УДК 004.056.53

РЕЛЯЦИОННАЯ МОДЕЛЬ ОЦЕНКИ БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Крахмалев Д.В., Левченков А.Н., Костенко П.И.
Ростовский военный институт ракетных войск им. М.И. Неделина, г. Ростов-на-Дону
E-mail: skf-mtuci@mail.ru

Аннотация. Рассматривается подход к оценке уровней безопасности программного обеспечения на основе моделирования с последующим расчетом эталонного метрического профиля спецификаций.
Ключевые слова. Реляционная модель, безопасность, программное обеспечение.

Решение проблемы безопасности программного обеспечения (далее – ПО) в настоящее время не удается достигнуть прямым путем, так как для систем высокого уровня сложности практически невозможно избежать риска наличия потенциально опасных функций вследствие ошибки либо сознательно деструктивного поведения многочисленных участников процесса разработки программного обеспечения.

Созданные ранее методы проверки качества программного обеспечения [1] малоэффективны для оценки его безопасности.

В статье рассматривается подход к оценке уровней безопасности ПО на основе моделирования с последующим расчетом эталонного метрического профиля спецификаций.

Формальное построение базовой модели

Модель программного обеспечения автоматизированной системы строится с применением нетрадиционного для этой области математического аппарата – теории отношений [2,3,4]. Преимущество подобного подхода состоит, прежде всего, в том, что такая модель программного обеспечения может быть легко реализована в виде реляционной базы данных, содержащей информацию о структуре и характеристиках модулей программного средства. Дальнейшая работа с этой моделью может производиться при помощи операций реляционной алгебры, а на практике – в среде какой-либо из существующих систем управления базами данных, которая возьмет на себя рутинную техническую сторону выполнения операторов преобразования содержащихся в базе данных отношений.

В соответствии с общей теорией систем базовая модель (1) *R* ПО 1представляется как теоретико-множественное отношение

$$R \subset x\{D_i; i \in I\} = \Pi, \qquad (1)$$

заданное на семействе множеств

$$\overline{D} = \{D_i; i = I\}.$$

Интерпретировав элементы каждого из множеств D_i как значения атрибутов процессов передачи, обработки и хранения информации между модулями ПО, их характеристики, типы данных и операции, могут быть получены элементы множества Π как упорядоченные наборы значений атрибутов. В результате множество Π в таблице 1 представляется как отношение, доменами которого являются совокупности элементов множеств $D_i \in \overline{D}$, а кортежами (наборами) п-элементы множества Π:

Таблица 1

Наборы значений атрибутов множества Π

$\Pi =$

	D_1	D_2		D_n
π_1	$d_{1.1}$	$d_{1.2}$	…	$d_{1.n}$
π_2	$d_{2.1}$	$d_{2.2}$	…	$d_{2.n}$
⋮	⋮	⋮	⋮	⋮
π_m	$d_{m.1}$	$d_{m.2}$	…	$d_{m.n}$

В общем случае, среди множества наборов имеются такие, которые для моделируемого ПО являются семантически некорректными. Исключив их из рассматриваемого декартового произведения Π, получено множество (отношение) $R \subset \Pi$, описывающее свойства структуры и процессы передачи, обработки и хранения информации (типов данных) модулями программного средства.

Таким образом, базовая модель (2) ПО, на которой будут основываться все дальнейшие построения, представляется в виде унарного теоретико-множественного отношения:

$$\mathrm{E} = \langle R, \Pi \rangle, \qquad (2)$$

где: $R \subset \Pi$.

Далее осуществляется привязка построенной модели к учету основных особенностей реальной картины операций: *иерархическая подчиненность операций* - (не существует априорно заданного уровня детализации реальных операций), с одной стороны, последовательность логически связанных между собой операций может быть представлена в виде единой операции более высокого уровня, с другой стороны, каждая крупная операция при необходимости может быть детализована с использованием более мелких), *естественная последовательность операций* (на практике одни операции выполняются строго после других, и исследование этой последовательности играет важную роль в последующих результатах). В результате с целью модификации модели для учета описанных особенностей вводится система служебных атрибутов, обязательных при представлении каждой операции (система абсолютной нумерации), после чего приводится методика построения модели программного процесса, где программный процесс представляется в виде дерева операций:

$$T = \left\{ O, O\langle i_1 \rangle, O\langle i_1, i_2 \rangle, ..., O\langle i_1, i_2, ..., i_n \rangle \right\},$$

