Эссе по курсу защита информации

студента 014гр. Черкес И.В.

Intel LaGrande Technology

«Технология LaGrande — это аппаратная защита будущего», — утверждает Давид Гроурок (David Grawrock), ведущий специалист по системам безопасности из корпорации Intel и один из главных идеологов LT.

Официально технология LaGrande была объявлена полтора года назад на Форуме Intel для разработчиков (IDF), проходившем в калифорнийском городе Сан-Хосе. Однако тогда информация о ней была крайне скупа, хотя представители Intel и сказали, что технология будет присутствовать уже в процессорах с ядром Prescott. Более подробно об LT было рассказано осенью 2003 года на Форумах в Сан-Хосе и Москве, а на недавнем февральском IDF различным аспектам LT и сопутствующим решениям было посвящено сразу несколько докладов в рамках технических сессий. По тому интересу, который проявили к докладам посетители IDF (а это, как правило, ведущие специалисты отрасли со всего мира), и по многочисленным вопросам к докладчикам можно сделать вывод о больших перспективах LT.

LT — технология дней грядущих, а не настоящих. Ее практическая реализация невозможна без создания принципиально новой операционной системы, хотя сама LT базируется именно на *аппаратных* изменениях платформы — процессора, чипсета, видеоускорителя и пр. Поэтому пользоваться преимуществами технологии мы сможем не раньше 2005 года — когда выйдет Windows Longhorn (и аналогичные варианты Linux), которая проектируется с учетом всех требований LaGrande.

Intel оценивает эффективность LT как высокую, если технология применяется для защиты от программных атак (вирусов, троянских коней, атак по локальной сети), информация статично хранится на компьютере, а пользователь «добропорядочен». Несколько меньшая защита обеспечивается, если с данными активно работают, могут происходить различные нештатные ситуации (например, потеря питания во время работы) или злоумышленники имеют свободный доступ к внутренностям компьютера.

Наконец, LaGrande практически не способна защитить от «аппаратных» атак, связанных с непосредственным подключением к «железу» специальных устройств (например, аппаратных «жучков», записывающих все, что вводится с клавиатуры компьютера). Кстати, от нерадивых пользователей (стирающих свои же документы или отключающих защиту в приложениях), а также от любителей записывать

пароли на бумажке и от прочих «человеческих факторов» LT тоже не спасает.

Масштабы задумки впечатляют — фактически LT защищает буквально все аспекты функционирования компьютерной системы и требует для своей реализации изменений в архитектуре центрального процессора, контроллера памяти и контроллеров ввода-вывода (AGP и USB), добавления в систему особого устройства — модуля TPM, существенной переработки BIOS и операционной системы, а в идеале — еще и использования специальных видеокарты, клавиатуры и мыши! К южному мосту по шине LPC подключается модуль TPM. Спецификации TPM, кстати, разрабатываются отдельно от LaGrande; технология уже продается на рынке, правда, только для целей защищенного хранения данных и проверки подлинности аппаратного обеспечения ПК. LT для своей работы требует TMP версии не ниже 1.2. Самое удивительное, что по всем пунктам уже выработаны необходимые спецификации, поддерживаемые крупнейшими участниками рынка. Intel и AMD обеспечивают поддержку LT в своих процессорах и чипсетах, Microsoft делает операционную систему New Generation of Secure Computing Base (NGSCB), а судя по участию в альянсе TCG таких имен, как Nvidia, ATI, Phoenix, ALi, SiS и еще двух десятков не менее известных, за остальные слагаемые LT можно не волноваться. Конечно, доводка технологии займет немало времени, но калибр взявшихся за дело фирм не оставляет сомнений, что LT пойдет в массы.

