В. А. Букасов

АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ ЗАЩИТЫ ОТ КОПИРОВАНИЯ

На многих форумах, посвященных компьютерной безопасности, можно встретить разные вариации одного и того же вопроса: как лучше защитить программу, чтобы ее не взломали? Конечно, лучше всю защиту спроектировать и написать самому, но это желательно делать с самого начала разработки программы, да и для качественной защиты у программиста должны быть специальные знания. Зачастую этих знаний не хватает, а защитить как-то нужно. В этом случае обычно обращаются к программным средствам защиты от несанкционированного копирования, или протекторам. Они в своем большинстве не требуют никаких особых навыков, содержат множество приемов защиты и легко могут обработать с нужными настройками разработанную программу всего после пары нажатий кнопок мыши. Но так ли они хороши на самом деле и могут ли остановить опытного аналитика? На современном рынке программных средств защиты предоставлено множество решений, далее рассмотрим несколько наиболее популярных из них и проанализируем их достоинства и недостатки.

Armadillo/SoftwarePassport [1]. Защита работает как для архитектуры x86, так и для x64, как с EXE-файлами, так и с динамически загружаемыми библиотеками. Стоит отметить, что, хотя для защиты x64-файлов в пользовательском интерфейсе существуют средства, задействовать их на практике не удалось. Не удалось встретить и программ с включенными этими настройками, из чего можно сделать вывод, что на момент исследования этого протектора еще не все его возможности были перенесены на x64-архитектуру. Помимо стандартных возможностей защиты (шифрование и сжатие кода и данных программы, перенаправление таблицы импорта), были разработаны и такие достаточно оригинальные в свое время приемы, как блокировка отладчика (debug blocker) и наномиты (nanomites). Некоторое время назад протектор был переименован из Armadillo в SoftwarePassport, тогда же в него была добавлена и более гибкая система лицензирования, позволяющая осуществить привязку копии программы к компьютеру, ограничить время работы или число запусков, отключить некоторые функции при отсутствии ключа, настроить различные связанные с лицензией предупреждения и многое другое.

Говоря об онлайн-распространении, стоит упомянуть и про Digital River. Это отдельная компания, которая предоставляет маркетинговые услуги для созданного разработчиками программного обеспечения. Если пользователь решает купить программу, он может перейти на страницу заказов на сайте прямо из защищенного приложения и сразу в онлайне приобрести его. При этом пользователю не придется вводить никакие ключи самостоятельно, после покупки

протектор сам свяжется с сервером Digital River и получит все необходимые ключи. При этом разработчик может посмотреть детальную статистику на серверах Digital River по работе пользователей с его программой.

К недостаткам стоит отнести слабое прогрессирование продукта. Новые версии появляются регулярно, но качественных изменений в плане защиты практически никаких нет. На данный момент для этой защиты существует несколько публичных автоматических распаковщиков, но все они зачастую работают некорректно, особые сложности возникают у них с наномитами. Стоит еще добавить, что взломанные версии этой защиты для x86 регулярно появляются в открытом доступе.

ASProtect [2]. Защита предназначена для архитектуры x86 и работает как с EXE-файлами, так и с динамически загружаемыми библиотеками. Один из первых протекторов, который стал выносить части программы, в частности код с оригинальной точки входа, разбавляя их мусорными командами, в динамически выделяемую память. Впоследствии простое преобразование кода превратилось в виртуальную машину. Протектор также обладает стандартными возможностями защиты, такими как сжатие и шифрование кода и данных защищаемого приложения и перенаправление таблицы импорта. В зависимости от версии (параллельно развиваются две ветки: 1.x и 2.x) поддерживаются схемы лицензирования с ключами разной длины. Что примечательно, ранние версии протектора были несколько раз взломаны из-за неправильного использования криптографических алгоритмов. Система лицензирования достаточно стандартная с поддержкой привязки копии защищаемой программы к компьютеру, ограничения времени работы приложения, поддержкой черного списка ключей и расшифровкой помеченных маркерами участков кода только при наличии верного ключа.

