УДК 681.3.06

ФОРМАЛЬНЫЙ БАЗИС И МЕТАБАЗИС ОЦЕНКИ СООТВЕТСТВИЯ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ ОБЪЕКТОВ ИНФОРМАТИЗАЦИИ

Максим Иванович Гришин
руководитель группы испытаний
ЗАО «НПО «Эшелон»
Адрес: 107023, Москва, ул. Электрозаводская, 24
Тел.: +7(495)645-38-09. E-mail: mail@cnpo.ru

Алексей Сергеевич Марков
кандидат технических наук, доцент, CISSP
генеральный директор ЗАО «НПО «Эшелон»
доцент МГТУ им. Н.Э. Баумана
Адрес: 107023, Москва, ул. Электрозаводская, 24
Тел.: +7(495)645-38-09. E-mail: mail@cnpo.ru

Александр Владимирович Барабанов
руководитель группы испытаний
ЗАО «НПО «Эшелон»
Адрес: 107023, Москва, ул. Электрозаводская, 24
Тел.: +7(495)645-38-09. E-mail: mail@cnpo.ru

В статье предложен подход к разработке формализованной методики испытаний средств защиты информации по требованиям безопасности информации. Даны рекомендации по оптимизации процедуры испытаний средств защиты информации.

An approach to the development of security test procedures for information security tools is presented. The recommendations for optimizing the test procedure are obtained.

Ключевые слова: информационная безопасность, защита информации, средство защиты информации, средство вычислительной техники, сертификация, тестирование, испытания.

Keywords: information security, information protection, information security tools, means of computer facilities, certification, conformity assessment, security testing.

Введение

Обязательная оценка соответствия средств защиты информации (СЗИ) объектов информатизации проводится в форме сертификационных испытаний СЗИ. Процедура оценки соответствия применяется также на этапах приемосдаточных испытаний, аттестации объектов информатизации, контроля эффективности защиты информации. Требования к СЗИ определены в руководящих и нормативно-методических документах регуляторов, но при этом методики испытаний либо отсутствуют, либо носят описательный характер, что затрудняет автоматизацию и оптимизацию процессов оценки соответствия СЗИ. В статье рассмотрен подход к формализации общих и частных методик оценки соответствия СЗИ, позволяющий определить факторы, связанные со временем, стоимостью и полнотой испытаний СЗИ.

1. Формальный метабазис оценки соответствия

Под комплексом СЗИ понимается совокупность программных и технических элементов систем обработки данных, способных функционировать самостоятельно или в составе других систем, и предназначенное для предотвращения или существенного затруднения несанкционированного доступа к информации [1]. СЗИ включают ряд подсистем безопасности, таких как идентификация, аутентификация, разграничение доступа, контроль целостности, протоколирование и другие механизмы противодействия актуальным угрозам информационной безопасности.

Рассмотрим процесс испытаний СЗИ. Пусть $R=\{r_i\}$ – множество требований, предъявляемых к СЗИ Σ, а $T=\{t_i\}$ – множество тестовых процедур, проверяющих реализацию предъявляемых требований.

Под *методом разработки тестовых процедур* будем понимать отображение: $M{:}\Sigma\times R\to T$. Функция M на основе требования $r_i\in R$ и информации о реализации СЗИ Σ, подлежащего процедуре тестирования, выполняет генерацию тестовой процедуры $t_i\in T$, выполняемой для проверки удовлетворения СЗИ требованию . Как правило, функция для испытываемого СЗИ является биективным отображением.

Каждая тестовая процедура $t_i\in T$ характеризуется следующими элементами: цель выполнения, последовательность выполняемых действий, результаты выполнение тестовой процедуры, подлежащие регистрации, критерий принятия положительного решения.

