УДК 004.056.5

Адаптация существующих способов верификации для программно-аппаратных СЗИ

Т. М. Каннер
Закрытое акционерное общество "ОКБ САПР", Москва, Россия

Рассматриваются особенности верификации программно-аппаратных средств защиты информации (СЗИ). Предложен адаптированный способ верификации программно-аппаратных СЗИ, реализующий процедуры формальной оценки критичности выявленных в ходе тестирования ошибок в их функциях безопасности и алгоритмы расчета критичности ошибок, используемые перед внедрением СЗИ в информационную систему. Описана программная реализация предложенного способа, представляющая собой систему поддержки принятия решений, позволяющую автоматически оценить критичность выявленных в таких средствах защиты ошибок.

Ключевые слова: верификация программно-аппаратных СЗИ, адаптированный способ верификации программно-аппаратных СЗИ, программа "Верификатор программно-аппаратных СЗИ".

Реализация функций безопасности в программно-аппаратном средстве защиты информации включает проектирование (в том числе формирование требований к функциям безопасности СЗИ), непосредственно разработку программного обеспечения и аппаратной компоненты СЗИ (включая ее внутреннее ПО), выполняющих функции безопасности, верификацию перечисленных компонент, включающую их тестирование, исправление найденных при тестировании ошибок, финализацию и выпуск средства защиты [1].

Этап проектирования связан с выработкой требований к функциям безопасности программно-аппаратного СЗИ. Требования к конкретному средству защиты формируются на основании задач конечного пользователя с использованием требований-регуляторов в области защиты информации (ФСТЭК и ФСБ России) к классу СЗИ, которому оно соответствует. Далее выполняется разработка программной и аппаратной компонент СЗИ, реализующих требуемые функции безопасности. Вне зависимости от вида программно-аппаратного СЗИ [2] до этапа финализации и выпуска его функции безопасности должны быть проверены на соответствие предъявляемым требованиям. Такая проверка проводится как при первом выпуске СЗИ, так и при последующих обновлениях его версий. Для этого выполняется тестирование функций безопасности программно-аппаратного СЗИ, на основании результатов которого проводятся верификация и, если это необходимо, исправление найденных при тестировании ошибок [1].

Таким образом, все перечисленные этапы, кроме первого, как правило, приходится неоднократно повторять в процессе жизненного цикла программно-аппаратного СЗИ. Однако наибольшее внимание необходимо уделять верификации как основополагающему этапу, от которого зависит выпуск программно-аппаратного СЗИ.

Верификация включает в себя не только тестирование СЗИ (с выявлением некорректного поведения, указывающего на наличие в нем ошибок, фиксацией проявления ошибок, локализацией проявлений ошибок, анализом локализованных проявлений ошибок, фиксацией ошибок и особенностей), а также классификацию ошибок и особенностей (определение их типа, возможности компенсации, степени критичности) и принятие решения об итогах верификации и о возможности финализации или возвращении СЗИ на доработку [1].

Для средств защиты, в том числе программно-аппаратных, процесс верификации имеет некоторые особенности, касающиеся анализа влияния ошибок, обнаруженных при выполнении реализованных функций безопасности, на защищенность системы, в которой эти средства используются.

Ошибки, выявленные при тестировании функций безопасности СЗИ, необходимо отнести к одному из следующих видов: несущественные опечатки и ошибки (например, опечатка в выводимом сообщении или некорректное название какой-либо функции), не влияющие на корректность выполнения функций безопасности; ошибки, приводящие к неработоспособности одной или нескольких

Каннер Татьяна Михайловна, руководитель обучающего центра.
E-mail: tatianash@okbsapr.ru

Статья поступила в редакцию 4 декабря 2017 г.

функций безопасности СЗИ; ошибки, приводящие к неработоспособности или нарушению защищенности системы, в которой используется средство защиты.

