддержки (обновления) программных продуктов.

Параметры лицензирования – множество параметров, определяющих процесс лицензирования. Их можно условно разделить на количественные (лицензионные ограничения) и качественные (степень защиты).

Уровень защиты (степень защиты)– сочетание параметров лицензирования, определяющих степень и характер привязки защищенного приложения к носителю лицензионной информации. Под носителем лицензионной информации понимается электронный ключ, в который записывается лицензия.

Возможности автоматической защиты

Автозащита имеет несколько режимов, позволяющих настроить процесс защиты, а также способ привязки защищаемого приложения к электронному ключу, частоту и характер производимых проверок и возвращаемых сообщений в случае неудачного завершения проверок.

Конечной целью является ограничение числа запусков или времени работы защищенного приложения и защита приложения от анализа и отладки.

Возможности автоматической защиты, в общем случае, можно классифицировать следующим образом:

- Схема лицензирования приложения
  - Возможность привязки к одному ключу любого количества защищенных приложений с независимыми друг от друга лицензиями
  - Наличие различных режимов лицензирования по локальной сети
  - Ограничение работы защищенного приложения:
    - По времени использования (для Guardant Time)
    - По количеству запусков (для всех типов ключей)
    - С использованием периодических проверок наличия ключа
    - С использованием принудительного завершения работы приложения через заданный интервал времени после обнаружения нарушения

- Способы привязки приложения к ключу:
  - К статическим данным ключа
  - С использованием алгоритмов ключа

- Защита приложения использует:
  - Шифрование кода и данных приложения
  - Технологию псевдокода (противодействие статическому и динамическому анализу)
  - Контроль целостности приложения

- Режимы работы автоматической защиты приложений:
  - с записью созданной лицензии в ключ
  - на основе ранее записанных в ключ данных
  - без привязки к электронному ключу

Последний режим предполагает, что привязка к ключу полностью реализуется при помощи Guardant API и необходимость дублирования вызовов ключа отсутствует.

В рамках данного урока рассматривается вариант работы с Мастером автозащиты, подразумевающий запись создаваемой лицензии в ключ в процессе защиты приложения.

Принцип работы автоматической защиты

В процессе работы утилита автоматической защиты вписывает в тело исполняемого файла защищаемого приложения модуль – **внутреннюю вакцину**. В момент запуска приложения внутренняя вакцина вызывает расположенную в отдельном файле **внешнюю вакцину**, которая в свою очередь занимается исполнением защищенного кода приложения и проведением необходимых проверок и преобразований. Описанный процесс проиллюстрирован на рисунке 1. Общий принцип работы автозащиты .NET и Native приложений схож, несмотря на радикальные отличия самих технологий.

Схема 1. Принцип работы Автозащиты

Автоматическая защита Guardant поддерживает 32-хразрядные Windows-приложения и предназначена для обработки исполняемых файлов Native-приложений (*.exe), а также .NET-сборок (*.exe, *.dll).

Дополнительная защищенность внешней и внутренней вакцины обеспечивается технологиями **псевдокода** и **обфускации**, специально разработанными специалистами компании «Актив».

Мастер лицензирования оперирует понятием **лицензия**. Под лицензией понимается сочетание параметров защиты приложений и данных, записываемых в ключ. Лицензия автоматически создается Мастером в процессе защиты, избавляя разработчика от необходимости подробно изучать архитектуру и особенности программирования ключа.

Мастер позволяет создавать, тиражировать и обновлять (в том числе, удаленно) лицензии в электронном ключе.

Порядок работы с Мастером автозащиты может быть схематично представлен следующим образом:

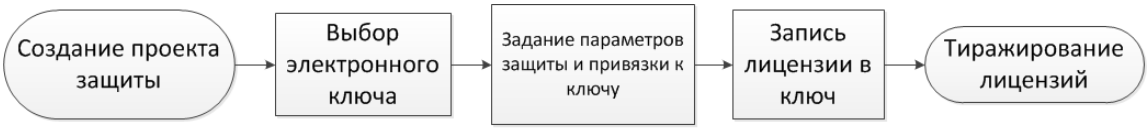


Схема2. Порядок работы с Мастером автозащиты

За наложением автозащиты следует **тиражирование лицензий,** включающее запись соответствующих проекту защиты лицензий в электронные ключи и окончательную подготовку копий защищенного приложения для их передачи конечным пользователям.