Цель испытания содержит изложение намерения о выполнения оценки соответствия СЗИ предъявляемым требованиям. *Последовательность выполняемых действий* определяет набор шагов, выполняемых специалистом по тестированию для приведения СЗИ в исходное состояние при выполнении тестовой процедуры и генерации входной последовательности, подаваемой на вход СЗИ. *Результаты выполнения тестовых процедур* регистрируются с использованием различных программных средств проведения испытаний: инструменты генерации и перехвата сетевого трафика, программы поиска информации во внешней памяти, программы проверки системы разграничения доступа. *Критерий принятия положительного решения* должен содержать эталонные результаты выполнения тестовых процедур. В ходе проведения испытаний выполняется сравнение эталонных и фактических результатов выполнения тестовой процедуры и принимается решение о соответствии или несоответствии СЗИ предъявляемому требованию.

Введем операторы выполнения требования и корректности результатов тестирования F_R для данных Σ, r_i, t_i.

Оператор выполнения требования r_i для Σ СЗИ $F_R{:}\Sigma\times R\to\{0,1\}$:

$$F_R(\Sigma,r_i)=\begin{cases}1, & \textit{требование } r_i \textit{ выполнено для } \Sigma,\\ 0 & \textit{в противном случае.}\end{cases}$$

Оператор корректности выполнения тестовой процедуры t_i для Σ СЗИ $F_C{:}\Sigma\times T\to\{0,1\}$:

$$F_C(\Sigma,t_i)=\begin{cases}1, & \textit{тест } t_i \textit{ выполнен успешно для } \Sigma,\\ 0 & \textit{в противном случае.}\end{cases}$$

Оператор F_C показывает, что для СЗИ Σ выполнение тестовой процедуры завершилось успешно: фактические результаты, зарегистрированные при выполнении теста, соответствуют эталонным значениям, указанным в описании тестовой процедуры.

Методикой испытаний СЗИ назовем набор из пяти объектов $\Im=\{\Sigma,R,M,F_R,F_C\}$, где R – множество требований, предъявляемых к СЗИ Σ, M – метод разработки тестовых процедур, F_R и F_C – операторы выполнения требования и корректности выполнения тестовой процедуры соответственно, а также для $\forall r_i\in R$ справедливо $F_R(\Sigma,r_i)\Rightarrow F_C(\Sigma,M(\Sigma,r_i))$.

Методика предусматривает наличие трех стадий: планирование, тестирование и анализ результатов. На стадии *планирования* выполняется анализ документации и особенностей работы СЗИ. Перед началом проведения тестирования эксперты должны установить, что в технической документации на объект испытаний разработчик декларирует соответствие СЗИ требованиям R, то есть $F_R(\Sigma,r_i)=1$ для $\forall r_i\in R$. На основании данных, полученных в ходе анализа документации, тестовых запусков СЗИ и предъявляемых требований, формируется множество тестовых процедур $T=\{t_i\}$, где $t_i=M(\Sigma,r_i)$. *Тестирование* АС выполняется с использованием набора тестовых процедур $T=\{t_i\}$. В результате тестирования для каждой тестовой процедуры определяются результаты выполнение тестовой процедуры t_i, подлежащие регистрации. На стадии *анализа* фактических и эталонных значений получают множество упорядоченных пар вида $(t_i,F_C(\Sigma,t_i))$. Для СЗИ Σ декларируется соответствие требованиям $R=\{r_1,r_2,...r_i,...,r_n\}$, если:

$$\sum_{i=1}^{n}\left[F_R(\Sigma,r_i)\cdot F_C(\Sigma,M(\Sigma,r_i))\right]=n,$$

то есть в ходе проведения испытаний установлено соответствие реальных возможностей СЗИ декларируемым в документации или нормативном документе.

2. Методика испытаний средств защиты информации на соответствие требованиям безопасности информации

Базовым документом, в котором предъявляются требования к комплексу СЗИ, является руководящий документ Гостехкомиссии России по средствам вычислительной техники [1]. Документ устанавливает 7 классов защищенности и содержит требования к реализации дискреционного и мандатного принципов контроля доступа, очистки памяти, изоляции модулей, маркировки документов, защиты ввода и вывода на отчуждаемый физический носитель информации, сопоставления пользователя с устройством, идентификации и аутентификации, регистрации

событий, обеспечения целостности и др. Рассмотрим формализованный порядок проверки для указанных наиболее ресурсоемких требований $R_{СЗИ} = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ указанного нормативного документа.

