УДК 004.007

А.С. Гордиенко, Д.Э. Ситников

Харьковская государственная академия культуры, Харьков

АНАЛИЗ МЕТОДОВ ПРОТИВОДЕЙСТВИЯ НАРУШЕНИЮ ЛИЦЕНЗИОННОЙ ЗАЩИТЫ НАСТОЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статье рассмотрены различные способы нарушения лицензионной защиты настольного программного обеспечения, предложены методы и конкретные практические рекомендации по противодействию взлому защиты. Ввиду обширности проблемы акцент сделан на классе условно бесплатного программного обеспечения. Исследованы отдельные уязвимости и причины их возникновения. Полученные результаты могут быть использованы при построении новых или усовершенствовании существующих систем защиты программных продуктов, а также при выборе готовых защитных программных решений.

Ключевые слова: *настольное программное обеспечение, условно бесплатное программное обеспечение, лицензионная защита, защита программного обеспечения, компьютерное пиратство.*

Введение

В 2009 году объем используемого нелегального программного обеспечения превысил 50 млрд. долларов США [1]. В большинстве развитых стран действуют государственные институты и профессиональные ассоциации, осуществляющие борьбу с компьютерным пиратством. Вопрос противодействия обходу лицензионной защиты коммерческого программного обеспечения актуален для компаний-производителей любого масштаба.

Цель данной статьи – проанализировать существующие способы обхода лицензионной защиты настольных приложений и методы противодействия им. В задачу входит обзор как технических, так и нетехнических подходов к решению задачи, а также формирование общих практических рекомендаций для производителей. Акцент делается на условно бесплатном коммерческом программном обеспечении (класса «Shareware»).

Способы нарушения защиты программного обеспечения и соответствующие методы противодействия

Практика противодействия обходу лицензионной защиты показывает, что при достаточном количестве затраченных ресурсов любое программное обеспечение можно взломать. Исключение составляют две категории, требующие принципиально большего количества ресурсов для взлома системы:

– приложения, где существенная часть логики выполняется на стороне сервера, и эту логику непросто эмулировать (примеры – многопользовательские онлайн-игры, сервисы облачного хранения данных);

– системы, где аппаратное обеспечение закрыто и контролируется той же компанией, что и программное обеспечение (примеры – некоторые игровые консоли, медиа центры).

Учитывая указанный факт, основной задачей защитных систем является максимальное усложнение взлома, а также повышение порога квалификации пользователя для возможности взлома. Ниже приведены примеры подходов к обходу лицензионной защиты в порядке возрастания сложности применения для конечного пользователя:

– введение нелегального серийного номера, распространяемого отдельно от программы (тут сложность, фактически, эквивалентна сложности легальной регистрации программы);

– использование генератора серийных номеров и/или ключей авторизации;

– применение для взлома программы-патчера, сигнатура которой занесена в антивирусные базы таким образом, что она будет удалена при нахождении (либо подмена оригинальных файлов распространяемыми «пропатченными»);

– блокирование обращения к серверу проверки регистрации (проще для взлома – по DNS-имени, сложнее – по IP-адресу, требует установки и настройки брандмауэра);

– получение полной копии программы, взломанной или вместе с инструментами для взлома.

Существует мнение, что, так как «непробиваемой» защиты добиться невозможно, то не стоит и препятствовать возможности взлома наиболее технически подкованными пользователями. Как пример, компания Microsoft известна лояльным отношением к частным нарушителям лицензионной защиты своего настольного программного обеспечения, что позволило ей занять доминирующее положение на рынке в ряде ниш [2].

Наиболее уязвимым звеном в механизме лицензионной защиты является условный переход, который определяет, выполняться ли программе или

нет, либо в каком режиме выполняться – полнофункциональном или демонстрационном. Вне зависимости от того, насколько сложный шифр применен при проверке серийного номера, результатом этой проверки будет простой ответ: «разрешить» или «запретить». Цель процесса взлома – повлиять на этот условный переход.

Следовательно, наиболее прямолинейный способ взлома – это прямое изменение логики перехода на обратную (грубо говоря, «if» на «if not» или «JE» на «JNE»). Для этой цели злоумышленники применяют методы декомпиляции и дизассемблирования.

