*В. КОНЯВСКИЙ*

# КОМПЬЮТЕР С «ВИРУСНЫМ ИММУНИТЕТОМ»

Вместе с огромными преимуществами компьютеры принесли в нашу жизнь и проблемы. Практически все они связаны с возможностью вмешаться в деятельность компьютера со стороны.

Борьба с хакерами стала «головной болью» значительной части населения, а слово «вирус» чаще ассоциируется не с медицинским термином, а с компьютерным. Безуспешность борьбы с хакерами заставляет искать причины не в программном обеспечении, как это принято сегодня, а в чем-то более глубинном.

Любой компьютер - это реализация (более или менее близкая) идеи «машины Тьюринга» [1].

Понятия «машина Тьюринга» и «алгоритм», вычислимость неразрывно связаны, определяются одно через другое. Само существование абстрактного «исполнителя», такого как машина Тьюринга, вселяет уверенность во всемогущество человека. Действительно, любая (точнее, рекурсивная, что и есть практически любая) задача может быть решена, если достаточно ресурсов (памяти и времени).

Универсальные машины (УМ), как минимум, должны выполнять элементарные операции, свойственные машине Тьюринга, а именно: перемещать управляющее устройство (головка чтения-записи) влево и вправо по ленте, читать и записывать в ячейки символы некоторого конечного алфавита.

Неотчуждаемая возможность «читать и записывать», которая делает операцию копирования в компьютере имманентной, полностью противоречит, как минимум, задачам защиты информации. Необходимое для «универсальности» свойство становится неприемлемым в конкретных условиях.

Если УМ выполняет любые программы, то, очевидно, она выполнит и вредоносную программу. Это не зависит от ее программного обеспечения, а определяется ее архитектурой. Универсальность компьютера обеспечивается архитектурно самой «конструкцией» машины Тьюринга, как мыслимой в абстракции, так и реализованной на практике. Способность выполнять вредоносные программы - это базовая, системная, архитектурная уязвимость всех компьютеров, построенных как машина Тьюринга. **Уязвимость - оборотная сторона универсальности. Машина Тьюринга архитектурно уязвима.** Архитектурно уязвимы и все виды компьютеров, которые мы используем, потому что они разрабатывались так, чтобы быть максимально универсальными. Этой уязвимостью мы платим за универсальность наших компьютеров. Мы эксплуатируем компьютеры, а хакеры эксплуатируют эту уязвимость.

Поскольку архитектуру нельзя изменить программным путем, то никакие программные средства не помогут нам защититься от хакеров надежно. Игра «кто кого» продолжается уже много лет, давая работу сотням тысяч специалистов по информационной безопасности, но не спасая нас от потерь.

Как же быть?

Если уязвимость в архитектуре, то и совершенствовать нужно архитектуру.

Классическими являются две архитектуры - архитектура фон Неймана [2] и Гарвардская архитектура [3]. Примером первой являются практически все настольные компьютеры, примером второй - практически все планшетные компьютеры и телефоны.

При разработке компьютера главное - понять, какая часть функций должна быть реализована аппаратно, а какая - программно. Правильный выбор этого соотношения позволил ПЭВМ с архитектурой фон Неймана многие годы занимать лидирующее положение. Однако сейчас практически достигнут предел эффективности данного технического решения. В последние годы техника совершила огромный скачок вперед. Наиболее заметное здесь изменение - опережающий рост решений на базе Гарвардской архитектуры. Если несколько лет назад объемы продаваемых процессоров типов x86 и ARM соотносились как 80:20, то уже сегодня 50:50, и эта тенденция усиливается[1]. Сложились условия для размышлений над усовершенствованием архитектуры.