2.1. Частная методика проверки дискреционного принципа разграничения доступа

Цель выполнения проверки состоит в определении степени соответствия СЗИ требованиям функциональных возможностей по реализации дискреционного принципа разграничения доступа. Введем определения, используемые при описании тестовой процедуры. Пусть $S = \{S_1, S_2, ..., S_i, ..., S_n\}$ – множество тестовых субъектов доступа и $O = \{O_1, O_2, ..., O_j, ..., O_m\}$ множество тестовых объектов доступа, $R = \{R_1, R_2, ..., R_k, ..., R_l\}$ – множество возможных прав доступа (например, чтение, запись, удаление и т.п.). Определим матрицу доступа $M = (m_{ij})$, $m_{ij} \subseteq R$, где m_{ij} – множество прав доступа тестового субъекта S_i к тестовому объекту O_j. Строка матрицы доступа соответствует субъекту S_i, а столбец – объекту O_j. На пересечении строки и столбца указывается множество прав доступа $m_{ij} \subseteq R$ соответствующего субъекта к данному объекту.

Оператор наличия права доступа субъекта к объекту в матрице $F_M : M, S_i, O_j, R_k \to \{0,1\}$:

$$F_M(M, S_i, O_j, R_k) = \begin{cases} 1, \text{если у субъекта } S_i \text{ есть право доступа } R_k \text{ к объекту } O_i, \\ 0, \text{в противном случае.} \end{cases}$$

Оператор фактического наличия права доступа субъекта к объекту $F_{fact} : S_i, O_j, R_k \to \{0,1\}$:

$$F_{fact}(S_i, O_j, R_k) = \begin{cases} 1, \text{если у субъекта } S_i \text{ есть право доступа } R_k \text{ к объекту } O_i, \\ 0, \text{в противном случае.} \end{cases}$$

Последовательность выполняемых действий может быть следующей:

1. Создание тестовых субъектов $S = \{S_1, S_2, ..., S_i, ..., S_n\}$ и объектов доступа $O = \{O_1, O_2, ..., O_j, ..., O_m\}$. Проверка должна проводиться для всех возможных субъектов и объектов доступа, перечень субъектов и объектов определяется на основе анализа документации на СЗИ.

2. Настройка правил разграничения доступа субъектов испытываемого СЗИ к тестовым защищаемым объектам. Данная операция заключается в настройке матрицы доступа $M = (m_{ij})$, $m_{ij} \subseteq R$. В ходе проведения тестирования проверяются все возможные права доступа субъектов к объектам, а так же их комбинации.

3. Тестирование настроек СЗИ: проверка фактического наличия права r_k у субъекта S_j по отношению к объекту O_i, таким образом, определяются все значения оператора $F_{fact}(S_i, O_j, R_k)$ для любых i, j, k. Проверка осуществляется с использованием тестовых попыток доступа (например, чтение, запись, удаление и т.д.) субъектов к объектам.

4. Сравнение фактических прав доступа с требуемыми правами, определенными в матрице доступа.

Результатами выполнения тестовой процедуры, подлежащие регистрации, являются:

1. Множество тестовых субъектов доступа $S = \{S_1, S_2, ..., S_i, ..., S_n\}$, множество тестовых объектов доступа $O = \{O_1, O_2, ..., O_j, ..., O_m\}$, $R = \{R_1, R_2, ..., R_k, ..., R_l\}$ множество возможных прав доступа.

2. Результаты настройки правил разграничения доступа – матрица доступа $M = (m_{ij})$, $m_{ij} \subseteq R$.