Декомпиляция предполагает трансляцию программы в исходный код, обнажающий логику приложения и доступный к редактированию. Декомпиляции могут быть подвергнуты приложения, написанные на языках программирования высокого уровня; наиболее подвержены декомпиляции приложения, основанные на технологиях Java, .NET, Adobe Air, Delphi. Для затруднения декомпиляции применяют запутывание кода (обфускацию), шифрование и упаковку исполняемых файлов.

Суть дизассемблирования состоит в преобразовании исполняемого кода в текст на языке ассемблера с целью отыскать и изменить критичные логические блоки. Эффективным инструментом поиска критичного условного перехода является применение интерактивных дизассемблеров-отладчиков, исполняемых на уровне ядра, что затрудняет их определение. Также дизассемблирование позволяет раскрыть детали реализации механизма лицензионной защиты, в том числе алгоритма проверки лицензионного ключа. Построение защиты от эффективного дизассемблирования является сложной технической задачей, для которой нет готовых решений. Используемые методы включают сокрытие условного перехода, определение присутствия отладчика (с помощью анализа выполняемого кода, состояний регистров процессора, точного времени выполнения инструкций и пр.) и преднамеренное нарушение работы отладчика [3].

Результат изменения логики программы обычно распространяется в виде файлов, которыми нужно заменить оригинальные, либо программы-патчера, которая модифицирует исполняемый или другие файлы. Алгоритмы проверки целостности программы и отдельных модулей, встроенные в приложение, могут существенно затруднить такой способ обхода защиты. Кроме того, крупными производителями широко используется добавление сигнатуры модифицированных файлов либо файлов-программ для взлома в базы распространенных антивирусных пакетов.

Однако, наибольшее количество взломов лицензионной защиты осуществляется не с помощью «переворачивания» условного перехода проверки на легальность использования, а с помощью непрямого воздействия на переменные, на которые опирается решение.

Одной из таких переменных является текущее время, которое определяет, истек ли срок действия лицензии или окончился ли пробный период. Изменение системной даты – простой способ обхода такого ограничения, доступный даже неопытным пользователям. Чтобы затруднить использование этого приема, рекомендуется запрашивать время с тайм-сервера, а вне его доступности – проверять, чтобы текущая дата не предшествовала ранее зафиксированной.

При проверке на легальность учитывается также ранее сохраненная приложением на компьютере информация. Особенно важна она в случае, если был нарушен доступ приложения к серверу, либо для проверки истечения пробного периода или срока действия лицензии. Злонамеренный пользователь может прибегнуть к удалению или изменению такой информации. Чтобы избежать этого, рекомендуется хранить данные в нескольких неочевидных местах в зашифрованном виде (например: в альтернативных каналах файла NTFS; специально предназначенной для этого области исполняемого файла; системном разделе реестра Windows и т.п.). Также можно переложить контроль над сохранностью данных на лицензионный сервис – процесс, запущенный в фоновом режиме и с расширенными привилегиями, ответственный за общение с сервером верификации и другие относящиеся к лицензиям задачи.

Другой важной переменной в логике проверки на легальность является серийный номер. Раскрытие злоумышленниками алгоритма проверки серийного номера часто приводит к распространению программы-генератора ключей, что усложняет контроль над лицензиями без строгой серверной верификации.

Рекомендуется использовать криптостойкие ассиметричные шифры, используя секретный ключ для генерации серийных номеров или их аналогов и интегрируя открытый ключ в приложение.

Производители часто используют один и тот же ключ на несколько лицензий, приобретенных одним покупателем, в целях удобства. Это открывает злоумышленникам возможность использования одного ключа на количестве компьютеров большем, нежели разрешено производителем. Чтобы противостоять такому приему, следует идентифицировать и зафиксировать на удаленном сервере каждую систему, где приложение было запущено. Наиболее надежным способом идентификации является привязка к аппаратным идентификаторам (можно рекомендовать использовать модель и серийный номер процессора, программно получаемые с помощью опкода CPUID [4]).

Львиная доля нарушений лицензионной защиты, связанных с лицензионным ключом, может быть решена с помощью механизма серверной верификации. Сбор данных на стороне сервера позволяет определить не только несанкционированно сгенерированные ключи, но и определить, какие из легальных ключей стали широко распространены.

