Московский физико-технический институт

(технический университет)

Эссе по курсу
«Основы защиты информации»
студента 014 группы
Жданова А.С.
на тему

Ошибки в реализации защиты в распространённых коммерческих программных продуктах.

Долгопрудный 2004

Введение

Огромной количество статей и книг по защите информации дают множество правил, строгое выполнение которых должно обеспечить требуемый уровень защиты. Однако, как показывает практика, точное следование правилам не всегда приводит к желаемому результату. Зачастую такая ситуация возникает вследствие жесткости накладываемых на математические модели ограничений. Часто приходится ослаблять защиту системы, чтобы не отогнать потенциального покупателя сложностью работы с ней. С другой стороны, зачастую программисты ошибаются, ведь для тестирования программы, имеющей дело с безопасностью, необходимо её тестирование во всевозможных режимах, а это практически невозможно.

Целью данной работы является собрание сведений об ошибках в реализации защиты программных продуктов и краткое описание этих ошибок. Попутно рассказывается о наиболее распространенных приёмах защиты.

Основные способы обеспечения защиты

Все методы защиты информации можно разделить на три класса:

1. административные
2. законодательные
3. технические

Законодательные методы определяют, кто должен иметь доступ к защищаемой информации, форму доступа, а также устанавливают ответственность за нарушения установленного порядка.

Административные методы заключаются в определении процедур доступа к защищаемой информации и их строгом выполнении.

Первые два класса методов довольно ненадёжны, например, соблюдение законодательных мер обусловлено только законопослушанием и страхом перед наказанием, а за соблюдением административных мер следят люди, которых можно подкупить или обмануть. В отличие от двух первых, технические методы позволяют в максимальной степени избавиться от человеческого фактора. В этом случае перед потенциальным противником ставится некоторая техническая задача, которую ему необходимо решить для получения доступа к информации, в то же время легитимному пользователю должен быть доступен более простой путь, позволяющий работать с предоставленной в его распоряжение информацией без решения сложных задач.

Замеченные ошибки, допущённые при организации защиты в некоторых коммерческих продуктах.

Аутентификация сетевых подключений в Windows 95/98

5 января 1999 года компания L0pht опубликовала статью о найденной уязвимости в реализации схемы запрос/ответ при подключении сетевых ресурсов Windows 95/98. Выяснилось, что при попытках подключения Windows 95/98 посылает один и тот же запрос в течение примерно 15 минут. За это время можно подключиться к сетевому ресурсу от имени пользователя, чью попытку аутентификации удалось перехватить.

Хеши паролей для LANMAN-аутентификации

В таких операционных системах, как Windows NT/2000, могут храниться две версии хэшей пароля. Одна версия используется собственными средствами безопасности Windows NT, а другая нужна для обеспечения совместимости с протоколом аутентификации LANMAN, который применяется в Windows 95/98.

Собственный хеш Windows NT достаточно стойкий. Однако процедура вычисления хэша LANMAN имеет несколько особенностей, значительно ослабляющих уровень защиты. Длина LANMAN-пароля не должна превышать 14 символов. При вычислении хэша, он разбивается на 2 части по 7 символов, и хэш для каждой части вычисляется отдельно. Таким образом, становится возможным полный перебор всех вариантов пароля. Если пароль короче 7 символов, то вторая часть остаётся пустой и порождает всегда одно и то же значение хэша, что позволяет по второй половине хэша сразу определить пароли, длина которых меньше 8 символов. Так как обе части пароля обрабатываются независимо, для паролей длины от 8 до 12 символов вторая часть может быть найдёна максимум перебором всех пятисимвольных комбинаций, что занимает совсем немного времени. Зачастую окончание пароля позволяет угадать весь пароль.

Перед вычислением хэша все буквы пароля переводятся в заглавные, что примерно в 9,4 раза сокращает количество возможных комбинаций (при использовании всех символов ASCII). Функция вычисления хэша LANMAN не использует подмешивания «соли» - случайной несекретной величины, делающим значение уникальным для каждого пользователя, даже если пароли совпадают.

После того как по хэшу LANMAN подобран пароль, можно за короткое время подобрать и пароль NT, если его длина не превышает 14 символов. На это потребуется не более $2^{14} =$ 16 384 вариаций прописных и строчных букв.

Подписание модулей расширения для программ семейства Adobe Acrobat

Программы семейства Adobe Acrobat имеют возможность подключать модули расширения, предназначенные для увеличения функциональности. Чтобы модуль расширения мог быть загружен в программу Adobe Acrobat Reader, он должен быть подписан разработчиком. А в некоторых режимах, связанных с поддержкой DRM (Digital Rights Management, управление цифровыми правами), разрешается загрузка только модулей, сертифицированных и подписанных компанией Adobe.

Исследователями компании ElcomSoft было установлено, что при проверке сертификата модуля расширения участвуют только некоторые поля заголовка переносимого исполняемого файла, не нарушающие целостность сертификата. Это приводит к возможности коррекции исполняемого кода таким образом, чтобы модуль расширения начал выполнять любые действия, включая опасные.

