ИПМ им.М.В.Келдыша РАН

Будущее-2024 • Труды конференции

А.П.Духвалов

Кибериммунитет = ключ к безопасному цифровому миру будущего

Рекомендуемая форма библиографической ссылки

Духвалов А.П. Кибериммунитет = ключ к безопасному цифровому миру будущего // Проектирование будущего. Проблемы цифровой реальности: труды 7-й Международной конференции (15-17 февраля 2024 г., Москва). — М.: ИПМ им. М.В.Келдыша, 2024. — С. 154-162. — https://keldysh.ru/future/2024/3-1.pdf https://doi.org/10.20948/future-2024-3-1

Размещено также видео выступления

Кибериммунитет = ключ к безопасному цифровому миру будущего

А.П. Духвалов

АО «Лаборатория Касперского»

Аннотация. Сложнейшие и сильно связанные между собой информационные системы в огромной степени определяют качество жизни. Банковские системы, промышленность, транспорт и другие ключевые аспектов современной цивилизации сильно зависят от надежности информационных систем. Однако, разработка информационных систем значительно отстает от требований по доверию и безопасности. Для реализации этих требований сейчас применяются большое число т.н. средств «наложенной» информационной безопасности: антивирусы, межсетевые экраны, анализаторы траффика и т.д. Однако применение их приводит к дополнительным проблемам: увеличение сложности, повышенные требования к производительности, энергопотреблению и другие. Предложенный подход позволяет разрабатывать информационные систем таким образом, чтобы они обеспечивали свою устойчивую работу в «агрессивной» информационной среде по факту свой разработки, т.е. в процессе эксплуатации не требовали дополнительных, наложенных средств защиты. По аналогии с иммунитетом живых организмов такое свойство информационных систем принято именовать кибериммунитетом.

Ключевые слова: информационные системы, информационная безопасность, наложенные средства безопасности, разработка информационных систем, конструктивная информационная безопасность, secure-by-design, кибериммунитет

Cybernetic immunity = the key to a secure digital world of the future

A.P. Dukhvalov

Kaspersky Lab

Abstract. Very complicated and highly interconnected information systems greatly determine the quality of life. Banking systems, industry, transportation, and other key aspects of modern civilization depend heavily on the reliability of information systems. However, the development of information systems lags far behind the requirements for trust and security. To implement these require-

ments, a large number of so-called "imposed" information security measures are now used: antiviruses, firewalls, traffic analyzers, etc. However, their application leads to additional problems: increased complexity, increased performance requirements, energy consumption, and others. The proposed approach makes it possible to develop information systems in such a way that they would ensure their stable operation in an "aggressive" information environment upon their development. I.e., they do not require additional, imposed means of protection during operation. By analogy with the immunity of living organisms, such a property of information systems is commonly referred to as cyberimmunity.

Keywords: information systems, information security, imposed security measures, information system development, constructive information security, secure-by-design, cyberimmunity

«Лучший способ предсказать будущее – это создать его!»

Питер Ф. Друкер

В современном развитом мире большинство возможностей доступно через посредство информационных технологий. Компьютеры проникают везде – госуслуги, личный и общественный транспорт, умный дом, библиотеки, банковские услуги, управление предприятием, бухгалтерия и т.д. Все это управляется компьютерами и доступ к этим и к абсолютному большинству других услуг и возможностей осуществляется через информационные технологии. Доступ к основному источнику знаний, интернету, без современных вычислительных устройств вообще не возможен. Практически у каждого есть смартфон, иногда и не один – весьма себе мощный компьютер. В 2023 г. для 95% процентов населения мира доступна мобильная связь. Потребление мобильного трафика к 2023 г. достигнет 110 ЕБ (эксабайт = 10^{18} байт) в месяц.

Однако цифровизация не исчерпывается этими девайсами. Сервера – центры обработки данных – обрабатывают эксабайты данных, и объем обрабатываемой информации постоянно растет.

