О подходе к обоснованию корректности проектирования монитора безопасности

Кривонос Ф. В.

Цнииатоминформ Минатома России

I

Монитор доступа, или, по-другому, монитор обращений, - и это общепринято - является ядром системы защиты. Но в то время, как каждая функция безопасности (аутентификация, аудит и т.д.) может быть локализована вплоть до фрагментов двоичного кода во множестве всего кода защищаемой системы, монитор является, скорее, архитектурной характеристикой, а именно, - таким свойством всех функций защиты, которое обеспечивает перехват всех обращений D субъектов S к объектам O с запросом операций из W, перед тем, как операция будет разрешена или отвергнута

D(S, O, W), где S – множество субъектов si (i∈ I),

O – множество объектов oj (j∈ J),

W – множество операций wk (k∈ K),

По-другому, монитор представляет собой периметр, первым встречающий запрос на всякую операцию (wk) перед ее возможным выполнением.

Главная задача монитора – контроль выполнения операции. Ошибка, то ли при проектировании, то ли при реализации может открыть «черный вход», через который могут выполняться несанкционированные запросы. Вопрос можно преднамеренно усложнить, если рассматривать тайные каналы. Монитор, в принципе, должен перехватывать запросы, которые влияют на изменение энтропии информации для субъекта. Но при такой широкой постановке вопроса вряд ли вообще возможно реализовать монитор.

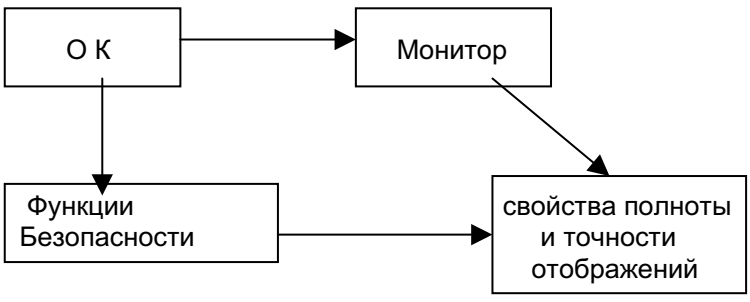
Какие отношения связывают мониторы разных представлений защищаемой системы? Они концентрированно выражены, на наш взгляд, в следующих постулатах такого всеоблемлющего (на сегодня) документа, как Общие Критерии (ОК) [1]. Следует учесть, что и ОК не низводит определение монитора к статичному коду продукта. То есть, архитектурное решение и здесь является определяющим, но приобретающим в ОК, на наш взгляд, правильное решение в виде динамики, которая дается через описание функций, через представления и отображения представлений того объекта, о мониторе которого мы говорим.

В ОК требуется, чтобы имелось достаточное число уровней представления объекта оценки (ОО) с необходимой степенью детализации для демонстрации того, что:

а) каждый уровень уточнения полностью отображает более высокие уровни (все функции, характеристики и режимы безопасности ОО, которые определены на более высоком уровне абстракции, необходимо наглядно представить на более низком уровне);

б) каждый уровень уточнения точно отображает более высокие уровни (не должно существовать функций, характеристик и режимов безопасности ОО, которые были бы определены на более низком уровне абстракции, но при этом не требовались бы на более высоком уровне).

Заметим, что полнота и точность отображений относится ко всем функциям безопасности в ОК, но эти же свойства полноты и точности распространяются и на монитор.



II

Представленные в ОК схемы правильного поуровневого проектирования (полного и точного) функций защиты можно сопроводить следующей моделью для такой обобщенной «функции» как монитор доступа (заметим, что для функций безопасности существует много различных моделей, и эти модели оказали влияние на функциональные требования безопасности ОК).

Модель монитора доступа

Вычислительная система характеризуется большим числом параметров. При моделировании такой системы выделяется группа существенных параметров, так что вся система проектируется на это подпространство.

В каждый момент времени набор параметров-координат представляет собой "точку" в этом пространстве состояний. Инициатором перехода из одного состояния в другое (из точки в точку) является выполнение команд (операторов) к текущему состоянию. Последовательность команд называется программой или, по-другому, функцией.