Использование автозащиты

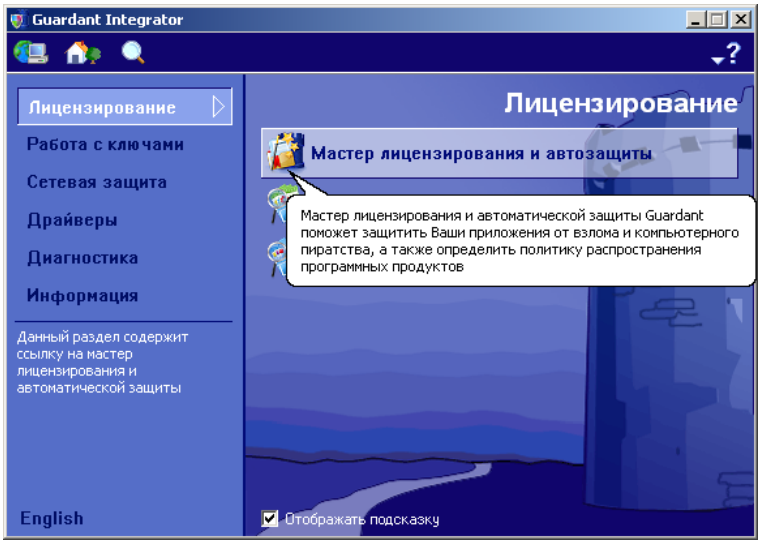
Для работы нам потребуется установленный Комплект разработчика Guardant и один электронный ключ любой модели.

Для наглядности будем защищать утилиту диагностики из комплекта драйверов Guardant **grddem32.exe**. После инсталляции Комплекта разработчика и драйверов она будет располагаться в директории *“%WinDir%\system32\”*.

Чтобы случайно не испортить рабочую копию утилиты, рекомендуется скопировать ее в какую-нибудь папку, например, *C:\GrdTEST*.

Шаг 1

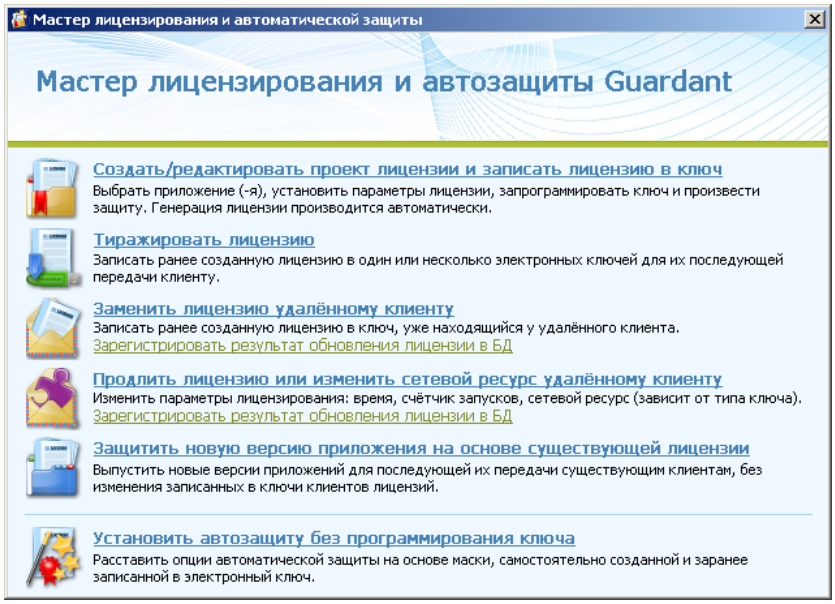
Из **Интегратора Guardant** запускаем Мастер лицензирования и автозащиты:



Шаг 2

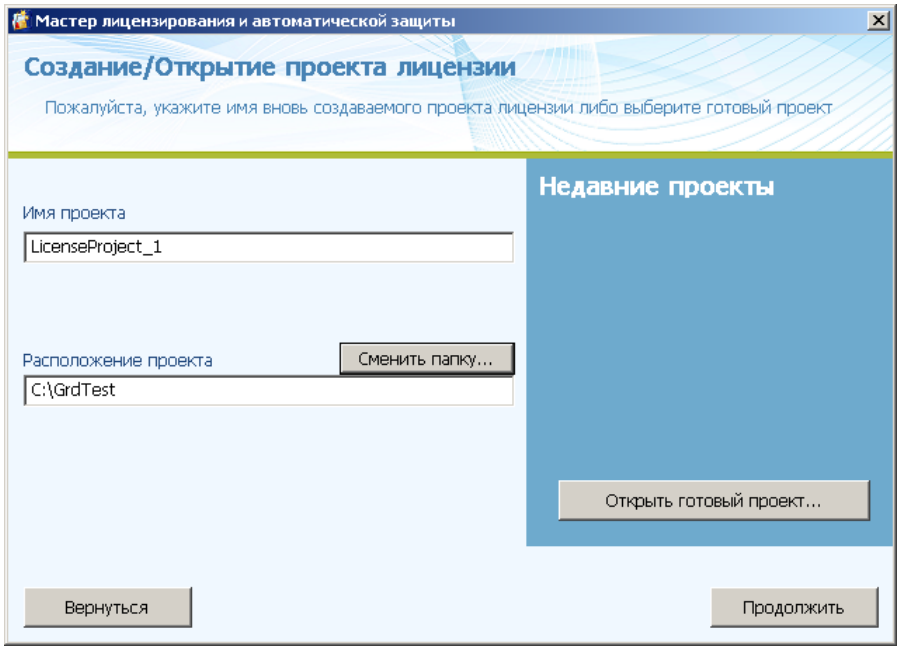
В главном окне Мастера представлены различные варианты защиты и лицензирования приложений, включая работу с лицензиями в ключах, находящихся у конечных пользователей.

На данном этапе нас будет интересовать пункт **Создать / редактировать проект лицензии и записать лицензию в ключ**. Он позволяет быстро и без необходимости изучения дополнительных технологий защитить приложение с заданными параметрами привязки к электронному ключу:



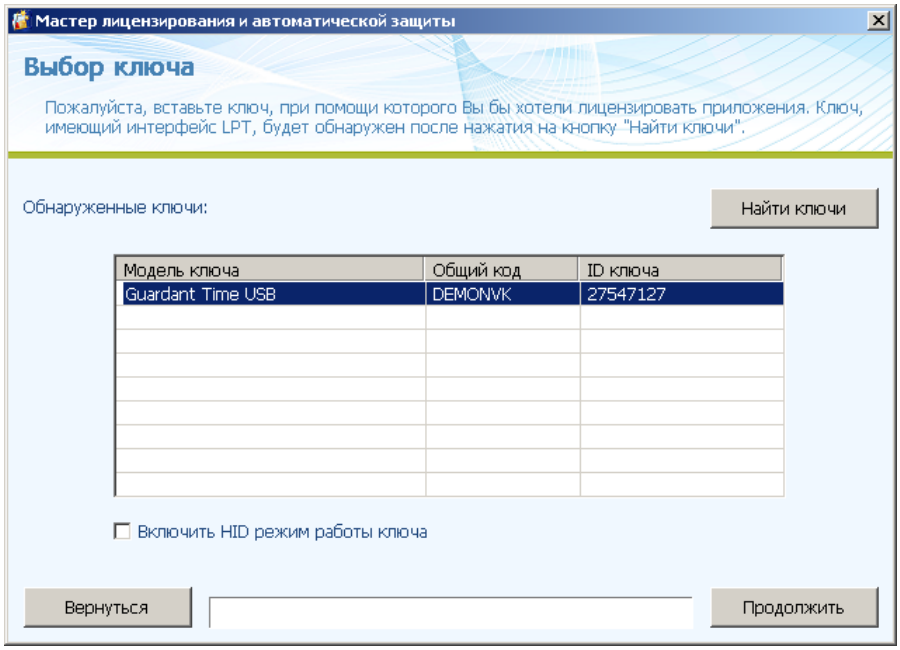
Шаг 3

Далее необходимо выбрать имя проекта и расположение рабочей директории:



Шаг 4

После подсоединения ключа Guardant Мастер автоматически его обнаруживает и отображает в списке:

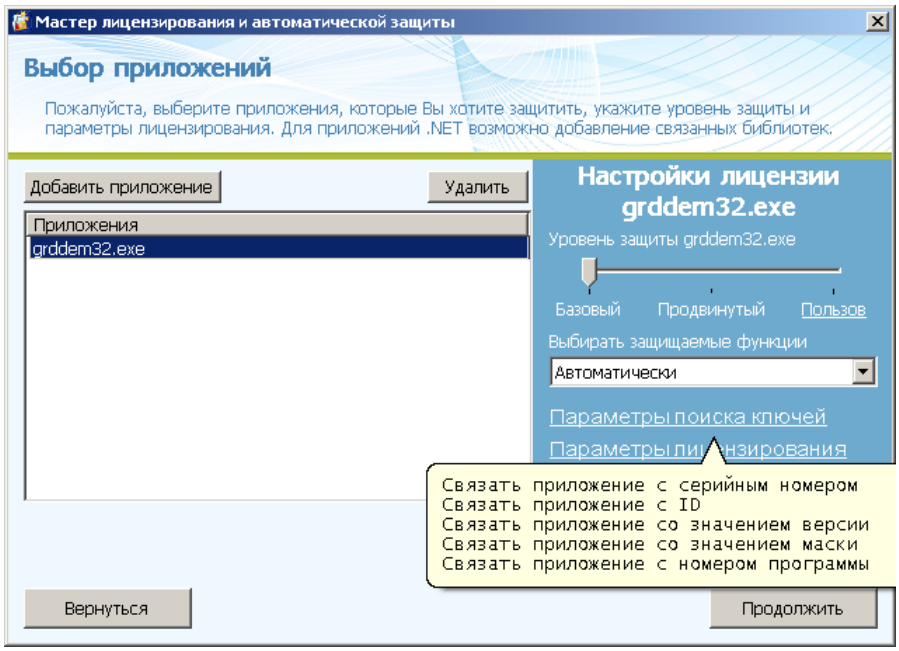


Если ключей обнаружено несколько, необходимо выбрать тот, на котором будет производиться защита.

На данном этапе можно также активировать HID-режим работы ключа, что позволит конечному пользователю использовать его без предварительной установки драйверов Guardant.

Шаг 5

Настало время выбрать приложения, подлежащие защите, и включить их в проект лицензии. Добавляем **grddem32.exe**:



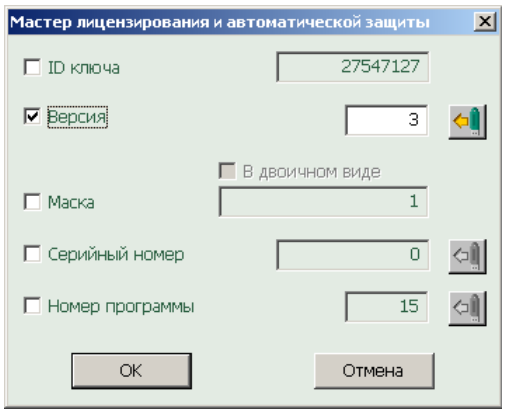
Шаг 6

Для каждого из внесенных в проект лицензии исполняемых файлов в отдельности можно задать параметры лицензирования, а также непосредственно процесса защиты приложения.

