А.С.Сизак, В.Н.Тарасов

ОЦЕНКА ЗАЩИЩЕННОСТИ И НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

В статье затрагиваются такие вопросы как классификации уровней защищенности ЭВМ, оценка защищенности программных продуктов от несанкционированного доступа и определения их надежности.

Вопрос оценки защищенности информации в ЭВМ, несмотря на ряд существующих решений, остается на сегодняшний день достаточно актуальным.

Вопросам оценки защищенности информации посвящено много литературы. Первыми завершившимся выпуском нормативов документов в этой области являются работы, проводимые в США.

Следуя по пути интеграции, Франция, Германия, Нидерланды и Великобритания в 1991 г. приняли согласованные "Европейские Критерии" оценки безопасности информационных технологий (Information Technology Security Evaluation Criteria) версию 1.2.

Министерство обороны США выработало ряд классификаций для определения различных уровней защищенности ЭВМ. Они изложены в "Оранжевой книге" или в "Оценочных критериях защищенности вычислительных систем".

Безопасность информации — важнейшая характеристика автоматизированной системы. Но в настоящее время не имеет единой шкалы единиц измерения. Оценочная шкала, необходима для определения ценности информации для определения уровня безопасности системы и ее пригодности в эксплуатации той или иной структуре.

Как работают документы?

Механизм одобрения для защищенных систем основан на принципе создания перечня оцененных изделий, в который включены изделия с определенной степенью качества. Защищенные системы оцениваются по запросам их изготовителей и помещаются в перечень оценочных изделий по шести уровням защищенности. В случае необходимости потребитель может выбрать из перечня подходящее к его требованиям изделие, либо обратиться с просьбой оценить необходимое ему изделие, не входящее в перечень оцененных.

Оценка защищенности информации в вычислительных системах «Оценочных критериев» основывается на классификации потенциальных угроз, которые делятся на три класса: безответственен пользователь, попытки несанкционированного проникновения и факт несанкционированного проникновения.

Под *безответственностью пользователя* понимаются такие действия аккредитованного лица, которые приводят к нелояльным или преступным результатам.

Попытка несанкционированного проникновения — термин, означающий использование нарушителем плохого управления системой, а также несовершенства системы защиты. То же самое можно сказать о системах, где все пользователи имеют одинаковый доступ к файлам. В этом случае возможны действия, которые полностью законны, но могут иметь не предвиденные последствия и нежелательные результаты для владельцев и управляющих вычислительными системами.

Проникновение подразумевает полный обход всех видов системного контроля для достижения несанкционированного доступа.

"Европейские Критерии" рассматривают следующие составляющие информационной безопасности:

- конфиденциальность — защиту от несанкционированного получения информации;
- целостность — защиту от несанкционированного изменения информации;
- доступность — защиту от несанкционированного доступа.

"Чтобы объект оценки можно было признать надежным, необходима определенная степень уверенности в наборе функции и механизмов безопасности. Степень уверенности называется *гарантированностью,* которая может быть большей или меньшей в зависимости от тщательности проведения оценки. Гарантиро-

ванность затрагивает два аспекта — *эффективность* и к*орректность* средств безопасности".

В 1992 г. Гостехкомиссий России (ГТК РФ) выпущен пакет временных руководящих документов по защите информации от НСД в атом тарированных системах (АС) и средствах вычислительной техники (СВТ) содержащий концепцию защиты, термины и определения, показатели защищенности, классификацию СВТ и АС по уровням защищенности. Однако и они по концепции защиты и оценки немногим отличаются от "Оценочных критериев" США.

Критерии оценки защищенности информации, используемые в "Оранжевой книге", в "Европейских Критериях" и "Положении ГТК РФ", не всегда учитывают или не учитывают совсем следующие параметры защиты:

- деление средств защиты на средства зашиты от случайных и преднамеренных НСД, имеющих различную физическую природу, характер воздействия и точки приложения в объекте защиты;
- образование системы взаимосвязанных преград, замыкающихся вокруг предмета защиты и препятствующих обходу преград нарушителем;
- время жизни информации, обнаружения и блокировки НСД;
- ожидаемое время преодоления преграды нарушителем.

