Лабораторная работа № 2

Тема: «Защита программного обеспечения от несанкционированного доступа» (3 часа).

Цель работы: получение практических навыков защиты программного обеспечения от несанкционированного доступа с помощью пароля.

Содержание отчета

1. Название и цель работы;
2. Индивидуальное задание согласно варианту;
3. Листинг программы;
4. Ответы на контрольные вопросы.

Методические указания

Необходимость использования систем защиты (СЗ) ПО обусловлена рядом проблем, среди которых следует выделить:

- незаконное использование алгоритмов, являющихся интеллектуальной собственностью автора,
- при написании аналогов продукта (промышленный шпионаж);
- несанкционированное использование ПО (кража и копирование); несанкционированная модификация ПО с целью внедрения программных злоупотреблений;
- незаконное распространение и сбыт ПО (пиратство).

Существующие системы защиты программного обеспечения можно классифицировать по ряду признаков:

- метод установки;
- используемые механизмы защиты;
- принцип функционирования.

Системы защиты ПО по методу установки можно подразделить:

- на системы, устанавливаемые на скомпилированные модули ПО;
- системы, встраиваемые в исходный код ПО до компиляции;
- комбинированные.

Системы первого типа наиболее удобны для производителя ПО, так как легко можно защитить уже полностью готовое и оттестированное ПО (обычно процесс установки защиты максимально автоматизирован и сводится к указанию имени защищаемого файла и нажатию "Enter"), а потому и наиболее популярны. В то же время стойкость этих систем достаточно низка, так как для обхода защиты достаточно определить точку завершения работы "конверта" защиты и передачи управления защищенной программе, а затем принудительно ее сохранить в незащищенном виде.

Системы второго типа неудобны для производителя ПО, так как возникает необходимость обучать персонал работе с программным интерфейсом (API) системы защиты с вытекающими отсюда денежными и временными затратами. Но такие системы являются более стойкими к атакам, потому что здесь исчезает четкая граница между системой защиты и как таковым ПО.

Наиболее живучими являются комбинированные системы защиты. Сохраняя достоинства и недостатки систем второго типа, они максимально затрудняют анализ и дезактивацию своих алгоритмов.

По используемым механизмам защиты СЗ можно классифицировать на:

- системы, использующие сложные логические механизмы;
- системы, использующие шифрование защищаемого ПО
- комбинированные системы.

Системы первого типа используют различные методы и приёмы, ориентированные на затруднение дизассемблирования, отладки и анализа алгоритма СЗ и защищаемого ПО. Этот тип СЗ наименее стоек к атакам, так как для преодоления защиты достаточно проанализировать логику процедур проверки и должным образом их модифицировать.

Более стойкими являются системы второго типа. Для дезактивации таких защит необходимо определение ключа дешифрации ПО.

Самыми стойкими к атакам являются комбинированные системы.

По принципу функционирования СЗ можно подразделить на:

- упаковщики/шифраторы;
- СЗ от несанкционированного копирования;
- СЗ от несанкционированного доступа (НСД).

Упаковщики/шифраторы. Их цель защита ПО от анализа его алгоритмов и несанкционированной модификации. Для достижения этого используются алгоритмы компрессии данных; шифрование данных, алгоритмы мутации, запутывание логики программы, приведение ОС в нестабильное состояние на время работы ПО и др.

СЗ от несанкционированного копирования осуществляют "привязку" ПО к дистрибутивному носителю (гибкий диск, CD ...). Данный тип защит основывается на изучении работы контроллеров накопителей, их физических показателей, нестандартных режимах разбивки, чтения/записи и т.п.

СЗ от НСД. Несанкционированным доступом (НСД) к информации ПК будем называть незапланированное ознакомление, обработку, копирование, применение различных вирусов, в том числе разрушающих программные продукты, а также модификацию или уничтожение информации в нарушение установленных правил разграничения доступа. В защите информации ПК от НСД можно выделить три основных направления:

- специальные технические средства опознавания пользователя;
- специальное программное обеспечение по защите информации;
- специальные средства защиты информации ПК от несанкционированного доступа.

Первым шагом в обеспечении безопасности информации является возможность проверки подлинности любого пользователя. Гарантированная проверка личности пользователя является задачей различных механизмов идентификации и аутентификации.

Идентификация основана на назначении каждому пользователю (группе пользователей) определенного отличительного признака - идентификатора, и его сравнении с утвержденным перечнем. Процесс проверки личности пользователя получил название аутентификации. Он происходит с помощью предъявляемого пользователем особого отличительного признака - аутентификатора, присущего именно ему.

