Защита 1С-конфигураций

Для того чтобы приступить к защите программного обеспечения на платформе 1С:Предприятие необходимо:

- Скачать набор инструментов для лицензирования Guardant SLK
- Скачать утилиту защиты WA:

Порядок действий для защиты конфигурации

Подготовка конфигурации

1. Определение списка модулей для защиты.
   Защитить можно общие модули, модули отчетов и обработок, а также модули менеджеров объектов. Названия модулей не должны содержать синтаксических ошибок и директив процессора. Также не допускается использование методов глобального контекста "Выполнить" и "Вычислить". Недопустима установка проверки ключа для функций, не содержащих ни одной переменной (будет вызвано исключение "Функция слишком простая для защиты").
2. Установка пароля на защищаемые модули.
   Выберите пункт меню: **Текст** – **Установить пароль**.
   Это необходимо для компиляции исходного текста в опкод.
   Можно установить любой пароль, только его необходимо запомнить.

   В процессе установки защиты исходный текст модуля из результирующей конфигурации будет удалён.
3. Сохранение библиотеки защиты.
   Это необходимо для функционирования системы защиты.
   Образ библиотеки рекомендуется сохранить в макете типа "Двоичные данные".

   Установить систему защиты можно только при использовании операционной системы Windows.
4. Создание функции *GetUSBKeyComponentLocationV3*.

   Функция GetUSBKeyComponentLocationV3 не нужна, если не требуется процедура проверки ключа.

   Эта функция должна возвращать полное имя макета, в котором хранится образ библиотеки защиты, или полное имя файла библиотеки защиты.
   Также можно использовать в макете zip архив с файлами библиотек и файлом манифеста.
   Функция *GetUSBKeyComponentLocationV3* вызывается однократно перед загрузкой библиотеки защиты.

Пример реализации функции GetUSBKeyComponentLocationV3

```
// GetUSBKeyComponentLocationV3
//
// :
//    ,
//
 GetUSBKeyComponentLocationV3()

     = ..;

     = ?( = , ..("V3_Hasp_zip"), );
     = ?( = , ..("V3_Guardant_zip"), );

      =
         =  ;

         . = .Windows_x86
             = "Win32";
         . = .Windows_x86_64
             = "Win64";
         . = .Linux_x86
             = "Linux32";
         . = .Linux_x86_64
             = "Linux64";
        ;
         = ?( = , ..("V3_Hasp_" + ), );
         = ?( = , ..("V3_Guardant_" + ), );
   ;

      =
        ("      ");
   ;

     .();
```

При необходимости в защищаемом модуле может быть размещена функция *GetUSBKeyObjectNative(ModuleID)*. При интеграции проверки ключа в защищаемый модуль содержимое этой функции будет замещено таким образом, что функция будет возвращать используемый объект компонента. Данную функциональность можно использовать, например, для получения информации о статусе лицензии.
Если значение параметра *ModuleID* неопределенно, то проверка лицензии не будет выполняться, а будет инициализирован объект компонента. Этот объект может использоваться для вызова методов GetInfo и Update.

Пример реализации функции GetUSBKeyObjectNative(ModuleID) для отладки

```
// GetUSBKeyObjectNative
//
//
//    ModuleID    -
//
// :
//
//
//
 GetUSBKeyObjectNative(ModuleID)
     USBKeyObject, USBKeyComponentFileName, USBKeyComponentName;

    USBKeyComponentName = "AddIn.HASP_DEMOMA.USBKeyLibV3";

        USBKeyObject = (USBKeyComponentName, );

        USBKeyComponentFileName = GetUSBKeyComponentLocationV3();
          AttachAddIn(USBKeyComponentFileName, "HASP_DEMOMA", AddInType.Native)
             "   " + USBKeyComponentFileName;
        ;
        USBKeyObject = (USBKeyComponentName, );
    ;

     ModuleID <>

        Scope = GetUSBKeyScope(ModuleID);
          USBKeyObject.Login(ModuleID, Scope)
            (USBKeyObject.GetLastError());
        ;
    ;

     USBKeyObject;
 //GetUSBKeyObjectNative
```

Пример использования функции GetUSBKeyObjectNative(ModuleID)

```
//
//
//
//    ModuleID    -
//
// :
//
//
//        XML.
//
//:
//  "GetSessionInfo"        HASP
//
 (ModuleID)
     = GetUSBKeyObjectNative(ModuleID);
     .GetSessionInfo("<haspformat format=""sessioninfo""/>");
```

При необходимости в защищаемом модуле может быть размещена функция *GetUSBKeyScope(ModuleID)*. При наличии эта функция будет вызвана в процессе инициализации объекта лицензирования. Если параметры поиска не требуются, то она должна вернуть параметры поиска ключа в виде строки "Неопределенно".