Перечисленные виды ошибок неравнозначны с точки зрения работы СЗИ, поэтому необходима определенная градация всех найденных ошибок относительно их влияния на выполнение основной задачи — защиты информационных ресурсов и обеспечения безопасности данных.

Как правило, шкала критичности ошибок определяется индивидуально для конкретного средства защиты. При этом корректность оценки критичности ошибок является основополагающим фактором при принятии решения о возможности финализации СЗИ.

В качестве примера рассмотрим шкалу критичности ошибок, найденных при тестировании функций безопасности одного из известных программно-аппаратных СЗИ. Выделим несколько типов ошибок [1]:

- Ошибки интерфейса: недочеты в удобстве пользовательского интерфейса, корректности отображения всех его элементов, опечатки в системных сообщениях и т. п. (не влияют на выполнение функций безопасности СЗИ, функциональность и защищенность системы, в которой применяется СЗИ). Исправление таких ошибок необходимо для обеспечения комфортной работы конечного пользователя, и их наличие не опасно для защищаемой системы. Соответственно им присваивается минимальный уровень критичности.
- Ошибки, ограничивающие функциональность средства защиты без нарушения его функций безопасности (накладывают некоторые ограничения на функциональность средства защиты, не подвергая опасности защищаемую систему в целом: либо некорректно функционирующие возможности СЗИ не отвечают непосредственно за безопасность защищаемой системы, либо отсутствие этих функций можно компенсировать за счет других средств и мер без снижения уровня защищенности). Таким ошибкам присваивается средний уровень критичности.
- Ошибки, связанные с нарушением функций безопасности СЗИ (способные повлиять на защищенность системы или данных из-за нарушения функций безопасности средства защиты, которое создает предпосылки для успешной реализации атаки с использованием возникшей уязвимости). Являются наиболее критичными (присваивается максимальный уровень критичности). Наличие даже одной ошибки этого типа может привести к завершению верификации запретом финализации и выпуска продукта, кроме тех случаев, когда это нарушение можно компенсировать дополнительными средствами и мерами.

При анализе полученных в результате классификации ошибок необходимо определить выполняются ли требования, предъявленные к функциям безопасности рассматриваемого средства защиты, или есть ошибки с максимальным уровнем критичности, которые не позволяют принять решение о начале финализации и выпуске СЗИ. При отсутствии наиболее критичных ошибок выносится решение о выпуске средства защиты, в противном случае СЗИ отправляется на доработку с последующим повтором этапов тестирования и верификации.

Однако принятие решения о выпуске СЗИ именно на основании наличия/отсутствия критических ошибок СЗИ не может гарантировать качественной оценки работоспособности рассматриваемого средства защиты. Поэтому необходимо использовать более сложные методы, позволяющие дать расширенную оценку его работоспособности.

Адаптированный способ "Верификация программно-аппаратных СЗИ, реализующих функции безопасности, основанная на классификации обнаруженных ошибок, анализе степени их критичности и влияния на защищенность системы или данных"

Для принятия решения о возможности выпуска СЗИ рекомендуется использовать методы из теории оптимизации и принятия решений [3], в которых данная задача может рассматриваться как задача принятия решения в условии полной определенности (наличие определенного типа ошибки ведет к "ухудшению" СЗИ в некоторой известной степени). Для решения такой задачи можно использовать, например, известный алгоритм простого выбора, основанный на методах Саати, Коггера и Ю [4], оценивая СЗИ (именуемые альтернативами) по критериям оценки с учетом "важности" каждого из них, рассчитывая при этом некоторую сумму — значение обобщенного критерия.

