# грабеж защищенного медиа контента в висте

крис касперски ака мыщъх, no-email

**магнатам медиа индустрии страшно не нравится бесконтрольное копирование аудио и видео и они всячески пытаются ему помешать. большой переполох среди пользователей вызвала новая инициатива Microsoft по созданию защитного механизма нового поколения, использующего зашифрованный цифровой поток, охраняемый по всему пути его следования и расшифровываемый только в видео-карте. сразу же возникает вопрос — как дальше жить и чем эту защиту ломать?**

## *введение или много шума из ничего*

Первое правило защиты информации гласит — если информацию можно воспроизвести, ее можно и скопировать. Помешать этому может только перепроектирование всего оборудования, задействованного в обработке сигнала. Вообразим себе такую систему в которой на DVD-диске записан зашифрованный файл, передаваемый по всем компьютерным шинам в закодированном виде и расшифровываемый только… в видео-карте! При попытке просмотра защищенного, но не разрешенного цифрового потока, видеовыходы автоматически отключаются (показывая равнодушный черный экран) или изображение умышленно искажается так, что никакого удовольствия от его просмотра мы все равно не получим.

Даже если мы перехватим цифровой поток по пути его следования от DVD к видео-карте, нам все равно не удастся ничего с ним сделать, ведь против шифровки не попреешь! Но это в теории. На практике же… чтобы расшифровать видео-поток достаточно добыть ключ.

Допустим, ключ спрятан непосредственно в самой видео-карте, причем спрятан так хорошо, что ни через прошивку, ни каким либо другим программным путем его считать нельзя. При условии выбора стойкого криптоаглоритма, у нас нет никаких шансов расшифровать цифровой поток, чтобы записать его на жесткий диск, например. Остается только грабить видео выход непосредственно с самой видео-карты, который… в свою очередь тоже обещает быть зашифрован, окончательно расшифровываясь лишь перед непосредственным выходом на экран. Если это будет LCD монитор, то у нас остается возможность подключиться непосредственно к выходам матрицы, сняв готовый к употреблению сигнал. Конечно, придется поднапрячься, но… оно того стоит. В теории. На практике же. Если одним и тем же ключом расшифровываются _все_ диски и этот ключ в явном виде хранится в куче "разнокалиберного" оборудования, _никакие_ ухищрения не позволяет удержать его в секрете! Следовательно, ключ должен передаваться в карту извне и храниться на самом DVD диске, что позволяет извлечь его, расшифровывая видео-поток вручную.

Но если раньше все это были догадки, основанные на смутных и противоречивых слухах, просочившиеся по неофициальным каналам, то теперь Microsoft достаточно подробно документировала систему защиты. Как и следовало ожидать, она оказалась смесью программно-аппаратных решений, причем расшифровка цифрового потока и отключение видеовыходов осуществляется драйверами карты, то есть _программно_

Хакеры пьют пиво и торжествуют (уже обдумывая как они будет грабить цифровые потоки), а пользователи по-прежнему гоняют ослов, записывают файлы на жесткие диски, обмениваясь ими с друзьями.

## *защита цифрового потока извне и изнутри*

В процессе разработки защиты нового поколения, Microsoft не родила ни одной умной идеи, зато изобрела множество аббревиатур, обрушив на программистов целый ворох новых терминов, объяснение которых можно найти в прилагаемом к статье глоссарии.

Ключевыми компонентами защитного механизма является: Защищенный Медиа Путь (он же Protected Media Path или, сокращенно, PMP), Защищенное Аудио Пользовательского Режима (Protected User-Mode Audio или PUMA), Защищенный Видео Путь (Protected Video Path или PVP), Управление Защитой Защищенного Видео Пути (Protected Video Path-Output Protection Management или PVP_OPM) и Безопасный Аудио Путь (Secure Audio Path или SAP). Это же крышей поехать можно, столько разных слов, а все равно никакого толку! Кстати, главный признак ненадежности защиты — ее запутанность и абсолютная непрозрачность. Такое чувство, что плаваешь в мутной воде зловонной реки, заболотившиеся много лет тому назад. Но это все лирика. Переходим к обсуждению технических деталей.