Способов атаковать компьютерную систему, по сути, всего два. Первый — «вынести» из нее все хранящиеся данные и потом уже с ними разбираться. Но если данные зашифрованы, а ключи хранятся в надежном месте (классический пример — программа использует вводимый пароль), то этот способ, как правило, бесполезен. Более того, в LT предусмотрено и специальное защищенное хранилище — Trusted Platform Module (TPM), прочитать данные из которого сможет лишь записавшая их туда программа (взломать TPM без специального оборудования практически невозможно). TPM можно «привязать» к конкретному компьютеру — в модуле хранится специальный хэш, вычисляемый на основе информации об основном оборудовании компьютера, и его можно настроить так, чтобы при несовпадении сохраненного и вычисленного при загрузке ОС хэшей модуль отказался выдавать данные. Так что копировать защищенную информацию (или красть винчестер с нею) бесполезно даже при наличии всех необходимых паролей. То же самое можно сказать и о попытке установить специальные «хакерские железки», маскирующиеся под компоненты системы (видеокарту, например, — она имеет прямой доступ в оперативную память компьютера). Правда, если ставится задача — уничтожить информацию или нарушить работу компьютера, то никакая LT в случае физического воздействия на оборудование, увы, не поможет.

Второй (и гораздо более распространенный) способ атаки — внедрить в систему «шпиона» или, если нужно нарушить функционирование системы, — «диверсанта». Тем или иным способом в систему запускается

хакерская программа, пытающая либо тихо «вытащить» данные, либо осложнить работу системы. Первая линия обороны здесь — не допустить самой возможности запуска вредоносного приложения. Любая операционная система сегодня позволяет запустить любое приложение по выбору пользователя (с его правами доступа), а печальный опыт эпидемий сетевых червей ярко показывает, насколько это слабое звено. Кроме того, запустить хакерское приложение может другой пользователь того же компьютера. Наконец, в код программы может быть незаметно внедрен вирус. Поэтому хотелось бы автоматически ограничить запуск опасных программ.

Проблему несколько смягчают антивирусы, но в базе данных антивируса может и не найтись конкретной вредоносной программы (особенно если ведется целенаправленная атака на систему), и тогда он окажется бесполезен. А вот технология LaGrande, по крайней мере теоретически, позволяет организовать идеальную защиту. Во-первых, можно «подписать» все приложения и просто не разрешать запуск неподписанных приложений. Во-вторых, можно воспользоваться хорошо отработанным на системах сетевой безопасности принципом «обучения» — сначала мы «учим» систему, указывая, какие приложения можно запускать (система сохраняет соответствующую информацию в TPM), а затем переключаем ее в «рабочий» режим, запрещая запуск всех других приложений (или хотя бы научив ее предупреждать пользователя при попытке запуска подозрительного кода). Этот способ немного лучше, но его сложнее реализовать и с ним неудобно работать. Так что защита от запуска не идеальна и к тому же не решает проблему целиком.

Существует целый ряд атак, направленных на «подкуп» честной программы, когда вредоносный код «внедряется» в работающее приложение. Он перехватывает управление, и программа начинает вкалывать на своего «поработителя» (защита от запуска здесь не поможет — программа уже запущена). Так что мы переходим ко «второй линии обороны» — защите работающих приложений от внешнего воздействия.

Рассмотрим функционирование современной многозадачной операционной системы. На «верхнем» уровне «живет» множество одновременно работающих приложений. С окружающей средой они общаются через программные интерфейсы системы. В свою очередь, интерфейсы уже взаимодействуют с ядром ОС. Именно оно отвечает за все «системные» задачи — обеспечивает одновременное функционирование многих приложений, обработку и перенаправление возникающих событий, выделение и защиту памяти и многое, многое другое. Для работы с внешними устройствами в состав ядра включают драйверы, через которые ядро «общается» с чипсетом и подключенными к нему устройствами. Самый нижний уровень — непосредственно компьютерное «железо». Как можно атаковать прикладное приложение?

Начнем сверху — с непосредственной атаки на приложение. Способ более чем популярный (и почти единственно возможный в семействе Unix-систем). Никакой защиты на этом уровне технология LaGrande не предоставляет.

