17. Угрозы безопасности ПО. Программные закладки. Троянские программы. Клавишные шпионы

Обеспечение безопасности автоматизированных информационных систем зависит от безопасности используемого в них программного обеспечения и, в частности, следующих видов программ:

- обычных программ пользователей;
- специальных программ, рассчитанных на нарушение безопасности системы;
- разнообразных системных утилит и коммерческих прикладных программ, которые отличаются высоким профессиональным уровнем разработки и тем не менее могут содержать отдельные недоработки, позволяющие захватчикам атаковать системы.

Программы могут порождать проблемы двух типов:

- могут перехватывать и модифицировать данные в результате действий пользователя, который к этим данным не имеет доступа
- используя упущения в защите компьютерных систем, могут или обеспечивать доступ к системе пользователям, не имеющим на это права, или блокировать доступ к системе законных пользователей.

Чем выше уровень подготовки программиста, тем более неявными (даже для него) становятся допускаемые им ошибки и тем более тщательно и надежно он способен скрыть умышленные механизмы, разработанные для нарушения безопасности системы.

Целью атаки могут быть и сами программы по следующим причинам:

- В современном мире программы могут быть товаром, приносящим немалую прибыль, особенно тому, кто первым начнет тиражировать программу в коммерческих целях и оформит авторские права на нее.
- Программы могут становиться также объектом атаки, имеющей целью модифицировать эти программы некоторым образом, что позволило бы в будущем провести атаку на другие объекты системы. Особенно часто объектом атак такого рода становятся программы, реализующие функции защиты системы.

Рассмотрим несколько типов программ и приемы, которые наиболее часто используются для атак программ и данных.

Эти приемы обозначаются единым термином — «программные ловушки».

К ним относятся «программные люки», «троянские кони», «логические бомбы», атаки «салями», скрытые каналы, отказы в обслуживании и компьютерные вирусы.

Люки в программах

Использование люков для проникновения в программу — один из простых и часто используемых способов нарушения безопасности автоматизированных информационных систем.

Люком называется не описанная в документации на программный продукт возможность работы с этим программным продуктом. Сущность использования люков состоит в том, что при выполнении пользователем некоторых не описанных в документации действий он получает доступ к возможностям и данным, которые в обычных условиях для него закрыты (в частности, выход в привилегированный режим).

Люки чаще всего являются результатом забывчивости разработчиков. В качестве люка может быть использован временный механизм прямого доступа к частям продукта, созданный для облегчения процесса отладки и не удаленный по ее окончании.

Люки могут образовываться также в результате часто практикуемой технологии разработки программных продуктов «сверху вниз»: в их роли будут выступать оставленные по каким-либо причинам в готовом продукте «заглушки» — группы команд, имитирующие или просто обозначающие место подсоединения будущих подпрограмм.

Наконец, еще одним распространенным источником люков является так называемый **«неопределенный ввод»** — ввод «бессмысленной» информации, абракадабры в ответ на запросы системы. Реакция недостаточно хорошо написанной программы на неопределенный ввод может быть, в лучшем случае, непредсказуемой (когда при повторном вводе той же

неверной команды программа реагирует каждый раз по-разному); гораздо хуже, если программа в результате одинакового «неопределенного» ввода выполняет некоторые повторяющиеся действия, — это дает возможность потенциальному захватчику планировать свои действия по нарушению безопасности.

Неопределенный ввод — частная реализация прерывания. То есть в общем случае захватчик может умышленно пойти на создание в системе некоторой нестандартной ситуации, которая бы позволила ему выполнить необходимые действия. Например, он может искусственно вызвать аварийное завершение программы, работающей в привилегированном режиме, с тем чтобы перехватить управление, оставшись в этом привилегированном режиме.

Борьба с возможностью прерывания, в конечном счете, выливается в необходимость предусмотреть при разработке программ комплекса механизмов, образующих так называемую «защиту от дурака».

Смысл этой защиты состоит в том, чтобы гарантированно отсекать всякую вероятность обработки неопределенного ввода и разного рода нестандартных ситуаций (в частности, ошибок) и тем самым не допускать нарушения безопасности компьютерной системы даже в случае некорректной работы с программой.