Пример использования функции GetUSBKeyScope(ModuleID)

```
// GetUSBKeyScope
//
//
//    ModuleID    -
//
// :
//
//
//
//
//:
//  "GetInfo"        HASP
//
//
 GetUSBKeyScope(ModuleID)
     ;
    KeyID = .KeyID.();
     (KeyID)
         = "<?xml version=""1.0"" encoding=""UTF-8"" ?>
            |<haspscope>
            |    <hasp id=""" + KeyID + """ />
            |</haspscope>";
    ;
     ;
```

5. Сохранение конфигурации.
   Выберите пункт меню: **Конфигурация – Сохранить конфигурацию в файл...**
6. Запуск конфигурации.
   Запустите конфигурацию. При запуске открывается форма для установки защиты.

Использование формы защиты конфигурации

1. В поле **Имя исходного файла** введите полный путь к ранее сохраненной конфигурации для защиты.
2. В поле **Имя файла библиотеки защиты** введите полное имя файла используемой библиотеки защиты.

   - При установке защиты требуется наличие установленного ключа защиты той серии, под которой выполняется защита.
   - Если используется модульное лицензирование, то в ключе должны быть доступны эти модули.
3. Нажмите кнопку **Анализ файла конфигурации**. После завершения процедуры будет построено дерево модулей, которые можно защитить. Также отобразится информация о режиме хранения сессий ключа для различных типов модулей. Если для конкретного модуля хранения сессии будет выбран отличный от режима тип модуля, то эта информация отобразится в столбце **Хранение сессий ключа**.

   Информацию о режиме хранения сессии нельзя редактировать.

   Варианты хранения сессий:
   - в переменной: лицензия хранится в переменной модуля, на каждый модуль создаётся отдельное хранилище сессий ключа. Этот режим используется для защиты отчетов и обработок.
   - комбинированное хранение: используется единое в рамках сеанса хранилище сессий. В зависимости от текущего режима работы модуля автоматически выбирается хранение сессий ключа, либо в переменной модуля, либо с использованием модуля с признаком "Повторное использование возвращаемых значений", установленным в значении "На время сеанса". Также у модуля, который используется для хранения сессий должны быть установлены следующие свойства: Сервер – Да, Клиент – Нет, ВнешнееСоединение – Нет.
4. Если необходимо проверить ключ в модулях, у которых определён признак хранения сессий ключа, как "Комбинированный", то требуется указать имя общего модуля, который будет использоваться для хранения сессий ключа.
5. Укажите функции, при вызове которых должен проверяться ключ защиты. Для выбранных функций необходимо указать номер (ID) компонента. Разработчик защищаемой конфигурации сам определяет набор и номера компонентов, при заведении продуктов в системе лицензирования Guardant Station.
6. При необходимости отключите учет лицензий по каждому модулю отдельно.
   Возможные случаи применения:
   - учет лицензий только по базовой поставке продукта, для дополнительных модулей контролируется только наличие ключа без подсчета количества мест;
   - учет количества лицензий вообще не выполняется, а контролируется только лишь наличие ключа.

7. Установите защиту путём модификации конфигурации. Для этого нажмите кнопку **Установить защиту**. Изменённая конфигурация сохраняется в том же каталоге, в котором расположен исходный файл и имя аналогичное исходному файлу с добавлением текущих значений даты и времени. В окне сообщений выводится полный путь к созданному защищённому файлу конфигурации.
   В процессе установки защиты системы могут отображаться различные информационные сообщения.

Установка защиты в пакетном режиме

Для установки защиты на конфигурацию в пакетном режиме предназначен скрипт *DoProtect3.js*, который расположен в каталоге шаблонов, куда была установлена поставка "Защита конфигураций 3".

При запуске скрипта без параметров, или если не все обязательные параметры указаны, выводится окно с описанием всех параметров и со значениями уже установленных параметров.

При указании строки соединения базы "Защита конфигураций 3" недопустимо указание пробелов и кавычек внутри строки соединения. При наличии пробелов в параметре, параметр должен быть заключен в кавычки.

Параметры должны быть указаны без пробела между ключом и значением параметра, например:

```
DoProtect3.js -LF"R:\ .txt" -CSSrvr=srv:3541;Ref=CodeGuard83; -SRR:\1Cv8.cf ...
```

Использование защищенной конфигурации

Для возможности использования защищенной конфигурации у пользователя должен быть в наличии соответствующий ключ защиты и установлен Guardant Control Center на машине с ключом. Система защиты конфигураций не имеет каких-либо дополнительных настроек.

При отсутствии лицензии на выполнение защищенной функции при выполнении такой функции будет сгенерировано исключение с описанием ошибки.

В случае если проверка ключа не интегрировалась, поведение алгоритмов защищаемой конфигурации не меняется.

Особенности лицензирования сессий

Если выполнение модуля происходит в контексте сервера программы 1С: Предприятие, то должны использоваться только сетевые лицензии в режиме распределения (конкуренции) по подключениям – в противном случае учет лицензий будет не корректен.
Если выполнение модуля происходит в контексте толстого клиента, можно использовать режим распределения (конкуренции) по рабочим станциям.
В общем случае учет лицензий осуществляется для каждого экземпляра модуля (т.е. для каждого сеанса – отдельная лицензия). При этом, если несколько модулей конфигурации одновременно используют один и тот же Feature ID – для каждого защищенного модуля будет выделена отдельная лицензия. Таким образом, не стоит защищать различные модули конфигурации, используя один Feature ID.