На вершине пирамиды защитных компонентов гордо реет MIG (Media Interoperability Gateway — Инетероперабельный Медиа Шлюз), предоставляющий приложениям доступ к защищенному медиа-контенту и управляющий политикой использования и воспроизведения медиа-контента в изолированных защищенных процессах, гарантирующих, что медиа-контент будет использован строго в соответствии с набором разрешений/запретов, установленных его законным владельцем. Защищенные процессы отличаются от всех остальных, что с ними нельзя манипулировать посредством API-функции ReadProcessMemory/WriteProcessMemory, даже с правами администратора. Тем не менее, в них можно проникнуть через ядро. На x86-64 машинах для этого понадобится подписанный драйвер (который подписывать, естественно, никто не собирается, ну разве что за подкуп), однако, всегда существует возможность нажать <F8> при загрузке системы, отключив проверку цифровой подписи. Конечно, это не слишком-то удобно, но для грабежа медиа-контента вполне подойдет.

Компонент, ответственный за управление защитой защищенного видео пути (PVP-OPM) гарантирует, что выходы видео-карты, установленной в PC, снабжены соответствующей защитой, которую требует лицензионное соглашение с обладателем защищенного медиа-контента. Он же управляет остальными защитными схемами такими как: защита широкополосного цифрового контента (High-Bandwidth Digital Content Protection или HDCP), печально известная Macrovision, проявляющая себя неоправданными искажениями на некоторых моделях телевизоров и домашних кинотеатров, Общая Система Аналогового Управления Копированием (Copy Generation Management System-Analog или CGMS-A) и т. д. Как и предыдущий компонент, PVP-OPM представляет собой чисто программное решение, которое легко может быть взломано. После недолго пыхтения в дизассемблере, хакер навсегда отучит этого зверюгу отключать видеовыходы все зависимости идет ли по ним разрешенный медиа-контент или нет.

Полнодоступная шина защищенного пользовательского виде пути (protected video path-user accessible bus или PVP_UAB) вполне соответствует своему пугающему названию и шифрует медиа-контент по пути его следования чрез шину PCI Express к целевому графическому адаптеру. Эта стадия не является обязательной и шифрование задействуется только в тех случаях, когда владелец медиа-контента считает, что среди потенциальных пользователей его продукции найдется идиот, воткнувший в шину эмулятор карты и перехватывающий цифровое содержимое в чистом виде. Типа грабеж среди бела дня. Шифрование естественно, выполняется программно и легко отключается элементарной правкой драйвера.

Модуль защищенного аудио пользовательского режима (PUMA) обеспечивает "безопасное" окружение для воспроизведения аудио, блокируя аудио-выходы при необходимости. Естественно, все это происходит на программном уровне.

От производителей видео-карт требуется "всего лишь" включать механизм управления видеовыходами и гарантировать, что драйверы, управляющие картой, не будут модифицированы злостными хакерами, а для этого драйвера должны быть пройти процедуру сертификации и получить цифровую подпись, удостоверяющую целостность их содержимого.

Таким образом, грабеж цифрового контента сводится к нейтрализации механизма Patch-Guard, контролирующего целостность ядра Windows (см. статью "взлом patch-guard", опубликованную в этом же номере), отключению механизма проверки цифровой подписи и модификации видео-драйвера, доходчиво объясняющей ему, что отключать видеовыходы это очень нехорошо и совсем не по коммунистически.

Обобщив все вышесказанное, мы получаем следующую схему, демонстрирующую маршрут передвижения цифрового медиа-контента и все превращения, которым он подвергается на пути своего следования.

