ВЫБОР И ОПРЕДЕЛЕНИЕ ФУНКЦИИ БЕЗОПАСНОСТИ ПРИ ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КРИТИЧЕСКИ ВАЖНЫХ ОБЪЕКТОВ ИНФОРМАТИЗАЦИИ

Сивко Б. В.
Кафедра микропроцессорной техники и информационно-управляющих систем, Белорусский государственный университет транспорта
Гомель, Республика Беларусь
E-mail: {bsivko}@gmail.com

Рассмотрены вопросы формализации и определения функции безопасности при верификации программного обеспечения критически важных объектов информатизации. Приведены способы поиска и выбора функции безопасности на основании технического задания, ограниченности ресурсов, используемой стратегии обеспечения безопасности и общих требований к характеристикам системы. Обоснована возможность изменения функции безопасности для проведения более эффективного доказательства корректности. Введено понятие контрольного списка особенностей системы, используемого для определения функции безопасности.

Введение

В настоящее время разрабатываемые критически важные объекты информатизации (КВОИ) представляют собой микроэлектронные аппаратно-программные комплексы (АПК), а их разработка, верификация и последующая эксплуатация должны удовлетворять принятому в соответствующей отрасли уровню безопасности. Программное обеспечение (ПО) данных АПК является неотъемлемым компонентом, влияющим на безопасность системы в целом, и при этом в настоящее время отсутствуют единые, универсальные и общепризнанные способы доказательства безопасности ПО. В связи с этим практикуется подход, заключающийся в применении комплекса методов и средств по повышению уровня безопасности на всех этапах жизненного цикла системы, а разработка новых способов верификации является актуальной задачей.

Выбор и определение функции безопасности (ФБ) относится к этапу валидации, выполняя формализацию задачи доказательства безопасности и прямым образом влияет на качество последующей верификации.

Описываемые в данной работе подходы основаны на опыте верификации КВОИ систем железнодорожной автоматики и телемеханики (СЖАТ) [1–3], которые подлежат обязательному анализу на безопасность функционирования. Верификация успешно проводилась с привлечением формальных методов [4] в лаборатории «Безопасность и ЭМС технических средств» (БЭМС ТС) Белорусского государственного университета транспорта (БелГУТ).

I. Особенности определения функции безопасности

Согласно стандартам разработки КВОИ [5] верификация данных устройств должна проводиться независимо от коллектива разработчиков. Во время анализа на безопасность независимой организацией необходимо определить доказываемые свойства системы и проверить их на корректность, выполнив полный цикл обратной разработки (reverse engineering). В данном процессе первым шагом является определение ФБ, выполнение которой верифицируется в последующем.

Определение и выбор ФБ имеет ряд особенностей и проблем. Прежде всего независимое определение ФБ используется для последующего анализа корректности предоставленных разработчиками доказательных документов. Исходный код ПО и конкретные схемные решения АПК являются опорной информацией, а их анализ может сформировать функцию поведения, противоречащую спецификациям. Во время анализа ПО на безопасность может оказаться, что принятая ФБ не является необходимой или достаточной, когда заданная ФБ слишком строга и доказательство безопасности провести невозможно, или напротив, слишком слаба, из-за чего уменьшается вероятность нахождения ошибок ПО. При этом обратная разработка является трудоемким процессом и актуальной задачей является создание эффективных методов и средств, позволяющих проводить анализ на безопасность с приемлемыми затратами как времени, так и ресурсов.

II. Предлагаемые подходы

Накопленный автором опыт верификации КВОИ микропроцессорных СЖАТ говорит о том, что определение доказываемой функции безопасности разрабатываемых и существующих АПК необходимо проводить на основании:

– Используемой стратегии обеспечения безопасности – например, во время проектирования может быть использована стратегия

применения логических элементов с несимметричными отказами (h_1-надежные элементы [6]) и система должна придерживаться её во время всего жизненного цикла;
- требований безопасности ко всей системе – например, верифицируемый компонент должен выдерживать заданные временные диапазоны сигналов взаимодействия с другими компонентами системы;
- требований безопасности к рассматриваемому АПК – например, система обязана сохранять инвариант и переходить в безопасное состояние в случае внутренних сбоев.