В аппаратную часть нужно включать то, что снижает стоимость, редко изменяется, расширяет возможности и используется постоянно. Нужно также устранить имеющиеся архитектурные уязвимости, тем более что сейчас совсем не выглядит экзотической мысль о том, что в процессе работы структура компьютера может динамически изменяться. Или, например, структура вначале может быть такой, как конечный авто-

***КОНЯВСКИЙ Валерий Аркадьевич*** *- доктор технических наук, профессор НИУ ВШЭ и НИЯУ МИФИ, зав. кафедрой «Защита информации» МФТИ, научный руководитель ФГУП ВНИИПВТИ, научный консультант ОКБ САПР.*
*Адрес: 115114, г. Москва, 2-й Кожевнический пер., 8*
*e-mail: 001@pvti.ru*

[1] *Оценка дана акад. Б.А. Бабаяном (INTEL) в беседе с автором, июнь 2015 г.*

мат, а потом, на следующем этапе, стать «универсальным исполнителем» по Тьюрингу.

Отличительной особенностью архитектуры фон Неймана является то, что команды и данные не разделяются, они передаются по единому общему каналу **(рис. 1).**

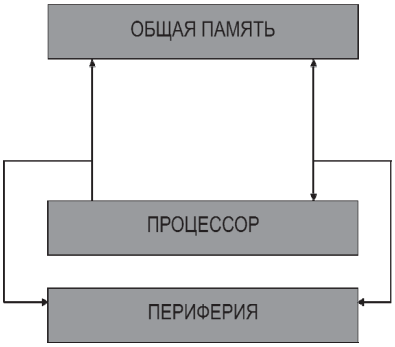


***Рис. 1.*** *Архитектура фон Неймана*

Гарвардская архитектура предполагает наличие разных каналов для команд и данных **(рис. 2)**.

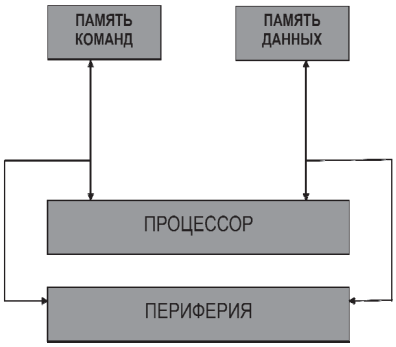


***Рис. 2.***
*Гарвардская архитектура*

Такая схема взаимодействия требует более сложной организации процессора, но обеспечивает более высокое быстродействие, так как потоки команд и данных становятся не последовательными, а параллельными, независимыми.

Однако и в случае компьютера фон Неймановского типа, и компьютера с Гарвардской архитектурой организация потоков команд и данных такова, что архитектурная уязвимость присуща каждому из них. Гибкость, универсальность и в одном, и в другом случае обеспечивается возможностью изменения последовательности команд и данных (двунаправленные стрелки от процессора к памяти) - независимо от того, в одной памяти они лежат или разделены. В свою очередь, возможность изменения последовательности команд и данных создает и возможность для несанкционированного вмешательства вредоносного программного обеспечения - это и есть основная архитектурная уязвимость.

На использовании этой уязвимости основаны практически все современные хакерские атаки, которые, в основном, сводятся к атаке на «перехват управления». Схема атаки обычно выглядит так:

s1) внедряется и размещается в оперативной памяти вредоносное ПО (ВрПО);

s2) внедряется и размещается в оперативной памяти вредоносный обработчик прерываний;

s3) записывается в долговременную память ВрПО и обработчик прерываний;

s4) с помощью любого доступного механизма вызывается прерывание, например, с помощью DDOS-атаки;

s5) внедренный ранее обработчик прерываний срабатывает и передает управление ВрПО;

s6) ВрПО выполняет свою функцию, например, реализует разрушающее программное воздействие (РПВ).

Здесь s1 - s3 - это шаги по подготовке атаки, s4 - инициирование атаки, s5 и s6 - собственно использование архитектурной уязвимости.