Решать поставленную задачу с помощью такого математического аппарата можно за счет того, что:

- описанная задача является задачей многокритериального выбора в условиях определенности с малым числом критериев и альтернатив;
- в качестве критериев оценки можно использовать как качественные (наличие/отсутствие ошибки), так и количественные (количество тех или иных ошибок) характеристики, т. е. при необ-

ходимости цель задачи может быть изменена без изменения применяемого аппарата;

- при фиксации ошибок во время тестирования всегда можно выявить необходимую дополнительную информацию (количество ошибок, степень их "важности" и т. д.);
- для тестирования и верификации важно получить однозначный ответ – возможно или нет осуществить выпуск СЗИ (например, зная некоторую допустимую грань критичности ошибок в средстве защиты), а указанные методы больше всего подходят в данном случае, когда другие могут предложить множество оптимальных ответов (так называемое множество Парето [4]).

Алгоритм простого выбора, основанный на методах Саати, Коггера и Ю, предполагает первоначально построить вектор весов (весовых коэффициентов) для критериев оценки (в нашем случае это критерии f_1, f_2 и f_3 — наличие ошибок трех уровней критичности) — $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, обладающий свойством нормированности ($\Sigma\, \alpha_i = 1$, $i = 1, 2, 3$).

Для этого по заданной шкале критичности вначале определяются отношения попарного превосходства критериев оценки между собой: $\alpha_{12} = \alpha_1 / \alpha_2$, $\alpha_{13} = \alpha_1 / \alpha_3$, $\alpha_{23} = \alpha_2 / \alpha_3$ (т. е. от качественных характеристик критериев делается переход к количественным отношениям), а затем с помощью решения линейного уравнения, полученного из условия нормированности, определяются сами значения весовых коэффициентов α_1, α_2 и α_3.

Далее аналогично рассчитываются значения частных критериев оптимальности, соответствующие альтернативам:

$$\alpha^{(1)} = (f_1(\text{alt}_1), \ldots, f_1(\text{alt}_k)),$$
$$\alpha^{(2)} = (f_2(\text{alt}_1), \ldots, f_2(\text{alt}_k)),$$
$$\alpha^{(3)} = (f_3(\text{alt}_1), \ldots, f_3(\text{alt}_k)),$$

где $\text{alt}_1, \ldots, \text{alt}_k$, $k \in N$ — оцениваемые альтернативы (СЗИ). Т. е. на основе качественных и количественных характеристик альтернатив делается переход к количественным отношениям:

$$\alpha^1_{12} = \alpha^1_1 / \alpha^1_2, \ \ldots, \ \alpha^1_{k-1k} = \alpha^1_{k-1} / \alpha^1_k,$$
$$\alpha^2_{12} = \alpha^2_1 / \alpha^2_2, \ \ldots, \ \alpha^2_{k-1k} = \alpha^2_{k-1} / \alpha^2_k,$$
$$\alpha^3_{12} = \alpha^3_1 / \alpha^3_2, \ \ldots, \ \alpha^3_{k-1k} = \alpha^3_{k-1} / \alpha^3_k.$$

После этого с учетом условия нормированности векторов $\alpha^{(1)}, \alpha^{(2)}$ и $\alpha^{(3)}$ решается система линейных уравнений и определяются значения $f_1(\text{alt}_1)$, $f_2(\text{alt}_1)$, $f_3(\text{alt}_1), \ldots$, $f_1(\text{alt}_k)$, $f_2(\text{alt}_k)$, $f_3(\text{alt}_k)$ — весовые коэффициенты значимости того или иного вида ошибок в конкретной альтернативе.

Для оценки критичности ошибок в альтернативах необходимо вычислить и сравнить значения обобщенных критериев по следующей формуле линейной свертки:

$$J(\text{alt}_i) = \sum \alpha_j f_j(\text{alt}_i), \ \ j = 1, 2, 3, \ \ i = 1, \ldots k, \ \ k \in N.$$

Большее значение обобщенного критерия соответствует наличию больших по общей критичности ошибок в той или иной альтернативе, поэтому задача сводится к минимизации значения обобщенного критерия.

