

И.В. АНАНЧЕНКО, А.А. МУСАЕВ  
**ЗАЩИТА ПРИЛОЖЕНИЙ, ВЫПОЛНЯЕМЫХ ТОРГОВЫМ  
ТЕРМИНАЛОМ METATRADER, КЛЮЧАМИ SENTINEL HASP**

---

*Ананченко И.В., Мусаев А.А. Защита приложений, выполняемых торговым терминалом MetaTrader, ключами Sentinel Hasp.*

**Аннотация.** Рассмотрена задача защиты торговых приложений (советников), выполняемых под управлением программы MetaTrader 4, с использованием программных и аппаратных ключей серии Sentinel Hasp. Защита устанавливается на файл динамической библиотеки (dll), обращение к функциям которой осуществляется из исполняемого файла (ex4).

**Ключевые слова:** электронный рынок «Forex», защита приложений, метатрейдер, ключи Hasp.

*Anantchenko I.V., Musaev A.A. Securing applications running trading platform MetaTrader keys Sentinel Hasp.*

**Abstract.** The problem of protecting trade applications running under the MetaTrader 4, using the software and the hard-key series of martial Sentinel Hasp. Protection is set to file a dynamic library (dll), function calls that are from the executable file (ex4).

**Keywords:** Sentinel Hasp, Forex, protection programs, MetaTrader.

---

**1. Введение.** Программы для торговли на рынке Forex выполняющиеся на торговой платформе MT4 разрабатываются на языке программирования MQL 4 и выполняются после компиляции, как программы формата ex4 [1,2]. Файл структуры ex4 содержит только исполняемый код без исходного кода программы что, по замыслам разработчиков программной среды, является достаточной преградой для внесения изменений в файл, полученный в результате компиляции. Это относится, в том числе, и к изменению участков кода отвечающих за защиту программы от несанкционированного использования. Например, привязка программы к конкретному номеру счета или к дилинговому центру, предоставляющему услуги; возможность работать только с демосчетами (используя функцию проверки IsDemo) и т.д. Однако практика использования показала, что построить защиту только с использованием кода ex4 фала проблематично. Рассмотрим причины этого и варианты усиления защиты с использованием дополнительных программных и программно-аппаратных средств [3,4].

**2. Ограниченность защиты основанной только на использовании внутреннего программного кода ex4 файла.** В функции проверки init() сравнивается значение вычисленного производного значения от номера счета AccountNumber() и полученного ранее от продавца ключа (переменная Key), который был передан покупателю. Если зна-

чения равны, то функция возвращает ноль и программа работает дальше в штатном режиме. В противном случае программа засыпает “навечно” - Sleep(93700000).

```
#property copyright "mctrewards.ru" ....
int init() { ....
if (!(IsDemo() || IsTesting() || IsOptimization())) {
if (6.0 * (1.0 * (2.0 * (8.0 * AccountNumber() + 678.0) + 16.0)) !=
Key) {ls_0 = "Для торговли на реальном счете нужен лицензионный
ключ.";
ls_8 = "Вы можете получить ключ на сайте mctrewards.ru";
Print(ls_0); Print(ls_8); MessageBox(StringConcatenate(ls_0,
"\n", ls_8, "\n"), "Ошибка", MB_ICONHAND); Sleep(93700000);} ...
return (0);}
```

Установка рассмотренного программного кода защиты занимает минимум времени, но злоумышленнику очень легко взломать защиту, если он может восстановить файл с кодом ex4 к исходному виду mq4.

Практика показала, что создание декомпиляторов выполняющих обратное преобразование файлов из формата ex4 в mq4 возможно. Были созданы, в том числе, и коммерчески распространяемые, декомпиляторы. Например, до недавнего времени в сети Интернет можно было купить программу EX4-TO-MQ4 decompiler V4.0.427, позволяющую получать функционально идентичные исходному тексту программы тексты программ на языке MQL. Имена переменных в результате декомпиляции могут не совпадать с оригинальными. При восстановлении исходного кода возможно получение другой последовательности функционально идентичных операторов. Объясняется это тем, что одни и те же шаги алгоритма могут быть реализованы с использованием разных программных операторов. Однако полученный исходный код после его компиляции дает тот же исполняемый код формата ex4. В результате массового использования декомпиляторов, выполняющих преобразование из формата ex4 в исходный формат mq4, на рынке прикладного программного обеспечения сложилась ситуация позволяющая любому желающему с минимальными усилиями получать исходный текст программы. Данная ситуация совершенно недопустима для разработчиков и продавцов коммерческих программ для торговли на Forex, стоимость которых в среднем составляет от 150-300 и выше долларов США за один экземпляр программы [5].