Следующий уровень атак — через программные интерфейсы или ядро системы. Идея здесь такова: ядро (а в операционных системах корпорации Microsoft — еще и большая часть всевозможных интерфейсов) работает в так называемом нулевом кольце (ring 0) системной защиты. Проще говоря, весь этот код находится в привилегированном положении, и никакие ограничения безопасности на него не действуют. Так что если мы не можем взломать приложение — мы взломаем его окружение. Принципы атаки остаются прежними, ведь и ядро, и программные интерфейсы — в общем-то, вполне обычные программы. И если ядро (сравнительно небольшое по размерам и объему выполняемых функций) можно сделать очень надежным (что, собственно говоря, и реализовано в Unix-системах), то в программных интерфейсах исключить ошибки и уязвимости практически нереально. В Unix проблем это не вызывает — всевозможные системные библиотеки просто наследуют права доступа от работающего с ними приложения, и атаковать, по сути дела, оказывается нечего. С продукцией Microsoft ситуация иная: в погоне за производительностью компания давно перевела в ring 0 почти все критичные к скорости компоненты операционной системы, например DirectX. Ради той же скорости действия приложения в интерфейсах обычно еще и не проверяются на допустимость — фактически большинство функций может вызвать любая программа и с любыми параметрами. В результате у хакерского приложения появляется возможность обойти защиту.

Кроме того, ОС от Microsoft (все по той же причине — из-за внесения программных интерфейсов в ring 0) нередко позволяют напрямую обращаться к оборудованию компьютера. Это может быть чтение или модификация видеопамяти (хакерская программа «видит» все, что видит пользователь, и способна подделывать изображение на экране), перехват нажатия кнопок на клавиатуре и движений мыши или, наоборот, их эмуляция (перехватываются вводимые пароли, хакерская программа имитирует работу человека, чтобы использовать нормальную программу в своих целях). Есть и весьма экзотические виды атак — например, использование контроллера DMA. Этот контроллер имеет свободный доступ к памяти и программируется пользователем, так что хакерской программе нужно всего лишь «попросить» DMA залезть в недоступную ей память. В Unix все эти атаки практически невозможны, поскольку любые действия приложения, проходящие через ядро (а любая непрямая атака проходит через ring 0, в котором в Unix-системах «живет» только ядро), строго контролируются операционной системой, которая не позволит напрямую обращаться к оборудованию без надлежащих прав доступа.

Итак, главная причина введения LaGrande, как ни печально, — необходимость компенсировать недостатки операционных систем

семейства Microsoft Windows. Нет, LT, конечно, помогает и в других случаях, но если бы дело ограничивалось только этим, то безопасность работы возросла бы ненамного (что, кстати, ставит под вопрос целесообразность введения поддержки LaGrande в Unix) и не потребовалась бы такая сложная система защиты. Но работать в «самой популярной» операционной системе хочется многим, а переделывать Windows уже поздно (проще совершенно новую ОС написать). Так что смиримся с этим и посмотрим, что нам предлагается в качестве решения в LT.

А предлагается фактически «сдать» операционную систему без боя, но зато выстроить оборону вокруг критических приложений. По запросу приложения ему предоставляется безопасная область памяти, «залезть» в которую не может никто — ни другая программа, ни сама операционная система, ни даже любое внешнее оборудование (вроде DMA-контроллера). К программе приставляют небольшого помощника (кто-то же должен выполнять в этой области основные функции ОС), и мы получаем то, что Intel называет доменом, — изолированный участок пространства с собственной маленькой операционной системой и единственным запущенным приложением. Домен может свободно общаться с окружающей средой (кроме других защищенных доменов, конечно), а вот окружающая среда повлиять на него не может. Вернее, почти не может — ведь обычно программе, работающей в домене, требуется как-то общаться с пользователем, путь к которому пролегает по «враждебной территории» операционной системы. Хакерская программа по-прежнему может подслушивать беседу пользователя и защищенной программы; более того — обманывать их (представляясь пользователю нужной ему программой или выдавая свои действия за действия пользователя программе). В итоге в такую схему вынужденно добавляется необходимость организации специальных защищенных каналов связи домена с видеокартой, мышкой, клавиатурой — поэтому полноценная поддержка LaGrande, строго говоря, потребует замены и этих устройств. В общем, взявшись защищать программу, а не операционную систему в целом, мы вынуждены возводить вокруг нее «великую китайскую стену».