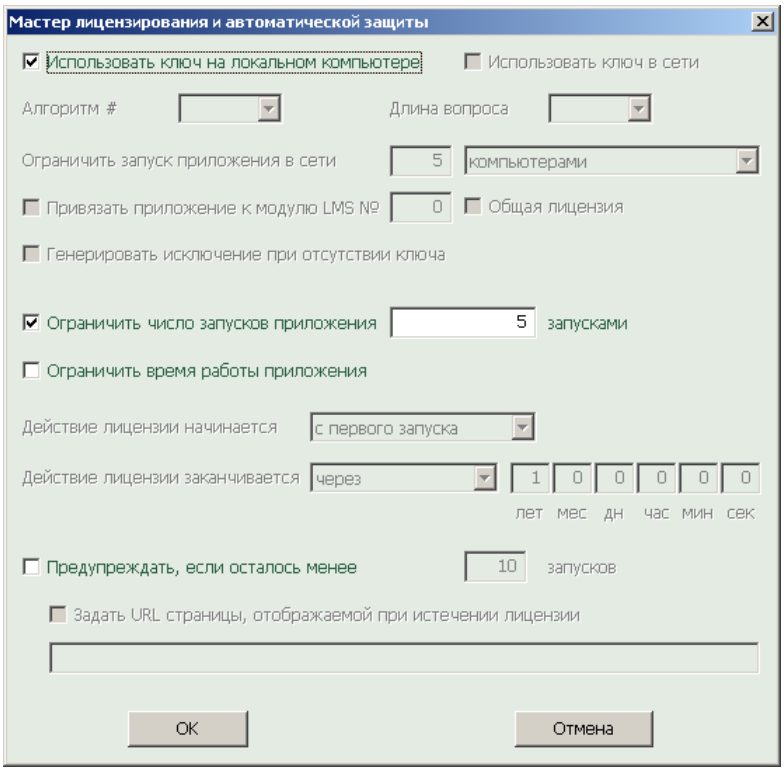
Предлагается выбрать один из профилей предустановленных настроек, установив **уровень защиты** в положение **Базовый** или **Продвинутый**, либо задать все параметры, самостоятельно, задав тем самым **Пользовательский** набор. Для нашего эксперимента вполне подойдет и **Базовый** вариант, однако при реальной защите приложения, скорее всего, возникнет необходимость в **Пользовательском**.

Теперь необходимо установить параметры привязки к ключу, функционально разделенные на три группы.

Первая группа параметров позволяет скорректировать **условия поиска ключа**. В качестве примера, установим в поле **Версия** значение 3. В результате в процессе программирования ключа в поле **Версия** будет записано число 3, а защищенное приложение будет контролировать значение этого поля и не запустится в случае, когда значение в ключе будет меньше 3:



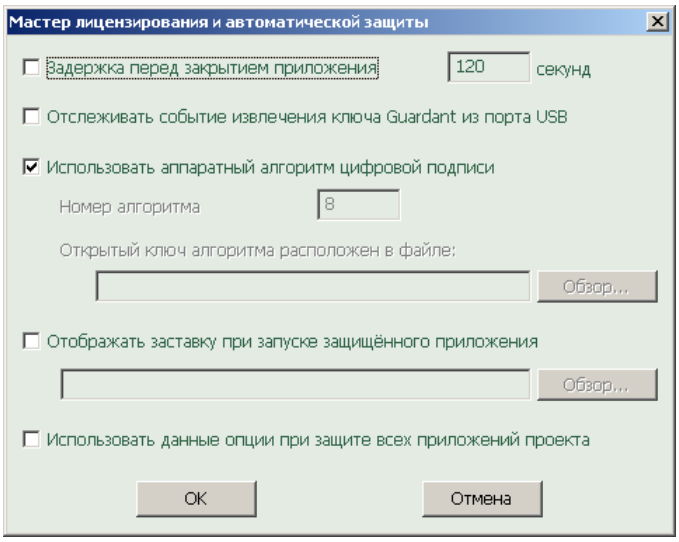
Вторая группа параметров позволяет установить **параметры лицензирования** защищаемого приложения.



Отметка **Использовать ключ на локальном компьютере** означает, что ключ не сетевой и защищенное приложение не будет пытаться обнаружить его по сети.