Для того чтобы обезвредить шаги s1 и s2, обычно используются антивирусные программы. Иногда это бывает полезным, но только иногда. Невозможно с помощью антивирусных программ выявить все ВрПО. Более того, специалистам известны конструкции ВрПО, которые нельзя обнаружить. Можно даже сказать, что компьютерные вирусы и в целом ВрПО удается обнаружить только в силу их несовершенства. В общем случае всегда можно разработать такое ВрПО, которое не может быть обнаружено с помощью антивирусных программ сигнатурного поиска, эвристических анализаторов и поведенческих блокираторов.

Блокирование последствий выполнения шага s3 производится при последующей загрузке с помощью механизмов контроля целостности - по сути, ревизоров, определяющих, есть ли изменения в составе данных. Иногда эта проверка выполняется с помощью тех же наборов антивирусных программ, но это слабое решение, так как проверка должна выполняться до загрузки ОС, а программы, в том числе и антивирусные, работают под управлением ОС.

Генерация события на шаге s4 частично блокируется с помощью специальных средств анализа трафика, устанавливаемых как в сети, так и на клиентских компьютерах. Важно то, что пока нет средств, позволяющих гарантированно блокировать эту уязвимость.

Негативные последствия шагов s5 и s6 блокируются с помощью механизмов контроля запуска задач (процессов, потоков). Это очень эффективные механизмы, но реализующие их средства довольно дорогие, и для их настройки нужно быть специалистом в компьютерных технологиях и информационной безопасности.

Поскольку некоторые из перечисленных функций безопасности должны выполняться до загрузки операционной системы, то их нельзя реализовать программно, а можно только с помощью сложного устройства.

Сложность его связана именно с фон Неймановской архитектурой защищаемого компьютера - нужно добавить неизменяемую память, разделить потоки команд и данных, исполнить контрольные процедуры в доверенной среде до запуска ОС и многое другое [12].

Однако в компьютерах, использующих Гарвардскую архитектуру, потоки команд и данных уже разделены. Если это так, то нельзя ли это обстоятельство использовать для упрощения и удешевления защитных механизмов?

Нужно только сделать память неизменяемой (тогда нет необходимости использовать сложные механизмы контроля целостности программ и данных до старта ОС), а контрольные процедуры в этом случае можно исполнять под управлением проверенной и неизменяемой ОС.

Эти функции легко реализовать, если обеспечить движение команд и данных только в одном направлении - из памяти в процессор **(рис. 3).**

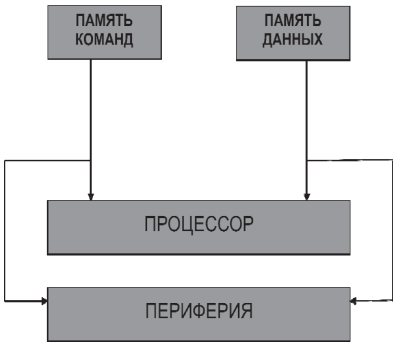


***Рис. 3.*** *Гарвардская архитектура с памятью RO*

Очевидно, что такая архитектура обеспечит неизменность ОС, программ и данных.

Если вернуться при этом к схеме атаки, описанной выше, то видно, что шаг s3 не может быть выполнен, поэтому и сама атака (шаги s5 и s6) тоже не исполнится. Такой компьютер приобретет значительный «вирусный иммунитет», так как вредоносное ПО не будет фиксироваться на компьютере.

Недостатком при этом будет то, что придется дорабатывать практически все программное обеспечение, так как разработчики существующего ПО не ограничивают себя в использовании операций записи в память. Для работы практически всех программ необходима возможность записи.

Для того чтобы можно было использовать без доработок все ранее разработанное ПО, необходимо предложенную архитектуру дополнить блоками сеансовой памяти, в которой и будут исполняться программы **(рис. 4)**.

