Предисловие

За последние несколько лет криптография в частности и технологии обеспечения безопасности вообще непрерывно повышали свою значимость для пользователей Windows и производителей программного обеспечения. С другой стороны функции обеспечения безопасности (и криптографические функции в том числе) 32-разрядных систем семейства Windows стали сопоставимы с аналогичными функциями больших компьютерных платформ, где такие функции издавна имели высокий приоритет. Теперь же, с широким внедрением .NET, реализация функций обеспечения безопасности на персональных компьютерах стала делом более простым, чем когда-либо ранее. Разумеется, все еще требуются значительные усилия, для того чтобы понять и освоить базовые концепции, а также приобрести навыки, необходимые для обращения с соответствующими функциональными возможностями .NET. (Кстати, это как раз и есть тема данной книги.) И хотя многие из этих функций были доступны и раньше в скрытой форме библиотек Windows, именно появление платформы .NET делает программирование с использованием криптографии и других технологий обеспечения безопасности гораздо более простым, а его результаты – гораздо более мощными, чем когда-либо ранее. Платформа .NET Security Framework позволяет использовать широкий набор специальных классов, которые относительно нетрудно освоить, и все эти возможности мы исследуем в нашей книге.

Книга призвана исчерпывающе осветить все практические вопросы в реализации криптографических и иных, связанных с безопасностью, функциональных возможностей в приложениях .NET. Она представляет собой эффективное учебное пособие, содержащее множество ясных и наглядных примеров исходного кода.

Организация книги

Структура книги включает в себя десять глав и пять приложений. Глава 1 вводит читателя в тематику криптографии и безопасности на платформе .NET и содержит еще нетехнический обзор тех тем, которые гораздо детальней освещены в последующих главах. Также в этой первой главе приведены рассуждения о структуре книги и о том, как в ней соотносятся темы криптографических функций и других функций, связанных с безопасностью. Цель этой главы не состоит в том, чтобы достичь сколько-нибудь глубокого понимания или изучить примеры кода, но она должна привести читателя к концептуальному пониманию криптографических и

других, связанных с безопасностью, технологий, на платформе .NET. Глава 2 обеспечивает знакомство с теорией, достаточное для более глубокого понимания материала последующих глав. Главный ее посыл состоит в том, что все технологии безопасности основываются, в конечном счете, на криптографии, а для многостороннего понимания криптографии необходимо вначале освоить несколько базовых теоретических криптографических концепций. Главы 3, 4, 5 и 6 содержат детально проработанные примеры программирования в .NET, относящиеся к симметричным алгоритмам, асимметричным алгоритмам, цифровым подписям и XML-криптографии, соответственно. Главы 7 и 8 описывают программирование в .NET с использованием концепции идентификации пользователей и концепции прав доступа к коду, соответственно. Глава 9 введет вас в тему программирования на основе технологии ASP.NET, а глава 10 познакомит с программированием Web-служб на основе ASP.NET.

Все аспекты криптографии и безопасности в .NET рассматриваются в подходящем контексте и в должной последовательности в приложениях, где они в реальности чаще всего применяются. Приложения к книге описывают несколько дополнительных тем таких, как способы атаки на выполняющийся код и некоторые математические темы, связанные с криптографией.

Эта книга задумана, как практическое учебное пособие с множеством примеров программ, которые иллюстрируют конкретные вопросы и концепции. Мы здесь концентрируемся скорее на вопросах практического программирования задач безопасности в .NET, чем на системном администрировании. Эта книга обеспечивает достаточно общей информации, чтобы читатель понял, почему вопросы безопасности и криптография столь важны при разработке современного программного обеспечения. Цель книги состоит в том, чтобы научить читателя создавать серьезные приложения с использованием платформы .NET Security Framework. Эта книга является частью серии «The Integrated .NET Series» от Object Innovations и Prentice Hall PTR.

