Кибериммунный подход к проектированию

как практическая реализация концепции Security-by-Design

Алексей МАТЮШИН,
старший аналитик по информационной безопасности, «Лаборатория Касперского»

Екатерина РУДИНА,
руководитель группы аналитиков по информационной безопасности, «Лаборатория Касперского»

Кибербезопасность развивается в формате технологической гонки злоумышленников и ИБ-специалистов, причем последние зачастую играют роль догоняющих. Сегодня, когда цифровые технологии проникли и в промышленность, и в энергетику, и в транспорт, и в финансовый сектор, и в другие отрасли, а также в повседневную жизнь людей, киберинцидент может нанести масштабный ущерб. В этих условиях «догонять» – неприемлемый вариант.

Классический подход к решению названной проблемы предполагает применение внешних средств защиты на изначально небезопасной системе. Альтернатива – внедрение так называемых конструктивно безопасных систем (англ. Secure-by-Design), позволяющих уменьшить поверхность атаки и предотвратить распространение угрозы по инфраструктуре. О них и поговорим здесь.

Проблемы безопасности современных киберсистем

Для современных киберсистем характерно взаимопроникновение информационных (ИТ) и операционных (ОТ) технологий: промышленных, транспортных, финансовых, научных и др. Киберсистемы становятся все более сложными, увеличивается объем программного кода, определяющего их работу. Например, по данным McKinsey & Company [1], современные автомобили содержат в десять раз больше строк кода, чем в 2010 г., а на борту автомобилей премиальных марок свыше 100 млн строк кода.

Усложнение систем усугубляется их растущей гетерогенностью, глобальной доступностью и высокоскоростной передачей больших объемов данных. Так, если раньше мобильные сети связи ориентировались на передачу контента по схеме «пользователь – сервис» или «пользователь – пользователь», то с сетями 5G на передний план выходят высокоскоростные коммуникации «киберсистема – киберсистема».

Конвергенция ОТ и ИТ изменила само понятие безопасности. Ранее киберсистемы использовались для обработки данных и под безопасностью понимались аспекты их целостности, конфиденциальности и доступности. Сегодня важность также приобретают функциональная безопасность, устойчивость, надежность работы и другие аспекты безопасности системы, определяемые обработкой данных [2] – рис. 1.

Проблемы кибербезопасности напрямую влияют на способность систем выполнять свои критические функции, что влечет финансовые последствия – риски кибербезопасности при разработке, сопровождении и поддержке систем растут. Согласно исследованию

«Лаборатории Касперского» [4], 53% организаций отказались от новых бизнес-проектов из-за неспособности справиться с рисками кибербезопасности, а 74% столкнулись с отсутствием подходящего защитного решения.

Рост киберрисков и затрат на компенсацию обусловлен несколькими факторами, среди которых:

- обширная поверхность атаки и высокая скорость ее распространения между компонентами системы;
- высокая сложность моделирования угроз;
- низкий уровень доверия к безопасности компонентов системы;
- высокая стоимость единственного (первого) инцидента;
- высокая стоимость и временные затраты, а иногда и невозможность обновления программного обеспечения.

Разработка безопасных киберсистем и их поддержка сопряжены с рядом сложностей, включая:

- отсутствие единого понимания целей деятельности: заказчики не могут описать критерии безопасной системы, а разработчики внедряют меры безопасности без понимания их необходимости и достаточности;
- высокую сложность и стоимость доказательства свойств безопасности;
- высокие затраты на квалифицированных сотрудников, особенно специалистов на стыке отраслевых дисциплин и кибербезопасности.

При разработке программных или программно-аппаратных продуктов вопрос безопасности возникает ближе к концу проекта, поскольку требования безопасности традиционно рассматриваются как нефункциональные. В этом случае защитные меры, реализованные в отрыве от основного функционала продукта, нередко оставляют лазейки для злоумышленника и не позволяют эффективно противостоять атакам.

Наиболее распространенный подход к решению подобных проблем – применение наложенных средств безопасности. Лучшие отраслевые практики включают также аудит безопасности и управление обновлениями компонентов OT- и ИТ-сетей, обучение персонала основам кибербезопасной работы и постоянный мониторинг изменений ландшафта угроз. Эти решения и методы остаются популярными и востребованными для активно эксплуатируемых систем и технологий, но есть и другой способ усилить защиту – на уровне архитектуры.