3. Результаты проверки фактического наличия права r_k у субъекта S_j по отношению к объекту O_i – значения оператора $F_{fact}(S_i, O_j, R_k)$ для всех i, j, k.

Определим критерий принятия положительного решения: в результате сравнения фактических прав доступа с требуемыми правами, определенными в матрице доступа, должно быть получено совпадение фактических и требуемых прав доступа:

$F_M(M, S_i, O_j, R_k) = F_{fact}(S_i, O_j, R_k)$ для $\forall\, i,\ j,\ k$

Проверка установленных прав доступа осуществляется путем осуществления попыток «явного» и «скрытого» доступа субъектов к объектам с фиксацией результатов проведенных попыток доступа: успешная или неуспешная.

Анализ полученных данных заключается в сравнении результатов проведенных попыток доступа с ожидаемыми результатами, которые отражены в тестовой матрице доступа.

Аналогично проверке дискреционного разграничения доступа проводится *проверка сопоставления пользователей и устройств*, но объектами доступа в данном случае будут различные устройства ввода-вывода.

2.2. Частная методика проверки мандатного принципа разграничения доступа

Введем определения, используемые при описании тестовой процедуры.

Пусть $S = \{S_1, S_2, ..., S_i, ..., S_n\}$ – множество тестовых субъектов доступа и $O = \{O_1, O_2, ..., O_j, ..., O_m\}$ – множество тестовых объектов доступа,

$M = \{m_1, m_2, ..., m_k, ..., m_l\}$ – множество классификационных меток (классификационных уровней) субъектов и объектов доступа (классификационные метки являются иерархическими:

$m_1 < m_2 <,..., m_{k-1} < m_k < m_{k+1},..., m_{l-1} < m_l$), m_{Oi} – классификационная метка i-го объекта доступа, m_{Sj} – классификационная метка j-го субъекта доступа.

Введем оператор $F_{READ} : S_i, O_j \to \{0,1\}$ проверки наличия права на чтение
и оператор $F_{WRITE} : S_i, O_j \to \{0,1\}$ проверки права на запись у субъекта S_i к объекту O_j:

$$F_{READ}(S_i, O_j) = \begin{cases} 1, \text{если у субъекта } S_i \text{ есть право доступа на чтение к объекту } O_j, \\ 0, \text{в противном случае.} \end{cases}$$

$$F_{WRITE}(S_i, O_j) = \begin{cases} 1, \text{если у субъекта } S_i \text{ есть право доступа на запись к объекту } O_j, \\ 0, \text{в противном случае.} \end{cases}$$

Последовательность выполняемых действий включает в себя следующие действия:

1. Создание тестовых субъектов
$S = \{S_1, S_2,..., S_i,..., S_n\}$, и объектов доступа
$O = \{O_1, O_2,..., O_j,..., O_m\}$.

2. Присвоение классификационных меток $M = \{m_1, m_2,..., m_k,..., m_l\}$ субъектам доступа (задается отображение $F_S : S \to M$, которое позволяет вычислять классификационный уровень любого субъекта доступа, т.е. $F_S(S_i) = m_{Si}$).

3. Присвоение классификационных меток $M = \{m_1, m_2,..., m_k,..., m_l\}$ объектам доступа (отображение $F_O : O \to M$, позволяет вычислить значения классификационного уровня любого объекта доступа $F_O(O_j) = m_{Oj}$).

4. Выполнение следующих тестовых попыток доступа субъектов к объектам:

– чтение данных из объектов доступа субъектами доступа, то есть вычисление значений $F_{READ} : S_i, O_j \to \{0,1\}$ для $\forall i, j$;

– запись данных субъектами доступа в объекты доступа, то есть вычисление значений $F_{WRITE} : S_i, O_j \to \{0,1\}$ для $\forall i, j$.

5. Проверка полученных результатов на соответствие правилам мандатного разграничения доступа.