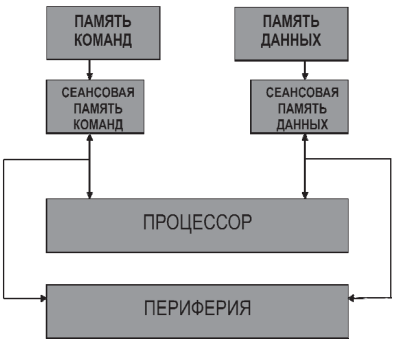


***Рис. 4.*** *Гарвардская архитектура с сеансовой памятью*

Таким образом, архитектура компьютера будет отличаться на разных этапах - сначала она такая, как на рисунке 3, а потом такая, как на рисунке 4.

Фактически архитектура изменяется от этапа начальной загрузки к этапу функционирования. Совмещая рисунки, получаем изменяемую архитектуру Гарвардского типа **(рис. 5)**.

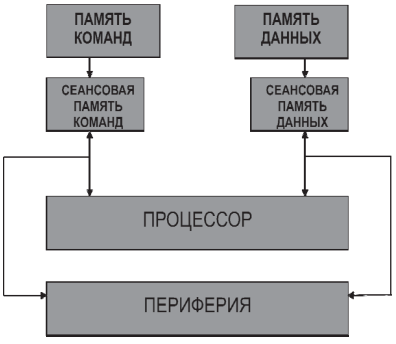


***Рис. 5.*** *Новая Гарвардская архитектура*

Предложенная нами архитектура получила название «новая Гарвардская» архитектура. Она отличается тем, что в ней используется память, для которой установлен режим «только чтение». При загрузке команды и данные размещаются в сеансовой памяти, в которой и исполняются. Начальная загрузка и копирование кодов в сеансовую память могут выполняться как последовательно, так и параллельно - суть разделения этапов от этого не меняется **(рис. 6)**.

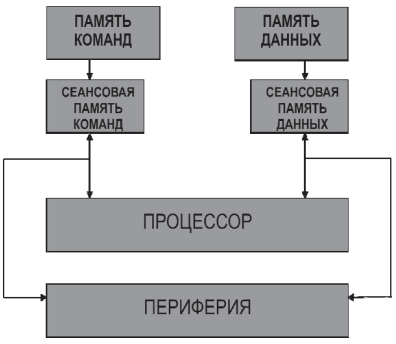


***Рис. 6.*** *Новая Гарвардская архитектура с общей сеансовой памятью*

Конечно, эта схема описана условно, и в реальных компьютерах все немного сложнее.

Однако можно уверенно сказать, что владельцы таких компьютеров чувствуют себя намного более защищенными от атак хакеров.

Новая архитектура характеризуется динамической изменяемостью, что обеспечивает защищенность и эффективность, неизменность операционной системы, «вирусный иммунитет». Она не мешает возможности применения адаптированных стандартных ОС и всего программного обеспечения, написанного для них.

Основных преимуществ у новых компьютеров два - высокий уровень «вирусного иммунитета» и возможность создания и поддержки доверенной среды и использования в ней всего ранее наработанного программного обеспечения.

Важно то, что на основе описанной архитектуры можно создавать компьютеры для всех видов информационного взаимодействия, при которых дове-

ренность и защищенность взаимодействия важна - от дистанционного банковского обслуживания (ДБО) [5] и защищенных «облаков» [6,7] до «интернета вещей». Сейчас на основе описанной архитектуры разработаны и серийно выпускаются 7 типов компьютеров - МКТ, МКТ+, МКTrusT, МКcard, МКcard-long, AQ-МК, TrusTPAD. Их особенности описаны в [8-14], а сами компьютеры - в [15]. Разработка новых видов компьютеров продолжается.