Примеры программ

Лучший способ изучить какую-то серьезную библиотеку классов (например, такую, как .NET Security Framework) состоит в том, чтобы изучить и написать на ней много программ. В этой книге содержится большое число небольших программ, которые иллюстрируют каждую из используемых на практике возможностей .NET Security Framework в отдельности, что облегчает их понимание. Программы (полностью или частично) приводятся в тексте книги (с переведенными комментариями и сообщениями), и все они имеются в программном приложении к книге (с оригинальными комментариями и сообщениями). Эти примеры программ представлены в виде самораспаковывающегося архивного файла на Web-сайте этой книги. После распаковки архива будет создана структура каталогов по пути (по умолчанию) C:\OI\NetSecurity. Все примеры про-

грамм (они начинают появляться в книге, начиная с главы 2) распределены по каталогам Chap02, Chap03 и так далее. Все примеры, относящиеся к одной главе, находятся в отдельных папках внутри каталога соответствующей главы. Имена папок ясно идентифицируют содержащиеся в них примеры программ.

Эта книга является частью серии «The Integrated .NET Series». Примеры программ для других книг этой серии находятся в своих каталогах внутри каталога \OI, поэтому все программы из всех книг серии будут находиться в одном месте. Эти программы предназначены только для целей обучения, и их нельзя использовать ни в каком программном продукте. Все программы, включая инструкции по их использованию, предоставляются на условиях «как есть», и любого рода претензии к ним не принимаются.

Web-сайт

Web-сайт для книг этой серии находится по адресу http://www.objectinnovations.com/dotnet.htm.

Ссылка на этот Web-сайт приведена для загрузки примеров программ для этой книги.

Благодарности

Питер Торстейнсон (Peter Thorsteinson)

Мы хотели бы поблагодарить Джилл Хэрри (Jill Harry) из Прентис Холл за ее поддержку в начинании этого проекта. Также мы благодарны редактору серии Роберту Обергу (Robert Oberg) за его ценную помощь.

Дж. Гнана Арун Ганеш (G. Gnana Arun Ganesh)

Я хотел бы поблагодарить моих родителей Дж. А. Гнанавел (G. A. Gnanavel) и Дж. Н. Вадивамбал (G. N. Vadivambal) за их безграничную любовь, терпение, поддержку и воодушевление. Также я благодарю мою сестру Дж. Дж. Сарадха (G. G. Saradha) за ее любовь, нежность и товарищескую поддержку. Моя глубочайшая благодарность адресована моему доброму ангелу, д-ру Роберту Обергу (Robert Oberg), который воодушевлял меня на протяжении всего этого замечательного проекта. Также я благодарен мистеру Анидо Ди (Anido Dey), мистеру Нарайана Рао Сурапанени (Narayana Rao Surapaneni) и мистеру Виноду Кумару (Vinod Kumar) за то, что они воодушевляли меня и побуждали двигаться вперед. Я хотел бы поблагодарить моего соавтора, Питера Торстейнсона (Peter Thorsteinson), за его руководство и помощь. Наконец, позвольте мне поблагодарить Всевышнего за то, что он предоставил мне такую возможность.

Мы хотели бы поблагодарить Эмили Фри (Emily Frey), Карен Гетман (Karen Gettman) и всех наших редакторов за их конструктивную помощь в усовершенствовании этой книги. Также мы хотели бы поблагодарить всех наших рецензентов за их детальные комментарии, которые во многом помогли обновить содержание книги.

Дж. Гнана Арун Ганеш (G. Gnana Arun Ganesh) является обладателем звания Microsoft .NET MVP (Most Valuable Professional – «наиболее ценный профессионал»), разработчиком, автором и консультантом по .NET. Также он ведет группу .NET Technology в Arun Micro Systems, которая занимается различными фазами технологии .NET. Он работал с технологией Microsoft .NET начиная с ее первоначальной бета-версии. Арун получил степень бакалавра электроники и коммуникационной техники в университете Бхаратьяра (Bharathiar), в колледже Kongo Engineering. Он является автором руководства .NET Reference Guide, опубликованного в InformIT. Он один из авторов Object Innovations (материалы для курса обучения фундаментальным программным технологиям). В качестве автора на темы .NET, он опубликовал более 50 статей о технологии .NET на различных Web-сайтах, посвященных этой теме. Как активный участник рецензирования в Prentice Hall, он написал множество технических рецензий, начиная с рецензии на книгу «C#: How to Program», написанную Харви и Полом Дейтел (Harry and Paul Deitel). Вот уже более двух лет, как администратор Arun Micro Systems, он обеспечивает онлайновое обучение .NET по всему миру.