Таким образом, люк (или люки) может присутствовать в программе ввиду того, что программист:

- забыл удалить его;
- умышленно оставил его в программе для обеспечения тестирования или выполнения оставшейся части отладки;
- умышленно оставил его в программе в интересах облегчения окончательной сборки конечного программного продукта;
- умышленно оставил его в программе с тем, чтобы иметь скрытое средство доступа к программе уже после того, как она вошла в состав конечного продукта.

Люк — первый шаг к атаке системы, возможность проникнуть в компьютерную систему в обход механизмов защиты.

«Троянские кони»

Существуют программы, реализующие, помимо функций, описанных в документации, и некоторые другие функции, в документации не описанные. Такие программы называются «троянскими конями».

Вероятность обнаружения «троянского коня» тем выше, чем очевиднее результаты его действий (например, удаление файлов или изменение их защиты). Более сложные «троянские кони» могут маскировать следы своей деятельности (например, возвращать защиту файлов в исходное состояние).

«Логические бомбы»

«Логической бомбой» обычно называют программу или даже участок кода в программе, реализующий некоторую функцию при выполнении определенного условия или в определенное время. Этим условием может быть, например, наступление определенной даты или обнаружение файла с определенным именем.

«Взрываясь», «логическая бомба» реализует функцию, неожиданную и, как правило, нежелательную для пользователя (например, удаляет некоторые данные или разрушает некоторые системные структуры). «Логическая бомба» является одним из излюбленных способов мести программистов компаниям, которые их уволили или чем-либо обидели.

Атака «салями»

Атака «салями» превратилась в настоящий бич банковских компьютерных систем. В банковских системах ежедневно производятся тысячи операций, связанных с безналичными расчетами, переводами сумм, отчислениями и т. д.

При обработке счетов используются целые единицы (рубли, центы), а при исчислении процентов нередко получаются дробные суммы. Обычно величины, превышающие половину рубля (цента), округляются до целого рубля (цента), а величины менее половины рубля

(цента) просто отбрасываются. При атаке «салями» эти несущественные величины не удаляются, а постепенно накапливаются на некоем специальном счете.

Как свидетельствует практика, сумма, составленная буквально из ничего, за пару лет эксплуатации «хитрой» программы в среднем по размеру банке может исчисляться тысячами долларов. Атаки «салями» достаточно трудно распознаются, если злоумышленник не начинает накапливать на одном счете большие суммы.

Скрытые каналы

Под скрытыми каналами подразумеваются программы, передающие информацию лицам, которые в обычных условиях эту информацию получать не должны. В тех системах, где ведется обработка критичной информации, программист не должен иметь доступа к обрабатываемым программой данным после начала эксплуатации этой программы.

Если захватчик имеет возможность доступа к компьютеру во время работы интересующей его программы, скрытым каналом может стать пересылка критичной информации в специально созданный в оперативной памяти компьютера массив данных.

Скрытые каналы наиболее применимы в ситуациях, когда захватчика интересует даже не содержание информации, а, допустим, факт ее наличия (например, наличие в банке расчетного счета с определенным номером).

Отказ в обслуживании. Большинство методов нарушения безопасности направлено на то, чтобы получить доступ к данным, не допускаемый системой в нормальных условиях. Однако не менее интересным для захватчиков является доступ к управлению самой компьютерной системой или изменение ее качественных характеристик, например, получить некоторый ресурс (процессор, устройство ввода-вывода) в монопольное использование или спровоцировать ситуацию клинча для нескольких процессов.

Различают пассивные и активные угрозы.

Пассивные угрозы (нарушение конфиденциальности данных, циркулирующих в сети) — это просмотр и/или запись данных, передаваемых по линиям связи. К ним относятся:

- просмотр сообщения;
- **анализ графика** — злоумышленник может просматривать заголовки пакетов, циркулирующих в сети, и на основе содержащейся в них служебной информации делать заключения об отправителях и получателях пакета и условиях передачи (время отправления, класс сообщения, категория безопасности, длина сообщения, объем трафика и т. д.).