Эффективным методом обхода серверной верификации является блокирование доступа к серверу. Большинство настольных приложений сохраняет функциональность без доступа к Интернету, потому однажды получив возможность запускать программу (возможно, даже легально), злоумышленник может воспроизвести ту же последовательность действий на любом количестве компьютеров (часто – с тем же лицензионным ключом), получая таким образом возможность создавать неограниченное количество несанкционированных инсталляций.

Так как большинство настольных компьютеров сегодня имеет доступ к Интернет, можно предложить такие методы противодействия обходу серверной верификации:

– использование нескольких серверов верификации, обновление списка серверов, сложные правила выбора серверов – например, верификация с помощью одного из серверов только по наступлению определенной даты;

– доступ к серверам по IP-адресу, а не по DNS-имени – это исключает простой метод блокировки с помощью переназначения DNS (например, с помощью файла hosts); как альтернатива – анализ файла hosts и автоматическое очищение его от упоминаний серверов верификации;

– добавление серверов в исключения установленных брандмауэров;

– усложнение протокола и шифрование общения с сервером верификации для затруднения его реверс-инжиниринга и эмуляции;

– скрещивание функции верификационного сервера с «полезными» онлайновыми функциями (например, автоматическим обновлением), так, чтобы было невозможно отделить верификационные запросы;

– отслеживание публично распространяемых или часто применяемых ключей и блокирование их как на стороне сервера, так и в «черных списках», включаемых в новые версии программного обеспечения;

– в дополнение к предыдущему пункту – препятствование распространению устаревших версий программного обеспечения (таким образом, злоумышленнику придется распространять устаревшую версию программы вместе с ключом).

Частые обновления программного обеспечения могут стимулировать покупку легальной копии. Последняя версия не всегда будет доступна в нелегальных источниках, в особенности, если вносить в каждой версии изменения в механизм защиты, препятствующие использованию ранее эффективных средств обхода.

Регулярные обновления программного обеспечения можно отнести к средствам организационной защиты сложных корпоративных приложений наряду с «горячими линиями» поддержки, системами обучения пользователей, сервисным обслуживанием, предварительным отбором покупателей. Такие дополнительные услуги обычно направлены на корпоративных покупателей. Они призваны увеличить привлекательность легального приобретения и сформировать образ нелицензионного приложения как продукта «второго сорта».

Также следует отметить ряд других нетехнологических методов уменьшения нелегального использования:

– поиск и закрытие источников распространения программы, в первую очередь, в Интернет. Большинство ресурсов удаляет нелегальное программное обеспечение по первой просьбе правообладателя;

– показательное судебное преследование отдельных пользователей-нарушителей;

– упоминание владельца лицензии в интерфейсе программы: при запуске, в окне «о программе», постоянно в заголовке окна программы и т.п. Помимо некоторого психологического дискомфорта для нелегального пользователя, это упрощает проведение проверок.

Для повышения уровня защищенности приложения могут быть использованы методы скрытой проверки и удаленного блокирования. Скрытая проверка подразумевает процедуру проверки подлинности, альтернативную стандартной и запускаемую не сразу при регистрации, а уже в процессе использования программы (например, случайным образом раз в месяц, начиная через неделю после установки). Удаленное блокирование – это скрытая в программе функция, которая может заблокировать ее работу по сигналу с сервера (обнулить статус регистрации и предложить легально приобрести приложение). Эти методы опираются на тот факт, что при взломе лицензионной защиты злоумышленник обыкновенно использует наиболее доступный способ обхода, не прибегая к всесторонней проверке результатов. Следовательно, включив в программу один или несколько альтернативных методов защиты (таких как использование других серверов, проверки целостности), которые не будут проявлять себя сразу же при взломе, можно впоследствии получить дополнительных легальных пользователей, заблокировав взломанные приложения. Следует отметить, что комбинация скрытой проверки и удаленного блокирования является эффективным методом стимуля-

ции к их применению для приложений каждодневного использования (например, электронных словарей) и компьютерных игр (где блокировка может наступать в середине игрового процесса), но менее применима в «одноразовых» программах.

В 90-х и 2000-х годах широко использовался метод контроля над легальностью установленного программного обеспечения с помощью наличия оригинального носителя [5] (CD-диск, DVD-диск, реже дискета). Носитель периодически запрашивался или требовался все время для работы программы. Применялись различные методы верификации оригинальности носителя и механизмы защиты от копирования. Ко времени написания данной статьи этот метод остается актуальным по большей части для игровых консолей, так как на персональных компьютерах получили распространение многофункциональные эмуляторы оптических дисководов, а также развилась электронная дистрибуция без участия физического носителя.