Пусть ***A*** - пространство состояний: ***A={(x1,x2,...,xN)}.*** Определим в ***A*** четыре непересекающихся подмножества:

C **-** свободная зона; ***З*** - закрытая зона; ***Г*** - граница, "разделяющая" ***C*** **и** ***З*****;**

H - оставшиеся в ***A*** точки.

Пусть ***K*** - множество команд (операторов) перевода состояний.

Определение 1. Вышеприведенную совокупность ***M={A, C, З, Г, K}*** будем называть моделью (системой) защиты.

Определение 2. Путем в ***M*** называется последовательность попарно соседних точек из ***A:*** ***a1,a2,a3,...an***, таких, что ***a2=к1(a1), a3=к2(a2), ..., an=кn-1(an-1),*** где ***к1,к2,к3,...,кn-1*** **-** последовательность команд из ***K*** (программа).

Определение 3. Обходом в ***M*** называется путь, начинающийся в ***C***, заканчивающийся в ***З***, который не пересекает границу ***Г*** (можно потребовать, чтобы и обратные пути из ***З*** в ***C*** пересекали границу ***Г***).

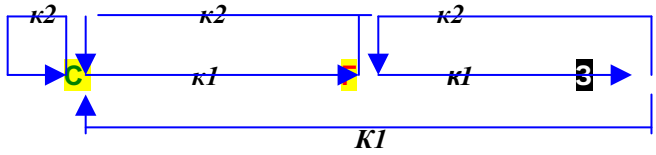
Определение 4. Система ***M*** называется безопасной, если в ней нет обхода (а граница ***Г*** называется, тогда, монитором).

Примеры "вырожденных" систем безопасности: ***C={∅ }*** *(*пустое мн-во); ***Г={∅}; З={∅}.***

Самая элементарная безопасная система: ***C, З, Г*** - одноточечные множества, т.е.: ***C={c}; З={з}; Г={г}, K={к1,к2}.***

Правила применения команд: **к1(*c*)=*г*, к1(*г*)=*з*, к1(*з*)=*c*; к2(*c*)=*c*, к2(*г*)=*c*, к2*(з)*=*г*.**

Схема элементарной безопасной системы (ЭБС)



Критерий наследования безопасности

Для правильного перехода от простых моделей безопасности к более сложным введем понятие правильного отображения моделей, называемое морфизмом моделей.

Определение 5. Морфизмом моделей ***M1={A1,C1,Г1,31,K1}*** и ***M2={A2,C2,Г2,32,K2}*** называется пара отображений пространств состояний ***f1: A1 -> A2,*** и команд ***f2: K1 -> K2,*** которые согласованы с действиями команд: ***f1(k*a) = f2(k)*f1(a).***

Определение 6. Отображение ***f: A -> B*** называется изолированным на подмножестве ***C*** из ***A***, если никакой элемент из дополнения ***A\C*** "не слипается" с элементами из ***C*** при этом отображении.

Теперь можно сформулировать некоторое достаточное условие, при выполнении которого вопрос о безопасности системы сводится к вопросу о безопасности менее сложной системы.

Утверждение. Пусть ***f=(f1,f2)*** - морфизм моделей ***M1*** и ***M2*,** причем ***C1*** отображается в ***C2, 31*** в ***32*,** а ***f1(Г1)*** покрывает всю ***Г2*** (эпиморфно при ***f1: Г1 -> Г2*).** Пусть **f** изолировано на **Г1**. Тогда, если ***M2*** модель без обхода, то и ***M1*** модель без обхода.

Доказательство.

Предположим противное, и ***@={a1,a2,...,an}*** - обход в ***M1***. Тогда @ не пересекается с ***Г1***, а ***a1*** принадлежит ***C1***, ***an*** принадлежит ***31***. Так как ***f*** изолированно на ***Г1, f1(@)*** не пересекается с ***f1(Г1*)**, а т.к. ***f*** эпиморфно, ***Г2*** лежит в ***f(Г1*)** и:

- ***f1(a1)*** принадлежит ***C2***, ***f1(an*)** принадлежит ***32***,
- ***f1(@)*** не пересекается с ***Г2*,**

т.е. ***f1(@)*** - обход в ***M2***. Противоречие. Следовательно, ***M1*** - безопасна.