4 июня 2003 года

Глава 1

Криптография и безопасность в .NET

Нечасто вы встретите книгу, в которой вопросы криптографии обсуждались бы одновременно с вопросами безопасности и при этом с одинаковым вниманием. Эти две темы, на первый взгляд, принадлежат к совершенно разным областям и обычно рассматриваются по отдельности. В конце концов, насколько важны для системного администратора проблемы криптографии, и как часто он задумывается над трудностями разбиения на множители произведения двух больших простых чисел? А как часто математику приходится думать о конфигурации системы с точки зрения безопасности, о таких, например, вещах, как управление доступом к ветвям реестра Windows или к виртуальным каталогам Internet Information Server (IIS)? Книги по криптографии, как правило, отличаются обилием математики и основываются на теоретических подходах. В противоположность им, книги по безопасности компьютерных систем ориентированы не на программиста, они посвящены практическим рецептам выполнения таких, например, операций, как установка сервера сертификатов, создание учетных записей и тому подобное. Между этими двумя крайностями мы видим программиста .NET, озабоченного проблемами, которые по своей природе не относятся ни к математике, ни к системному администрированию.

Тем не менее, программисты становятся все более заинтересованными во внедрении в свои программы как криптографических функций, так и функций, связанных с безопасностью системы. С одной стороны все функции обеспечения безопасности так или иначе основываются на криптографическом «фундаменте». В сущности, все реально применяемые протоколы и технологии безопасности такие, как Kerberos, шифрованная файловая система Windows, пакет Microsoft Certificate Server и все классы .NET Security Framework, полностью основываются на криптографических примитивах. С другой стороны любое программное обеспечение, так или иначе связанное с системой безопасности, в какой-то момент времени обязательно соприкасается с конфигурацией параметров безопасности конкретной системы, на которой оно работает. В этой главе мы рассмотрим вопросы криптографии и безопасности в технологии .NET и получим общее представление о том, как эти две темы связаны между собой с точки зрения .NET-программирования.

В следующих главах мы более детально изучим эти темы.

Природа этой книги

Эта книга написана специально для программистов, интересующихся вопросами криптографии и безопасности, но не для системных администраторов. Мы, следовательно, очень мало внимания уделим здесь тем навыкам, которые требуются профессиональному «сисадмину». Однако каждый программист должен иметь некоторое представление о задачах администрирования, если он хочет разрабатывать эффективные приложения, и программист, работающий в области безопасности, тут не исключение. Поэтому наша книга все же исследует некоторые аспекты администрирования в той мере, в какой они имеют отношение к .NET-программированию в области безопасности. С другой стороны, книга не адресована профессиональным математикам и криптографам[1] и не слишком углубляется в теоретические вопросы криптографии, однако уделяет все же некоторое внимание теории, поскольку даже ограниченное понимание теории криптографии весьма полезно для программиста.

В результате эта книга реализует смешанный подход к освещению предмета, сочетающий в себе основы теории криптографии и вопросы администрирования систем с точки зрения безопасности на платформе .NET. В первой главе мы начнем с обзора наиболее важных концепций безопасности и криптографии на платформе .NET, приведя примеры того, как реализуется система безопасности в приложениях .NET. В главе 2 мы рассмотрим теоретические основы криптографии, начиная с устройства шифра и принципов криптоанализа простейших шифровальных систем, известных с древних времен. Далее в главах 2, 4 и 5 мы продвинемся дальше, изучив технику программирования .NET применительно к трем основным криптографическим системам, используемым в наше время: симметричные и асимметричные системы, а также цифровая подпись. В этих трех главах приводятся развернутые примеры программного кода, реализующего все эти системы при помощи классов .NET Security. В главе 6 продолжится исследование вопросов, связанных с шифрованием и цифровой подписью, но уже в контексте технологии XML. Главы 7 и 8 продемонстрируют основные техники программирования, реализующие такие концепции .NET, как управление доступом на основе механизма ролей и управление доступом кода к ресурсам на основе свидетельств в программах .NET. Разумеется, реалии распределенных приложений и среды Internet делают особенно важными, с точки зрения безопасности, многие специфические аспекты программ, и в главах 9 и 10 мы рассмотрим вопросы безопасности технологии ASP.NET и Web-сервисов, основанных на .NET.

---

[1] Криптограф – это тот, кто разрабатывает и анализирует алгоритмы шифрования (но не тот, кто просто использует готовые библиотеки для встраивания в свою программу криптографических функций).