Активные угрозы (нарушение целостности / доступности ресурсов и компонентов сети) — несанкционированное использование устройств, имеющих доступ к сети для изменения отдельных сообщений или потока сообщений. К ним относятся:

- отказ служб передачи сообщений — злоумышленник может уничтожать или задерживать отдельные сообщения или весь поток сообщений;
- «маскарад» — злоумышленник может присвоить своему узлу или ретранслятору чужой идентификатор и получать или отправлять сообщения от чужого имени;
- внедрение сетевых вирусов — передача по сети тела вируса с его последующей активизацией пользователем удаленного или локального узла;
- модификация потока сообщений — злоумышленник может выборочно уничтожать, модифицировать, задерживать, переупорядочивать и дублировать сообщения, а также вставлять поддельные сообщения.

МОДЕЛЬ УГРОЗ И ПРИНЦИПЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Использование при создании программного обеспечения КС сложных операционных систем, инструментальных средств разработки ПО импортного производства увеличивают потенциальную возможность внедрения в программы преднамеренных дефектов диверсионного типа.

Помимо этого, при создании целевого программного обеспечения всегда необходимо исходить из возможности наличия в коллективе разработчиков программистов - злоумышленников, которые в силу тех или иных причин могут внести в разрабатываемые программы РПС.

Характерным свойством РПС в данном случае является возможность внезапного и незаметного нарушения или полного вывода из строя КС. Функционирование РПС реализуется в рамках модели угроз безопасности ПО, основные элементы которой рассматриваются в следующем разделе.

Подход к созданию модели угроз технологической безопасности ПО

Один из возможных подходов к созданию модели технологической безопасности ПО АСУ может основываться на обобщенной концепции технологической безопасности компьютерной инфосферы, которая определяет методологический базис, направленный на решение, в том числе, **следующих основных задач**:

- создания теоретических основ для практического решения проблемы технологической безопасности ПО;
- создания безопасных информационных технологий;
- развертывания системы контроля технологической безопасности компьютерной инфосферы.

Модель угроз технологической безопасности ПО должна представлять собой официально принятый нормативный документ, которым должен руководствоваться заказчики и разработчики программных комплексов.

Модель угроз должна включать:

- полный реестр типов возможных программных закладок;
- описание наиболее технологически уязвимых мест компьютерных систем (с точки зрения важности и наличия условий для скрытого внедрения программных закладок);
- описание мест и технологические карты разработки программных средств, а также критических этапов, при которых наиболее вероятно скрытое внедрение программных закладок;
- реконструкцию замысла структур, имеющих своей целью внедрение в ПО заданного типа (класса, вида) программных закладок диверсионного типа;
- психологический портрет потенциального диверсанта в компьютерных системах

В указанной Концепции также оговариваются необходимость содержания в качестве приложения банка данных о выявленных программных закладках и описания связанных с их обнаружением обстоятельств, а также необходимость периодического уточнения и совершенствования модели на основе анализа статистических данных и результатов теоретических исследований.

На базе утвержденной модели угроз технологической безопасности компьютерной инфосферы, как обобщенного, типового документа должна разрабатываться прикладная модель угроз безопасности для каждого конкретного компонента защищаемого комплекса средств автоматизации КС. В основе этой разработки должна лежать схема угроз, типовой вид которой применительно к ПО КС представлен на рис.1.2.

Наполнение модели технологической безопасности ПО должно включать в себя следующие элементы:

- матрицу чувствительности КС к "вариациям" ПО (то есть к появлению искажений)
- энтропийный портрет ПО (то есть описание "темных" запутанных участков ПО)
- реестр камуфлирующих условий для конкретного ПО
- справочные данные о разработчиках и реальный (либо реконструированный) замысел злоумышленников по поражению этого ПО.

На рис.1.3 приведен пример указанной типовой модели для сложных программных комплексов.

1.5.2. Элементы модели угроз эксплуатационной безопасности ПО

Анализ угроз эксплуатационной безопасности ПО КС позволяет, разделить их на два типа: случайные и преднамеренные, причем последние подразделяются на активные и пассивные. Активные угрозы направлены на изменение технологически обусловленных алгоритмов, программ функциональных преобразований или информации, над которой эти преобразования осуществляются. Пассивные угрозы ориентированы на нарушение безопасности информационных технологий без реализации таких модификаций.