Создатель архитектуры современных компьютеров и компьютеризированных устройств Джон фон Нейман при разработке принципов этой архитектуры в свое время представлял себе несколько огромных машин, которые удовлетворили бы мировую потребность в высокоскоростных расчетах. Однако цифровизация пошла по другому сценарию – сегодня в мире гигантское количество компьютеров построено в соответствии с фоннеймановской архитектурой. Огромное количество встроенных компьютеров используется для управления разнообразной техникой, начиная от холодильников и пылесосов и заканчивая производственными технологическими линиями, газопроводами и трансконтинентальными морскими лайнерами. Принципы, разработанные ученым, и среди них – использование общей памяти для хранения данных и программ – вероятно, привели нас на этот путь.

К примеру, в современном легковом автомобиле от 30 до 100 взаимодействующих контроллеров, управляющих всеми функциями и безопасностью в том числе. Руль и педали больше не связаны с колесами и тормозами с помощью механической связи. По данным портала iot.ru в 2023 г. количество подключенных к интернету вещей (IoT) устройств в мире достигнет 30 млрд.

Всё это не ради забавы, хотя и игрушки вносят свой значимый вклад в информационную среду. В основном цифровизация проводится ради повышения эффективности и качества, экономии энергии, совершенствования обработки информации, получения новых знаний, создания новых материалов, повышения качества управления и т.д. – все для благих целей.

Однако человечество давно усвоило одну важную мудрость: любые новые технологии приносят с собой не только положительные изменения жизни, но и проблемы. Вероятно, именно фоннеймановская архитектура (в разработке которой, нужно отметить, принимали участие и другие ученые) привела человечество на этот путь. Но она же, а именно возможность обращаться с кодом программы как с данными привела к созданию первых вирусов – сперва ради забавы, а затем превратила сферу хакерства из деятельности «во имя искусства» в огромную криминальную индустрию.

Кража и подтасовка информации, обвалы сайтов, воровство денег, социальные проблемы – только верхушка айсберга. Нарушение работы опасных производств, ущерб окружающей среде и здоровью людей, катастрофы государственного масштаба – это то, к чему ведет цифровизация в ее сегодняшнем развитии, если мы не будем оглядываться на проблемы информационной безопасности.

Луддиты ли мы? Нет, мы просто констатируем масштабы проблемы: Лаборатория Касперского в 1998 г. регистрировала 50 новых уникальных угроз в день, в 2008 г. их было уже 15 тыс. в день, в прошлом году более 400 тыс. новых угроз в день. Отметим, что количество угроз растет экспоненциально. А как человечество с ними сейчас борется? Умные люди придумывают десятки технологий для защиты. Перечислю некоторые:

– файловый антивирус / signatures / SRC-checker / … – application behavior blocker – system behavior blocker – whitelisting	– anonymizer – device control – script-checker – URL-filtering – statistic analyzer – IM-antivirus – secure payment – web-antivirus	– anti-DDOS – anti-SPAM – anti-fishing – parental control – update control – firewall – sand-boxing – …

Здесь далеко не все. На самом деле технологий защиты гораздо больше. И что, позволяют ли эти и все остальные защитные технологии с уверенностью утверждать, что проблем с информационными системами

становится все меньше и меньше? Вовсе наоборот – цифры показывают, что проблем становится все больше и больше.

В таблице перечислены технологии, которые включены в понятие, которое мы привычно называем термином «антивирус». Хотя, конечно, это давно уже вышло за рамки первоначального значения этого слова. Все они предназначены для защиты «традиционных» информационных систем: офисных сетей, серверов, рабочих или домашних компьютеров, смартфонов, планшетов и т.д. Конечно, существуют и более сложные и комплексные системы. Например, SIEM – Система управления безопасностью большой сетью крупного предприятия. EDR, XDR, MDR, SOAR, etc. – различные подходы и системы, позволяющие управлять безопасностью сложных информационных систем.

В целом неудачная с точки зрения безопасности архитектура современных компьютерных систем усугубляется чрезвычайной сложностью программного кода – того самого, который перемешан с данными. К примеру, проект ядра (только ядра) ОС Linux содержит десятки миллионов строк кода. И это не самый большой проект. Программное обеспечение, как и компьютерное «железо», состоит из сотен и тысяч компонентов, взаимодействующих между собой по огромному количеству протоколов и интерфейсов. В таком обилии существует ненулевая вероятность найти непредусмотренные разработчиками сценарии использования или наборы данных, обработка которых приводит к не предусмотренным на этапе разработки результатам. Чем и пользуются хакеры, и прочая околокомпьютерная нечисть.