С использованием описанного математического аппарата можно сравнивать между собой различные версии одного и того же СЗИ либо использовать определенную версию как эталон и выпускать средство защиты только при показателях, близких или превышающих показатели эталона (однако данная оценка может быть неточной из-за присутствия не выявленных в ходе тестирования СЗИ ошибок). Кроме того, можно ввести новый уровень иерархии критериев оценки — критичность самих функций безопасности, в которых выявлены ошибки того или иного уровня критичности. Для решения такой задачи можно использовать прежний математический аппарат и аналогичные расчеты.

На основании сказанного можно предложить адаптированный способ "Верификация программно-аппаратных СЗИ, реализующих функции безопасности, основанная на классификации обнаруженных ошибок, анализе степени их критичности и влияния на защищенность системы или данных". Этот способ имеет широкое назначение и применим не только для всех существующих видов программно-аппаратных СЗИ [2], но также и для программных (но не актуален для ПО или любых других программно-аппаратных средств). Данный способ позволяет оценить критичность ошибок, выполнить анализ полученных результатов и степени влияния ошибок на защищенность системы, на основании чего принять решение об успешном завершении тестирования или о возврате СЗИ на доработку. Согласно предложенному способу необходимо:

1) сформировать шкалу критичности ошибок для верифицируемого программно-аппаратного СЗИ (например, в соответствии с предложенным разделением ошибок на типы: несущественные и не влияющие на корректность выполнения функций безопасности; приводящие к неработоспособ-

ности одной или нескольких функций безопасности; приводящие к неработоспособности или нарушению защищенности системы, в которой используется СЗИ);

2) провести тестирование программно-аппаратного СЗИ, например на разных аппаратных платформах, с использованием средств виртуализации [5] и/или вспомогательных средств тестирования [6];

3) выявить и зафиксировать вычислимые ручные или автоматические проверки функций безопасности, результат завершения которых отрицателен [2];

4) выявить и зафиксировать те из оставшихся ручных или автоматических проверок функций безопасности, которые являются невычислимыми в результате отрицательного завершения проверок [2] из п. 3;

5) провести классификацию ошибок, полученных в результате отрицательного завершения ручных и автоматических проверок из пп. 3 и 4, в соответствии с выбранной шкалой критичности ошибок из п. 1;

6) принять решение о возможности успешного завершения тестирования или необходимости исправления выявленных ошибок с использованием положений теории оптимизации и принятия решений (например, алгоритма простого выбора, основанного на методах Саати, Коггера и Ю) [4] либо на основе других оценок;

7) пп. 1—6 проводить для зафиксированной версии программно-аппаратного СЗИ (программной и аппаратной компонент), в случае внесения любых изменений в СЗИ или исправления выявленных ошибок до окончания процесса верификации начать заново с п. 2.

В отличие от принятого в известных работах и нормативных документах вероятностного подхода к риск-менеджменту и оценке надежности информационных систем (ИС) [7] данный способ предлагает детерминированный подход, наиболее подходящий для целей верификации программных и программно-аппаратных СЗИ. Верификация СЗИ должна проводиться непосредственно до их внедрения в ИС. При этом должна определяться не вероятность отказа ИС или его частота, а целесообразность исправления уже выявленных конкретных ошибок, влияющих на защищенность системы или данных, непосредственно до внедрения средств защиты (они проявятся в ходе работы системы). Такой подход позволяет исключить ухудшение защищенности ИС еще до внедрения в нее СЗИ (т. е. является превентивной мерой обеспечения надежности), а также организовать регрессионное тестирование самих средств защиты.

Средства верификации программно-аппаратных средств защиты информации

На основании предложенного способа верификации программно-аппаратных СЗИ разработано программное обеспечение, предоставляющее возможность автоматического решения задачи оценки критичности выявленных в ходе тестирования ошибок в функциях безопасности, — программа "Верификатор программно-аппаратных СЗИ".