Конкретные механизмы идентификации и аутентификации могут быть реализованы на основе следующих средств и процедур защиты информации:

- Пароли;
- Средства биометрии;
- Интеллектуальные карты;
- Прекращение доступа пользователя в сеть после нескольких ошибок при регистрации, блокировка компьютера (клавиатуры) или автоматизированного рабочего места с помощью пароля;
- Криптография с уникальными ключами для каждого пользователя.

В качестве таких особенностей пользователей, вследствие максимальной простоты реализации, чаще всего используются пароли.

Обычные пароли не являются в полном смысле средствами защиты, они скорее относятся к механизму управления доступом. Пароли обеспечивают сохранение целостности программного обеспечения в составе вычислительной системы. Пароли должны быть просты для запоминания и не должны быть очевидными. Вопросно-ответные системы обеспечивают высокий уровень защиты, но требуют значительных ресурсов и времени работы вычислительной системы. Однако защита с помощью пароля может оказаться неэффективной, если требуется хранение копии пароля, которая может быть похищена хакером.

Пароль в обычном смысле этого слова не является средством защиты программного обеспечения, поскольку законный пользователь, которому вручен пароль, может оказаться хакером. При формировании пароля можно прибегнуть к помощи специального устройства, которое генерирует последовательности чисел или букв в зависимости от данных, которые задает пользователь. Такое устройство называется преобразователем информации. В действительности вычислительная система генерирует последовательность случайных чисел и требует, чтобы пользователь в течение короткого промежутка времени присвоил ей некоторое число. Системы такого рода не зависят от интерфейса с компьютером.

Большинство парольных СЗ ПО использует логические механизмы, сводящиеся к проверке правильности пароля / кода и запуске или не запуске ПО, в зависимости от результатов проверки.

Слабым звеном парольных защит является блок проверки правильности введённого пароля/кода. Для такой проверки можно сравнивать введённый пароль с записанным в коде ПО правильным либо с правильно сгенерированным из введённых дополнительных данных паролем. Возможно так же сравнение производных величин от введённого и правильного паролей, например их ХЭШ-функций, в таком случае в коде можно сохранять только производную величину, что повышает стойкость защиты. Путём анализа процедур проверки можно найти реальный пароль, записанный в коде ПО, найти правильно сгенерированный пароль из введённых данных либо создать программу для перебора паролей для определения пароля с нужной ХЭШ-суммой. Кроме того, если СЗ ПО не использует шифрования, достаточно лишь принудительно изменить логику проверки для получения беспрепятственного доступа к ПО.

Для всех парольных систем существует угроза перехвата пароля при его вводе авторизованным пользователем. Кроме того, в большинстве СЗ ПО данного типа процедура проверки используется лишь единожды, обычно при регистрации или установке ПО, затем система защиты просто отключается, что создаёт реальную угрозу для НСД при незаконном копировании ПО.

Системы "привязки" ПО при установке на ПК пользователя осуществляют поиск уникальных признаков компьютерной системы, либо они устанавливаются самой системой защиты. После этого модуль защиты в самом ПО настраивается на поиск и идентификацию данных признаков, по которым в дальнейшем определяется авторизованное или неавторизованное использование ПО. При этом возможно применение методик оценки скоростных и иных показателей процессора, материнской платы, дополнительных устройств, ОС, чтение/запись в микросхемы энергонезависимой памяти, запись скрытых файлов, настройка на наиболее часто встречаемую карту использования ОЗУ и т.п.

Документация, сопровождающая любое программное обеспечение, является субъектом авторского права и может выполнять функции защиты. Этому способствуют следующие факторы: её репродуцирование стоит достаточно дорого, особенно если оригинал выполнен в цвете и не может быть качественно воспроизведен одноцветным копировальным устройством; обычно программы распространяются, будучи представленными в машинном коде, что затрудняет анализ их структуры и обеспечивает определенную степень защиты. В последнем случае весьма важно, чтобы сохранялось сопровождение программы со стороны разработчика, особенно в тех случаях, когда программа не полностью отлажена.

Внешние средства активной защиты могут быть активизированы при возникновении многих условий. Такие внешние факторы включают также использование ключевых слов, чтобы вызвать распечатку названия программы или имени ее владельца. Хотя описанное и не является защитой от пиратства, это, тем не менее, способствует увеличению объема продаж, если число случайных копий уменьшится. Общепринятые сигналы тревог более сродни созданию среды защиты компьютера, когда требуется подтверждение подлинности операции, особенно при копировании.

Запуск распечатки этикетки или других деталей из защищенных участков программы осуществляется только при наличии ключевых слов.

ПРИМЕР РЕАЛИЗАЦИИ
Ввод пароля;
IF пароль введен неверно THEN
BEGIN
печать сообщения об ошибке;
сигнал тревоги;
END
ELSE
печать сообщения о разрешении доступа;
END PROGRAM.