Опасность подстерегает повсюду

Если начать думать обо всех возможных опасностях, то можно стать параноиком. В конце концов, среднестатистический гражданин крайне редко становится объектом расследования ЦРУ или ФБР (насколько мы знаем, конечно...) или мишенью международного шпионского заговора. Если дать волю воображению, то многие потенциальные угрозы покажутся просто притянутыми за уши. В самом деле, почему бы ни обернуть свою голову алюминиевой фольгой для того, чтобы злонамеренные инопланетяне не смогли прочитать ваши мысли? Тем не менее, как бы странно это ни звучало, угрозы безопасности наших данных подстерегают повсюду: чем важнее и ценнее данные, тем большее значение приобретают аспекты безопасности. На самом деле, в компьютерном мире данные на удивленье часто подвергаются самым разнообразным угрозам и рискам.

ПОСТАВЬТЕ СЕБЯ НА МЕСТО ЗЛОУМЫШЛЕННИКА

Наверное, вам доводилось слышать совет опытного рыбака: чтобы поймать рыбу, нужно думать, как рыба. Лично мне этот совет всегда казался немного странным, поскольку непонятно все же, как мыслит рыба. Однако этот совет весьма полезен, если ваш противник – человек. Сказанное особенно верно, если речь идет о защите от потенциального злоумышленника[1], атакующего вашу систему: очень полезно изучить ситуацию с его точки зрения и попытаться представить себе возможный ход его мыслей.

Проблема заключается в том, что хорошим ребятам вроде нас с вами очень трудно мыслить так, как мыслит изобретательный злоумышленник, в то время как на его стороне оказываются неисчерпаемые ресурсы: изобретательность, время, энергия и знания. Зачастую все, что мы в состоянии сделать, это бежать вдогонку за мыслью злоумышленника. Шансы в этой игре неравны: одному единственному противнику достаточно найти одну единственную уязвимость, и вся система в опасности. В противоположность этому, измученный защитник должен предусмотреть все возможные варианты нападения и для каждого варианта предпринять достаточные оборонительные меры. Для того чтобы получить общее представление о возможных угрозах, давайте рассмотрим несколько примеров потенциальных угроз.

---

[1] Мы используем определение «злоумышленник, атакующий вашу систему» в отношении любого лица, которое стремится получить недозволенный доступ к системе с целью кражи, подделки или уничтожения данных. Такого злоумышленника часто называют «крэкер» (взломщик). Очень часто в качестве синонима для слова «крэкер» используют слово «хакер», что абсолютно неверно, и чему мы обязаны сообщениям современных СМИ на технические темы. Под словом «хакер», вообще говоря, имеется в виду компьютерный эксперт или энтузиаст, отличающийся обширными знаниями, как правило, приобретенными самостоятельно, и предпочитающий в своей деятельности «партизанские», нерегулярные методы.

ПРИМЕРЫ УГРОЗ И СООТВЕТСТВУЮЩИХ ИМ КОНТРМЕР

Вероятно, нет никаких пределов числу всех возможных трюков и уловок, которые может применить злоумышленник, атакующий вашу систему. Угрозы, с которыми имеет дело программист, концептуально идентичны тем угрозам, которым подвергается пользователь электронной почты (e-mail). Например, большинство людей не шифруют свои электронные сообщения, что можно представить, как отправку по обычной почте открытки вместо вложенного в конверт письма. В этом может заключаться определенный риск, поскольку такие электронные сообщения могут быть легко перехвачены на вашем почтовом сервере или на одном из маршрутизаторов на пути сообщения. Другой вариант угрозы – почтовый вирус может причинить вам немало неприятностей, просто разослав ваши же, ранее отосланные сообщения, по всем адресам в вашей адресной книге – это может поставить вас в неудобное положение. Если бы корреспонденция, по крайней мере, важная ее часть, шифровалась, то подобных проблем не возникло бы. От перехваченных сообщений злоумышленник не получил бы никакой пользы, а действия описанного выше вируса не привели бы ни к чему, кроме рассылки бессмысленных для получателей данных. Существуют почтовые вирусы, которые создают разделяемые ресурсы (общие папки) на вашем компьютере, предоставляя тем самым доступ к вашим папкам всем пользователям вашей локальной сети. Если бы папки были зашифрованы, то не было бы никаких неприятностей. Конечно, вы вовремя обновили свой антивирус, тщательно сконфигурировали почтовую программу и установили все нужные «заплатки», устраняющие известные уязвимости в программном обеспечении, для того, чтобы вообще избежать встречи с почтовым вирусом. Но даже в этом случае шифрование почты и файловой системы создаст дополнительный уровень защиты, который очень пригодится в случае, когда все остальные меры окажутся тщетными. Примеры, о которых шла речь, иллюстрируют важность шифрования в мире электронной почты. По аналогии с этим должно быть понятно, что шифрование критических данных имеет огромное значение вообще в любых программных технологиях.