«Традиционные» информационные технологии не исчерпывают современный ландшафт. Необходимо думать о защите вновь появляющихся технологий. Распределенный реестр (block-chain), машинное обучение, искусственный интеллект, IoT, умные здания и города, беспилотный транспорт, системы управления технологическими процессами и т.д. У всех этих и многих других новшеств вместе с их преимуществами существуют и уязвимости к угрозам. Но для большинства из них или нельзя применять привычные методы защиты, или они становятся неэффективны.

– На производственную линию нельзя поставить антивирус, потому что он может заблокировать технологический процесс, что может привести к катастрофе.

– На ультразвуковой датчик, измеряющий расстояние, который по факту является микрокомпьютером антивирус поставить нельзя, потому что не существует антивируса, способного работать на ультразвуковом датчике.

– Систему искусственного интеллекта нельзя защитить антивирусом, потому что угрозы для нее существенно отличаются от тех, на которые рассчитан антивирус. Например, от отравления данных антивирус не спасет.

Попробуем проанализировать, откуда проблемы и почему сложилась такая ситуация. Фоннеймановская архитектура отличается, как мы уже сказали, смешением программного кода и данных. Нарушение безопасности связано с вмешательством в процесс работы системы путем манипулирования данными (возможно, с дальнейшей интерпретацией этих данных как кода). Большинство современных технологий защиты, включая те, что мы привычно называем «антивирусом», реактивны – они реагируют на факт реализации нарушителем той или иной техники вмешательства в работу программного кода. Это требует дополнительных ресурсов для реализации логики контроля операций на уровне программного кода, и, кроме того, сама логика может содержать неизбежные ошибки и упущения. Эти ошибки могут быть связаны как с появлением новых техник, применяемых нарушителем, так и со сложной логикой поведения системы, для которой не существует универсального алгоритма контроля. Каждое приложение выполняет функцию, отличную от других, каждый технологический процесс отличается от другого, и что уж говорить про системы искусственного интеллекта. Ограничения безопасности для систем связаны с функцией, которую они выполняют. Как правило, корректное и безопасное функционирование системы можно описать при помощи нескольких основных условий или инвариантов. Например, определенная информация находится на отдельном накопителе или в контейнере и остается недоступной для модификации, или параметры техпроцесса в любой момент времени удовлетворяют установленным ограничениям. Такое инварианты можно назвать целями безопасности для системы.

По существующей практике, в современном процессе разработки информационных систем основной упор делается на сценарии работы, которые позволят добиться главной, основной цели для которой разрабатывается софт, железо или любая другая информационная система.

Например, если разрабатывается бухгалтерская система, то для нее проектируются сценарии создания и корректирования плана счетов, совершения бухгалтерских проводок, справочников контрагентов и всех других важных для этой области действий. В системе прописываются характерные роли с соответствующими правами: бухгалтер, фин.директор, администратор, менеджер, продавец и пр. По сценариям разрабатывается техническое задание (ТЗ), проектируется архитектура, пишется софт, системы верификации, тестирования и отладки. Все это для того, чтобы все предусмотренные роли в соответствии с планируемыми сценариями смогли эффективно выполнять производственные задачи.

При таком подходе основные сценарии использования системы проходят все стадии жизненного цикла: анализ, ТЗ, архитектура, разработка, тестирование, верификация достижения требований. В этом и заключается основная цель разработчиков – добиться того, чтобы означенные сценарии в готовой системе работали «как часы».

Если вопросы информационной безопасности откладываются на этап внедрения и эксплуатации, то разработанный софт замечательно выполняет функции, ради которых он был разработан, но становится легко уязвимым для хакеров, которые легко находят не предусмотренные разработчиками сценарии работы.

В этом и состоит одна из основных проблем современного подхода к разработке информационных систем. Она особенно сильно проявляется для т.н. низкоуровневых систем, типа операционных систем, СУБД, сервисов, где количество сценариев и взаимодействующих компонентов очень велико и сами системы очень сложные и многосвязные.