Идентификация программы или отдельного модуля представляет интерес в том случае, когда другие методы защиты не приносят успеха. Широко обсуждаются проблемы авторского права для отдельной процедуры программы и взаимосвязь между идеей и способом ее реализации. Выделение объективных характеристик программы - довольно сложная процедура, тем не менее, признаки подобия двух программ или модулей, содержащихся в больших программах, указать можно. Проблема заключается в том, чтобы уметь идентифицировать программы, которые изменены хакером, погружены в другую программу или откомпилированы в машинный код.

Понятие "родимые пятна" используется для описания характеристик, появляющихся в результате естественного процесса разработки программы и относящихся к особенностям стиля программирования, ошибкам и избыточностям, которые не должны иметь места в независимо написанной программе. Каждое из них может служить убедительной уликой нарушения авторского права.

Отличительные метки относятся к таким признакам, которые не являются случайными, а вводятся специально, чтобы дать информацию об авторе или владельце авторского права. Другое использование идентификационных меток - выявление путей незаконного копирования или других злоумышленных действий. Термин "отличительная метка" относится к пассивным средствам защиты, которые при нормальном функционировании не проявляют себя по отношению к пользователю.

Одно из убедительных доказательств копирования - наличие скопированных ошибок. В каждой программе остаются избыточные части, которые были необходимы для отладки в процессе проектирования программного продукта, а затем не были удалены. Таким образом, в любой программе содержится встроенная улика, которая тем или иным способом сохраняет следы разработки.

Убедительность улики повышается, если отличительная метка, содержащая информацию о владельце авторского права, закодирована. Использование закодированных отличительных меток - довольно распространенная практика, т.к. при этом они остаются доступными и в машинном коде. Отличительные метки не являются в полной мере избыточными для того, кто организует контроль за данными, и в состоянии отделить на их фоне действительно избыточные данные.

Важная особенность отличительных меток заключается в том, что они не известны нарушителю.

Устройства регистрации событий, процедур или доступа к данным могут рассматриваться как часть общей системы защиты, причем как программ, так и данных. Подтверждение подлинности программы охватывает проблемы от установления идентичности функционирования текущей программы и ее оригинала до подтверждения адекватности средств защиты. Это важно, если используются устройства с низким уровнем защищенности, когда возможен обход проверок, связанных с защитой.

Использование водяных знаков как метода выявления подделки занимает особое место, поскольку препятствует созданию точной копии, которую пользователь не мог бы отличить от оригинала.

Психологические методы основаны на том, чтобы создать у нарушителя чувство неуверенности и психологического напряжения, заставляя его все время помнить, что в похищенном программном продукте могут сохраняться средства защиты. Поэтому полезно было бы дать объявление, что в программное обеспечение встроены механизмы защиты (независимо от того, так ли это на самом деле). Существует

огромное число хитроумных способов расстановки отличительных меток в программе и никакой хакер не может быть уверен, что ему удалось уничтожить все ключи и механизмы защиты.

ПРИМЕР РЕАЛИЗАЦИИ

Данная программа демонстрирует пассивную защиту с помощью выдачи на экран авторской этикетки.

```
program demo;
uses crt;
const st=' ЭТА ПРОГРАММА НАПИСАНА ';
st1='В ДЕМОНСТРАЦИОННЫХ ЦЕЛЯХ';
st2=' Защитником В.В. ';
st3=' ПРОДАЖА ЗАПРЕЩЕНА! ';
…
end.
```

Задание. Написать программы в соответствии с индивидуальным вариантом:

Вариант	Тематика
1.	Защита программ от несанкционированного доступа. Доступ к файлу данных по паролю.
2.	Защита программ с помощью пароля и авторской этикетки.
3.	Защита программ с помощью контрольного суммирования.
4.	Защита сопровождения: регистрация обращений.
5.	Защита программ от несанкционированного доступа. Пароль с шифрованием.
6.	Защита программ от несанкционированного доступа путем привязки ПО к ПК.
7.	Защита программ от несанкционированного доступа. Пароль на основании ХЭШ-суммы.
8.	Защита программ с помощью пароля и выдачи звукового сопровождения при НСД.
9.	Защита программ от НСД с помощью пароля и отличительных меток в программе.
10.	Защита программ от НСД с помощью пароля и распечатка данных о владельцах и защите ПО.

Контрольные вопросы

1. Принципы классификации систем защиты программного обеспечения (ПО).
2. Классификация систем защиты ПО по методу установки.
3. Классификация систем защиты ПО по используемым механизмам защиты.
4. Классификация систем защиты ПО по принципу функционирования.
5. Системы защиты от несанкционированного копирования.
6. Системы защиты от несанкционированного доступа.
7. Выделение объективных характеристик программы.