Установим ограничение на количество запусков защищенного приложения. Число 5 означает, что защищенное приложение можно будет запустить не более 5 раз.

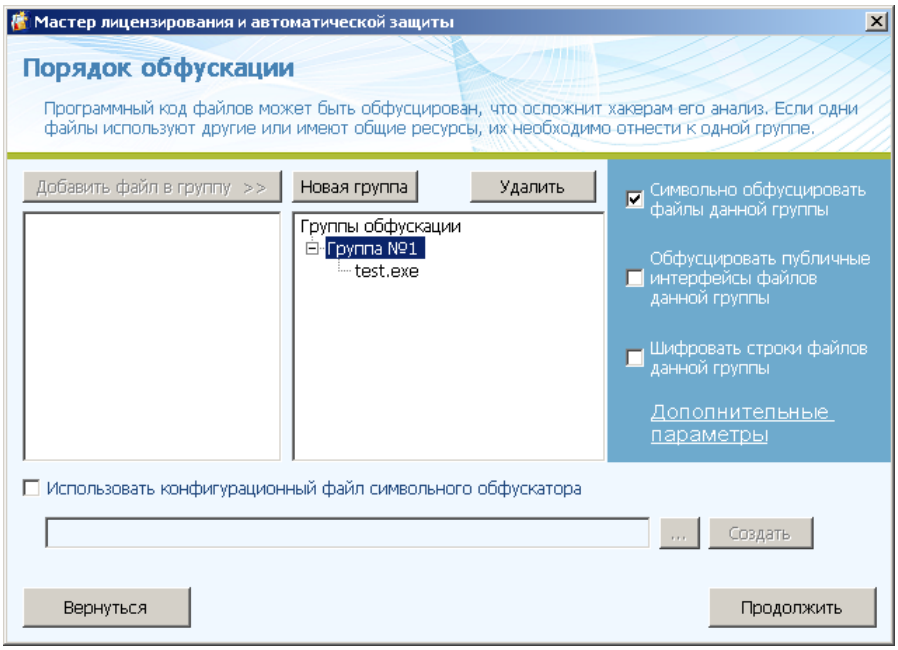
Последняя группа параметров содержит некоторые **дополнительные настройки** защиты приложения:



В целях повышения защищенности приложения будем использовать алгоритм цифровой подписи в составе процедуры периодического опроса ключа.

Шаг 7 (только для .NET приложений)

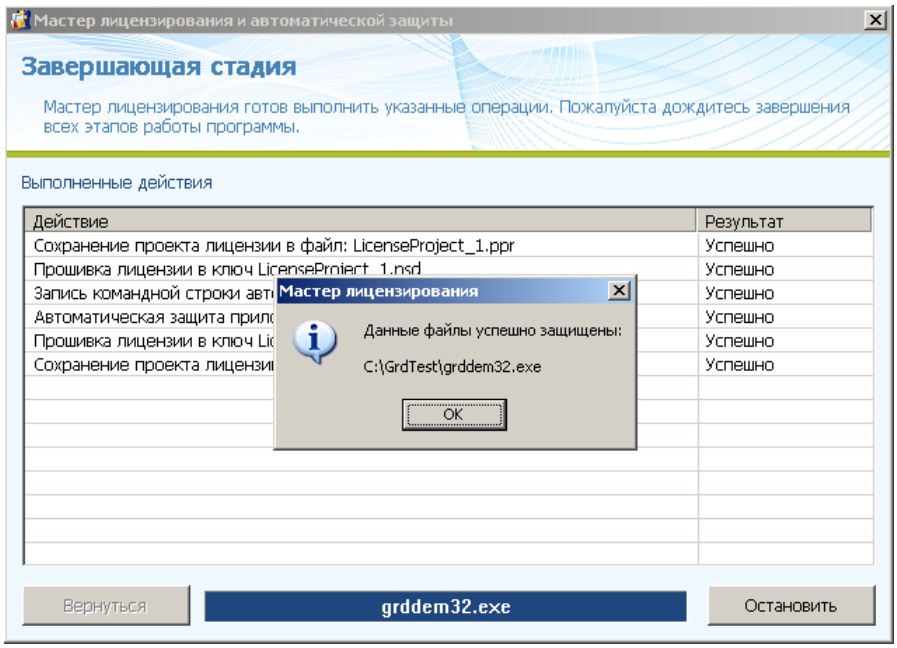
В случаях, когда защищается приложение **.NET** (**grddem32.exe** таковым не является) будет доступен следующий диалог мастера:



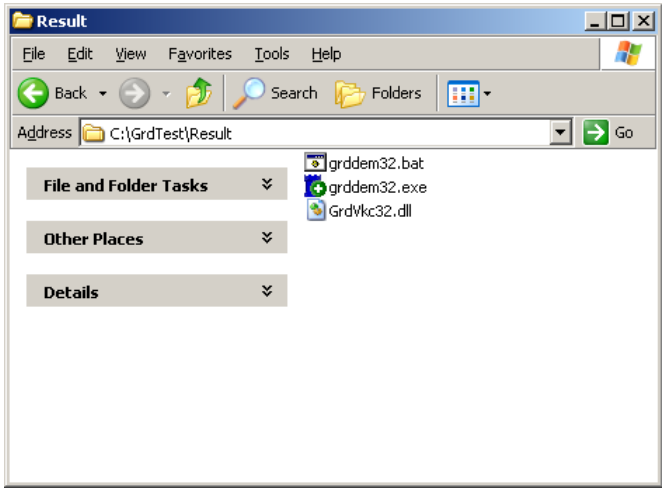
В нем предлагается установить **параметры обфускации** файлов и отдельных функций защищаемого приложения. При необходимости, можно разбить файлы приложения по разным группам обфускации и задать различные параметры обфускации для каждой группы.

Шаг 8

Приложение успешно защищено. В окне заключительной стадии работы Мастера сведены результаты выполняемых во время защиты процессов:



В результате получено защищенное приложение в комплекте с необходимой для работы библиотекой **GrdVkc32.dll** (т.н. **внешняя вакцина**).



На этом процесс защиты успешно завершен. Настало время проверить, как работает привязка защищенного приложения к ключу.

Примечание

1) В результате защиты создается **BAT-файл**, содержащий набор параметров командной строки автозащиты, который может быть использован для повторения процедуры защиты с теми же параметрами на том же электронном ключе (для защиты на другом ключе необходимо изменить значение параметра **/SPEC_ID**).

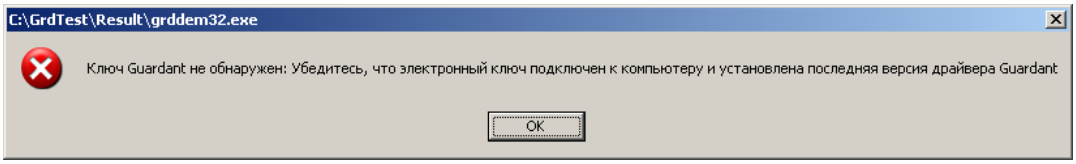
2) Информация о лицензии, записанной в ключ в процессе защиты приложения, автоматически сохраняется в базу данных (речь о которой также пойдет в одном из следующих уроков).