Описание данной уязвимости было опубликовано CERT (Computer Emergency Response Team, бригада неотложной компьютерной помощи), и Adobe обещала устранить дефект в ещё не вышедшей на тот момент версии Acrobat Reader 6. Но Acrobat Reader 6 уже вышел, и он продолжает загружать модули расширения, подписанные разработчиками старым уязвимым способом.

Компрометация сертификатов Microsoft Corporation

30 и 31 января 2001 года удостоверяющий центр компании VeriSign выдал два цифровых сертификата лицу, выдавшему себя путём обмана за работника компании Microsoft. Эти сертификаты могут быть использованы для подписания компонентов ActiveX, макросов MS Office и других исполняемых модулей. VeriSign добавил эти сертификаты в список отозванных сразу же после обнаружения обмана. Но сертификаты, выпускаемые VeriSign для подписания исполняемого кода, не содержат указания на центр распространения списков отмены (CRL Distribution Point, CDP). Из-за этого программное обеспечение Windows не способно автоматически получить информацию о том, что сертификат был отозван, пока Microsoft не выпустит, а пользователь не установит соответствующее исправление.

Программа eBook Pro

Разработчики программы eBook Pro во всю рекламируют своё детище как «единственный программный продукт во вселенной, способный обеспечить Вашей информации практически 100% защиту от взлома». Поверить рекламе, которая обещает именно то, что хочется получить, очень легко. Правда, вскоре выяснится, что, нажав комбинацию клавиш <Ctrl>+<A>, можно выделить весь видимый текст, а затем скопировать его в буфер обмена. Корме того, незащищенные копии HTML-страниц и картинок остаются после просмотра в директории, хранящей кэшированные файлы Internet Explorer. И, наконец, можно будет узнать, что защита заключается в наложении при помощи XOR на каждый байт защищаемых данных последовательно всех байтов строки «encrypted», что эквивалентно наложению однобайтовой константы.

Результаты тестирования ключей HASP

Аппаратные ключи для защиты программного обеспечения от несанкционированного тиражирования HASP (Hardware Against Software Piracy), производимые компанией Aladdin Knowledge Systems, Ltd., являются самыми распространёнными в России. По результатам тестирования, проведённой Национальной Тестовой Лабораторией США (NSTL), ключи HASP были названы лучшими 2 раза подряд. Согласно отчёту NSTL, датированному январём 1999 года, сравнительное тестирование ключей разных производителей велось по пяти категориям: безопасность, простота использования, совместимость, возможности сетевых ключей и универсальность. Ключи HASP оказались лидерами во всех пяти категориях, и, следовательно, стали безусловными победителями.

Казалось бы, такой серьёзной организации, как NSTL, можно доверять. Но есть несколько нюансов, ставящих под сомнение истинность вердикта, вынесенного NSTL.

Прежде всего, при тестировании сравнивались ключи только двух семейств HASP от компании Aladdin и Sentinel от компании Rainbow Technologies Inc. Не исключено, что ключи Rainbow Sentinel являются наиболее значимым конкурентом для Aladdin HASP, но на момент тестирования на рынке были представлены ключи и других производителей, сравнение с которыми не проводилось. Но самое главное, на момент опубликования отчёта NSTL в Интернете можно было найти большое количество статей, руководств и даже исходных текстов программ, в деталях описывающих внутреннее устройство ключей HASP, включая алгоритм вычисления секретной функции HaspCode и быстрого поиска пароля для доступа к ключу. Итак, существовал инструментарий, позволяющий при наличии физического доступа к ключу за пару минут получить всю информацию, необходимую для построения полного эмулятора, способного на любой корректный запрос к ключу вычислить ответ, совпадающий с ответом реального ключа.

Алгоритм шифрования A5

Шифр A5, применяемый для шифрования сеансов телефонной связи между трубкой абонента и базовой станцией в европейской системе мобильной цифровой связи GSM (Group Special Mobile), был разработан в 1989 году и существует в двух версиях: A5/1 – «сильная » версия шифра, разрешённая к применению только в некоторых странах, и A5/2 – «ослабленная» версия, разрешённая к свободному применению. В 1989 году широкая публикация алгоритмов не была распространённым подходом, и детали построения A5 оказались засекречены.

Но как бы строго не контролировались коммерческие секреты, широкое распространение продукции рано или поздно приводит к утечкам информации. В случае с GSM утечки начались в начале 90-х годов. Британская телефонная компания передала всю документацию Брэдфордскому университету, не потребовав от него подписать соглашение о неразглашении. Часть информации попала в Интернет, а к 1994 году основные детали алгоритма A5 стали общедоступны. В конце концов, кембриджские учёные Майк Роэ (Mike Roe) и Росс Андерсон (Ross Anderson) опубликовали в Интернете примерную схему алгоритма.

В начале 1999 года в ассоциации разработчиков смарт-карт (SDA) были полностью восстановлены и проверены на реальных тестовых векторах схемы алгоритмов A5/1 и A5/2. Почти сразу после этого была предложена атака, позволяющая вскрывать шифр A5/2 на персональном компьютере за 15 миллисекунд.