### **3. Усиление защиты ex4 файлов путем переноса критически**

**важного для выполнения торговой программы кода в функции, вызываемые из динамической библиотеки.** В качестве меры противодействия многие разработчики специализированного программного обеспечения (ПО) применяют решение, основанное на переносе критически важного для реализации успешной торговли функционала в динамически загружаемую dll библиотеку, к функциям которой выполняется обращение из исполняемого ex4 файла. Одной из поддерживаемых функций dll библиотеки может быть функция проверки кода регистрации, который является уникальным для каждого пользовательского счета. Структура dll библиотеки, исходный код реализован средствами Delphi, может быть, например, такой:

```
library xyz; {uses SysUtils, Classes ; } {$R *.res}
function
key(a0:integer;a1:integer;a2:integer;a3:integer;a4:integer;a5:integer;a6:i
nteger):integer;
    stdcall; begin if (a2=a1*7-784351) then key:=0 else key:=1;
    end;
function
price1(a0:integer;a1:integer;a2:integer;a3:integer;a4:integer;a5:integer;a
6:integer):integer;
    stdcall; begin ..... end; ...
exports key,price1,btp1,stp1,maxorders; begin end.
```

В библиотеку входят пять функций: key,price1,btp1,stp1,maxorders. Функция key отвечает за реализацию функции проверки, остальные функции реализуют часть функционала торговой программы. Без файла библиотеки программа работать не будет, но это не мешает злоумышленнику использовать и распространять файл библиотеки вместе с вызывающим ее скомпилированным кодом ex4.

Для подбора кода регистрации злоумышленнику потребуется больше времени, так как необходимо корректно вызвать функцию проверки из библиотеки и подобрать нужный вариант ключа для заданного номера счета.

Отметим, что имея декомпилированный вариант mq4 злоумышленнику достаточно найти всего один регистрационный номер для одного счета, чтобы использовать программу на всех счетах. Пример:

```
#import "fxlibra.dll"
int MACD_Signal(int a0, int a1, double a2, double a3, double a4, dou
ble a5, double a6);
#import
extern int KEY = 368440261;
```

```

// int li_0 = AccountNumber();
int li_0 = 3034383;
int li_4 = KEY;
if (MACD_Signal(li_0, li_4, gd_464, gd_472, gd_480, gd_488,
gd_496) == 3) { Print(MACD_Signal(li_0, li_4, gd_464, gd_472, gd_480,
gd_488, gd_496)); Comment(" Expert key is not activate"); return (0);
... if (IsTradeTime() && MACD_Signal(li_0, li_4, gd_464, gd_472,
gd_480, gd_488, gd_496) == 1 && gi_504 == FALSE) {...

```

При рассмотрении фрагмента видно, что злоумышленник не может избавиться от библиотеки, так как функция не только осуществляет проверку правомерности использования программы, возвращая значение 3 для счета без лицензии (при неверно заданном значении переменной key), но и формирует сигнал на покупку или продажу, возвращая 1 или 2 для этих случаев. Злоумышленник, зная корректную пару: номер счета и код регистрации, подставляет их при вызове функции.

Использование декомпиляторов dll библиотек и средств анализа динамически исполняемого программного кода (например, совместное использование IDA v6.1 и HexRays v1.5) позволяет получить исходный программный код dll библиотеки. Восстановленный исходный код более сложен для анализа, но, фактически, это проблему не решает, так как встает вопрос о защите уже dll, составляющей ПО, от несанкционированного использования.

**4. Вариант решения задачи защиты dll библиотеки, содержащей критически важный для работы торговой программы исполняемый код, с использованием программных и программно-аппаратных средств Sentinel Hasp.** В качестве одного из возможных вариантов защиты dll библиотек предлагается использовать продукцию производителя SafeNet: аппаратные устройства HASP HL, подключаемые к USB порту компьютера или программный аналог - ключ HASP SL.