***Литература:***

*1. Машина Тьюринга [Электронный ресурс]. - URL: https://ru.wikipedia.org/wiki/Машина Тьюринга.*

*2. Архитектура фон Неймана [Электронный ресурс]. - URL: https://ru.wikipedia.org/wiki/Архитектура фон Неймана.*

*3. Гарвардская архитектура [Электронный ресурс]. - URL: https://ru.wikipedia.org/wiki/Гарвардская архитектура.*

*4. Конявский В.А. Управление защитой информации на базе СЗИ НСД «Аккорд». - М.: Радио и связь, 1999. - 325 с., ил.*

*5. Конявский В.А. Защищенный микрокомпьютер МК-TRUST - новое решение для ДБО // Национальный банковский журнал. - 2014. - № 3. - С. 105.*

*6. Конявский В.А., Акаткин Ю.М. Мы не доверяем облаку или облако нам? // Information Security / Информационная безопасность. - 2014. - № 1. - С. 28-29.*

*7. Акаткин Ю.М., Конявский В.А. Безопасный доступ к корпоративным облачным приложениям. Information Security / Информационная безопасность. - 2014. - № 1. - С. 28-29.*

*8. Конявский В.А., Степанов В.Б. Компьютер типа «тонкий клиент» с аппаратной защитой данных: Патент на полезную модель № 118773. 27.07.12. Бюл. № 21.*

*9. Конявский В.А. Компьютер с аппаратной защитой данных от несанкционированного изменения: Патент на полезную модель № 137626. 20.02.2014. Бюл. № 5.*

*10. Конявский В.А. Мобильный компьютер с аппаратной защитой доверенной операционной системы: Патент на полезную модель № 138562. 20.03.2014. Бюл. № 8.*

*11. Конявский В.А. Мобильный компьютер с аппаратной защитой доверенной операционной системы от несанкционированных изменений: Патент на полезную модель № 139532. 20.04.2014. Бюл. № 11.*

*12. Конявский В.А. Мобильный компьютер с аппаратной защитой доверенной операционной системы: Патент на полезную модель № 147527. 10.11.2014. Бюл. № 31.*

*13. Конявский В.А., Акаткин Ю.М. Мобильный компьютер с аппаратной защитой доверенной операционной системы от несанкционированных изменений: Патент на полезную модель № 151264. 27.03.2015. Бюл. № 9.*

*14. Конявский В.А. Рабочая станция с аппаратной защитой данных для компьютерных сетей с клиент-серверной или терминальной архитектурой: Патент на полезную модель № 153044. 27.06.2015. Бюл. № 18.*

*15. Trusted Cloud Computers [Электронный ресурс].-URL: http://www.trustedcloudcomputers.ru.*

## НАША ИНФОРМАЦИЯ

# II Международный библиографический конгресс «Библиография: взгляд в будущее»

**Российская ассоциация электронных библиотек 6 октября 2015 года провела Интернет-трансляцию II Международного библиографического конгресса под девизом «Библиография: взгляд в будуще», который состоялся 6-8 октября 2015 года в РГБ.**

Организаторы мероприятия: РГБ, РНБ, Президентская библиотека им. Б.Н. Ельцина, Российская книжная палата, Российская библиотечная ассоциация, Библиотечная ассамблея Евразии.

В рамках конгресса состоялись пленарные и секционные заседания по следующим направлениям: «Общетеоретические и футурологические проблемы библиографии», «Библиографическая запись как основа формирования библиографических ресурсов», «Универсальные библиографические ресурсы», «Информационно-библиографическое обеспечение науки, техники, образования и культуры» и т.д.

Со вступительным словом к участникам конгресса обратился А.И. Вислый - генеральный директор Российской государственной библиотеки, президент Некоммерческого партнерства «Библиотечная ассамблея Евразии». С докладами выступили: В.П. Леонов - директор Библиотеки Российской академии наук в г. Санкт-Петербурге, М.Д. Афанасьев - директор Государственной публичной исторической библиотеки России, Б.Р. Логинов - генеральный директор Национального информационно-библиотечного центра ЛИБНЕТ и др.

В работе конгресса приняли участие около 400 человек из 10 стран.

**По материалам сайта: *http://aselibrary.ru***