Из-за отсутствия теории и расчетных соотношений в "Оценочных критериях ..." не приведены единицы измерения и количественная оценка защищенности информации в вычислительных системах.

Перечисленные факторы дают основания полагать, что "Оценочные критерии", "Европейские Критерии" и "Временное положение" ГТК РФ, использующие существующую концепцию защиты, действительно не дают адекватного представления о свойствах и взаимодействии звеньев защиты и, следовательно, о прочности защиты информации в вычислительной системе в целом.

Как известно любую защиту и систему контроля можно взломать или обойти, обладая неограниченным количеством материальных и временных параметров.

Прочность защитной преграды является достаточной, если ожидаемое время преодоления ее нарушителем больше времени жизни предмета защиты или больше времени обнаружения и блокировки его доступа при отсутствии путей скрытного обхода этой преграды.

С учетом возможного отказа системы контроля прочность преграды будет определяться по формуле

$$P_{сзи} = P_{обл}(1-P_{отк}) \cup (1-P_{обх1}) \cup (1-P_{обх2}) \cup \ldots \cup (1-P_{обхj})$$

где $P_{обл}$ и $P_{отк}$ определяются соответственно по формулам.

Вероятность обнаружения и блокировки несанкционированных действий нарушителя:

$$P_{обл} = (1-P_{нр}),$$

где $P_{нр}$ определяется как $P_{нр} = 1 - \frac{t_н}{T_{обл}}$,

отсюда $P_{обл} = \frac{t_н}{T_{обл}}$.

При $t_н > T_{обл}$ попытка несанкционированного доступа (НСД) не имеет смысла, так как она будет обнаружена наверняка. В этом случае $P_{обл}=1$.

Вероятность отказа системы определяется по формуле

$$P_{отк}(t) = e^{-\lambda t},$$

где λ — интенсивность отказов группы технических средств, составляющих систему обнаружения и блокировки НСД;

t — рассматриваемый интервал времени функционирования системы обнаружения и блокировки НСД;

Рассмотрим вероятность преодоления преграды нарушителем, со стороны законного входа в систему. Оценку прочности будем производить по выше описанной формуле с учетом ниже приведенных параметров.

$$P_{нр} = \frac{n}{A^S},$$

где A^s— количество возможных значений кодов паролей;

n — количество попыток подбора кода пароля, обычно в проекте допускается три попытки на случай возможных ошибок законного пользователя;

A — число символов в выбранном алфавите кода;

S — длина кода пароля в количестве символов.

Как можно измерять надежность программы ?

Поскольку имеется несколько подходов к количественному измерению надежности, рассмотрим их терминологически. Во-первых можно рассчитать величину экономического риска вызванного возможностью ошибок в программе. По сути этот риск определяется разделами инструкции, устанавливающими использование результатов программы при данном режиме эксплуатации, и вероятностями ошибок, влияющими на каждый тип использования. Такой риск может быть основан только на прогнозе возможных ошибок, что значительно снижает ценность подобного подхода. В то же время разумно оценить реальные потери за период эксплуатации вызванные ошибками. Если определить тенденцию изменения средних реальных потерь за конкретный период (например, за месяц или год), можно прогнозировать экономический риск на будущее.

Говоря о статистической надежности программы, измеряемой как дополнительная вероятность обнаружения новой ошибки, не учтенной в предыдущих коррекциях, при очередном обращении к программе Простейшей оценкой статистической надежности является величина

$$P(n) = 1 - \frac{f(n)}{n} \pm e(d,n),$$

или

$$P(n) > 1 - \frac{f(n)}{n} - e(d,n),$$

где n - количество выполненных обращений к программе;

f(n) - число обнаруженных ошибок;

e(d,n) - доверительный интервал оценки вероятности ошибки f(n)/n при заданном уровне значимости.

В качестве оценки дисперсии d с гарантией можно пользоваться максимально возможной дисперсией (= 1/4) двоичной случайной величины .