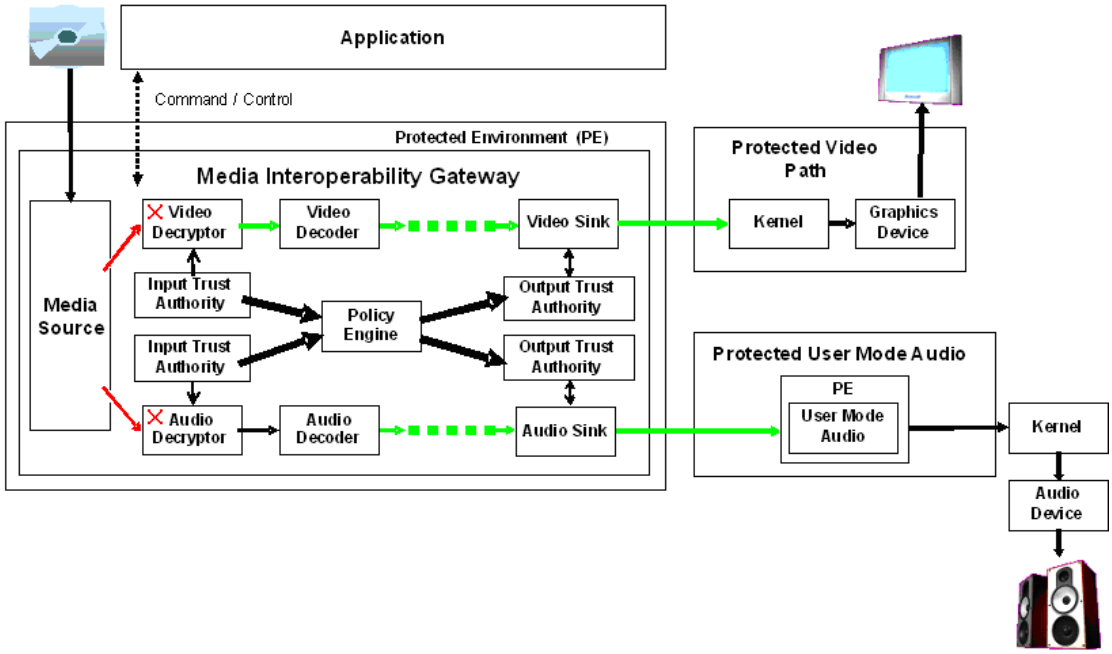


**Рисунок 1 структурная схема защиты медиа-контента, реализованная в висте**

Красным цветом показаны маршруты передачи зашифрованного цифрового потока, зеленым — уже расшифрованного и готового к употреблению. Что награбить расшифрованный поток, достаточно подключится к "выходу" видео и аудио декрупторов и… все!!! Остальные компоненты (типа аудио/видео декодеров при желании можно реализовать и самостоятельно).

Таким образом, защита не препятствует ни пиратам, ни продвинутым пользователям, а вот легальные потребители получают геморрой в виде неизбежных глюков, невозможности применения сторонних плейеров (защищенные процессы пока может создавать только штатный медиа-плейер), наконец, нагрузка на процессор и потребности в оперативной памяти резко возрастают, что рикошетом ударяет по качеству изображения. Ведь далеко не все могут позволить себе крутую машину и даже паршивый MPEG2 при выводе на телевизор с большим экранам требует приобретения аппаратного декодера, поскольку Pentium-4 с ним уже не справляется (естественно, к фильмам, ужатым до размеров половины CD-ROM это не относится).

Подробнее узнать о механизме защиты можно из официального документа Microsoft, лежащего на: http://download.microsoft.com/download/a/f/7/af7777e5-7dcd-4800-8a0a-b18336565f5b/PMP-sign.doc

## *два слова о COPP*

Сертифицированный Протокол Защищенного Выхода (Certified Output Protection Protocol или, сокращенно, COPP) позволяет приложениям защищать видео-поток на всем пути его следования от видео-карты до монитора или любого другого устройства отображения. Приложения могут использовать COPP, чтобы определить тип физического соединения, связывающего карту с монитором и наличие защиты от просмотра неразрешенного медиа-контента. Защитный механизм состоит из трех следующих компонентов:

- защиты широкополосного цифрового контента (High-Bandwidth Digital Content Protection или HDCP);
- общей системы аналогового управления копированием (Copy Generation Management System-Analog или CGMS-A);
- Защиты от Аналогового Копирования (Analog Copy Protection или ACP)

Если видео-карта поддерживает хотя бы один из этих механизмов, приложения могут задействовать протокол COPP для установки требуемого уровня защиты. Сам протокол COPP определяет набор соглашений, обеспечивающих установку безопасного (secure) коммуникационного канала с драйвером видео-карты, что является ключевым элементом общей стратегии взлома. Если мы распотрошим COPP, останется только модифицировать драйвер!

Протокол COPP использует Коды Аутентификационных Сообщений (Message Authentication Codes или, сокращенно MACs), для проверки целостности COPP-команд, циркулирующих между приложением (видео-плеером) и видео-драйвером. Приложения взаимодействуют с COPP посредством вызова методом IAMCertifiedOutputProtection, представляющего собой интерфейс DirectShow Video Mixing Renderer filter (VMR-7 or VMR-9).

Сам по себе протокол COPP не определяет никаких политик цифровых прав, применяемых к цифровому медиа-контенту. Так же, COPP не содержит в себе никаких защитных механизмов, контролирующих выход цифрового медиа-потока. Протокол COPP всего лишь предоставляет собой своеобразный "контейнер", обслуживающий подключенные к нему защиты, обеспеченные видео-картой и, в частности, может вывести список поддерживаемых ею методов — нехай медиа-плеер сам решает: достаточно ли этих защит для воспроизведения защищенного контента или нет. Очевидно, что слегка доработав COPP напильником, мы можем заставить его возвращать подложную информацию, позволяющую проигрывать защищенный медиа-контент на незащищенном оборудовании, допускающим его грабеж.