Обычно для повышения устойчивости информационных систем применяются упомянутые ранее технологии защиты. Они все характеризуются тем, что применяются к готовым изделиям, т.е. после разработки. По общему свойству их всех можно охарактеризовать как наложенные средства безопасности. Большинство продуктов в этой нише разработаны очень профессионально и имеют отличные эксплуатационные качества. И они действительно хорошо защищают от различных информационных угроз. Однако, судя по статистике роста информационных проблем, таких технологий недостаточно.

Рассмотрим некоторые отрицательные стороны применения наложенных средств. Во-первых, любые дополнительные средства требуют дополнительных вычислительных ресурсов. Т.е. при разработке необходимо выбирать процессор, дисковую подсистему, сетевые устройства так, чтобы они учитывали нагрузку не только от полезной функциональности, но и от наложенных средств безопасности. На ранних стадиях развития индустрии, когда такие дополнительные нагрузки не умели правильно рассчитывать, родилось мнение, что «антивирус тормозит». Дополнительные вычислительные мощности требуют дополнительных затрат на электроэнергию. Информационные системы становятся дороже. Это тоже в некоторых случаях является препятствием для применения наложенных средств.

Поскольку наложенные средства в абсолютном большинстве разрабатываются в отрыве от функциональности основной системы, то и требования, и особенности эксплуатации этих систем часто бывает трудно согласовать. Это тоже может являться фактором, препятствующим использованию надежных средств защиты. Особенно ярко это проявляется в промышленных информационных системах, когда блокировка негативной активности может привести к неработоспособности основной функциональности – контролю технологического процесса, что недопустимо. Эксплуатанты стоят перед трудным выбором: или оставить систему беззащитной или принять риск того, что в какой-то момент технологический процесс может остаться без контроля. Понятно, что в большинстве случаев выбирают первое.

И самая главная проблема с наложенными средствами состоит в том, что результирующая информационная система становится сложнее, значит оказывается менее надежна и более уязвима.

Индустрия информационных технологий не первая сталкивается с проблемами сложности и надежности. Авионика, атомная энергетика, автомобилестроение, химическая промышленность также имеют дело с ними. Можно приводить и другие примеры. Однако IT-индустрия – первая, где с вопросами надежности и безопасности столь массово сталкивается большинство населения и дома, и на работе.

В целом методы борьбы за устойчивость, применяемые в других индустриях, часто применимы и в IT. Конечно, с поправками на специфику. Да и в самой IT-индустрии накоплен богатый опыт. Можно сформулировать ряд принципов, которые в целом известны и приняты в разных отраслях.

– **Задание на безопасность** (ЗБ). Наличие ЗБ, включая цели безопасности, с самого начала разработки. Безопасность не может быть лучше, чем задание на безопасность. Любая разработка должна с самого начала ориентироваться на достижение целей безопасности в той же мере, что и на достижение функциональных целей. Другими словами, ТЗ для любой IT-системы должно содержать цели безопасности.

– **Разделение и изоляция**. Принцип, который декларирует разделение системы на независимые области (домены безопасности), каждому из которых приписываются определенные атрибуты безопасности, характеризующие соответствие заданию по безопасности или политике безопасности. В определенной мере этот принцип противоположен принципу фоннеймановской архитектуры, которая декларирует однородность среды исполнения кода и хранения данных, однако выражается шире как свойство, обеспечивающее такое разделение кода и распределение данных, отнесенных к конкретным функциям системы, которые позволят минимизировать количество непредусмотренных сценариев ее работы. Это свойство должна обеспечивать среда, в которой функционируют домены. Ключевое свойство – изоляция. Система должна допускать взаимодействие с доменами только посредством декларированных интерфейсов. И все взаимодействия должны быть наблюдаемыми (см.далее). Близкая по смыслу концепция MILS – multiply independent levels of security и её развитие DMILS = Dynamic MILS.

– **Наблюдаемость и полнота реализации контроля**. Complete Mediation. Все взаимодействия между доменами в системе должны быть наблюдаемы и оцениваться на предмет соответствия политике безопасности. Компоненты, которые обеспечивают проверку соответствия, несомненно, должны являться частью ядра безопасности.