Использование электронной подписи – еще один способ избежать излишних рисков. К сожалению, большинство пользователей электронной почты не используют электронную подпись в своей корреспонденции. Если вы не подписываете свои наиболее важные сообщения, то кто-нибудь может отправить сообщение от вашего имени, дискредитировав вас тем или иным образом. Если вы постоянно подписываете свои наиболее важные сообщения, то ваши корреспонденты будут ожидать наличия подписи и с подозрением отнесутся к любому фальшивому сообщению от вашего имени. Этот пример иллюстрирует роль цифровой подписи в электронной почте, но все сказанное можно распространить на любую область, в которой могут работать ваши программы.

ЛОЖНОЕ ЧУВСТВО БЕЗОПАСНОСТИ

К несчастью, людям свойственно считать само собой разумеющимся, что использование компьютера знакомым, привычным образом не заключает в себе никакой опасности, но это не так. Вот впечатляющий пример: летом 2002 года корпорация Microsoft в корейской версии пакета Visual Studio .NET непредумышленно распространила копию червя «Nimda». К счастью оказалось, что копия червя была включена в дистрибутив таким образом, что реальной опасности для чьей-либо системы он не представлял. Но кто вообще мог вообразить, что может быть что-то опасное в установке такого известного приложения, полученного от столь надежного поставщика? Эта новость послужила тревожным сигналом[1] для программистов во всем мире! Есть много примеров, доказывающих, что «само собой разумеющаяся» безопасность привычного программного обеспечения вовсе не разумеется сама собой. Как часто вам доводится слышать об обнаружении новых уязвимостей в программах и выпуске «заплат», ликвидирующих эти уязвимости? Увы, но подобные новости доходят до нас едва ли не ежедневно. А хорошая новость состоит в том, что .NET Security Framework и вообще вся платформа .NET способна обеспечить весьма эффективную защиту для программ и данных от множества потенциальных угроз. К сожалению, все проблемы безопасности невозможно решить раз и навсегда, однако технология .NET делает огромный шаг в этом направлении, позволяя создавать программы, гораздо более безопасные, чем когда-либо ранее.

Производители программного обеспечения, системные администраторы, программисты и пользователи – все мы должны проявлять бдительность и принимать необходимые меры предосторожности. Каждый должен помнить, что возникающее иногда чувство безопасности – ложное чувство. Совершенно очевидно, что вопросы безопасности играют важную роль, и это должны признать все люди, чья профессиональная деятельность связана с компьютерами. Особенно важным все это становится сейчас, когда компьютерные системы играют всевозрастающую роль во всех сферах жизни, и когда связи этих систем через Internet становятся все более глобальными.

---

[1] Много раньше, в истории развития системы UNIX, имел место еще один впечатляющий и убедительный пример подобного происшествия, когда стало ясно, что нельзя слепо доверять безопасности программного обеспечения, которое мы используем. Прочитайте на странице *http://www.acm.org/classics/sep95*, что сказал Кен Томпсон, «отец» UNIX, на эту тему.

Природа криптографии и других средств обеспечения безопасности

Центральной темой этой книги являются криптография и различные средства обеспечения безопасности на платформе .NET. Но если углубиться в эти темы, то легко потерять из виду главные вопросы, касающиеся самой природы криптографии и средств обеспечения безопасности:

- Почему криптография и средства обеспечения безопасности так важны?
- Что возможно и что невозможно сделать с их помощью?

Первый вопрос, в сущности, это вопрос типа «зачем нам это нужно?», а второй – это вопрос типа «что мы можем сделать при помощи этого?». Давайте, прежде чем углубиться в технические детали в последующих главах, рассмотрим эти два фундаментальных вопроса. И далее, когда вы будете читать остальную часть книги, не забывайте о существовании этих вопросов...