Чрезмерное усложнение систем противодействия обходу защиты приводит к появлению ложных определений нарушения защиты. Например, технология подтверждения подлинности операционной системы Windows Genuine Advantage от Microsoft, по заявлениям Microsoft, ошибочно определяла около 1% легальных копий как нелегальные (по некоторым оценкам – до 20%), что привело к необходимости введения третьего режима функционирования системы помимо «взломанного» и «подтвержденного легальным» – «неопределенного», и вынудила компанию в дальнейшем ослабить некоторые механизмы контроля [6].

Выводы

Таким образом, были рассмотрены различные способы обхода лицензионной защиты и обозначены методы противодействия им. В то время как было названо большое количество возможных мер защиты, нужно учитывать, что применять сразу их все на ранних этапах развития программного продукта чаще всего будет нецелесообразно. Лицензионная защита должна развиваться и укрепляться сообразно развитию программного продукта, отвечая на реальные попытки взлома и учитывая последние тенденции, а также согласуясь с потенциальным экономическим эффектом от внедрения методов защиты.

Список литературы

1. *BSA Global Software Piracy Study 2010 [Электронный ресурс] // Business Software Alliance USA – Режим доступа к ресурсу: http://portal.bsa.org/globalpiracy2009/index.html. – Загл. с экрана.*

2. *How Microsoft conquered China [Электронный ресурс] // CNNMoney.com – Режим доступа к ресурсу: http://money.cnn.com/magazines/fortune/fortune_archive/2007/07/23/100134488/. – Загл. с экрана.*

3. *A Review of Modern Day Software Copy Protection Solutions [Электронный ресурс] // Deadc0de's Security Blog – Режим доступа к ресурсу: http://www.deadc0de.info/2010/02/14/a-review-of-modern-day-software-copy-protection-solutions/. – Загл. с экрана.*

4. *IA-32 architecture - CPUID [Электронный ресурс] // Sandpile.org – Режим доступа к ресурсу: http://www.sandpile.org/ia32/cpuid.htm. – Загл. с экрана.*

5. *PC Game Piracy Examined [Электронный ресурс] // Tweakguides.com – Режим доступа к ресурсу: http://www.tweakguides.com/Piracy_8.html. – Загл. с экрана.*

6. *Windows Genuine Advantage's newest setting: "you might be a pirate" [Электронный ресурс] // Ars Technica – Режим доступа к ресурсу: http://arstechnica.com/old/content/2007/02/8922.ars. – Загл. с экрана.*

Поступила в редколлегию 1.06.2011

Рецензент: д-р техн. наук, проф. И.В. Гребенник, Харьковский национальный университет радиоэлектроники, Харьков.

АНАЛІЗ МЕТОДІВ ПРОТИДІЇ ПОРУШЕННЮ ЛІЦЕНЗІЙНОГО ЗАХИСТУ НАСТІЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

А.С. Гордієнко, Д.Е. Ситніков

Розглянуто різні способи обходу ліцензійного захисту настільного програмного забезпечення, запропоновані методи та конкретні практичні рекомендації щодо протидії злому захисту. Досліджено окремі уразливості і причини їх виникнення. Отримані результати можуть бути використані при побудові нових або вдосконалення існуючих систем захисту програмних продуктів, а також при виборі готових захисних програмних рішень.

Ключові слова: *настільне програмне забезпечення, умовне безкоштовне програмне забезпечення, ліцензійний захист, захист програмного забезпечення, комп'ютерне піратство.*

ANALYSIS OF METHODS FOR DESKTOP SOFTWARE LICENSING PROTECTION BREACH PREVENTION

A.S. Gordiyenko, D.E. Sitnikov

The article lists ways of breaching desktop software licensing protection and suggests methods and specific practical recommendations for maximizing anti-piracy security. Various vulnerabilities and their causes were analyzed. Obtained results can be used while constructing new or improving existing licensing systems and desktop software protection systems, as well as can help with choice of off-the-shelf protection solutions.

Keywords: *table software, public domain software, licensed defence, defence of software, computer piracy.*