– **Минимизация ядра безопасности** (иначе называемого доверенной вычислительной базой, или TCB = trusted computing base). Число компонентов и объем кода, от которого зависит выполнение задания по безопас-

ности и достижение целей безопасности, должны минимизироваться, насколько это возможно. Система может состоять из большого числа компонентов, но важно разработать такую архитектуру, в которой выполнение задания по безопасности будет зависеть от наименьшего количества компонентов. Именно эти компоненты необходимо наиболее тщательно тестировать и проверять, тогда выполнение задания по безопасности становится наиболее вероятным.

Такой подход позволяет строить большие сложные системы затрачивая разумное время на доказательство свойств безопасности.

– **Разумная декомпозиция.** Сложность – самый главный враг надежности и устойчивости, а значит и безопасности. Основной метод борьбы со сложностью – декомпозиция. Сложная система представляется как набор взаимодействующих подсистем (компонентов) с хорошо прописанными интерфейсами взаимодействия. Компоненты должны быть «слабосвязанными», т.е. представлять из себя обособленные, независимые сущности с хорошо определенным интерфейсом. Конечно, нужно соблюдать меру. Если переборщить с уменьшением связности, то в системе будет очень большое количество мелких подсистем, и значит система опять будет сложной для контроля. Необходимо придерживаться баланса между количеством подсистем и их сложностью. В свою очередь, каждую подсистему можно рассматривать как совокупность своих собственных подсистем. Т.е. мы представляем IT-систему как систему систем. И декомпозируем на несколько уровней до тех пор, пока не добьёмся приемлемого, контролируемого уровня сложности для каждой из подсистем. Компоненты должны быть достаточно обособлены, иметь четкий интерфейс (контракт, как принято говорить в софтверной разработке) и их функциональность во всех сценариях использования должна быть доказана тестами, статическим и динамическим анализом кода или иными проверками. Например, фазингом, пентестом или в предельном случае формальной верификацией. Особое внимание следует уделять компонентам, входящим в состав ТСВ.

Это основные принципы, которые легли в основу методологии, которую мы привыкли называть Secure-by-design. Самый близкий по смыслу русский аналог этого термина – Конструктивная информационная безопасность (КИБ).

Важное свойство такого подхода – доверенную систему можно создать из компонентов, часть из которых доверенные, а часть не доверенные. Доверенные – это те, в работе которых мы совершенно уверены на основании процесса разработки, тестирования верификации и т.д. Обеспечивать выполнение всех функций ЗБ должны только доверенные компоненты. А «не доверенные» – это те, которые не влияют на ЗБ, и значит их проверку можно осуществлять в меньшем объеме, что не повлияет на ЗБ.

Хочу подчеркнуть, что все эти принципы известны давно. Некоторые более 30 лет. Они уже доказали свою значимость как в теоретических ра-

ботах, так и в продуктах и технологиях многих компаний. Как я считаю, наша заслуга состоит в том, что мы органически соединили эти принципы в одной методологии.

Еще одно замечание. По опыту работы во многих проектах и обсуждений в разных сообществах возникает вопрос о соотнесении методологии с принципами РБПО = разработка безопасного программного обеспечения (SDLC). КИБ нацелена в первую очередь на архитектурные аспекты создания информационных систем, тогда как для РБПО главный фокус – правильная организация процесса разработки. Так что эти подходы гармонично дополняют друг друга. Есть и отличие. Оно заключается в том, что РБПО регламентирует процессы создания ПО и ограничивается этим. Применение КИБ шире – разработка информационных систем = компонентная электронная база, приборы, софт, облачные сервисы, корпоративные сети и др.

На основании этих принципов уже реализовано и сейчас создается несколько продуктов как нашей компании, так и других компаний. Тесты показывают, что эти продукты гораздо более устойчивы к внешним угрозам даже без применения наложенных средств безопасности.

На такие технологии и продукты, основанные на этих принципах, как нашей компанией, так и партнерами защищены более 70 патентов в России и других географиях. Идет работа над серией ГОСТов.

Представляемая концепция иногда также называется **Кибериммунитет**. Ее реализация позволяет любой программе, микросхеме, устройству, программно-аппаратному комплексу, информационной системе, разработанной на ее основе, нормально существовать, бесперебойно работать и предоставлять пользователям свою основную функциональность в современной «агрессивной информационной среде». Так же как делают это живые организмы в природе.