Вариант общей структуры набора потенциальных угроз безопасности информации и ПО на этапе эксплуатации КС приведен в табл.1.2.

Таблица 1.2

Рассмотрим основное содержание данной таблицы. Угрозы, носящие случайный характер и связанные с отказами, сбоями аппаратуры, ошибками операторов и т.п. предполагают нарушение заданного собственником информации алгоритма, программы ее обработки или искажение содержания этой информации.

Субъективный фактор появления таких угроз обусловлен ограниченной надежностью работы человека и проявляется в виде ошибок (дефектов) в выполнении операций формализации алгоритма функциональных преобразований или описания алгоритма на некотором языке, понятном вычислительной системе.

Угрозы, носящие злоумышленный характер вызваны, как правило, преднамеренным желанием субъекта осуществить несанкционированные изменения с целью нарушения корректного выполнения преобразований, достоверности и целостности данных, которое проявляется в искажениях их содержания или структуры, а также с целью нарушения функционирования технических средств в процессе реализации функциональных преобразований и изменения конструктива вычислительных систем и систем телекоммуникаций.

На основании анализа уязвимых мест и после составления полного перечня угроз для данного конкретного объекта информационной защиты, например, в виде указанной таблицы, необходимо осуществить переход к неформализованному или формализованному описанию модели угроз эксплуатационной безопасности ПО КС.

Такая модель, в свою очередь, должна соотноситься (или даже являться составной частью) обобщенной модели обеспечения безопасности информации и ПО объекта защиты.

К неформализованному описанию модели угроз приходится прибегать в том случае, когда структура, состав и функциональная наполненность компьютерных системы управления носят многоуровневый, сложный, распределенный характер, а действия потенциального нарушителя информационных и функциональных ресурсов трудно поддаются формализации.

После окончательного синтеза модели угроз разрабатываются практические рекомендации и методики по ее использованию для конкретного объекта информационной защиты, а также механизмы оценки адекватности модели и реальной информационной ситуации и оценки эффективности ее применения при эксплуатации КС.

Таким образом, разработка моделей угроз безопасности программного обеспечения КС, являясь одним из важных этапов комплексного решения проблемы обеспечения безопасности информационных технологий, на этапе создания КС отличается от разработки таких моделей для этапа их эксплуатации.

Принципиальное различие подходов к синтезу моделей угроз технологической и эксплуатационной безопасности ПО заключается в различных мотивах поведения потенциального нарушителя информационных ресурсов, принципах, методах и средствах воздействия на ПО на различных этапах его жизненного цикла.

Основные принципы обеспечения безопасности ПО

В качестве объекта обеспечения технологической и эксплуатационной безопасности ПО рассматривается вся совокупность его компонентов в рамках конкретной КС. В качестве доминирующей должна использоваться стратегия сквозного тотального контроля технологического и эксплуатационного этапов жизненного цикла компонентов ПО.

Совокупность мероприятий по обеспечению технологической и эксплуатационной безопасности компонентов ПО должна носить конфиденциальный характер. Необходимо обеспечить постоянный, комплексный и действенный контроль за деятельностью разработчиков и пользователей компонентов. Кроме общих принципов, обычно необходимо конкретизировать принципы обеспечения безопасности ПО на каждом этапе его жизненного цикла. Далее приводятся один из вариантов разработки таких принципов.

Принципы обеспечения технологической безопасности при обосновании, планировании работ и проектном анализе ПО

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

- **Комплексности обеспечения безопасности ПО**

Предполагает рассмотрение проблемы безопасности информационно - вычислительных процессов с учетом всех структур КС, возможных каналов утечки информации и несанкционированного доступа к ней, времени и условий их возникновения, комплексного применения организационных и технических мероприятий.

- **Планируемости применения средств безопасности программ**

Предполагает перенос акцента на совместное системное проектирование ПО и средств его безопасности, планирование их использования в предполагаемых условиях эксплуатации.

- **Обоснованности средств обеспечения безопасности ПО**

Заключается в глубоком научно-обоснованном подходе к принятию проектных решений по оценке степени безопасности, прогнозированию угроз безопасности, всесторонней априорной оценке показателей средств защиты.