И последний штрих. Чтобы пользователь мог убедиться, что он работает в защищенной системе с защищенной программой (а не вводит пароли в хакерскую фальшивку, имитирующую настоящее приложение), в LaGrande предусмотрены специальные возможности проверки подлинности. Подойдя к компьютеру, можно легко выяснить, стоит ли с ним работать (или пора звать специалиста по безопасности). Выглядит это примерно так: пользователь подключает к компьютеру специальную смарт-карту, карта проводит проверку системы (на основе данных модуля TPM о защищаемом приложении; информации LT о том, запущено ли искомое приложение, видно ли оно на экране и т. п.) и сигнализирует о результатах. Вместо смарт-карты можно использовать и специальные веб-сервисы, проводящие удаленную аттестацию системы.

Для полноты картины остается добавить, что пользователь может включать и выключать LaGrande в любой момент, даже не перезагружая систему, а внедрение технологии, видимо, произойдет через несколько лет, в процессорах с кодовым названием Tejas.

В таблице систематизировано все вышесказанное.

Проблемы безопасности и методы борьбы с ними		
Проблема	Решение	Аппаратная реализация
Возможность запуска пользователем вредоносного приложения и взлома защищенного приложения еще до запуска	Проверка подлинности запускаемого приложения («защищенный старт»)	TPM +операционная система NGSCB
Возможность внедрения в работающую программу чужеродного кода	В общем виде не существует, но можно предотвратить наиболее «популярные» атаки	Технология Execution Protection (непосредственно к LaGrande не относится)
Возможность получения прямого доступа к памяти приложения и, соответственно, возможность кражи данных либо изменения работающей программы	В Unix-системах атака затруднена. В Windows-системах защита обеспечивается запуском приложения в специальном изолированном домене («защищенное выполенение»)	Изменения в процессоре и контроллере памяти; ОС NGSCB и специальная BIOS с поддержкой доменов
Возможность перехвата и модификации вводимых пользователем данных, атака путем имитации децствий пользователя	В Unix атака затруднена. В Windows защита обеспечивается созданием специального защищенного канала ввода	Изменения в контроллере USB и устройствах ввода – мыши и клавиатуре; поддержка BIOS и NGSCB
Возможность перехвата и модификации выводимых пользователю данных	В Unix атака затруднена. В Windows защита обеспечивается созданием специального защищенного канала вывода	Изменения в контроллере AGP и в видеокарте; поддержка BIOS и NGSCB
Возможность маскировки хакерского приложения под нормальную программу	Специальные механизмы проверки подлинности запущенных приложений	TPM + специальные сервисы или устройства аттестации
Возможность добавления в систему специального хакерского оборудования либо неправильной настройки оборудований. Возможность кражи информации (в другой компьютер)	Проверка подлинности и корректной работы компьютера, на котором выполняется приложение	TPM (непосредственно к LaGrande не относится)
Возможность атаки при возникновении нештатных ситуаций в программе	В Unix атака затруднена. В Windows требуется усиление защиты	Ряд специальных решений в LT

Список литературы.

1. www.terralab.ru/system/20087
2. www.terralab.ru/system/23898
3. www.terralab.ru/system/29226
4. www.terralab.ru/system/30348
5. www.terralab.ru/system/32360
6. www.trustedcomputinggroup.org
7. www.microsoft.com/presspass/features/2002/jul02/024palladiumwp.asp
8. www.gnu.org/philosophy/can-you-trust.html/www.gnu.org/philosophy/can-you-trust.html
9. www.computerra.ru/offline/2003/491/26495
10. www.computerra.ru/offline/1999/283/2289