Существенный недостаток аппаратных ключей защиты – их относительно высокая стоимость (от \$15 и выше за один ключ) и необходимость физической доставки ключа покупателю. SafeNet, в рамках серии HASP SRM, предлагает решение, позволяющее работать с двумя типами ключей: программными (HL) и аппаратными (SL). Программные ключи по своей сути – аналоги аппаратных ключей, оформленные в виде файлов. Программный код внутри аппаратного ключа выполняется специализированным процессором. Этот же алгоритм может быть

реализован под управлением процессора компьютера, на котором выполняется защищаемая программа.

Одно из достоинств ключей HASP SL в том, что не требуется затрат на физическое изготовление ключа, поэтому стоимость программного ключа значительно ниже стоимости аппаратного (от \$5 за экземпляр и ниже). Фактически стоимость программного ключа – это затраты на разработку программного обеспечения системы защиты.

Лицензию для ключа HASP SL можно практически мгновенно доставить покупателю по каналам глобальной сети Интернет. Защита, построенная с использованием программного ключа HASP SL, подобна защите, реализованной с помощью аппаратного ключа HASP HL. Для установки обоих типов защит используется один и тот же программный инструментарий.

SafeNet предлагает в комплекте разработчика инструментарий, не требующий навыков программирования и реализующий так называемый вариант навесной защиты. Навесная защита устанавливается на готовый программный продукт с помощью утилиты Sentinel HASP Envelope. Альтернативным является вариант встраиваемой защиты, позволяющий максимально гибко использовать весь потенциал средств защиты, так как программная разработка защиты выполняется на уровне исходного кода.

Для установки встроенной защиты необходимо не только обладать навыками программирования и иметь исходный код программы, но для данного языка программирования должна быть специальный программный инструментарий, предлагаемый SafeNet. Функции для построения защиты программ Sentinel HASP API четко структурированы, что позволяет разработчику не тратить дополнительно время на подробное изучение реализации интерфейса, и дает возможность сразу приступить к построению защиты программного обеспечения. Например, можно использовать следующий фрагмент:

```
feature:= 15; status := hasp_login(feature, @vendor_code[1],  
handle);  
if status = HASP_STATUS_OK then begin Showmessage('Проверка  
подлинности ключа завершена успешно!'); end else begin  
Showmessage('Ошибка проверки подлинности ключа'); Form1.Close;  
Exit; end;
```

В приведенном примере *feature:=15* – определяет ID, соответствующий защищаемой программе (на одном ключе может храниться информация о нескольких защищенных программах с разными иден-

тификаторами ID), @vendor\_code – информация о коде разработчика, для каждой серии ключей код свой.

Алгоритм установки программной защиты на основе ключа HASP SL следующий:

1. Подготавливается dll библиотека, содержащая функции, реализующие критически важный код исполнения торговой стратегии. Защищаем dll библиотеку, используя Sentinel HASP Envelope и/или Sentinel HASP API, указав ID.

2. Используя Business Studio определяем ID. После этого определяем программный продукт (используя пункт меню New в Manage Products).

3. Для продукта в Business Studio определяем, будет ли он работать и с HASP SL и HASP HL ключами или только с HASP SL.

4. Подготавливаем демоверсию продукта (используя New Provisional в Manage Products).

5. Демоверсия программы поставляется клиенту и устанавливается им. На компьютере клиента устанавливается HASP SL ключ с ограниченной лицензией (можно установить срок от 0 до 90 дней).

6. Клиент опробует программный продукт и принимает решение о его покупке. Клиент использует специальное программное обеспечение регистрации RUS, отвечающее за возможность удаленного обновления функций ключа защиты. Программа RUS формируется слепок с информацией о параметрах компьютера клиента. Этот файл (с расширением s2v) клиент передается продавцу.

8. Получив файл, продавец обрабатывает файл, сгенерированный ответный файл формата v2c передает покупателю.

9. Покупатель, используя RUS, принимает полученный файл и в ключ HASP SL прописывается постоянная лицензия.

Передаваемый пользователю вместе с покупаемым комплектом программного обеспечения файл dll библиотеки на шаге 1 должен быть обработан утилитой HASP SRM Envelope шифрующей библиотеку dll с использованием алгоритма AES, после чего зашифрованный код помещен в защищенную оболочку-кокон, которая вызывается при обращении к dll файлу. Оболочка проверяет наличие программного SL или аппаратного ключа HASP HL. Если ключ присутствует и записанная в ключ лицензия действующая, то реализуется расшифровка загруженного в оперативную память dll кода и его выполнение. Использование ключа HASP HL Time, включающего таймер реального времени встроенный в ключ, позволяет реализовывать схемы продаж ПО на основе схем лизинга или аренды.