Некоторое время назад данный продукт был продан компании StarForce. Новые версии появляются нечасто, основные обновления связаны с исправлением недочетов и незначительными улучшениями качества защиты, каких-либо серьезных нововведений не было уже достаточно долгое время. Для него существуют несколько распаковщиков, недоступных широкой публике, но автору не известен ни один автоматический распаковщик, который полностью декомпилировал бы виртуальную машину, хотя автоматизированные средства для этого существуют.

EXECryptor [3]. Данный протектор работает для архитектуры x86 и способен обрабатывать EXE-файлы и динамически загружаемые библиотеки. Один из первых протекторов, в котором стал применяться полиморф вполне неплохого качества для защиты кода. Помимо полиморфа существует возможность использовать так называемый «конверт» со сжатием и шифрованием кода и данных программы и перенаправлением таблицы импорта. Также в списке возможностей защиты числится и виртуальная машина, но она откровенно слабая, может эмулировать только одну инструкцию за раз и способна эмулировать только небольшой набор инструкций общего назначения. Схема лицензирования достаточно стандартная, с использованием асимметричной криптографии, поддержкой черного списка ключей и возможностью расшифровывать помеченный маркерами код только при наличии верного ключа.

Разработка данного протектора, видимо, остановлена, поскольку обновлений не было уже около пяти лет. Хотя и ходят слухи, что новая версия должна скоро появиться, пока они ничем не подтверждены. Распаковщиков для этой защиты существует немного в открытом доступе, но и существующие неплохо справляются со своей задачей. Стоит также отметить, что последняя версия выложена взломанной в открытый доступ.

StarForce [4]. Защита предназначена для архитектур x86 и x64 и работает не только с EXE-файлами и динамически загружаемыми библиотеками, но и с драйверами. Ранее защита в своей работе использовала драйвер, который перехватывал отладочные прерывания, что мешало их использованию аналитиком в своих целях. На данный момент от этого метода защиты

отказались, теперь драйвер используется в основном для работы с диском и привязки копии защищаемой программы к компьютеру. Основная защита легла на виртуальную машину, которая является одной из самых сильных среди конкурентов, и файловую систему SFFS (StarForce File System). Собственная файловая система представляет собой зашифрованный контейнер, содержащий указанные разработчиком файлы. При обращении защищенного приложения к этим файлам протектор перехватывает обращения и перенаправляет их в контейнер, расшифровывая нужные файлы. Развивается данный протектор стабильно. В основном его используют для защиты игр и программ, распространяемых на CD/DVD с привязкой к носителю, но спектр возможностей значительно шире, в том числе онлайн-распространение продуктов с гибкой системой лицензирования, включающей в себя онлайн-сервера компании StarForce.

Не считая двух старых статей, какой-либо информации по компрометации этого протектора в публичном доступе нет, поскольку разработчики защиты регулярно просматривают форумы. Взломанные версии программ, защищенных этим протектором, появляются не слишком часто, в основном от одних и тех же групп, которые хорошо его исследовали. Такое положение дел, видимо, обусловлено тем, что на данный момент защищенные этим протектором программы не слишком распространены, а также тем, что само защищающее приложение не передается вместе с программами, а защита проходит на серверах компании.

Themida/WinLicense [5]. Данная защита, согласно информации на главной странице официального сайта, поддерживает EXE-файлы и динамически загружаемые библиотеки для архитектур x86 и x64. В реальности же никаких ссылок на версию для x64 или защищенных программ для этой архитектуры найти не удалось. На данный момент эта защита является одной из самых перспективных, поскольку уже включает в себя множество достаточно сильных способов защиты и неплохо развивается. Новые версии выходят регулярно, список изменений обычно велик, но реально большинство этих изменений носит «косметический» характер и на сложность самой защиты практически не влияет. В защиту входит так называемый «конверт» с достаточно сильными антиотладочными приемами и один из самых сильных методов перенаправления импорта. Пользовательский код возможно защитить при помощи нескольких различных виртуальных машин, что значительно затрудняет работу аналитика. Для усложнения исследования обработчики в виртуальных машинах обработаны полиморфом вполне неплохого качества. WinLicense представляет собой ту же Themida, но с дополнительными настройками лицензирования, позволяющими делать достаточно стандартные вещи: привязывать копию программы к компьютеру, вносить ключи в черный список, расшифровывать помеченные маркерами части кода только при наличии ключа, ограничивать время работы программы и т. д.