Результаты выполнение тестовой процедуры, подлежащие регистрации, будут следующими:

1. Множество тестовых субъектов доступа $S = \{S_1, S_2,..., S_i,..., S_n\}$, множество тестовых объектов доступа
$O = \{O_1, O_2,..., O_j,..., O_m\}$, $M = \{m_1, m_2,..., m_k,..., m_l\}$ – множество классификационных меток (классификационных уровней) субъектов и объектов доступа.

2. Результаты настройки правил разграничения доступа – отображения $F_S : S \to M$, $F_O : O \to M$ (классификационные метки субъектов и объектов доступа).

3. Результаты проверки фактического наличия прав на запись и чтение у субъекта S_j по отношению к объекту O_i значения операторов $F_{READ} : S_i, O_j \to \{0,1\}$ и $F_{WRITE} : S_i, O_j \to \{0,1\}$ для $\forall i, j$.

В результате проверки правил мандатного разграничения доступа имеем следующие данные:

– субъект доступа S_j с классификационным уровнем m_{Sj} может осуществлять чтение данных из объекта доступа O_i с классификационным уровнем m_{Oi} тогда и только тогда, когда классификационный уровень субъекта доступа O_i выше, либо равен классификационному уровню объекта доступа S_j, т.е. для $\forall i, j$, $F_{READ}(S_i, O_j) = 1$ тогда и только тогда, когда $m_{Sj} \geq m_{Oi}$;

– субъект доступа S_j с классификационным уровнем m_{Sj} может осуществлять запись данных в объект доступа O_i с классификационным уровнем m_{Oi} тогда и только тогда, когда классификационный уровень объекта доступа O_i выше, либо равен классификационному уровню субъекта доступа S_j, т.е. для $\forall i, j$ $F_{WRITE}(S_i, O_j) = 1$, тогда и только тогда, когда $m_{Oi} \geq m_{Sj}$.

Аналогично проводится проверка защиты ввода и вывода на отчуждаемый физический носитель информации.

2.3. Частная методика проверки реализации очистки памяти

Введем определения, используемые при описании тестовой процедуры.

Пусть $A = \{a_1, a_2,..., a_i,..., a_l\}$ – множество участков памяти подлежащих проверке (оперативная память, разделы на жестком диске, внешние носители и т.д.), S – тестовая последовательность символов, используемая при проверке (уникальна для каждого участка памяти А). В ходе описания проверки будем использовать оператор наличия тестовой последовательности в памяти $F_{check} : a_i, S \to \{0,1\}$:

$$F_{check}(a_i, S) = \begin{cases} 1, \textit{последовательность } S \textit{ присутствует на участке } a_i, \\ 0, \textit{в противном случае.} \end{cases}$$

Последовательность выполняемых действий следующая:

1. Настройка средств очистки памяти.

2. Помещение тестовых данных в память (уникальная последовательность текстовых символов S).

3. Определение расположения тестовых данных в памяти (адрес, сектор диска и т.д.).

4. Перераспределение (освобождение) памяти с использованием штатных средств испытательного стенда.

5. Контроль наличия тестовых данных в памяти (повторный поиск и обращение по адресам определенным на начальном этапе) – определение значения $F_{check}(a_i, S)$.

6. Анализ полученных результатов.

7. Применение критериев оценки.

Результаты выполнения тестовой процедуры, подлежащие регистрации, следующие:

1. Результаты настройки средств очистки памяти.

2. Результаты контроля наличия тестовых данных в памяти после ее перераспределения (освобождения).

Критерий принятия положительного решения следующий: последовательность символов S, помещенная в память, после освобождения (перераспределения) памяти не была обнаружена, т.е. $F_{check}(a_i,S)=0,\forall a_i$.

2.4. Частная методика проверки реализации изоляции модулей

Обеспечение изоляции модулей следует из наличия у каждого процесса (запущенного от имени какого-либо пользователя) отдельного адресного пространства и невозможности прямого доступа к данному пространству процессами, запущенными от имени других пользователей.