Для применения данной программы ей требуется передать входные данные с результатами тестирования функций безопасности программно-аппаратных СЗИ в определенном формате. При поступлении входных данных программа "Верификатор программно-аппаратных СЗИ" позволяет оценить критичность выявленных в ходе тестирования ошибок и степень их влияния на защищенность ИС. При этом входные данные можно либо задавать вручную (определяя критичность всех выявленных ошибок), либо передавать на вход результаты программ тестирования функций безопасности программно-аппаратных СЗИ [6]. В последнем случае критичность ошибок определяется автоматически и выставляется в результатах программ тестирования в зависимости от проверок функций безопасности, которые завершились с отрицательным результатом.

В качестве объектов оценки могут выступать различные версии одного и того же СЗИ, протестированные во всевозможных ОС и с различными исполнениями аппаратной компоненты. В результате становится возможным сравнить тестируемое программно-аппаратное СЗИ как с предыдущими его версиями для проведения регрессионного тестирования, так и с некоторой абстрактной "эталонной" версией (в которой не выявлено никаких ошибок) для оценки степени влияния найденных ошибок на качество средства защиты.

При этом существуют два режима работы программы "Верификатор программно-аппаратных СЗИ": базовый (оценка СЗИ проводится только на основе выявленных ошибок разного уровня критичности); расширенный (оценка СЗИ проводится также на основе критичности ошибок, но с учетом уровня критичности функций безопасности, в которых они были выявлены).

Базовый режим работы может быть использован для любых программно-аппаратных СЗИ, тогда как расширенный – в отношении тех видов, для которых разработаны программы тестирования в [6] (СКЗИ и подсистемы разграничения доступа различных производителей). Оба режима работы позволяют также ранжировать объекты оценки по степени влияния выявленных ошибок

на качество программно-аппаратного СЗИ. Также в обоих режимах работы программы "Верификатор программно-аппаратных СЗИ", кроме оценки непосредственно критичности выявленных ошибок, может учитываться и их количество. Применение того или иного способа оценки (только по критичности ошибок или по критичности и количеству) зависит от конкретных целей верификации.

Предположим, что производится верификация очередной версии ПСКЗИ ШИПКА [8] (v3.6.0.0) на основе выявленных в работе программ тестирования ошибок. Для сравнения используются результаты тестирования предыдущих версий (v3.3.0.0, v3.4.0.0 и v3.5.0.0) и некая "эталонная" версия СЗИ. При этом в ходе работы программ тестирования для указанных версий СЗИ зафиксировано следующее количество влияющих на защищенность ИС критических ошибок, ошибок в функциях безопасности и некритических ошибок (зададим их в виде вектора, значения элементов которого соответствуют количеству перечисленных видов ошибок): v3.3.0.0 — (1, 1, 1); v3.4.0.0 — (2, 1, 1); v3.5.0.0 — (0, 1, 5); v3.6.0.0 — (0, 0, 10).

В своей работе программа "Верификатор программно-аппаратных СЗИ" использует предложенный математический аппарат теории оптимизации и принятия решения. При этом решается задача оценки критичности выявленных ошибок в функциях безопасности программно-аппаратного средства защиты и ранжирования альтернатив (версий СЗИ) по критериям оценки.

В базовом режиме работы программа "Верификатор программно-аппаратных СЗИ" учитывает только критичность выявленных в ходе тестирования ошибок в функциях безопасности СЗИ и их количество. При этом для работы программы задаются указанные количественные данные о выявленных ошибках (результаты работы программы тестирования), а также определяется шкала критичности, описывающая отношения "важности" (превосходства) одного критерия над другим. В соответствии со шкалой Саати [4] в качестве результата попарного сравнения превосходства ошибок принятых уровней критичности целесообразно задать следующие значения: критические ошибки в сравнении с ошибками в функциях безопасности — 3 (среднее превосходство), критические ошибки в сравнении с незначительными — 9 (абсолютное превосходство), ошибки в функциях безопасности в сравнении с некритическими — 7 (сильное превосходство). По заданным значениям оценки превосходства программа "Верификатор программно-аппаратных СЗИ" автоматически вычисляет числовые значения весовых коэффициентов для критериев оценки. Важно отметить, что шкала критичности должна быть определена только один раз и в дальнейшем может без изменений применяться для решения новых аналогичных задач оценки критичности ошибок программно-аппаратных СЗИ.