- **Достаточности безопасности программ**

Отражает необходимость поиска наиболее эффективных и надежных мер безопасности при одновременной минимизации их стоимости.

- **Гибкости управления защитой программ**

Требующая от системы контроля и управления обеспечением информационной безопасности ПО способности к диагностированию, опережающей нейтрализации, оперативному и эффективному устранению возникающих угроз в условиях резких изменений обстановки информационной борьбы.

- **Заблаговременности разработки средств обеспечения безопасности и контроля производства ПО**

Заключается в предупредительном характере мер обеспечения технологической безопасности работ в интересах недопущения снижения эффективности системы безопасности процесса создания ПО.

- **Документируемости технологии создания программ**

Подразумевает разработку пакета нормативно-технических документов по контролю программных средств на наличие преднамеренных дефектов.

Принципы достижения технологической безопасности ПО в процессе его разработки

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

- Регламентации технологических этапов разработки ПО, включающей упорядоченные фазы промежуточного контроля, спецификацию программных модулей и стандартизацию функций и формата представления данных.
- Автоматизации средств контроля управляющих и вычислительных программ на наличие дефектов, создания типовой общей информационной базы алгоритмов, исходных текстов и программных средств, позволяющих выявлять преднамеренные программные дефекты.
- Последовательной многоуровневой фильтрации программных модулей в процессе их создания с применением функционального дублирования разработок и поэтапного контроля.

- Типизации алгоритмов, программ и средств информационной безопасности, обеспечивающей информационную, технологическую и программную совместимость, на основе максимальной их унификации по всем компонентам и интерфейсам.
- Централизованного управления базами данных проектов ПО и администрирование технологии их разработки с жестким разграничением функций, ограничением доступа в соответствии со средствами диагностики, контроля и защиты.
- Блокирования несанкционированного доступа соисполнителей и абонентов государственных сетей связи, подключенных к стендам для разработки программ.
- Статистического учета и ведения системных журналов о всех процессах разработки ПО с целью контроля технологической безопасности.
- Использования только сертифицированных и выбранных в качестве единых инструментальных средств разработки программ для новых технологий обработки информации и перспективных архитектур вычислительных систем.

Принципы обеспечения технологической безопасности на этапах стендовых и приемо-сдаточных испытаний

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

- Тестирования ПО на основе разработки комплексов тестов, параметризуемых на конкретные классы программ с возможностью функционального и статистического контроля в широком диапазоне изменения входных и выходных данных.
- Проведения натурных испытаний программ при экстремальных нагрузках с имитацией воздействия активных дефектов.
- Осуществления "фильтрации" программных комплексов с целью выявления возможных преднамеренных дефектов определенного назначения на базе создания моделей угроз и соответствующих сканирующих программных средств.
- Разработки и экспериментальной отработки средств верификации программных изделий.
- Проведения стендовых испытаний ПО для определения непреднамеренных программных ошибок проектирования и ошибок разработчика, приводящих к невыполнению целевых функций программ, а также выявление потенциально "узких" мест в программных средствах для разрушительного воздействия.
- Отработки средств защиты от несанкционированного воздействия нарушителей на ПО.
- Сертификации программных изделий АСУ по требованиям безопасности с выпуском сертификата соответствия этого изделия требованиям технического задания.

Принципы обеспечения безопасности при эксплуатации программного обеспечения

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

- Сохранения и ограничения доступа к эталонам программных средств, недопущение внесения изменений в них.
- Профилактического выборочного тестирования и полного сканирования программных средств на наличие преднамеренных дефектов.
- Идентификации ПО на момент ввода его в эксплуатацию в соответствии с предполагаемыми угрозами безопасности ПО и его контроль.
- Обеспечения модификации программных изделий во время их эксплуатации путем замены отдельных модулей без изменения общей структуры и связей с другими модулями.
- Строгого учета и каталогизации всех сопровождаемых программных средств, а также собираемой, обрабатываемой и хранимой информации.
- Статистического анализа информации обо всех процессах, рабочих операциях, отступлениях от режимов штатного функционирования ПО.
- Гибкого применения дополнительных средств защиты ПО в случае выявления новых, непрогнозируемых угроз информационной безопасности.