Последовательность выполняемых действий:

1. Запуск программ (процессов) от имени различных пользователей.

2. Попытки доступа к памяти процессов запущенных от своего имени.

3. Попытки доступа к памяти процессов запущенных от имени других пользователей (субъектов).

4. Попытки доступа к физической памяти ПЭВМ.

5. Анализ результатов.

6. Применение критериев оценки.

Результатами выполнения тестовой процедуры, подлежащие регистрации, являются факты попыток доступа к памяти процессов, запущенных от имени других пользователей, и попыток доступа к физической памяти ЭВМ.

Критерий принятия положительного решения следующий: доступ к памяти процессов, запущенных от имени других пользователей, и к оперативной памяти в ходе испытаний получен не был.

2.5. Частная методика проверки механизмов идентификации и аутентификации субъектов доступа

Введем определения, используемые при описании тестовой процедуры. Пусть A – алфавит паролей и идентификаторов пользователей СЗИ. Обозначим пользователя $id\in ID\subseteq A^*$, пароль $id\in ID\subseteq A^*$. Учетная запись пользователя $id\in ID\subseteq A^*$ характеризуется следующим кортежем $usr_i=(id_j,pwd_k)$. Введем оператор корректности учетных данных $F_{AUT}:USR\to\{0,1\}$:

$$F_{AUT}(usr)=\begin{cases}1, \textit{доступ к СЗИ получен},\\ 0\ \textit{в противном случае}.\end{cases}$$

При проверке корректности реализации механизмов идентификации и аутентификации субъектов может быть использована следующая последовательность выполняемых действий:

1. Включение механизма идентификации и аутентификации СЗИ и создание множества учетных записей субъектов доступа $USR=\{usr_1,usr_2,...\}$.

2. Выполнение запросов на идентификацию и проведение аутентификации с использованием различных сочетаний учетных данных: зарегистрированный/незарегистрированный идентификатор, верный/неверный пароль $try_i=(id_j,pwd_k)$.

3. Анализ полученных данных.

Результаты выполнение тестовой процедуры, подлежащие регистрации:

1. Конфигурация СЗИ множество USR учетных записей субъектов доступа.

2. Полученные результаты тестовых запросов на идентификацию и аутентификации: множество $\{F_{AUT}(try_1),F_{AUT}(try_2),...\}$.

Критерии принятия положительного решения:

1. После ввода зарегистрированного идентификатора и пароля пользователю предоставляется доступ к защищаемым ресурсам:
$F_{AUT}(try_i)=1\Leftrightarrow try_i\in ADM$.

2. После ввода незарегистрированного идентификатора и/или неверного пароля пользователю отказывается в доступе к защищаемым ресурсам: $F_{AUT}(try_i)=0\Leftrightarrow try_i\notin ADM$.

Проверка надежности связывания идентификации и аутентификации пользователя со всеми его действиями осуществляется в ходе проведения проверок дискреционного и мандатного принципов разграничения доступа. По окончании каждой из проверок анализируется содержимое журналов регистрации событий.

Проверка считается успешно выполненной, если журнал аудита содержит записи обо всех попытках доступа и всех действиях любых пользователей (в том числе администраторов безопасности), осуществленных в ходе проверок по предыдущим пунктам. При этом в каждой регистрационной записи присутствует информация об идентификаторе пользователя, который был предъявлен при попытке доступа, и/или под которым пользователь был зарегистрирован в системе и выполнял различные действия.

2.6. Частная методика проверки обеспечения целостности

Пусть $FILE = \{file_1, file_2, ..., file_n\}$ множество файлов СЗИ (конфигурационные файлы, программные модули). Введем операторы нарушения целостности F_{MOD} и контроля целостности файлов СЗИ F_{INT}.

