Защита Guardant для ОС GNU Linux

Современные ключи Guardant Sign/Time и Guardant Code, позволяют защищать приложения, запускаемые в ОС GNU/Linux на аппаратных платформах i386 и x86_64. Для этого в комплект разработчика включены статические (**libgrdapi.a**) и динамические библиотеки (**libgrdapi.so**) соответствующей разрядности, которые реализуют функционал Guardant API.

Защита Native-приложений GNU/Linux

Для сборки защищаемого приложения, необходимо слинковать защищаемое приложение с библиотекой Guardant API.
Рекомендуется использовать компилятор GCC 4-ой версии, однако, возможно использовать и более ранние версии GCC, и другие компиляторы, например, Intel C++ Compiler (ICC).
Для компиляции с библиотекой Guardant API необходимо выполнить следующее (на примере файла с исходным текстом программы — foobar.c):

```
$ gcc [-I<____GrdAPI.h>] -c foobar.c -o foobar.o
$ gcc [-L<___libgrdapi.a>] foobar.o -o foobar -lpthread -lgrdapi
```

или

```
$ gcc [-I<____GrdAPI.h>][-L<___libgrdapi.a>] foobar.c -o foobar-lgrdapi -lpthread
```

Обратите внимание, что библиотека Guardant API использует библиотеку **pthread** - POSIX Threads, поэтому для сборки приложений необходимо использовать соответствующую библиотеку.

Установка ключей Guardant в операционных системах GNU/Linux

Ключи Guardant работают в ОС GNU/Linux (в том числе и в HID-режиме) без установки дополнительных драйверов и демонов, требуя лишь обеспечить имя и разрешение доступа к файлу устройства. Для обращения к ключу используются соответственно Linux USB Device Filesystem или Linux USB HID Device Interface (в случае HID-режима).

Для работы с ключами в ОС GNU/Linux необходимо добавить правило для штатного средства обработки HotPlugging. На большинстве современных дистрибутивов, таким средством является udev (https://ru.wikipedia.org/wiki/Udev). В комплект разработчика включен набор правил для udev (архив **udev-rules.tar.gz**). Указанные правила предписывает udev установить права на чтение и запись для файла устройства, представляющего электронный ключ Guardant в системе. Это позволяет обращаться к ключу с привилегиями любого пользователя системы.

> **Внимание!**
>
> Для случаев нетипичного конфигурирования устройств, обратитесь к разделу **имена и доступ к устройствам ▼**

Установка правил для udev

Для ключей как в драйверном, так и в HID-режиме, и в случае использования файлов-устройств USB Device Filesystem.

Скачать архив *udev-rules.tar.gz* , распаковать и выполнить установку правил:

```
$ tar -xvf udev-rules.tar.gz
$ ./install.sh
```

После успешной установки правил нужно отсоединить электронный ключ от USB-порта и подсоединить повторно.

Ключ готов к работе с защищенным Linux-приложением.

Имена и доступ к устройствам

Для ключей, работающих в драйверном режиме

Обращение к ключу происходит через Linux USB Device Filesystem. Подробную информацию см. в файле **linux/Documentation/usb/proc_usb_info.txt** из документации к Linux. Для успешной работы с ключом в системе нужно разрешить доступ на чтение/запись к файлу устройства.

Для ключей, работающих в HID-режиме

Обращение к ключу происходит через Linux USB HID Device Interface (драйвер **usbhid**). Подробную информацию см. в файле **linux/Documentation/usb/hiddev.txt** из документации к Linux. Для успешной работы с ключом в системе нужно изменить имена соответствующих устройств на **/dev/grdhidN** и разрешить доступ на чтение/запись к файлу устройства.

Переменные окружения

Для настройки Guardant API под GNU/Linux следует пользоваться следующими переменными окружения:

GRD_IPC_NAME	директория, в которой процессы будут создавать/открывать для чтения и записи файлы, используемые для синхронизации доступа к ключу. Если переменная не задана, используется значение по умолчанию (**/tmp**)
USB_DEVFS_PATH	директория **LinuxUSBDeviceFilesystem** (точка монтирования или директория, содержащая дерево соответствующих устройств). Если переменная не задана, будет использоваться **/dev/bus/usb** (если существует), иначе – **/proc/bus/usb**

Запуск защищенных Windows-приложений в среде Wine

Для работы приложений Windows, защищённых ключами Guardant Sign/Time и Guardant Code под Wine (www.winehq.org), необходима библиотека **grdwine.dll.so**. Для этого в комплект разработчика включен проект библиотеки для Wine — **grdwine**, распространяемый под свободной лицензией GNU Lesser General Public License version 2.1 (поставляется в двух вариантах - в виде скомпилированных библиотек **grdwine-0.5.7-bin.tar.gz** и в виде пакета с исходными текстами – **grdwine-0.5.7.tar.gz**).