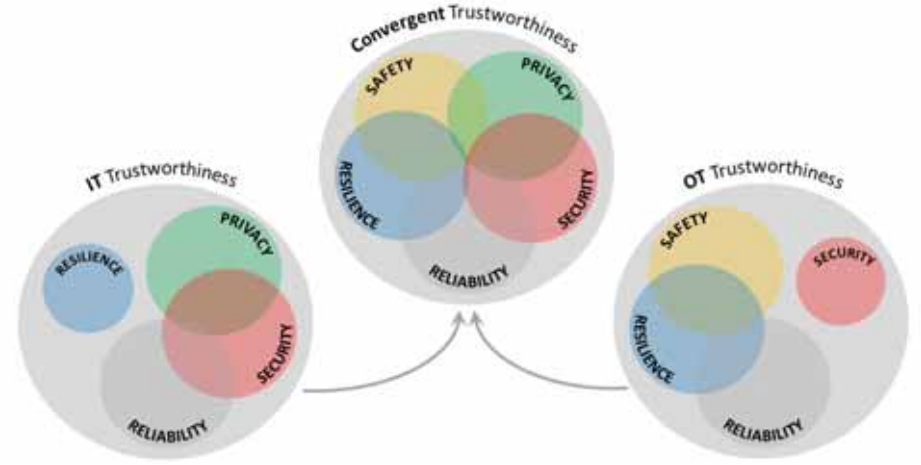

Рис. 1. *Изменение понятия и содержания безопасности [2]*

Конструктивная информационная безопасность — Security-by-Design

Альтернативный подход – разработка конструктивно безопасных систем. Его суть заключается в проектировании киберсистем, в которых меры безопасности интегрированы в архитектуру и программный код и являются его частью. В данном случае аспекты безопасности принимаются во внимание, начиная с самых ранних стадий разработки, – требования безопасности приравниваются к функциональным требованиям и влияют на выбор архитектуры решения и аппаратной базы.

Этот подход применяется в прикладных сферах, традиционно относящихся к критическим, например, в авиастроении концепция архитектуры IMA (Integrated Modular Avionics) полностью соответствует принципам Security-by-Design. Тем не менее широкого распространения подход пока не получил, чему есть несколько причин:

- подход недостаточно проработан методически, поэтому его практическое применение сводится к экспертизе конкретных разработчиков, что препятствует массовому использованию, поскольку процесс разработки сложно оценивать, прогнозировать и финансировать;
- интеграция безопасности в архитектуру и дизайн системы – весьма затратная процедура, поскольку требует привлечения высококвалифицированных специалистов и увеличивает сроки разработки.

Таким образом, необходима простая и экономически эффективная методология, которая позволит воплощать безопасность как результат не «персонального искусства» отдельных разработчиков, а спланированной деятельности. На решение этой задачи претендует кибериммунный подход к разработке от «Лаборатории Касперского».

Кибериммунный подход к разработке киберсистем

Кибериммунный подход – это эволюционное развитие технологий обеспечения безопасности, основу которого составляют

теоретическая база и мировые практики построения безопасных систем в области промышленности, транспорта, управления. Подход объединяет экономически эффективную методологию разработки киберсистем с архитектурными требованиями к ним, что позволяет применять его в различных сферах.

Целью кибериммунного подхода является создание кибериммунной системы – киберсистемы, декларированные активы которой защищены от нежелательных событий при любых условиях, даже под атакой, при условии заданных ограничений.

Кибериммунный подход состоит из двух частей, нацеленных на методическое обеспечение подхода Security-by-Design:

- требования к организации разработки (процессные требования): какие действия необходимо и достаточно предпринять и какие результаты получить, чтобы обеспечить экономически эффективную разработку безопасной архитектуры;
- требования к архитектуре и дизайну системы – базовые концепции, которые должны быть заложены в архитектуру и дизайн, чтобы обеспечить высокий уровень безопасности системы и высокий уровень уверенности в её безопасности.

Обе составляющие кибериммунного подхода далее рассмотрим более подробно.

Процессные требования кибериммунного подхода

Кибериммунный процесс предъявляет требования к результатам каждого этапа разработки, но не регламентирует жестко, как именно выполняется работа, ограничиваясь методическими рекомендациями. Это соответствует принципу «не важен процесс, важен результат».