Это соответствует подходу к программе как к черному ящику, иногда выдающему ошибки. Недостатком такого подхода является неявно использованная модель «урны с возвратом шаров», не учитывающая корректировки кортежа после обнаружения каждой новой ошибки. Ситуация исправляется, если учитывать только новые ошибки, не компенсируемые ранее сделанными корректировками инструкции.

Более сложной, но иногда оправданной оценкой статистической надежности может служить байесовская оценка вероятности верного срабатывания кортежа при задании некоторой экспертной оценки априорной вероятности ошибки и использовании статистической выборки отказов при обращениях к программе. Этот метод хорош, если есть серьезные основания для прогноза частоты ошибок в исходном кортеже. Не останавливаясь на конкретных формулах, отметим, что теоретически использование обеих оценок не вполне корректно, так как процесс, включающий коррекцию кортежа по каждой ошибке, не стационарен, а это предполагается в обосновании методов оценки вероятности.

Надо отметить также не вполне конструктивный, но логически безупречный подход к определению комбинаторной надежности программы, определяемой как отношение числа вариантов исходных данных, на которых программа срабатывает верно, к общему числу вариантов исходных данных. В условиях динамической корректировки кортежа эта надежность постоянно растет. Однако оценить ее можно только статистически, используя формулы, приведенные выше, для «препарированной» статистической выборки результатов обращений к программе, из которой выброшены повторные варианты исходных данных. В общем случае возникает та же трудность с потерей стационарности процесса при корректировках. Но имеется одна ситуация, где эта трудность не возникает: статистическая оценка комбинаторной надежности, полученная на основе многократного тестирования программы, работающей безошибочно на всей серии тестовых обращений. Это соответствует стадии тестирования программы в процессе отладки, когда каждая обнаруженная ошибка исправляется на уровне программы или инструкции, а потом тестирование начинается заново по полной программе. В этом случае на каждом прогоне тестов корректировок не возникает, и процесс возникновения ошибки остается стационарным. Оценка тогда дает $P(n) > 1 - e(\frac{1}{4}, n)$. Более тонкая оценка может быть основана на знании внутренней структуры программы.

Нестабильность работы программы естественно измерять числом зарегистриро-

ванных ошибок за определенный период эксплуатации, т.е. числом внесенных в инструкцию корректировок за этот срок. В период устойчивой работы оборудования количество внесенных корректировок f можно оценить численным интегралом по времени от нестабильности D(i), измеренной на последовательных интервалах времени длительности h(i)

$$f == D(l)h(l) + ... + D(n)h(n).$$

Поскольку технического износа у программы, инструкции и режима эксплуатации нет то до наступления износа компьютера нестабильность монотонно падает за счет корректировок. По мере износа компьютера (старение технической части среды) возникают специальные корректировки для обхода машинных сбоев и поломок, т.е. нестабильность начинает расти. При нормальном режиме эксплуатации предусматривается своевременная замена оборудования, и рост нестабильности ограничен введением предосторожностей на период освоения новых приборных средств.

Наиболее сложно измерить надежность объектно-ориентированного программного продукта. Как и процедурные, так и объектно-ориентированные программы могут рассматриваться как комплексы, состоящие из различного набора компонент.

При процедурном подходе это очевидно. Для объектно-ориентированного подхода это не столь очевидно. Объектные компоненты могут не иметь явных алгоритмов, а быть, например, описанием новых типов данных. Однако отсутствие явного описания методов работы данных не говорит об отсутствии метода вообще: благодаря принципу наследования, соответствующий метод обработки будет находиться в одном из классов-предков, но метод всегда имеется.

Второй особенностью моделирования надежности объектно-ориентированных программ является первоочередность понятия данных. Для того чтобы уравнять в правах данные и методы, сопоставим дополнительную тривиальную функцию. В зависимости от доступности данных тривиальные функции могут быть как публичными, так и приватными. Изменение данных осуществляется обращением к этой же тривиальной функции, но с аргументом, определяющим новое значение данного.