Оператор нарушения целостности $F_{MOD} : FILE \to \{0,1\}$:

$$F_{MOD}(file) = \begin{cases} 1, \text{целостность файла нарушается при проведении испытаний,} \\ 0 \text{ в противном случае.} \end{cases}$$

Оператор контроля целостности файлов СЗИ $F_{INT} : FILE \to \{0,1\}$:

$$F_{INT}(file) = \begin{cases} 1, \text{целостность файла нарушена,} \\ 0 \text{ в противном случае.} \end{cases}$$

Обозначим $FILE^{\Delta} = \{file_1^{\Delta}, file_2^{\Delta}, ..., file_n^{\Delta}\}$ множество файлов СЗИ, модифицированных в ходе проведения испытания. При этом выполняется модификация файла $file_i$ в файл $file_i^{\Delta}$. При проверке корректности реализации механизма контроля целостности СЗИ может быть использована следующая последовательность выполняемых действий:

1. Настройка средств контроля целостности (реакция на нарушение целостности, способ контроля целостности, период проверки, условие проверки и т.д.) и идентификация множества файлов СЗИ $FILE = \{file_1, file_2, ...\}$.
2. Внесение изменений в файлы СЗИ (изменение конфигурации, подмена (модификация) исполняемых файлов и т.п.) – получение множества измененных файлов $FILE^{\Delta} = \{file_1^{\Delta}, file_2^{\Delta}, ...\}$.
3. Инициализация проверки целостности файлов СЗИ (создание условий, при которых СЗИ осуществляет контроль целостности).
4. Анализ реакции СЗИ на нарушение целостности своей программной или информационной части.

Результаты выполнение тестовой процедуры, подлежащие регистрации:

1. Множество файлов СЗИ $FILE = \{file_1, file_2, ...\}$.
2. Множество модифицированных файлов СЗИ $FILE^{\Delta} = \{file_1^{\Delta}, file_2^{\Delta}, ...\}$.
3. Реакции СЗИ на нарушение целостности: $F_{INT}(file_1^{\Delta}), F_{INT}(file_2^{\Delta}), ...$

Критерий принятия положительного решения: СЗИ обнаружены все факты нарушения целостности: $F_{INT}(file_i^{\Delta}) = F_{MOD}(file_i)$ для $\forall i \in [1, n]$.

3. Рекомендации по оптимизации процедуры проведения испытаний

Опыт проведения испытаний различных СЗИ, проводимых аккредитованной испытательной лабораторией, показывает, что основной проблемой является рост временных и материальных затрат на выполнение типовых операций при росте сложности разрабатываемых СЗИ и большим количеством платформ и сред, на которых данные СЗИ могут функционировать. Например, при проведении проверки механизма дискреционного разграничения доступа необходимо выполнить $|S| \cdot |O| \cdot |R|$ типовых операций.

В общем случае можно показать, что время τ_{Σ}, затрачиваемое на проведение испытаний, носит экспоненциальный характер роста: $\tau_{\Sigma} \sim n \cdot v^{w}$, где n – число проверяемых требований, w – число факторов тестирования (например, «учетная запись пользователя»), v – число возможных значений фактора тестирования.

Задача оптимизации процедуры проведения испытаний СЗИ может быть сформулирована следующим образом. Пусть $\tau : T \times \Sigma \to \mathbb{R}_0$ – время, затрачиваемое экспертами на тестирование СЗИ Σ с использованием тестовой процедуры t_i. Пусть отображение вида $C : R \times \Sigma \to \mathbb{R}_0$ – затраты на проведение тестирования СЗИ Σ требованиям R. Задача оптимизации выглядит следующим образом (минимизация времени тестирования при ограничениях на затраты):

$$\begin{cases} \sum_i \tau(t_i, \Sigma) \to \min, \\ \sum_i C(r_i, \Sigma) \le C_M, \end{cases}$$

где C_M – ограничения, накладываемые на затраты.

В качестве методических приемов, позволяющих решить задачу оптимизации, могут быть предложены следующие.

