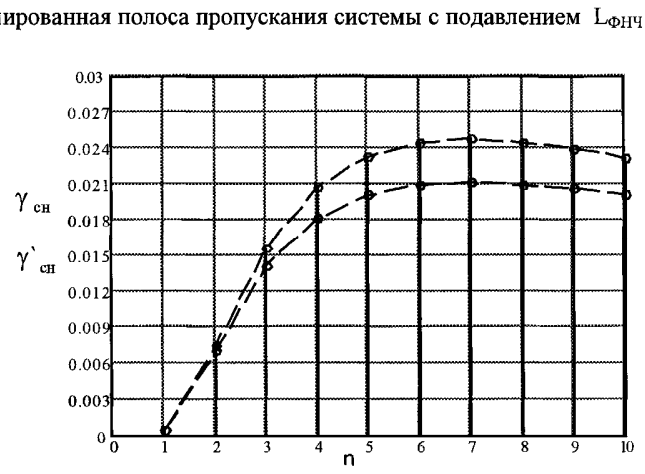

Рис.7. *График разных типов дискриминаторов*

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Шахгильдян В.В., Ляховский А.А.* Система фазовой автоподстройки частоты. – М.: Связь. 1972. –447с.
2. *Малахов А.Н.* Флюктуации в автоколебательных системах. – М.: Наука, 1968. – 660с.
3. *Спилкер Дж.* Цифровая спутниковая связь: Пер. с англ./Под ред. В.В. Маркова. –М.: Связь, 1979. –592с.

УДК 621.039

В.В. Коробкин

ОЦЕНКА БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ УПРАВЛЯЮЩИХ СИСТЕМ МЕХАТРОННЫХ КОМПЛЕКСОВ

Общеизвестно, что для безопасного функционирования мехатронного комплекса, его управляющая система должна содержать элементы защит и блокировок, позволяющих в случае отказа любого компонента комплекса действовать таким определенным образом, чтобы комплекс оставался в безопасном состоянии или переходил в такое состояние. В настоящее время, сложные защиты и блокировки в управляющих системах реализуются с помощью программного обеспечения (ПО).

Таким образом одной из наиболее важных задач является оценка безопасности ПО, позволяющая доказать – безопасность функционирования ПО адекватно и полно отвечает общей безопасности мехатронного комплекса, функционирующего в конкретной среде.

В отличие от технических средств, ПО имеет ряд важных особенностей, определяющих подход оценке его безопасности, а именно:

- в ПО наблюдаются как систематические, так и случайные ошибки;
- систематические ошибки в ПО могут оставаться скрытыми до тех пор, пока не наступит подходящая комбинация обстоятельств в процессе эксплуатации мехатронного комплекса, что может привести к опасному отказу;
- отказ ПО может не быть нейтрализован даже при наличии резервных каналов управляющей системы в случае, если эти каналы имеют идентичные про-

граммные версии. Это обуславливает соблюдение принципа разнообразия (диверсности) при разработке ПО;

- управляющие системы мехатронных комплексов, как правило, являются сложными электронными системами, состоящими из нескольких подсистем, в каждой из которых может использоваться ПО, отличающееся по функциональному назначению. Это обуславливает асимметричность требований к различным компонентам ПО и соблюдение принципа открытости (транспарентности) применяемого системного и разрабатываемого прикладного ПО.

Снизить риски развития аварийных ситуаций, опасных отказов или свести к минимуму их последствия, возможно только при учете следующей сложной совокупности:

- оценки политики безопасности, принятой при разработке мехатронного комплекса в целом [1];

- оценки политики безопасности и проектной оценки (количественный и качественный анализ) безопасного функционирования всех компонентов ПО [2];

- адекватности постулированных эксплуатационных процедур ПО управляющей системы требованиям безопасного функционирования объекта управления мехатронного комплекса [3].

Поэтому, для оценки безопасности ПО должен применяться **комплексный подход.** Под этим подразумевается охват мер всех уровней, направленных на обеспечение информационной (полнота и достоверность данных от сенсорной системы, передача данных по каналам связи, представление информации человеку-оператору, документирование и т.д.) и функциональной безопасности (способности к правильному функционированию) на всех этапах жизненного цикла ПО мехатронного комплекса.