Проверка результатов

Опыт 1

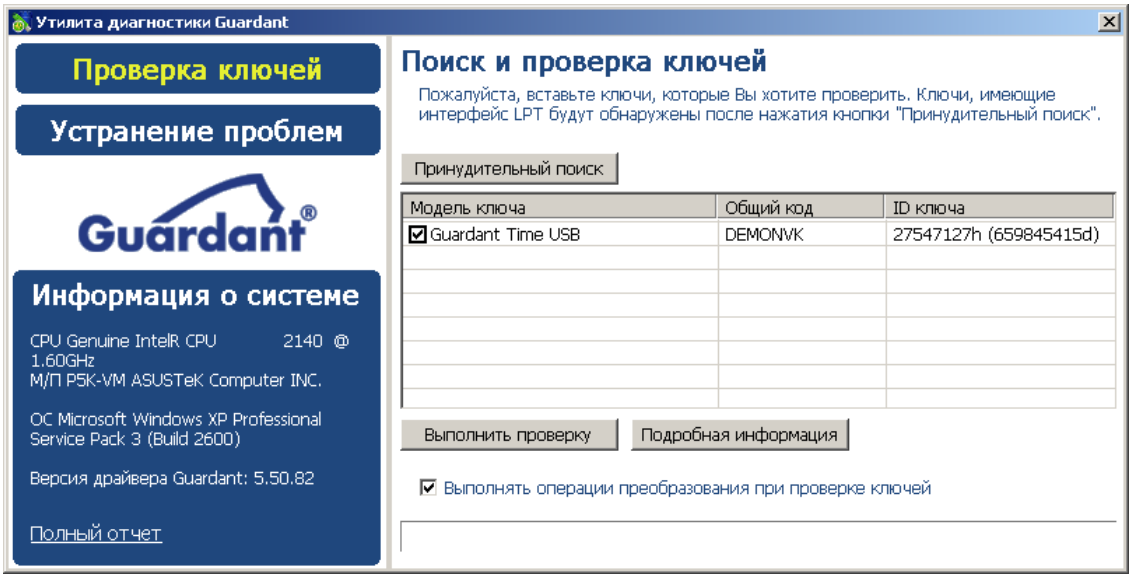
Отсоедините ключ от порта и запустите приложение без ключа (защищенное приложение находится в папке *C:\GRDTEST\Result)*.

Программа не запустится, сообщив об отсутствии электронного ключа Guardant.



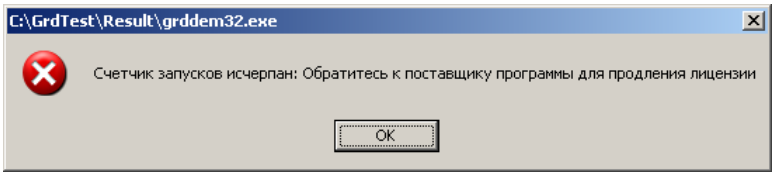
Опыт 2

Подсоедините ключ к порту и запустите приложение. Оно должно запуститься и обнаружить ключ:



Опыт 3

Запустите защищенное приложение еще несколько раз. После 4-й попытки должен исчерпаться счетчик запусков, и приложение не запустится с сообщением об ошибке:



Заключение

В рамках данного урока был рассмотрен порядок работы с Мастером лицензирования и автозащиты на примере защиты Native приложения платформы Win32. Наложение автозащиты приложений .NET происходит по аналогичному сценарию. Отличие составляет лишь установка некоторых опций (в силу радикального различия самих платформ и, соответственно, технологий Guardant, используемых для защиты приложений).

Очевидная простота в эксплуатации автоматической защиты приложений может привести к соблазну использования ее в качестве основного механизма защиты.

Следует, однако, понимать, что использование даже таких мощных средств автоматизации защиты, в качестве единственного (или основного) метода защиты допустимо только в одной и следующих ситуаций:

- Когда исходный код защищаемого приложения недоступен
- В случае недостатка ресурсов для разработки собственной защиты при помощи Guardant API
- При необходимости защиты неосновных модулей приложения, анализ которых потенциально может сообщить дополнительную информацию для проведения успешной атаки на механизмы защиты основных компонентов

При реализации защиты необходимо придерживаться принципа ее равнопрочности. Автозащита в этом случае может играть важную, но лишь второстепенную роль, и предназначена для защиты приложений от анализа и дизассемблирования.

Дополнительные источники информации

При возникновении вопросов, на которые вам не удалось найти ответа в этом пособии, рекомендуем обратиться к следующим дополнительным источникам информации:

WWW: http://www.guardant.ru

Web-сайт разработчика содержит большой объем справочной информации об электронных ключах Guardant.

Служба технической поддержки:

e-mail: hotline@guardant.ru

тел. +7(495)925-77-90