Стоимость аппаратных ключей HASP HL зависит от типа и условий закупки устройств. Расходы на аппаратные ключи вполне приемлемы в общей сумме накладных расходов на производство дорогостоящего ПО для торговли на Forex, особенно для программ стоимостью выше \$350.

Использование программного ключа HASP SL позволяет минимизировать расходы на защиту для более дешевого ПО (\$50-\$100), так как стоимость лицензии для работы с ключом составляет порядка 220 руб. на лицензию при закупке партии от 10 лицензий (<http://protect-online.ru/price>).

**Заключение.** Сравнение предлагаемой схемы использования аппаратной защиты показывает, что ключи HASP HL обеспечивают защиту не хуже, чем их аппаратные аналоги от конкурирующих фирм (например, ключи серии GUARDANT), но процесс установки и использования программы с защитой более удобен для конечного пользователя ПО. Защита на основе HASP SL показала хорошие результаты в сравнении с программными средствами защиты от других разработчиков (например, Enigma Protector 3.80). Предлагаемая схема защиты на ключах HASP HL и HASP SL успешно реализована в коммерческих торговых роботах Crisis v 1.2 ([forex-crisis.com](http://forex-crisis.com)), Gladiator и др. [4,6].

### Литература

1. *Мусаев А.А.* Методы построения робастифицированных систем анализа торговых ситуаций // Труды СПИИРАН. Вып. 14. 2010. С. 187–215.
2. *Мусаев А.А.* Корреляционный анализ процессов изменения состояния фондовых и валютных рынков // Труды СПИИРАН. Вып. 18. 2011. С. 5–18.
3. *Ананченко И.В., Волков А.А.* Защита программ для торговли на рынке Forex ключами HASP // Lambert Academic Publishing. 2013. С. 112.
4. *Ананченко И.В., Купченко А.* Разработка защищенных программ для торговой платформы MT4 // Lambert Academic Publishing. 2013. С. 116.
5. *Ананченко И.В.* Защити свою программу ключом HASP // Lambert Academic Publishing. 2012. С. 80.
6. *Мусаев А.А., Ананченко И.В.* Математические и информационные технологии на рынке «Forex» // Lambert Academic Publishing. 2013. С. 180.

**Мусаев Александр Азерович** — д.т.н., профессор; ведущий научный сотрудник, научно-исследовательской группы информационных технологий в образовании СПИИРАН, научный консультант, ОАО Специализированная инжиниринговая компания «Севзапмонтажавтоматика». Область научных интересов: анализ данных, управление и прогнозирование в сложных динамических системах, стохастические хаотические системы. Число научных публикаций — 176. [amusaev@szma.com](mailto:amusaev@szma.com), [www.szma.com](http://www.szma.com); СПИИРАН, 14-я линия В.О., д. 39, г. Санкт-Петербург, 199178, РФ; р.т. +7(812)350-5885, факс +7 (812)350-1113.

**Musaev Alexander Azerovich** — Dr. in Appl. Math., professor; leading researcher, Education

Information Technology Group, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), expert, public corporation Specialized engineering company "Sevzapmontageautomatica". Research interests: data analysis, complicated dynamic systems prognosis and control, stochastic chaos systems. The number of publications — 176. amusaev@szma.com, www.szma.com; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)350-5885, fax +7(812)350-1113.

**Ананченко Игорь Викторович** — доцент кафедры системного анализа Санкт-Петербургского государственного технологического института (технического университета). Область научных интересов: защита программ и данных, управление и прогнозирование в экономике и финансах, разработка программ для торговли на рынке Forex. Число научных публикаций — 24. igor@anantchenko.ru, www.mctrewards.ru; СПбГТИ(ТУ), Московский проспект, дом 26, г. Санкт-Петербург, 198150, РФ; д.т. +7(921)320-1586.

**Anantchenko Igor Viktorovitch** — associate professor of systems analysis SITSPR. Research interests: the protection of programs and data management and forecasting in economics and finance, program development for trading on Forex. The number of publications — 24. igor@anantchenko.ru, www.mctrewards.ru; SITSPR, Moscow Avenue, Building 26, St. Petersburg, 198150, Russia; home phone +7 (921) 320-1586.

Рекомендовано ИГИТО СПИИРАН, рук. Тишков А.В., к.ф.-м.н., доц.  
Статья поступила в редакцию 21.02.2013.