Следовательно, в процессе работы над проектом по разработке ПО необходимо оценить принятые архитектурные, проектные и конфигурационные решения, чтобы убедиться в их достаточности и полноте в плане выполнения функциональных требований, предъявляемых в целом к функциям мехатронного комплекса и вытекающим из них функциям ПО, особенно к тем, которые обеспечивают безопасную работу всего комплекса в течение его жизненного цикла.

Средством оценки служит анализ представленной технической документации, освещающей эти решения. Необходимо проверить, отражает ли проект мехатронного комплекса в целом, а также совокупно архитектура и конфигурация технических и программных средств управляющей системы требования безопасности, которые предъявляются к различным подсистемам и компонентам мехатронного комплекса.

Когда требования безопасности определены и задокументированны, основной задачей этапа базового анализа становится проверка соответствия реализованных в проекте механизмов безопасности этим требованиям.

При этом должны быть выделены и проанализированы следующие аспекты:

- границы и определенность всех подсистем управляющей системы;

- распределение функциональности по этим подсистемам;

- распределение степеней доверия к этим подсистемам;

- определены внутренние и внешние интерфейсы подсистем и их функциональность, обеспечиваемая через определенные интерфейсы;

- определена внутренняя взаимосвязь между подсистемами и проходящие по межсоединениям информационные потоки;

- определены внешние системы, интерфейсы и взаимосвязи с ними;

- определены информационные потоки между внешними системами и управляющей системой мехатронного комплекса;

- определены требуемые функции безопасности и относящиеся к ним уровни соответствия комплексу требований по безопасности объекта управления и управляющей системы;

- определена архитектура программного обеспечения для всех функций: информационной, управления и безопасности;

- совместно разработчиками объекта управления, технических и программных средств управляющей системы рассмотрена, определена и описана архитектура безопасности.

Требования безопасности могут быть сформулированы с различной степенью детализации. В некоторых случаях требования только определяют необходимость наличия некоторого механизма безопасности, например, такого, как аутентификация удаленных пользователей. В других случаях требования могут определять необходимость использования конкретной схемы аутентификации. В обоих ситуациях проверяется наличие соответствующих механизмов безопасности и их адекватность существующим угрозам.

Важным вопросом анализа механизмов безопасности является выбор уровня детализации. В общем случае базовый анализ должен выполняться на двух уровнях – информационном и функциональном.

Функциональный уровень – это уровень абстракции, представленный спецификациями функций ПО. Это относится как к внутренним механизмам безопасности (например, защита программных модулей от внесения в них несанкционированных изменений), так и к внешним (например, физическим и административным мерам защиты программных модулей), хотя последние обычно не определяются в функциональных спецификациях. Для многих программно-технических средств (например, используемые в приводах “инверторы” или “интеллектуальные” датчики) функциональных спецификаций просто не существует либо они бывают неполны. Поэтому для того, чтобы определить функциональные спецификации ПО таких программно-технических средств, приходится изучать принципы их функционирования по имеющейся в наличии эксплуатационной документации.

Информационный уровень включает в себя вопросы обеспечения целостности получаемой и предоставляемой ПО управляющей системы информации, ее конфиденциальности и доступности. Однако при комплексной оценке безопасности ПО [4] нельзя ограничиться только этими характеристиками. Необходимо также учесть такие характеристики как «надёжность», «сопровождаемость», а также ряд смежных субхарактеристик (в составе других характеристик), которые косвенно затрагивают атрибуты безопасности всего комплекса, в котором они функционируют. Для примера в табл. 1 приведены метрики субхарактеристик внешнего и внутреннего качества «защищённость» и качества в использовании «безопасность», в которых будет более точно учтены проблемы безопасности, затрагивающие ПО.

Во многих случаях бывает недостаточно одного базового анализа. Примерами могут служить случаи, при которых:

- во время базового анализа обнаруживаются проблемы, требующие проведения дальнейших исследований;

- ПО имеет высокую степень критичности или основные механизмы безопасности встроены во внутренние функции, которые не видны на функциональном уровне.