Прежде, чем начать взаимодействовать с COPP'ом, приложение должно последовательно выполнить следующие шаги:

- получить цепочку сертификатов драйвера (а то вдруг попадется "левый" драйвер?!);
- построить DirectShow playback graph, воспроизводящий (to render) видео-поток через Video Mixing Renderer filter (VMR);
- запросить VMR для IAMCertifiedOutputProtection интерфейса;
- вызывать IAMCertifiedOutputProtection::KeyExchange, возвращающий 128-битной случайное число, сгенерированное видео-драйвером вместе с цепочкой сертификатов, содержащей 2048-битный публичный RSA-ключ, которым был подписан драйвер;
- проверить цепочку сертификатов и если цепочка сертификатов неверна, прекратить работу и уйти на покой;
- проверить список аннулированных сертификатов (certificate revocation list или CRL) и, если хотя из сертификатов уже успел "засветиться" в этом листе, прекратить работу;
- извлечь из цепочки сертификатов публичный RSA-ключ;
- инициализировать COPP-сессию;
- сгенерировать 128-битный AES сессионный ключ, который будет использован для подписи данных и проверки, что подписанные данные не были модифицированы;
- сгенерировать два криптографических секретных 32-битных случайных числа. Первое представляет собой номер последовательности статуса, а второе — номер последовательности команд. Каждый раз, когда приложение посылает команду или запрос статуса, соответствующие номера последовательности увеличиваются на единицу, и возвращаются вместе с запросом назад к приложению;
- объединить 128-битное случайное число, полученное от видео-драйвера, с AES сессионным ключом, номерами последовательности статуса и команд. зашифровать этот байтовый массив публичным ключом драйвера и передать полученный результат методу IAMCertifiedOutputProtection::SessionSequenceStart;
- начать отправить COPP-команды и запросы статуса;
- поинтересоваться наличие защитных механизмов, а так же их типом путем вызова метода IAMCertifiedOutputProtection::ProtectionStatus;
- установить заданные уровни защиты путем вызова метода IAMCertifiedOutputProtection::ProtectionCommand;
- периодически опрашивать текущий уровень защиты, немедленно прекращая воспроизведение, если локальный уровень защиты вдруг неожиданно измениться или произойдет какая-нибудь другая проблема типа этой.

Как видно, основные защитные компоненты реализованы на программном уровне и сводятся к серии проверок, полагающихся на криптографию, но забывающие о том, что выполнять все эти шаги, в общем-то, совершенно необязательно. Фактически, защита цифрового медиа-контента закладывается на честность реализации плеера, который кстати говоря, хоть и защищен от модификации, но и эта защита выполнена на программном, а отнюдь не аппаратном уровне.

Подробнее обо всем этом можно прочитать в спецификации на COPP-протокол "Using Certified Output Protection Protocol (COPP)", описание которого доступно всем желающим: msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwmt/html/using_certified_output_protection_protocol_copp__bwjn.asp, там же можно найти и готовы примеры, значительно упрощающие укрощение этой заразы.

## *заключение*

Как и следовало ожидать, ничего путного у Microsoft не получилось. Вместо изящной, безглючной и по-настоящему надежной защиты, мы получили уродливое переплетение множества модулей, непонятно что делающих и для какой цели созданных. То есть, цель как раз очень хорошо понятна, но вот ее реализация…

В общем, хакеры без работы не останутся, так что пользователи могут не волноваться.

## *>>> врезка кинозалы vs DVD*

До сих пор существует заблуждение, что своей основной доход киностудии собирают в кинотеатрах, а выпуск DVD составляет ничтожную долю прибыли. Ну это когда как… В частности, фильм "DOMINO", на съемки которого ушло порядка $50 миллионов долларов собрал в кинотеатрах всего лишь… $1 миллион, но попав на DVD сразу же получил культовый статус и до сих пор не собирается сокращать объемы продаж. Аналогичная история произошла и с "Fight Club", и с "Dead Man" и со многими другими фильмами, рассчитанными на специфическую аудиторию (иди отвергаемыми рядом кинотеатров из-за излишней жестокости, например).