В декабре 1999 года израильскими математиками Ади Шамиром (Adi Shamir) и Алексом Бирюковым (Alex Biryukov) была опубликована ещё одна работа, в которой описан нетривиальный, но по теоретическим расчётам очень эффективный метод вскрытия алгоритма A5/1. Этот метод требует 2^{48} предварительных вычислений и позволяет отыскать ключ за 1 секунду на персональном компьютере, имеющем 128 Мбайт оперативной памяти и 150 Гбайт дискового пространства, путём анализа выхода алгоритма в течение первых двух минут телефонного разговора.

Взлом 128-битового шифрования в Netscape

Компания Netscape разработала протокол SSL и реализовала его в своём браузере. Данные, передаваемые посредством SSL, зашифровывались алгоритмом RC4 со 128-битовым ключом. 17 сентября 1995 года Йен Голдберг (Ian Goldberg) о том, что ему в сотрудничестве Дэвидом Вагнером (David Wagner) удалось обнаружить уязвимость в процедуре выбора 128-битового ключа для алгоритма RC4. Недостаток процедуры заключался в том, что начальное состояние генератора псевдослучайных чисел основывалось на трёх значениях: идентификаторе процесса, генерирующего ключ, идентификаторе его родительского процесса и текущем времени. Учитывая то, что значительную часть информации о номерах процессов и времени можно было предугадать, пространство возможных ключей сократилось с 2^{128} до 2^{20}, и на поиск ключа шифрования уходило всего 25 секунд.

Генератор псевдослучайных чисел в InfoZIP

Согласно спецификации ZIP, после загрузки ключа необходимо зашифровать 12 байт до того, как начать шифровать данные файла. Последний из этих 12 байт является младшим байтом контрольной суммы файла, которая хранится в заголовке архива в

незашифрованном виде. Это позволяет определять неправильный пароль с вероятностью 255/256 после расшифровки всего 12 байт. Остальные байты выбираются случайным образом.

Существует открытая реализация библиотеки для работы с архивами формата ZIP, называемая InfoZIP. В этой библиотеке из 12 дополнительных байт не один, а два последних байта содержат младшие байты контрольной суммы файла. Для генерации случайных байт используется алгоритм, идентичный rand() из стандартной библиотеки MS Visual C++. Зная 4 байта выхода этого алгоритма, можно получить начальное состояние генератора и полностью предсказать выход.

Псевдослучайные байты, полученные при помощи этого генератора, используются в InfoZIP таким образом, что для каждого файла в архиве генерируется 10 байт и один из них хранится в архиве в незашифрованном виде. Это позволяет при наличии в архиве пяти файлов, зашифрованных с одним паролем и подряд (без повторной инициализации генератора), найти начальное состояние генератора. А зная выход генератора и значение двух младший байт контрольной суммы для каждого из пяти файлов, можно определить ключ шифрования этих файлов менее чем за час.

Цифровая подпись ElGamal в библиотеке FGInt

FGInt представляет собой библиотеку для работы с большими целыми числами и включает в числе прочих поддержку алгоритма цифровой подписи ElGamal. Но вычисление и проверка это подписи в FGInt были реализованы с некоторыми отступлениями от спецификации.

Во-первых, при проверке целостности подписи должна выполняться проверка того, что значения двух составляющих подписи r и s не превышают значения использованного модуля p. На случай если та проверка не выполняется, в книге “Handbook of Applied Cryptography” приводится алгоритм атаки, позволяющий при наличии одного подписанного сообщения вычислить цифровую подпись для любого другого сообщения.

Во-вторых, в подписи должно использоваться не само сообщение, а его хэш. Причина этого заключается в том, что без использования хэш-функции оказывается возможным подобрать сообщение, соответствующее заданному значению подписи, вычисленному определённым образом. Данная атака также описана в книге «Handbook of Applied Cryptography». А если подписывается значение хэша, то для отыскания подходящего сообщения придётся обратить ещё и хэш, что при использовании стойкой криптографической хэш-функции сделать почти невозможно.

Описанные выше недостатки в FGInt были обнаружены и успешно использованы представителями группы CORE, что позволило им выпустить генераторы регистрационных кодов к нескольким версиям программы SmartWhois, защищённой с использованием цифровой подписи ElGamal с ключом длиной 960 бит, реализованной через библиотеку FGInt.

Вывод

Как видно из вышесказанного, зачастую людям свойственно ошибаться. Даже в реализациях хорошо зарекомендовавших себя алгоритмов, возможны ошибки, которые приводят к неработоспособности систем защиты, построенных на основе этих алгоритмов.

Дело ещё более осложняется в случае отказа разработчика от публичного обсуждения алгоритма. Зачастую такие алгоритмы имеют существенные изъяны, которые в конечном итоге всё-равно будут обнаружены. Только открытое и всестороннее исследование алгоритма – правильное направление на пути построения действительно защищённой системы.

Список литературы

1. Скляров Дмитрий, «Искусство защиты и взлома информации», изд. «БХВ-Петербург», 2004
2. Шнайер Б., «Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си», изд. «ТРИУМФ», 2002
3. Масленников Михаил, «Практическая криптография», изд. «БХВ-Петербург», 2002