Почему криптография и средства обеспечения безопасности так важны

Все мы знаем много примеров из бизнеса, из истории войн, а порой даже из личного опыта, что встречаются ситуации, в которых немного более высокий уровень секретности очень помог бы делу и позволил бы избежать серьезных проблем. Можно вспомнить много случаев, когда дело обошлось бы без неприятностей и неудобств, если бы была проявлена чуть большая осторожность. Конечно, шифрование в состоянии обеспечить вам безопасность и секретность[1] в том, по крайней мере, что касается компьютерных данных. Существует четыре основных аспекта, в которых рассматривается безопасность данных: секретность, аутентификация, целостность и подтверждение обязательств (nonrepudiation). Очевидно, что секретность во многих случаях может иметь большое значение. Понятно, что секретность необходима в ситуациях, когда критическую информацию необходимо скрыть от противника. Вы без труда также представите себе ситуацию, когда очень важно точно знать – с кем именно вы имеете дело (проблема аутентификации). Не менее важно иногда увериться в том, что информация, которую вы получили от кого-то или кому-то отослали, не может быть изменена злонамеренным третьим лицом (проблема целостности). Наконец, вам может потребоваться уверенность в том, что некто, с кем вы договорились о чем-то, не откажется от своих слов (подтверждение обязательств). Для решения всех перечисленных

[1] В большинстве стран сокрытие или утаивание сведений, имеющих отношение к уголовному расследованию, является преступлением, поэтому в деле засекречивания данных необходим разумный подход: одно дело осторожность и предусмотрительность, но совсем другое – создание помех правосудию.

проблем (то есть секретности, аутентификации, целостности и подтверждения обязательств) можно реализовать соответствующие протоколы безопасности, использующие цифровые подписи и цифровые сертификаты, а так же симметричные алгоритмы шифрования, криптографические хеши и коды аутентификации сообщений (MAC – Message Authentication Codes).

ЗАЧЕМ БЕСПОКОИТЬСЯ, ЕСЛИ ВАМ НЕЧЕГО СКРЫВАТЬ?

Зачем беспокоиться о безопасности данных, если вам нечего скрывать? Этот вопрос иногда задают люди, которые наивно полагают, что в защите своей конфиденциальности нуждаются только преступники, террористы и шпионы, чьи грязные секреты нуждаются в сокрытии. Эта ошибочная точка зрения исходит из предположения, что обычному честному человеку мало, что необходимо скрывать в своей жизни, и что энергичная забота о конфиденциальности свидетельствует о нечистой совести. Здесь необходимо четко сознавать, что конфиденциальность – это совершенно нормальная потребность всех законопослушных граждан, что особенно верно, когда речь идет о гражданах государства, чьи власти далеки от идеала.

Чтобы немного яснее представить себе все аспекты этой проблемы, вообразите на минуту, что вам отказано в праве на конфиденциальность. Например, ваше правительство запретило гражданам отправлять по почте письма в закрытых конвертах и обязало публиковать медицинские и банковские данные всех людей – как бы вы чувствовали себя в подобной ситуации? Как бы вам понравился запрет на сохранение в тайне каких-либо сведений о себе, пусть даже с риском стать в результате жертвой преступников? Как бы вы чувствовали себя, если вся ваша переписка по электронной почте и все сведения о ваших посещениях сайтов в Web оказались бы доступными для всех интересующихся на поисковых сайтах, таких, например, как *www.google.com*? Конечно, вам придется признать, что право на конфиденциальность совершенно нормально и законно для вполне обычных, законопослушных граждан, и что определенная степень конфиденциальности является фундаментальным и неотъемлемым правом всех людей.

КАТЕГОРИИ БЕЗОПАСНОСТИ

Мы могли бы привести множество примеров из практики, однако, чтобы не повторяться и не путаться, рассмотрим общие категории, на которые можно разбить проблемы безопасности. Вам обязательно придут на ум из сообщений новостей или даже из собственной практики примеры каждой из этих категорий:

- утечки интеллектуальной собственности, нарушенные контракты, сорванные сделки;
- созданный злоумышленником программный код – почтовые вирусы, «логические бомбы», обычные программные вирусы и «троянские» программы;

[. . .]