Отсюда следует, что при развитии (детализации, т.е., расширении) системы необходимо, прежде всего, следить за границей ***Г*,** прообраз всех ее элементов должен целиком содержаться в границе прообраза самой системы.

Приведенная модель, как представляется, является некоторой формализацией требований ОК применительно к корректности отображений для монитора доступа.

Таким образом, ОК дает возможность разрабатывать формальные подходы и подтверждает свою методологическую роль.

Можно от системы команд ***{к_1,к_2,...к_n}*** перейти к другой, производной от нее, системе команд, когда каждая из команд является функцией (цепочкой) на предыдущем наборе команд, т.е. к системе ***{l_1,l_2,...l_m},*** где

l_i=(k_i1,k_i2,...,k_is), l_i(x)=(k_i1*k_i2*...k_is)(x).

Это означает переход от модели ***M1(A,C,3,Г,K)*** к модели ***M2(A,C,3,Г,L)*** с «макросами». Можно поставить вопрос о том, как взаимозависимы между собой вопросы о безопасности для ***M1*** и о безопасности для ***M2.***

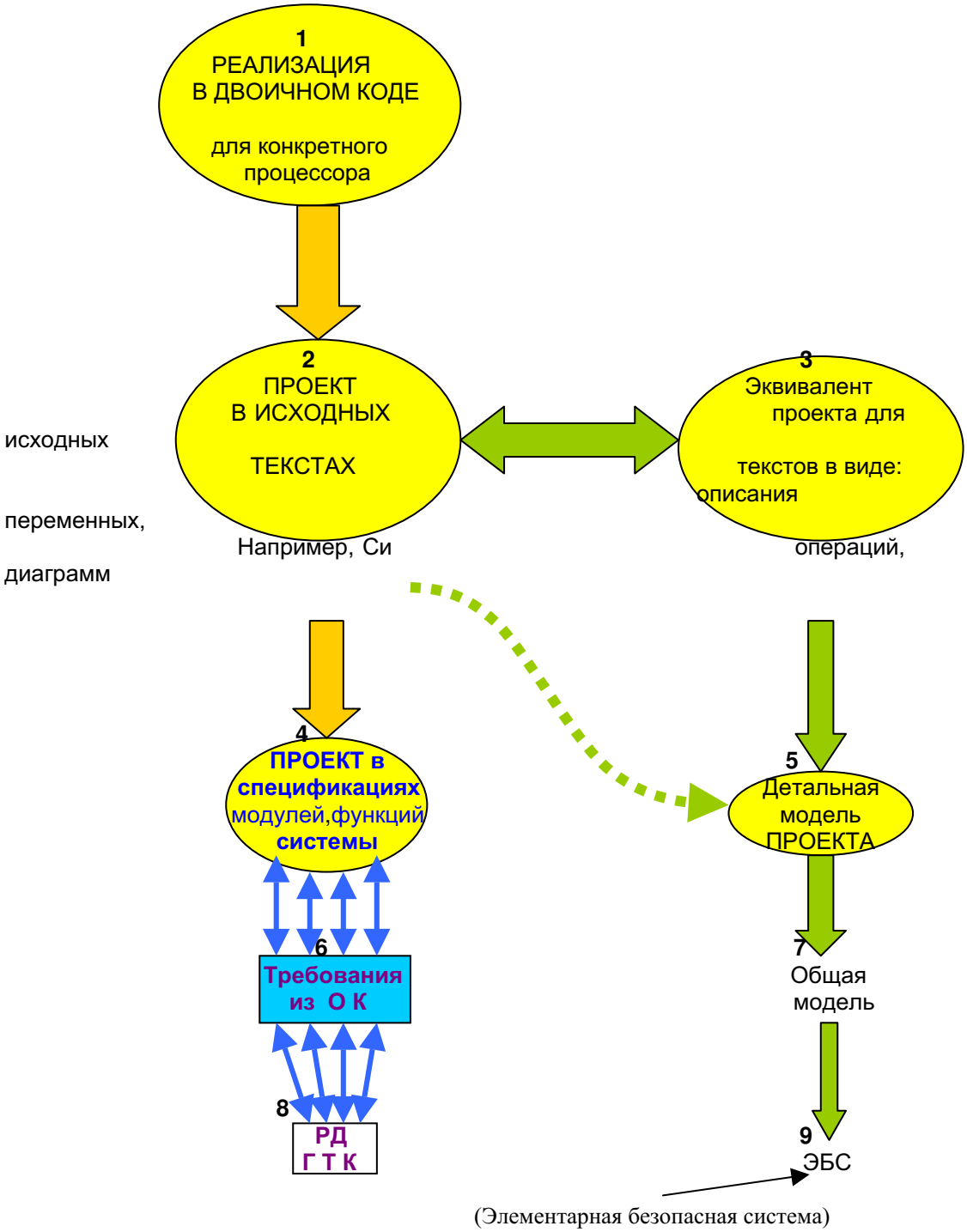
III

Модель монитора отражает некоторые положения ОК. Но напрашивается идея применить подобную конструкцию при проектировании реального продукта. Какой вид примет новая модель, в которой можно реально наследовать правила безопасности – это задача дальнейших исследований. Сейчас же укажем схему разработки, следуя которой разработку монитора можно считать корректной.

Процесс разработки, как он рассматривается в ОК, основан на уточнении требований безопасности, выраженных в задании по безопасности. Каждый последующий уровень уточнения представляет декомпозицию проекта с его дополнительной детализацией. Самым низким по степени абстракции уровнем является непосредственно реализация ОО.

Критерии уверенности в безопасности из ОК идентифицируют следующие уровни абстракции проекта: функциональная спецификация, проект верхнего уровня, проект нижнего уровня и реализация.