Во время проведения валидации нельзя гарантировать необходимость и достаточность ФБ, но можно её изменить и повлиять на особенности проводимого доказательства корректности в дальнейшем. Предлагаются следующие способы изменения:

- Усиление ФБ – в этом случае определяется более точное и детерминированное представление о том, как работает система; более строгая функция может быть более простой для доказательства; снижается общая анализируемая сложность системы.
- ослабление ФБ – данная функция может оказаться достаточной для необходимого доказываемого уровня безопасности;
- разбиение ФБ на несколько составляющих – доказательство нескольких простых функций может быть проще, чем одной сложной;
- выбор такой ФБ, которая доказывает отсутствие сразу нескольких опасных факторов.

Опыт верификации КВОИ говорит о том, что ФБ формируется, прежде всего, на основании произошедших аварий и катастроф в прошлом [7]. Описанные выше способы не способны учитывать опыт эксплуатации систем, а существующие практики и рекомендации [5, 8, 9] по большей части не указывают обстоятельства, из-за которых введена рекомендация. В связи с этим для решения проблемы на данном этапе предлагается использование контрольного списка особенностей, представляющего собой набор пар «условие» и «необходимая проверка». В данном случае для верификации новых систем необходимо определить их свойства, и, если свойства выполняют какие-либо условия списка, то требуется провести указанную проверку на безопасность. Ниже пример контрольного списка:

- Работа в режиме 24/7 – доказательство инварианта безопасного поведения системы;
- система реального времени – доказательство временных параметров функционирования;
- использование преобразования типов – доказательство корректности преобразования в такое же или в более безопасное состояние;
- использование динамических объектов – доказательство отсутствия утечек и переполнения объема доступной памяти.

Использование контрольного списка позволяет не только использовать опыт эксплуатации КВОИ, но и улучшить формализацию процесса определения ФБ и сделать первый шаг в создании методики по определению ФБ для конкретной предметной области.

Заключение

Определение ФБ во время проведения верификации является важным этапом анализа на безопасность, а её выбор представляет собой компромисс между имеющимися ресурсами и доказываемыми свойствами. Выполненная в настоящее время в лаборатории «БЭМС ТС» БелГУТа работа по доказательству безопасности ПО микропроцессорных СЖАТ показала правильность принятого подхода определения ФБ для конкретных систем. Опыт верификации и анализ особенностей других предметных областей показывают, что данные способы имеют потенциал для применения для множества АПК, относящихся к КВОИ.

1. Сивко, Б. В. Доказательство корректности блока телеуправления 16-1 диспетчерской централизации «Нёман» / Б. В. Сивко // Вестник БелГУТа: Наука и Транспорт. – 2012. – №1 (24). – С.18–21.
2. Харлап, С. Н. Верификация программного обеспечения микропроцессорной светооптической светодиодной системы / С. Н. Харлап, Б. В. Сивко // Вестник БелГУТа: Наука и Транспорт. – 2012. – №1 (24). – С.22–25.
3. Сивко, Б. В. Доказательство корректности программного обеспечения многопроцессорных устройств связи с объектами железнодорожной автоматики и телемеханики // Вестник БелГУТа: Наука и Транспорт. – 2012. – №2 (25). – С.27-30.
4. Butler, R. W. «What is Formal Methods?» / R. W. Butler // NASA LaRC Formal Methods Program – 2001.
5. David Smith, J. «Safety Critical Systems Handbook. A Straightforward Guide to Functional Safety, IEC 61508 and Related Standards, Including Process IEC 61511 and Machinery IEC 62061 and ISO 13849» / J. David Smith, G. L. S. Kenneth // Elsevier Ltd. – 2010.
6. Сапожников, В. В. Дискретные устройства железнодорожной автоматики и телемеханики. / Ю. А. Кравцов, Вл. В. Сапожников // М. Транспорт – 1988.
7. Leveson, N. G. Software safety in embedded computer systems. / N. G. Leveson // Communications of the ACM, 34(2):34–46 – February, 1991.
8. Gavin, M. Guidelines for The Use of The C Language in Vehicle Based Software / M. Gavin // The Motor Industry Software Reliability Association, – ISBN 978-0-9524156-2-6 – October, 2004.
9. JPL Coding Standard for Flight Software Written in the C Programming Language [Electronic resource] / Jet Propulsion Laboratory California Inst. of Technology – 2009; Mode of access: http://lars-lab.jpl.nasa.gov/jpl_coding_standard_c.pdf Date of access: 21.09.2013.