Распаковщики для данного протектора в публичном доступе существуют только для старых версий, но и те работают далеко не всегда корректно. Практически все появляющиеся в открытом доступе способы скомпрометировать защиту достаточно быстро исправляются, что говорит о том, что разработчики следят за форумами. Стоит отметить, что многие из последних версий были взломаны и выложены в открытый доступ.

VMProtect [6]. Данный протектор предназначен для работы с архитектурами x86 и x64, причем способен защищать не только EXE-файлы и динамически загружаемые библиотеки, но и драйверы. Это практически первая защита, которая работала по принципу виртуальной машины. Виртуальная машина и по сей день является основным элементом защиты. Относительно недавно помимо виртуальной машины в качестве дополнительной защиты стало возможно поместить программу в так называемый «конверт», который на этапе защиты шифрует и сжимает код и данные программы, а во время запуска приложения расшифровывает обратно. Также «конверт» отвечает и за восстановление таблицы импорта, которая перенаправляется на собственные переходники, и за обнаружение факта отладки различными способами. Помимо виртуальной

машины существует возможность защитить код при помощи полиморфа, который преобразует исходные инструкции и разбавляет их мусорными командами, но качество полиморфа на данный момент не слишком хорошее.

В последнее время протектор стал развиваться значительно быстрее, улучшилось качество полиморфа, появилась возможность накрытия «конвертом» с достаточно сильной антиотладкой. Стоит также отметить, что несколько последних версий этой защиты были взломаны и выложены в открытый доступ.

И напоследок вкратце рассмотрим основные тенденции развития программных средств защиты. Сами по себе защиты все меньше стараются полагаться на недокументированные возможности в целях антиотладки, в частности, либо отказываются от драйверов вообще, либо возлагают на них гораздо меньше функций. Вызвано это, скорее всего, развитием архитектуры x64 и значительными изменениями в операционной системе. Сама защита все больше и больше стала полагаться на преобразование кода: в меньшей части полиморф, в большей части — виртуальные машины, поскольку именно они являются одним из наиболее сложно поддающихся анализу методов защиты. И постепенно защиты смещаются в сторону онлайн-технологий. Системы лицензирования все чаще становятся онлайн-ориентированными. Некоторые приложения, предназначенные для работы в Интернете, вообще могут не содержать ряда функций, а делать запросы к серверу, который заодно проверяет и легальность копии. Понятно, что в этом случае взлом практически невозможен. Также стоит отметить, что некоторые защищенные игры, распространяемые на CD/DVD, могут вообще не содержать основного EXE-файла, при первом запуске он будет загружен с сервера, который и проверит регистрацию. А также сервер следит за тем, чтобы одним ключом не пользовалось много людей.

Как видно, идеальной защиты не существует. Практически все протекторы будут рано или поздно взломаны, поэтому основная задача разработчика — не пытаться создать неломаемую защиту, а сделать взлом экономически нецелесообразным.

СПИСОК ЛИТЕРАТУРЫ:

1. Протектор Armadillo/SoftwarePassport. URL: http://www.siliconrealms.com.
2. Протектор ASProtect. URL: http://asprotect.com.
3. Протектор EXECryptor. URL: http://www.strongbit.com.
4. Протектор StarForce. URL: http://www.star-force.ru.
5. Протектор Themida/WinLicense. URL: http://www.oreans.com.
6. Протектор VMProtect. URL: http://www.vmprotect.ru.