Имеющиеся «уровни проектирования» монитора и отображения между ними можно представить в виде следующей «обратной» диаграммы последовательных отображений, каждое из которых, предположительно, подчиняется правилу наследования безопасности, аналогичному правилу в рассмотренной модели.



Интуитивно существующие отображения <РЕАЛИЗАЦИЯ В ДВОИЧНОМ КОДЕ> =><ПРОЕКТ В ИСХОДНЫХ ТЕКСТАХ> => <ПРОЕКТ в спецификациях модулей, функций, системы>, с вложением требований Общих Критериев и Требований РД Гостехкомиссии России в алгоритмическую структуру <ПРОЕКТА в спецификациях модулей, функций, системы> должны, каким-то образом, перейти в формализованную модель, например, это должен быть переход <ПРОЕКТ В ИСХОДНЫХ ТЕКСТАХ> => <Детальная модель проекта>. Сегодня существуют продукты (компилятор переднего плана для языка Си), преобразующие исходный текст на языке Си, в некоторый вид представления, с одной стороны, семантически эквивалентного семантике исходного текста, с другой стороны, достаточно формализованного для того, чтобы отобразить это представление в детальную формализованную модель.

Тогда, в соответствии с некоторыми формализованными правилами, реализующими критерий наследования безопасности, мы сможем построить цепочку отображений вплоть до

элементарной безопасной системы. Отображенная на схеме подветвь (**2 → 3 → 5**), может реализовать переход <ПРОЕКТ В ИСХОДНЫХ ТЕКСТАХ> => <Детальная модель проекта> с дальнейшим продолжением до конечной элементарной безопасной системы ЭБС.

В заключение отметим, что правила проверки корректности отображений на каждой паре уровней (например, пара <1. реализация в двоичном коде> - <2. проект в исходных текстах>, пара <2. проект в исходных текстах> - <3. эквивалент исходников в виде диаграмм> и т.д.) будут возможны, если при проектировании и программировании уже следовать некоторым строгим правилам.

Литература.

1. Common Criteria for Information Technology Security Evaluation. Version 2.1. August 1999.