Таблица 1

Метрики субхарактеристик

Метрика	Формула	Примечания [5]
1. Внешние метрики безопасности: 1.1 протоколирование доступа	X = A / B; A = число «фактов доступа пользователя к системе и данным», зафиксированных в протоколе системы; B = число «фактов доступа пользователя к системе и данным», которые были произведены во время оценки	1. Рекомендуется использовать «тесты на проникновения» для эмуляции атак на систему 2. Под записью протокола «факт доступа пользователя к системе и данным» может подразумеваться запись «факт обнаружения вируса» для обеспечения антивирусной безопасности системы 3. Метрика носит экспериментальный характер
1.2 контролируемость доступа	X = A / B; A = число обнаруженных видов несанкционированного доступа B = число видов несанкционированного доступа в спецификации	1. Необходимо проверить способность системы определять факты несанкционированного доступа при неправильном применении функций системы 2. Рекомендуется использовать «тесты на проникновения» для эмуляции атак на систему; 3. Метрика носит экспериментальный характер
1.3 предотвращение повреждения данных	a) X = 1 – A / N; A = число фактов существенного повреждения данных N = число видов тестов, при помощи которых пытались спровоцировать факт повреждения данных; b) Y = 1 – B / N B = число фактов незначительного повреждения данных c) X = A / T или B / T; T = время выполнения операции	1. Необходимо проверить корректность работы системы при неправильном применении её функций; 2. Необходимо построить классификацию эффекта от событий повреждения данных 3. Для вычисления внешних метрик следует использовать информацию, доступную извне системы Порядок подсчёта событий здесь отличается от порядка подсчёта событий при оценке аналогичных внутренних данных; 4. Рекомендуется использовать «тесты на проникновения» для эмуляции атак на систему; 5. Метрика носит экспериментальный характер 6. Резервирование данных – один из наиболее эффективных способов предотвращения фактов повреждения данных, однако, резервирование данных относится к метрике «надёжность»

Продолжение табл. 1

Метрика	Формула	Примечания [5]
2. Внутренние метрики безопасности: 2.1 протоколирование доступа	X = A / B A = число типов доступа, которые были зарегистрированы корректно, как определено в спецификации B = число типов доступа, которые должны регистрироваться по спецификации	
2.2 контроль доступа	X = A / B A = число требований контроля доступа, реализованных корректно, в соответствии со спецификацией B = число требований контроля доступа в спецификации	
2.4 предотвращение повреждения данных	X = A / B A = число реализованных механизмов защиты от повреждения данных B = число механизмов, требуемых по спецификации	Необходимо учитывать уровни безопасности при использовании этой метрики
2.5 криптографическая защита данных	X = A / B A = число реализованных механизмов B = число требуемых механизмов по спецификации	Криптографическая защита данных может касаться, например, данных в открытых базах данных или общедоступных данных
3. Метрики безопасности качества в использовании: 3.1 безопасность пользователей и их здоровья	X = 1 – A / B; A = число пользователей, сообщивших о наличии проблем; B = число пользователей;	Проблемы со здоровьем могут включать: травмы от многократно повторяющихся мышечных напряжений, утомление, головная боль и т.д.
3.2 безопасность людей, задействованных в использовании системы	X = 1 – A / B; A = число людей, подверженных риску B = число людей, задействованных в использовании продукта	

Окончание табл. 1

Метрика	Формула	Примечания [5]
3.3 экономический ущерб;	X = 1 – A / B; A = число событий экономического ущерба; B = общее число использования системы;	Также можно учитывать ситуации, где был риск экономического ущерба;
3.4 повреждение прочего ПО;	X = 1 – A / B; A = число событий повреждения прочего ПО; B = общее число использования системы;	Также можно учитывать ситуации, где был риск повреждения прочего ПО; метрика также может быть вычислена как X = суммарная стоимость повреждённого ПО / время использования.

Для этого необходимо проводить детальный анализ, который концентрируется на оценке эффективности реализации механизмов безопасности. ПО исследуется с трех точек зрения:

- оценки правильности функционирования механизмов безопасности;
- оценки эксплуатационных характеристик, таких как надежность, производительность и эффективность;
- оценки устойчивости к попыткам взлома.

При детальном анализе используется множество подходов, выбор которых определяется скорее существующими угрозами и их последствиями, чем общими характеристиками и критичностью ПО.