На основе входных данных разработанная программа "Верификатор программно-аппаратных СЗИ" позволяет автоматически оценить критичность выявленных в ходе тестирования ошибок в альтернативах. При использовании предложенного математического аппарата также учитываются количественные показатели выявленных ошибок. Пример результатов такой верификации заданных ранее версий ПСКЗИ ШИПКА приведен на рис. 1 (вместо значений весовых коэффициентов для удобства указывается процентное отношение).

Из представленных результатов видно, что ПСКЗИ ШИПКА v3.6.0.0 имеет наиболее близкое к "эталонной" версии значение обобщенного критерия, а также наименьшую из всех остальных версий критичность выявленных ошибок. Это говорит о положительной тенденции при проведении регрессионного тестирования: по сравнению с предыдущими версиями количество ошибок СЗИ выросло, при этом общая их критичность уменьшилась. Таким образом, верификацию ПСКЗИ ШИПКА v3.6.0.0 можно завершить, а выявленные ошибки учесть при разработке уже следующей версии СЗИ.

	Важность	ПСКЗИ ШИПКА v3.3.0.0	ПСКЗИ ШИПКА v3.4.0.0	ПСКЗИ ШИПКА v3.5.0.0	ПСКЗИ ШИПКА v3.6.0.0	Эталонный вариант
Оценка критичности ошибок в функциях безопасности СЗИ	100.0%	17.5%	46.3%	13.7%	12.6%	10.0%
Влияющие на защищенность ИС	65.5%	10.4%	39.3%	5.3%	5.3%	5.3%
Нарушающие работу функций безопасности	29.0%	6.7%	6.7%	6.7%	4.5%	4.5%
Несущественные ошибки	5.5%	0.3%	0.3%	1.7%	2.9%	0.2%

Рис. 1. Результаты верификации ПСКЗИ ШИПКА на основе ошибок, выявленных при работе программ тестирования (базовый режим)

Из результатов верификации, приведенных на рис. 1, видно, что версия ПСКЗИ ШИПКА v3.4.0.0 имеет наибольшую критичность выявленных ошибок (за счет большего количества ошибок, влияющих на защищенность ИС или данных). В соответствии с этим верификация данной версии была прервана в момент проведения, а СЗИ отправлено на доработку. Результатом доработки стала версия v3.5.0.0, в которой исправлены все наиболее критические ошибки, но допущено несколько новых ошибок наименьшего уровня критичности. Однако за счет того, что общая критичность всех выявленных ошибок сократилась (в том числе по сравнению с версией v3.3.0.0), верификация ПСКЗИ ШИПКА v3.5.0.0 была успешно завершена.

Необходимо пояснить, что значения обобщенного и частных критериев (по конкретным видам ошибок) для "эталонной" версии СЗИ на рис. 1 имеют ненулевые значения за счет использованного для решения задачи математического аппарата и принятой в нем шкалы превосходства (даже с учетом нулевых количественных показателей всех критериев оценки). Приведенные значения критериев для "эталонной" версии необходимо трактовать как минимально допустимые значения, которые потенциально не могут быть превышены в любой реальной версии программно-аппаратного СЗИ.