1. *Совмещение отдельных видов испытаний*. При проведении испытаний СЗИ рекомендуется выполнять совмещение некоторых видов испытаний. Например, процедура тестирования подсистемы регистрации событий может быть совмещена с процедурой тестирования подсистемы разграничения доступа и идентификации/аутентификации.

2. *Тестирование с использованием методов комбинаторного покрытия*. Тестирование всех возможных комбинаций входных воздействий, позволяющее полностью исследовать поведение программы, является очень трудоёмким. Тестирование методом комбинаторного покрытия – подход, позволяющий вместе со снижением затрат на тестирование повысить его эффективность, а так же снизить вероятность ошибок в программном обеспечении. Основная идея данного метода состоит в следующем. Отказ в работе ПО, в большинстве случаев, может быть вы-

зван не некорректным значением одного входного параметра, а комбинацией двух или более входных параметров (от 2 до 6 параметров в соответствии с эмпирическими результатами). Метод комбинаторных покрытий на практике может быть применим как для тестирования конфигураций, так и тестирования входных параметров.

3. *Использование программных средств тестирования.* Для сокращения временных затрат при проведении испытаний необходимо использовать программные средства, позволяющие автоматизировать процедуру тестирования. Могут использоваться как инструментальные средства, широко представленные на современном рынке программного обеспечения, так и программы собственной разработки, написанные на языках сценариев [2, 3].

Заключение

Рассмотренное в работе решение задачи формализации общей методики и процедур испытаний подсистем безопасности комплексных СЗИ позволяет упростить автоматизацию оценки соответствия СЗИ и изделий в защищенном исполнении.

Основной проблемой, с которой сталкиваются организации при проведении испытаний, является рост временных и материальных затрат, связанный с большим количеством однотипных проверок функциональных возможностей объекта испытаний на возможных областях входных данных. Предложенные методические приемы позволяют сократить затраты на проведение оценки соответствия и решить задачу минимизация времени оценки соответствия при заданных ограничениях на затраты.

Концептуальный подход к формализации процедуры оценки соответствия может быть рекомендован при проведении разного рода испытаний программных и технических средств и систем в защищенном исполнении [4-10].

Литература

1. Законодательно-правовое и организационно-техническое обеспечение информационной безопасности АС и ИВС / Котенко И.В., Котухов М.М., Марков А.С. и др. СПб: ВУС, 2000. 190 с.

2. Барабанов А.А. Инструментальные средства проведения испытаний систем по требованиям безопасности информации // Защита информации. Инсайд. 2011, №1. С. 2-4.

3. Марков А.С., Миронов С.В., Цирлов В.Л. Опыт тестирования сетевых сканеров уязвимостей // Информационное противодействие угрозам терроризма, 2005. № 5. С. 109-122.

4. Барабанов А.В., Марков А.С., Фадин А.А. Сертификация программ без исходных текстов // Открытые системы. СУБД. 2011. №4. С. 38-41.

5. Марков А.С., Миронов С.В., Цирлов В.Л. Выявление уязвимостей в программном коде // Открытые системы. СУБД. 2005. №12. С. 64-69.

6. Марков А.С., Маслов В.Г., Цирлов В.Л., Олексенко И.А. Тестирование и испытания программного обеспечения по требованиям безопасности информации // Известия Института инженерной физики. 2009. №2(12). С. 2-6.

7. Подход в тестировании, сокращающий время разработки программного обеспечения / Д.Ю.Бурлак, А.В.Седаков, А.Н.Сударенко, П.С.Соколов // Известия Института инженерной физики. 2010. №2(16). С. 29-32.

8. Gourlay J. S. A Mathematical Framework for the Investigation of Testing // IEEE Transactions on Software Engineering. 1983, Vol. SE-9, №6. P. 686-709.

9. Kuhn R., Kacker R., Lei Y. Practical Combinatorial Testing // NIST Special Publication 800-142. Washington: U.S. Government Printing Office, 2010. 75 p.

10. Tian J. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement. Wiley, 2005. 440 p.