где корень дерева (операция O) соответствует процессу в целом, а каждое из последующих подмножеств $\{O\langle i_1, i_2, ..., i_n \rangle\}$ представляет собой последовательность операций n-ого уровня детализации, для каждой из которых последовательность индексов составляет уникальный квалифицированный номер. На основании полученного дерева операций проводятся исследование поддеревьев операций и вводится понятие цепи операций и обобщенных типов операций, которые непосредственно используются при исследовании основных метрических показателей:

удельного веса операций определенного обобщенного типа в данной цепи операций;

средней длины непрерывной подцепи операций определенного обобщенного типа;

показательности смежности операций в зависимости от их обобщенного типа:

частоты появления операции одного обобщенного типа непосредственно после операции другого обобщенного типа;

среднего расстояния между операциями двух выбранных обобщенных типов;
количественного соотношения операций двух выбранных обобщенных типов.

Метрически показатели предложено разделять на показатели спецификаций (рассчитываются аналитическим путем на основе идеализированного дерева операций, полученного путем пошаговой детализации формализованной системы спецификаций), программные показатели (рассчитываются аналитическим путем на основе фактического дерева операций, полученного путем анализа исходного текста программ), фактические показатели (получается путем проведения непосредственных измерений при выполнении реального ПО в специализированных условиях, например, с использованием программного эмулятора), обобщенные показатели (для класса ПО получается путем проведения непосредственных измерений фактических показателей для некоторой репрезентативной выборки программных продуктов данного класса с последующим усреднением результатов в рамках выбранной метрики).

Введенная система метрических показателей процесса зависит от ряда параметров:
выбора способа разбиения доменов на классы;
выбора конкретной цепи операций процесса;
выбора совокупности показателей из общего набора.

Для получения точных характеристик программного процесса необходимо совместно исследовать метрические показатели из различных выборок. В результате вводится понятие метрического профиля, под которым предложено понимать упорядоченный кортеж метрических показателей $\langle P_i \rangle$, каждый из которых получен для определенного фиксированного разбиения доменов и определенным образом выделенной цепи операций. При этом понятие метрического профиля уже не привязывается к какой-либо из цепей операций, а является характеристикой процесса в целом, а для расчета метрического профиля необходимо манипулировать с полной реляционной моделью операций.

Интегральная оценка уровня безопасности программного продукта осуществима на основе оценки расстояния всех трех типов получаемых метрических профилей между собой, а также расстояния от центров шаров, представляющих обобщенные классы программного обеспечения.

Реляционная модель программного обеспечения

При построении реляционной модели программного обеспечения вводится фиксированная градация уровней иерархической зависимости операций обработки информации: процессы; подпроцессы; укрупненные операции; типовые операции.

Поставив в соответствие атрибуту D_i модели *R* под номером один типы всех режимов, второму – процессы, третьему – операции и обозначив их через *A*1, *A*2 и *A*3 соответственно, получим, что кортежу отношения *R* соответствует одна элементарная передача или преобразование любого типа информации.

Таблица 2

Множества атрибутов

$A_y = \{A_{y1}, A_{y2}, ..., A_{yn}\}$	характеристики ПО средств управления вычислительного процесса (ВП)
$A_a = \{A_{a1}, A_{a2}, ..., A_{ak}\}$	характеристики системы адресации ВП
$A_d = \{A_{d1}, A_{d2}, ..., A_{dm}\}$	характеристики данных ВП
$A_s = \{A_{s1}, A_{s2}, ..., A_{sl}\}$	служебные признаки ВП
$A_p = \{A_{p1}, A_{p2}, ..., A_{pi}\}$	параметры и типы системы преобразований

На рисунке 1 представлена модель R ПО с учетом введенных атрибутов $A_1, A_2, A_3, A_y, A_a, A_d, A_s, A_p$:

A_1	A_2	A_3	A_y	A_a	A_d	A_s	A_p
k_1	p_1	c_1	t_1	g_1	l_1	m_1	ν_1
k_1	p_1	c_2	t_2	g_2	l_2	m_2	ν_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$k_{\max}$	p	c	t	g	l	m	ν_i

пересылка — операция — процесс — режим

Рис.1. Соответствие атрибутов модели типовым операциям

Для построения моделей типовых операций R_0, процессов R_p и режимов R_r используется аппарат реляционного исчисления. Запросы к модели должны содержать отличительные информационные признаки. К ним относятся признаки, включающие в себя значения режима, процесса, операции, которые обозначаются через W_k, W_j и Wi соответственно.

Модель типовой операции R_0 может быть положена из (1) отношения R с помощью следующего запроса

$$T_0 = \left(r\left[A_1 - A_p\right]\right): P_r \wedge \left(r\left[A_1\right] = W_k\right) \vee \left(r\left[A_2\right] = W_j\right) \wedge \left(r\left[A_3\right] = W_i\right) \quad (3)$$

Соответственно для R_p и R_r

$$T_p = \left(r\left[A_1 - A_p\right]\right): P_r \wedge \left(r\left[A_1\right] = W_k\right) \vee \left(r\left[A_2\right] = W_j\right), \quad (4)$$

$$T_r = \left(r\left[A_1 - A_p\right]\right): P_r \wedge \left(r\left[A_1\right] = W_k\right). \quad (5)$$

Адекватное модельное отражение особенностей конкретных программных процессов (3,4,5) достижимо на основе дополнения модельных описаний операций служебными атрибутами, отражающими их иерархическую зависимость и естественный порядок следования.

Большинство операций оценки уровня безопасности ПО может быть сведено к расчету, соответствующим образом выбранного расстояния в пространстве метрических профилей, представляющих собой упорядоченные кортежи метрических показателей, каждый из которых получен для определенного фиксированного разбиения доменов и определенным образом выделенной цепи операций.

Оценка уровня безопасности специального программного обеспечения включает несколько этапов:

Этап №1: на основе изучения спецификаций должен быть определён и уточнён класс, к которому должно относиться оцениваемое программное средство. Если это не сделано ранее, для данного класса должна быть произведена репрезентативная выборка программных средств и на основе её анализа должны быть рассчитаны обобщенные метрические показатели.

Этап №2: спецификации программного средства должны быть приведены к формализованному виду и на основе полученных формализованных описаний должно быть выполнено моделирование с последующим расчетом эталонного метрического профиля спецификаций.

Этап №3: исходные тексты программного средства, если они имеются, следует преобразовать в реляционную модель операций, после чего необходимо выполнить расчет соответствующего эталонного метрического профиля.

Этап №4: производятся непосредственные измерения фактического метрического профиля программного средства.

Этап №5: производится сравнение насчитанного фактического метрического профиля с эталонными профилями и с обобщенным профилем класса программного обеспечения путем вычисления метрических разностей.

Этап №6: Производится оценка попадания вычисленных разностей в диапазон допустимых значений для каждого класса защищенности программного обеспечения.

Этап №7: На основе полученной оценки может быть сделан вывод об отнесении программного обеспечения к тому или иному классу защищенности и окончательно оформлены документы отчётности.

В процессе проведения оценки уровня безопасности ПО оформляются протоколы испытаний по каждому показателю. По завершении оценки составляется официально утверждаемое экспертное заключение об уровне защищенности конкретного программного обеспечения.

Литература

1. Оценка качества программных средств. ГОСТ - 28195 - 89.
2. Левченков А.Н., Бабарицкий А.Н., Киба А.В., Синтез реляционной модели специального программного обеспечения // Информационная безопасность: Материалы VII междунар.науч.-практ. Конф. Таганрог, 2005. С.
3. Лобко В.Т. Инструментальное средство экспериментальной оценки метрик качества и безопасности программ // Известия высших учебных заведений. Северо-Кавказский регион. Технические науки, № 2, 1997.
4. Уткин Л.В., Шубинский И.Б. Нетрадиционные методы оценки надежности информационных систем. С-Пб.: Любавич, 2000. –173 с.