В расширенном режиме работы программа "Верификатор программно-аппаратных СЗИ" позволяет учитывать не только критичность ошибок и их количество, но и критичность функций безопасности, в которых эти ошибки были выявлены в ходе тестирования функций безопасности программно-аппаратного СЗИ. В данном режиме в иерархическую структуру решаемой задачи добавляется новый уровень частных критериев, характеризующих тестируемые функции безопасности СЗИ, для которого критерии с ошибками будут являться уже вложенными локальными критериями (для каждого критерия вышестоящего уровня иерархии). В таких условиях для ПСКЗИ ШИПКА, как и для других СКЗИ, целесообразно рассматривать следующие функции безопасности, которые будут представлять критерии вышестоящего уровня: идентификация и аутентификация для доступа к функциям безопасности; генерация ключей и ключевых пар; шифрование и расшифрование; подпись и проверка подписи; экспорт и импорт открытых ключей; экспорт и импорт закрытых или симметричных ключей.

При этом наличие ошибок в идентификации и аутентификации для доступа к функциям безопасности, а также в экспорте/импорте закрытых и симметричных ключей будет иметь наибольший приоритет, в генерации ключей — приоритет ниже, в шифровании/расшифровании и подписи/проверке подписи — еще ниже, а в экспорте и импорте открытых ключей приоритет минимален. Полученные ранее отношения «важности» (превосходства) для самих видов ошибок (выявленных в соответствующей функции безопасности), а также шкала критичности могут быть использованы без изменений. В соответствии с этим в расширенном режиме работы программы "Верификатор программно-аппаратных СЗИ" для тех же входных данных были получены результаты, представленные на рис. 2.

В данном режиме работы результаты верификации ПКСЗИ ШИПКА иные:

- Общие критичности ошибок версий v3.3.0.0 и v.3.4.0.0 не так сильно отличаются друг от друга за счет того, что в v3.4.0.0 все критические ошибки были выявлены в наименее важной функции безопасности. Возможно, что при таких результатах верификацию этой версии ПСКЗИ ШИПКА можно было бы завершить за счет того, что по сравнению с v3.3.0.0 в ней были исправлены критические ошибки в более значимой функции безопасности;
- Версия v3.5.0.0 имеет меньшую общую критичность ошибок, чем v3.6.0.0, за счет того, что в последней было выявлено множество незначительных ошибок в двух наиболее значимых функциях безопасности. СЗИ v3.6.0.0 в таких обстоятельствах, возможно, было бы целесообразней отправить на доработку.

	Важность	ПСКЗИ ШИПКА v3.3.0.0	ПСКЗИ ШИПКА v3.4.0.0	ПСКЗИ ШИПКА v3.5.0.0	ПСКЗИ ШИПКА v3.6.0.0	Эталонный вариант
Оценка критичности ошибок в функциях безопасности СЗИ	100.0%	20.7%	20.9%	19.6%	20.2%	18.6%
И/А для доступа к ФБ	23.8%	6.2%	4.2%	4.2%	5.0%	4.2%
Генерация ключевой информации	19.0%	3.8%	3.8%	3.8%	3.9%	3.8%
Шифрование и расшифрование	14.3%	2.8%	2.8%	3.2%	2.8%	2.8%
Подпись и проверка подписи	14.3%	2.7%	3.1%	3.2%	2.7%	2.7%
Экспорт и импорт открытых ключей	4.8%	0.5%	2.5%	0.7%	0.5%	0.5%
Экспорт и импорт закрытых ключей	23.8%	4.7%	4.6%	4.6%	5.3%	4.6%

Рис. 2. Результаты верификации ПСКЗИ ШИПКА на основе ошибок и различного уровня критичности функций безопасности, в которых они зафиксированы