## РЕФЕРАТ

### *Ананченко И.В., Мусаев А.А.* **Защита приложений, выполняемых торговым терминалом MetaTrader, ключами Sentinel Hasp.**

Программы для торговли на рынке Forex, выполняющиеся на торговой платформе MT4, разрабатываются на языке программирования MQL 4 и выполняются после компиляции как программы формата ex4. Практика использования программ показала, что построить защиту только с использованием кода ex4 файла практически невозможно. При восстановлении исходного кода с помощью декомпилятора возможно получение последовательности функционально идентичных операторов. В результате массового использования декомпиляторов, выполняющих преобразование из формата ex4 в исходный формат mq4, на рынке прикладного программного обеспечения сложилась ситуация позволяющая любому желающему с минимальными усилиями получить исходный текст программы. Данная ситуация совершенно недопустима для разработчиков и продавцов коммерческих программ для торговли на Forex.

В качестве меры противодействия многие разработчики специализированного программного обеспечения (ПО) применяют решение, основанное на переносе критически важного для реализации успешной торговли функционала в динамически загружаемую dll библиотеку, к функциям которой выполняется обращение из исполняемого ex4 файла. Одной из поддерживаемых функций dll библиотеки может быть функция проверки кода регистрации, который является уникальным для каждого пользовательского счета.

Использование декомпиляторов dll библиотек и средств анализа динамически исполняемого программного кода позволяет получить исходный программный код dll библиотеки. Восстановленный исходный код более сложен для анализа, но это частичное решение проблемы, так как встает вопрос о защите dll библиотеки от несанкционированного использования.

В качестве варианта защиты dll библиотек предлагается использовать продукцию производителя SafeNet: аппаратные устройства HASP HL, подключаемые к USB порту компьютера или программный аналог - ключ HASP SL. Программный код внутри аппаратного ключа выполняется специализированным процессором. Этот же алгоритм может быть реализован под управлением процессора компьютера, на котором выполняется защищаемая программа.

Рассмотрен алгоритм установки программной защиты с использованием ключа HASP SL. Сравнение предлагаемой схемы использования аппаратной защиты показывает, что ключи HASP HL обеспечивают защиту не хуже, чем их аппаратные аналоги от конкурирующих фирм. Процесс установки и использования программы с защитой более удобен для пользователя. Защита с помощью ключей HASP SL показала хорошие результаты в сравнении с программными средствами защиты других разработчиков. Предлагаемая схема защиты с помощью ключей HASP HL и HASP SL успешно реализована в коммерческих торговых роботах.

## SUMMARY

### *Anantchenko I.A., Musaev A.A.* **Securing applications running trading platform MetaTrader keys Sentinel Hasp.**

Software for trading in the Forex running on MT4 trading platform developed in the programming language MQL 4 and executed after compiling a program format ex4. The use of the program showed that building protected only with code ex4 halcyon impossible. When restoring the source code using a decompiler is possible to obtain the sequence is functionally identical to the operators. As a result of mass use of decompilers perform conversion from the original format ex4 to mq4, market application software a situation allows anyone with minimal effort to get the source code. This situation is not acceptable for the developers and vendors of commercial software for trading in Forex.

As a countermeasure, many developers of specialized software (software) used a solution based on the transfer of critical importance for the implementation of successful trading functionality in dynamic link library dll whose functions referenced from executable ex4 file. One of the features supported dll libraries can function to check the registration code that is unique to each user account.

Decompilers use dll libraries and tools for analyzing dynamic executable code can get the source code of program dll libraries. Refurbished source code more complicated to analyze, but it is a partial solution to the problem, as there is a question of protecting dll libraries from unauthorized use.

Alternatively protect dll libraries are encouraged to use the manufacturer's products SafeNet: hardware devices HASP HL, under-clouded a USB port of a computer or software combination - key HASP SL. The code runs inside a hardware key specialized processor. The same algorithm can be implemented under the control of the processor of the computer on which the program is protected.

The algorithm set of software protection using the key HASP SL. A comparison of the proposed scheme using the hardware protection shows that HASP HL keys provide protection not worse than their hardware counterparts from competing firms. The installation and use of the program to the protection of more user friendly. Protection with HASP SL keys shown good results in comparison with the programmatic means of protection other developers. The proposed protection scheme using HASP HL key and HASP SL successfully implemented in commercial trading robots.