После проведения анализа, для проверки наличия механизмов безопасности, реализованных в ПО всех подсистем управляющей системы мехатронного комплекса, необходимо проведение тестирования, при котором должно учитываться качество реализации защитных механизмов (например, реализация защит и блокировок). Обычно для проверки наличия механизмов безопасности бывает достаточно проведения тестирования по методу «черного ящика». Такая проверка особенно актуальна для приобретенных программных и программно-технических средств (например, операционных систем, “интеллектуальных” датчиков и т.д.).

При тестировании ПО необходимо особо обращать внимание на следующие вопросы:

- работоспособности механизмов безопасности;
- проверки правильности обработки недопустимых параметров функций;
- обработки исключительных ситуаций;
- мониторинга механизмов безопасности и регистрация событий, связанных с безопасностью;
- проверки правильности функционирования средств администрирования.

Реализованные в ПО механизмы безопасности должны быть адекватно защищены как от ошибок пользователей, так и от внутренних ошибок. Следовательно, при тестировании особое внимание должно уделяться системным интерфейсам, через которые могут распространяться эти ошибки:

- система-человек (сообщения оператору);
- человек-система (команды от оператора, процедуры);
- система-система (внутренние функции системы);
- процесс-система (системные вызовы);

- процесс-процесс (межмодульное взаимодействие ПО).

Здесь может использоваться большинство известных методов тестирования. Тестирование может быть либо внешним (метод «черного ящика»), либо внутренним (тестирование отдельных программных модулей и связей между модулями), в зависимости от типа интерфейса, который подвергается тестированию. Тестирование должно быть выполнено группой экспертов (на этапе верификации), проводящих анализ и разработчиками.

По окончанию тестирования должен быть составлен отчет выполнения требований по безопасности, в котором сопоставляются задокументированные в спецификациях требования и результат выполнения этих требований как отдельным ПО подсистем, так и в целом ПО управляющей системы мехатронного комплекса.

Следует подчеркнуть, что выбор методов оценки ПО не гарантирует сам по себе того, что будет достигнуто полное соответствие комплексу требований по безопасности. Производящие оценку специалисты должны учитывать:

- обоснованность и согласованность выбранных методов, языков и инструментальных средств со всем цикле разработки ПО;

- хорошо ли методы, языки и инструментальные средства подходят для исключения дефектов и ошибок, возникающих при разработке ПО.

После окончания разработки ПО оценка функциональной и информационной безопасности продолжается на этапе эксплуатации всего мехатронного комплекса. На этом этапе необходимо выполнять протоколирование, непрерывное отслеживание работы технических и программных средств, отслеживание процедурных и административных регуляторов безопасности, обеспечивая обратную связь для корректирующих действий после внесения необходимых изменений в ПО. Как показывает опыт эксплуатации мехатронных комплексов, в начальном его периоде ПО управляющих систем подвержено изменениям и модификациям. Составными частями процессов изменения и модификации являются запросы на изменения, сервисные пакеты, любые применимые программные коррекции, а также специализированные требования интероперабельности или совместимости, выдвигаемые при добавлении новых или изменении существующих внутренних и внешних интерфейсов. При этом рассматриваются и анализируются все предлагаемые изменения в ПО, включая изменения политик, правил и процедур. При необходимости выполняется регрессионное тестирование. Если возможно значительное изменение остаточных рисков, то может потребоваться переоценка политики безопасности всего мехатронного комплекса, и, как следствие, ПО его управляющей системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р ИСО/МЭК 15408-2002 “Критерии оценки безопасности информационных технологий”.
2. IEC 61508-6 “Functional safety of electrical/electronic/programmable electronic safety-related systems. Part 6. Guidelines on the application of IEC 61508-2 and IEC 61508-3”.
3. IEC 60880-2:2000 “Software for computers important to safety for nuclear power plants. Software aspects of defense against common cause failures, use of software tools and of pre-developed software”.
4. http://www.securitylab.ru “Оценка характеристик безопасности в рамках процесса оценки качества программных средств в соответствии с международными стандартами ISO/IEC”, 16 сентября 2005.
5. ISO/ IEC 14598 «Software engineering – Product evaluation».