**Рисунок 2 кард из фильма "DOMINO"**

| компонент | тип требуемого сертификата | использование сертификата | пример разрешенного сценария воспроизведение | опции подписи |
|---|---|---|---|---|
| взаимодействующий с видео-драйвером в режиме ядра | подпись кода | подпись кода | HD DVD | KMCS1, WHQL2 |
| | PVP-OPM | запрос-отклик | HD DVD с интегрированным графическим адаптером | MFPMP3 |
| | PVP-UAB | запрос-отклик | HD DVD с дискетным графическим адаптером | MFPMP |
| | PVP-OPM в режиме совместимости | запрос-отклик | медиа-контент, требующий COPP'a на XP | MFPMP |
| драйвер ядра, не взаимодействующий с видео-драйвером | подпись кода | подпись кода | HD DVD | KMCS, WHQL |
| драйвер пользовательского режима, взаимодействующий с видео-драйвером | PMP-PE | подпись кода | воспроизведение защищенного контента через PMP | WHQL, MFPMP |
| драйвер ядра, взаимодействующий с аудио-драйвером или его компонентами | PUMA | подпись кода | SAP-контент с аудиосистемой, позволяющий задействовать эти требования | WHQL |
| драйвер пользовательского режиме или другие компоненты, участвующие в обработке объектов APO | PMP-PE | подпись кода | компоненты или APOs могущие обрабатывать защищенный контент | WHQL, MFPMP |
| плагины Media Foundation (кодеки, фильтры) | PMP-PE | подпись кода | плагины, обрабатывающие защищенный контент | MFPMP |

**Листинг 1 виды цифровых подписей, необходимых различным компонентам, взаимодействующих с защищенным медиа-контентом**

## *>>> врезка глоссарий*

- Advanced Access Content System (AACS):
  - Продвинутая Система Контроля Доступа: спецификация, описывающая управление медиа-контентом, записанным на оптических носителях следующего поколения и предназначенных для воспроизведения в PC и автономных проигрывателях;
- certification authority (CA):
  - Уполномоченный по Сертификации: "авторитет", предоставляющий сертификаты, подвергающие, что публичный ключ принадлежит именно тому, за кого он себя выдает, предъявляя настоящее или поддельное удостоверение личности;
- code-signing certificate:
  - Сертификат Подписывающий Код: сертификат, используемый для подписывания двоичных файлов;
- DRM attribute:

- DRM Атрибут: code-signing атрибут, предоставленный "Windows Logo Program" (здесь "Program" — это не программа, а компания), проверяющий, что драйвер выполнен в полном соответствии с требовании аппаратной Универсальной Аудио Архитектурой (Universal Audio Architecture или, сокращенно, UAA) и позволяющей драйверу обрабатывать защищенный медиа-контент;

- discrete versus integrated graphics:
  - Дискретная Графика Против Интегрированной: дискретные графические адаптеры представляют собой самостоятельные устройства, обычно вставляемые в материнскую плату через слоты расширения. интегрированные же графические адаптеры встроены непосредственно в сам чипсет;
- identified kernel:
  - Отождествленное Ядро: ядро, в котором находятся только драйвера, подписанные авторитетами, которым Microsoft безоговорочно доверяет;
- kernel-mode code signing (KMCS):
  - Подписывание кода в Режиме Ядра: процесс цифрового подписывания программного обеспечения, без которого оно не может быть загружено на уровень ядра;
- Media Interoperability Gateway (MIG):
  - Инетероперабельный Медиа Шлюз: расширяемый мультимедийный конвейер, заброшенный на вершину нового Media Foundation API и работающий внутри защищенного окружения (Protected Environment или, сокращенно, PE), которое, очевидно, защищенным не является;
- MIG plug-in:
  - MIG-плагины: компоненты обработки медиа-потока или защиты его содержимого, находящиеся внутри MIG-конвейера/PE и препятствующие открытому грабежу среди бела дня. примерами MIG-планинов являются кодеки и декрипиторы;
- participating driver:
  - Драйвер-Соучастник: любой компонент пользовательского уровня, загруженный внутрь PMP PE и имеющий доступ к незашифрованному защищенному медиа-контенту и транспортирующий его через всю систему Windows Vista PC к пункту конечного назначения, такому как монитор. понятное дело, что драйвер-соучастник может грабить столько медиа-контента, сколько только помещается на жестком диске;
- protected content:
  - Защищенный Контент: любой медиа-контент, защищенный какой-либо формой DRM (Digital Rights Management – Управление Цифровыми Правами);
- premium content:
  - цифровой контент следующего поколения, такой, например, как HD DVD или любой другой формат, защищенный стандартом AACS;
- Protected Environment (PE):
  - Защищенное Окружение: среда, защищенная от хакерского воздействия (ну, теоретически защищенная), в которой работают PMP-компоненты;
- Protected Media Path (PMP):
  - Защищенный Медиа-Путь: общий термин, охватывающий множество технологий и платформ, обеспечивающих устойчивую, интероперабельную обработку цифрового контента нового поколения на Windows Vista.