> **Важно!**
> Рекомендуемая к использованию версия Wine — 1.x.x. Корректная работа с более ранними версиями Wine не гарантируется. Загрузить последнюю версию Wine можно по адресу: https://www.winehq.org/download

> **Важно!**
> Библиотека предназначена только для работы с современными моделями ключей Guardant. С моделями линейки Stealth II и Stealth III эта библиотека **не может** быть использована.

Имеется **два варианта** установки библиотеки **grdwine.dll.so** под Wine: установка скомпилированных библиотек из **grdwine-0.5.7-bin.tar.gz** и компиляция библиотек из исходных текстов и их последующая установка из **grdwine-0.5.7.tar.gz**.

Установка скомпилированных библиотек

> **Важно!**
> Для установки скомпилированных библиотек требуется наличие заранее установленного wine.

1. Скачать архив с бинарными файлами библиотек *grdwine-0.5.7-bin.tar.gz*

2. Распаковать архив *grdwine-0.5.7-bin.tar.gz* и перейти в распакованную директорию

```
:
$ tar -xvf grdwine-0.5.7-bin.tar.gz
$ cd grdwine-0.5.7-bin
```

3. Перейти в распакованную директорию и запустить скрипт *install.sh* с правами администратора

После того как библиотека **grdwine.dll.so** распакована, необходимо перенести ее версию нужной разрядности в директорию с защищенным приложением ("x86" необходима для работы 32-битных *windows*-приложений, а "x86_64" - для работы 64-битных *windows*-приложений соответственно).
Затем файл библиотеки необходимо переименовать изменив расширение с **grdwine.dll.so** на **grdwine.dll**.

```
:
$ cp ./grdwine-0.5.7-bin/x86/grdwine.dll.so /home/user/samples/
/home/user/samples/ -
$ mv /home/user/samples/grdwine.dll.so /home/user/samples/grdwine.dll
```

Компиляция библиотек из исходных текстов и их последующая установка

1. Установка зависимостей на примере Ubuntu

Для сборки библиотеки на Ubuntu 16.04 требуется предварительно установить пакеты:

```
$ sudo apt-get install autoconf
$ sudo apt-get install wine
$ sudo apt-get install wine-dev
$ sudo apt-get install libc6-dev-i386 (   32-  )
```

Для Ubuntu 17.10 и 18.04 дополнительно установить:

```
$ sudo apt-get install wine64-tools (   64-  ,    wine32-tools)
$ sudo apt-get install wine32-tools (   32-  ,    wine64-tools)
```

2. Последовательность сборки библиотеки grdwine.dll.so (для 64-bit систем)

Скачать и распаковать пакет с исходными текстами *grdwine-0.5.7.tar.gz* :

```
$ tar -xvf grdwine-0.5.7.tar.gz
$ cd grdwine-0.5.7
```

Запустить ./bootstrap.sh

```
$ ./bootstrap.sh
```

Для компиляции и установки 32-разрядной библиотеки выполнить:

```
$ ./configure --with-wineincs=/usr/include --with-winedlls=/usr/lib/i386-linux-gnu/wine
$ make
# sudo make install
```

Для компиляции и установки 64-разрядной библиотеки выполнить:

```
$ ./configure --enable-win64 --with-wineincs=/usr/include --with-winedlls=/usr/lib/x86_64-linux-gnu/wine
$ make
# sudo make install
```

Важно!
Указанные в примере пути к заголовочным файлам и библиотекам Wine (опции **--with-wineincs** и **--with-winedlls**) могут меняться в зависимости от версии Wine, используемого дистрибутива Linux или
заданного префикса для установки (в случае, если Wine устанавливался из исходных кодов)

Важно!
Для запуска 32-разрядных приложений Windows в дистрибутивах Linux архитектуры x86_64 достаточно собрать только 32-разрядную библиотеку. Сборка 64-разрядной библиотеки требуется для запуска
64-разрядных приложений Windows.

Важно!

Если правила для udev не были установлены ранее, то их необходимо установить. Подсоедините ключ Guardant к USB-порту компьютера, защищенное приложение готово к работе.