 DOI: 10.22184/1992-4178.2021.208.7.126.128

Тестирование кибербезопасности встроенных систем с помощью их цифрового двойника

А. Демьянов[1]

УДК 004.9 | ВАК 05.13.19

Компьютерные системы, встроенные непосредственно в объект управления, как правило, имеют ограниченные ресурсы по производительности процессора и объему памяти и работают в режиме реального времени. По этой причине обеспечение встроенных систем серьезными средствами киберзащиты проблематично, если вообще возможно. В связи с ростом числа встроенных систем, подключенных к Интернету, задача обеспечения их кибербезопасности становится все более важной, особенно для промышленных объектов критической инфраструктуры. В статье рассматривается симулятор цифровых электронных систем Wind River Simics и его возможности для обнаружения уязвимостей в ПО встроенных компьютерных систем.

Для чего нужен цифровой двойник компьютерной системы

Цифровой двойник (модель) физического объекта создается для экспериментов над этим объектом, которые в реальности проводить опасно, дорого или просто невозможно. Цифровой двойник компьютерной системы (виртуальная аппаратная платформа) используется для отладки ее программного обеспечения методами, недоступными обычным средствам отладки ПО, например, с помощью останова всех компонентов системы одновременно и выполнения программы реверсивно (в обратном направлении) для поиска первопричины возникшего сбоя.

Реальное ПО на виртуальной аппаратной платформе

Главное необходимое свойство виртуальной аппаратной платформы – способность исполнять реальное ПО в двоичном коде (операционную систему, сетевой стек, BIOS) без какой-либо специальной модификации / перекомпиляции. Реальное ПО даже не подозревает, что оно исполняется не на реальном компьютере. На виртуальной платформе можно также запускать вирус и другое вредоносное ПО и отслеживать его действия без внедрения дополнительных средств трассировки исполнения, которые вредоносным ПО могут быть обнаружены, и оно может «затаиться».

[1] ООО «АВД Системы», avdsys@aha.ru.

Симулятор Wind River Simics – от чипа до системы

Симулятор цифровых электронных систем Simics был создан для того, чтобы вести разработку программного обеспечения встроенной компьютерной платформы, даже если сама платформа еще не существует. Например, операционная система реального времени Wind River VxWorks была заранее портирована на первый многоядерный процессор Freescale PowerPC с использованием Simics-модели этого процессора до появления работающих образцов процессора. Сегодня в библиотеке моделей Simics – десятки процессоров и сотни системных и периферийных устройств, из которых можно построить иерархическую модель системы-на-кристалле, модуля (платы), многоплатного блока и распределенной многоблочной системы.

Полный синхронный останов

В реальной аппаратной платформе невозможно выполнить останов всех компонентов системы одновременно, например, синхронно всех ядер многоядерного процессора. В виртуальной платформе происходит полный одновременный останов всех устройств и даже данных, передающихся в данный момент по шинам и интерфейсам.

Сохранение состояния и воспроизводимость проблемы

Состояние виртуальной платформы в момент останова может быть сохранено в файл (checkpointed) и передано

другим разработчикам для последующего возобновления исполнения ПО с той же точки. При каждом прогоне Simics обеспечивает полную повторяемость результата, поэтому любая проблема может быть воспроизведена. Таким образом, на базе Simics-модели можно организовать киберполигон для совместной работы нескольких экспертов, находящихся в географически различных местах.

Имитация неисправностей (fault injection)

В реальной компьютерной системе сложно имитировать аппаратные ошибки, поэтому обработчики ошибочных ситуаций тестируются недостаточно и потенциально могут стать уязвимостью, используемой для проникновения. В виртуальной платформе аппаратные ошибки имитировать просто, и тестирование обработчиков ошибочных ситуаций может быть выполнено в полном объеме. Механизм fault injection в симуляторе Simics позволяет изменять ячейку памяти, регистр устройства, данные с датчика или содержимое сетевого пакета, имитировать отключение порта или одной из плат многомодульной системы и др.

Реверсивное исполнение – анализируем первопричину, а не последствия

Если атака удалась, то с большой вероятностью злоумышленник ее повторит. Для того, чтобы от повторной атаки нельзя было защититься, нужно стереть следы своих действий. Механизм реверсивного исполнения в симуляторе Simics позволяет отследить действия вредоносного ПО, даже если в результате этих действий злоумышленник не оставил следов. Контрольная точка (checkpoint) для реверсивного исполнения может быть передана другим участникам киберрасследования.

От реагирования к киберохоте

Иногда результат работы вредоносного ПО выглядит как последовательность случайных сбоев, а на самом деле это может быть подготовка к атаке. Если такие подозрения возникают, то самое время перейти от реактивных действий (защищаемся, когда напали) к проактивной защите – киберохоте. Механизм fault injection в симуляторе Simics может быть использован для имитации уязвимостей, провоцирующих вредоносное ПО на активные действия и обнаружение себя. Действия же киберохотника не будут видны ПО, исполняющемуся на виртуальной платформе.

Раннее интеграционное тестирование

Практика непрерывной интеграции CI (Continuous Integration) подразумевает как можно более раннее и частое интеграционное тестирование – сборку отдельных программных компонентов, прошедших модульное тестирование, в единый комплекс и проведение тестов, ориентированных на выявление ошибок на стыке отдельных компонентов. Если для обычного компьютера, который есть на каждом столе, автоматизировать ежедневную сборку и тестирование не проблема, то разрабатываемая встраиваемая система может существовать в одном экземпляре и находится на стенде, доступ на который ограничен. Симулятор Simics позволит обеспечить каждое рабочее место цифровым двойником встраиваемой компьютерной системы для проведения раннего и регулярного интеграционного тестирования, что в конечном итоге значительно повысит качество конечного программного продукта, в том числе и его защищенность.