Заключение

Предложен адаптированный способ "Верификация программно-аппаратных СЗИ, реализующих функции безопасности, основанная на классификации обнаруженных ошибок, анализе степени их критичности и влияния на защищенность системы или данных". Данный способ рассматривает критичность ошибок не в части нарушения работоспособности объекта тестирования, а с точки зрения возможности нарушения защищенности ИС или данных при некорректной работе функций безопасности. При этом вводится новый уровень иерархии критериев оценки — критичность самих функций безопасности, а критичность каждой ошибки складывается из совокупности уровней критичности самой ошибки и функции безопасности, в которой она обнаружена (с учетом взаимозависимости вычислимости функций безопасности). Данный способ реализует процедуры формальной оценки критичности выявленных в ходе тестирования ошибок в функциях безопасности программно-аппаратных СЗИ и алгоритмы расчета общего значения критичности ошибок, используемые перед внедрением средств защиты в ИС и применяемые для сравнения как их различных версий между собой в рамках проведения регрессионного тестирования, так и с абстрактной "эталонной" версией в целях определения степени влияния выявленных ошибок на защищенность системы или данных и принятия решения о возможности выпуска конкретной версии СЗИ.

Описана разработанная программная реализация предложенного адаптированного способа верификации программно-аппаратных СЗИ — программа "Верификатор программно-аппаратных СЗИ", которая представляет собой систему поддержки принятия решений, позволяющую автоматически оценить критичность выявленных ошибок в функциях безопасности программно-аппаратных СЗИ после использования программ тестирования. Для учета различных целей верификации в разработанной программной реализации предусмотрено два режима работы, результаты которых могут отличаться в зависимости от «важности» функций безопасности, в которых выявлены ошибки разного уровня критичности. За счет зависимости расширенного режима работы от программно-аппаратного СЗИ в части реализуемых функций безопасности и их значимости в рамках верификации его можно применять в отношении средств защиты, для которых ранее разработаны программы тестирования (СКЗИ и подсистемы разграничения доступа различных производителей). Базовый режим работы можно без изменений применять в отношении любых программно-аппаратных, а также программных СЗИ.

Литература

1. *Каннер Т. М., Султанахмедов Х. С.* Особенности верификации средств защиты информации // Вопросы защиты информации. 2014. № 4. С. 55—57.
2. *Каннер Т. М.* Применимость методов тестирования ПО к программно-аппаратным СЗИ // Вопросы защиты информации. 2015. № 1. С. 30—39.
3. *Язов Ю. К., Соловьев С. В., Колесникова Е. В.* Перспективы развития систем поддержки принятия решений в области защиты информации: мат. XXI Межд. конф. "Комплексная защита информации". 17—19 мая 2016 г., — Смоленск. 2016. С. 78—81.
4. *Черноруцкий И. Г.* Методы оптимизации и принятия решений. — Спб.: Лань, 2001. С. 381.
5. *Каннер Т. М.* Особенности применения средств виртуализации при тестировании программно-аппаратных средств защиты информации // Информация и безопасность. 2015. Т. 18. № 3. С. 416—419.
6. *Каннер Т. М.* Эффективность применения средств тестирования программно-аппаратных СЗИ // Вопросы защиты информации. 2017. № 2. С. 9—13.
7. *Дроботун Е. Б.* Критичность ошибок в программном обеспечении и анализ их последствий // Фундаментальные исследования. 2009. № 4. С. 73—74.
8. *Конявская С. В.* Идеальное устройство. ПСКЗИ ШИПКА // Connect! Мир связи. 2008. № 10. С. 111.

Adaptation of existing verification methods for software and hardware DST

T. M. Kanner
Closed Joint Stock Company "OKB SAPR", Moscow, Russia

The article describes the features of verification of software and hardware data security tools (DST). An adapted method for verifying of such DST is proposed also implements procedures for formal assessment of the criticality of errors in the security functions of such protection tools detected during testing and algorithms for calculating the total error criticality used before the integration of a DST into an information system. The software implementation of the proposed adapted verification method is described, which is a decision support system that allows to automatically evaluate the criticality of detected errors in the protection tools.

Keywords: software and hardware DST verification, adapted verification method of software and hardware DST, program "Verifier of software and hardware DST".

Bibliography — 8 references. *Received December 4, 2017*