Введение тривиальной функции позволит, без потерь в общности, считать данные идеально надежными и, в дальнейшем, не обсуждать надежность данных. Весь элемент ненадежности перенесен с данных на тривиальные функции, которые, в свою очередь, являются некими методами, алгоритмами, надежность которых моделируется ранее разработанным методом.

Третьей особенностью объектных компонентов является наличие, как правило, многих входов, в отличие от процедурных компонентов, часто имеющих единственный вход. По сути, количество входов в объектно-ориентированный компонент определяется количеством публично доступных членов соответствующего класса.

Наличие нескольких входов в компонент важно потому, что нельзя говорить о надежности компонента в целом, а только о надежности, соответствующей определенному входу.

Таким образом, надежность компонента определяется не постоянным значением, а вектором, координатами которого являются надежности компоненты по каждому из входов.

В работе И.С. Кабак и Б.М. Позднеева описан один из методов оценки надежности программного продукта. Оценка надежности начинается с анализа его структуры, то есть с определения трех типов графов:

1. Графа G, определяющего структуру процедурной части объектно-ориентированной программы. Вершинами графа G являются либо программные модули (процедуры), либо методы классов данных. Граф G позволяет определить частотные характеристики V_i
2. Набора графов {G'i}, где каждый граф соответствует определенному классу данных. Каждый из графов набора описывается собственные (то есть, явно определенные) методы этого класса.
3. Графа G~, описывающего иерархию классов объектно-ориентированной программы. Использование этого графа позволяет определить для каждого класса заимствованные (то есть, неявно определенные) методы.

После того как структура программного продукта будет полностью определена, оценка надежности производится снизу-вверх.

Оценка надежности компонентов нижнего уровня.

Компонентами нижнего уровня являются компоненты, которые нельзя подвергнуть декомпозиции, так как они состоят из атомов-операторов исходного языка программирования и базовых классов данных.

Для каждого атома определяется статистически первоначальная вероятность $H = \sum_{i=1}^{N} v_i * h_j$ интенсивности потока отказов W_i

По структуре компонента определяется значения частотных коэффициентов Vi и по формуле подсчитывается первоначальная интенсивность потока отказов компонентов. При этом первоначальная интенсивность потока определяется не для компонента в целом, а для каждого его входа.

Оценка надежности компонентов верхних уровней.

Компоненты остальных уровней являются системами, состоящими из атомов и компонентов более низших уровней. Оценка их надежности проводится аналогично оценке компонентов низшего уровня, за исключением того, что для компонентов, входящих в их состав используется подсчитанная ранее первоначальная интенсивность потока отказов. Для оценки надежности программного обеспечения в целом, используется граф G.

Список использованных источников

1. Моисеенков И. Американская классификация и принципы оценивания безопасности компьютерных систем. КомпьютерПресс, 1992 №2,3
2. Гостехкомиссия РФ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа в автоматизированных системах и средствах вычислительной техники. М.,Воениздат,1992.
3. Галатенко В. Информационная безопасность. Открытые системы, 1995, №5(13).
4. Security Architecture for Open System Interconnection for CCITT Applications. Recommendation X. 800. – CCITT. Geneva, 1991.
5. Кабак И.С., Рапопорт Г.Н. Оценка надежности программного обеспечения по математической модели. Проблемы создания гибких автоматизированных поризводств под редакцией И.М. Макарова, К.В.Фролова, П.Н. Белялина, - М.: Наука, 1987.-стр. 236-245.
6. Кабак И.С., Либов Л.Я. Иммитационное моделирование надежности программного обеспечения. Сб. «Труды НИАТ», НИАТ, 1990
7. Г.Дийерс. Надежность программного обеспечения. М. Мир,1980.
8. Р.Лоцгботтом. Надежность вычислительных систем. Москва, Энергоатомиздат, 1985.
9. А.А. Штрик, Л.Г. Осовецкий, И.Г. Мессих. Структурное проектирование надежных программ встроенных ЭВМ. Ленинград, Машиностроение, 1989.