Библиотека моделей и разработка пользовательских моделей

В библиотеке моделей Simics – десятки процессоров (PowerPC, ARM, x86, MIPS, Sparc/Leon) и сотни системных и периферийных устройств: контроллеры/мосты/коммутаторы PCI и PCIe, контроллеры памяти, прерываний и прямого доступа, контроллеры дисковых интерфейсов SCSI, SATA, IDE, контроллеры Ethernet, USB, I^2C, CAN, устройства специализированных интерфейсов MIL-STD-1553, ARINC 429, Spacewire, часы реального времени и сторожевые таймеры, а также десятки коммерческих плат на базе этих процессоров и устройств

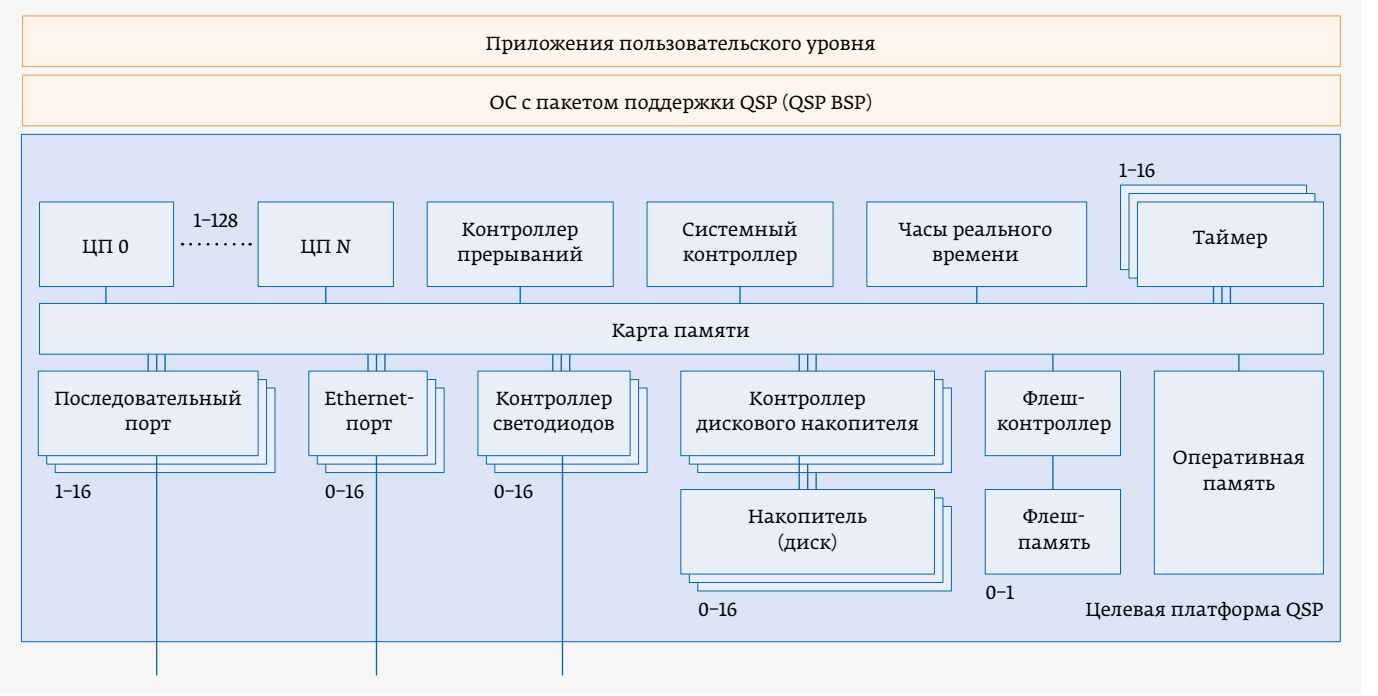

Рис. 1. Виртуальная платформа QSP симулятора Wind River Simics

(reference-дизайны и COTS-модули). В семейство продуктов Simics входит инструмент для разработки новых моделей, с помощью которого разработчик встраивает в модель поддержку механизмов сбора контрольной информации в точке останова, реверсивного исполнения и других механизмов симулятора Simics. Модели в Simics можно разрабатывать на языках C, C++, SystemC и Python и импортировать модели IP-XACT. Библиотечные модели и виртуальные платы являются дополнительными продуктами, но в базовую конфигурацию Simics входят три упрощенные синтетические виртуальные платформы QSP (Quick Start Platform) (рис. 1) на базе архитектур ARM, x86 и PowerPC, которые позволяют сразу начать работу в симуляторе.

ЗАКЛЮЧЕНИЕ

На цифровом двойнике встроенной компьютерной системы, созданном в симуляторе Wind River Simics, возможно:

- запускать реальное вредоносное ПО без нанесения ущерба компьютерной системе;
- отслеживать действия вредоносного ПО, не показывая ему, что оно находится под наблюдением;
- организовывать совместную работу нескольких экспертов по кибербезопасности, даже если они находятся в географически различных местах;
- переходить от реактивной к проактивной защите;
- обеспечивать раннее и регулярное интеграционное тестирование.