Цели безопасности

Более формальное определение кибериммунной системы звучит так: *это система, которая достигает установленных для нее целей безопасности во всех сценариях работы, при любых обстоятельствах, при выполнении заданных ограничений.*

Цель безопасности – условие или требование, относящееся к системе в целом, которое должно выполняться на протяжении всего ее жизненного цикла. Например, «при выполнении операций чтения, записи, передачи данных эти данные остаются неизвестными для неавторизированного круга лиц».

Определение целей безопасности – первый этап разработки кибериммунной системы. Под обозначенные цели безопасности затем разрабатывается архитектура и выполняется дизайн.

Цели безопасности могут определяться разными методами: для отраслей с высоким уровнем регуляции – путем анализа требований регулятора, для сфер, где регуляция практически отсутствует, – на основе анализа потребностей в безопасности, определенных владельцем или заказчиком. Эти потребности часто связаны с защитой активов системы, и на практике заказчику понятно, как формулировать задачи по безопасности, отталкиваясь от них. Таким образом, кибериммунный подход применим ко всем прикладным сферам разработки киберсистем.

Важный практический аспект подхода – противостояние «перекладыванию ответственности». Любые технические средства, технологии и инструментарий защиты, например криптографические ключи, пароли, сертификаты и др., не рассматриваются как актив системы и не должны фигурировать в поле зрения заказчика. С одной стороны, это дает возможность заказчику самому определять активы системы на своем языке, не перекладывая эту задачу на разработчика, с другой – не сужает задачу и не ограничивает разработчика в выборе средств и методов защиты.

Перед разработкой архитектуры под заданные цели безопасности каждая цель конвертируется в набор требований безопасности, которые детально описывают, что именно надо сделать, чтобы обеспечить достижение этой цели, т. е. формируются требования к функциям защиты. На данном этапе и появляются требования вроде обеспечения защиты криптографических ключей.

Моделирование угроз

Моделирование угроз активно используется в методологии создания кибериммунных систем. Чтобы этот инструмент был эффективен, важно понимать, к какому объекту применять моделирование, какие исходные данные и методы использовать, какие результаты нужно получить.

1. Техническое моделирование

Чаще всего под термином «моделирование угроз» понимают технические процедуры, которые опираются на базы знаний о типовых угрозах, векторах, техниках и тактиках кибератак (MITRE, STRIDE, БДУ ФСТЭК). Обязательный результат – артефакт под названием «модель угроз», который описывает источники угроз или модель нарушителя, перечень исходящих угроз, актуальные технические сценарии реализации. Частая цель такого моделирования – описание угроз для оценки рисков в терминах размера ущерба и вероятности его реализации. На основании этой информации можно спланировать меры противодействия.

Для выполнения полноценного технического моделирования угроз необходим вариант архитектуры системы и привлечение квалифицированных специалистов.

2. Верхнеуровневое моделирование

Верхнеуровневое моделирование угроз – менее формальная процедура, для которой необязательно иметь полностью описанную архитектуру системы. Такой тип моделирования угроз можно назвать «а что будет, если...».

Кибериммунный подход использует оба типа моделирования угроз: верхнеуровневое – для создания архитектуры под заданные

цели безопасности, формальное техническое – для проверки качества разработанной архитектуры, исправления ошибок или подтверждения того, что архитектура соответствует целям безопасности.

Создание архитектуры под цели безопасности

Подход Security-by-Design предполагает, что аспекты безопасности должны быть учтены при создании архитектуры, т. е. до этапа технического моделирования угроз. Выглядит немного парадоксально – меры защиты вырабатываются по итогам моделирования, для которого нужна архитектура, которую надо создать с учетом требований по безопасности! Как кибериммунная методология предлагает решать такую задачу?

Это основная идея кибериммунного подхода: еще до формального моделирования угроз необходимо определить критические части системы, непосредственно отвечающие за активы, и изолировать их от любого возможного нежелательного воздействия. Для этого следует контролировать взаимодействия критических частей системы со всеми остальными, делить их на более мелкие и оставлять среди критических только самое необходимое.

Данный процесс выполняется на всем протяжении формирования базовой архитектуры системы. Мы постоянно отвечаем на вопрос: «Что, если тот или иной компонент будет скомпрометирован?». В результате остается относительно небольшое число критических компонентов, которые не должны быть скомпрометированы. Их необходимо очень тщательно разработать, протестировать и верифицировать. Поверхность атаки на эти компоненты, следовательно на всю систему, будет мала, а стоимость атаки – высока.

Отрицает ли такой подход проведение полноценного технического моделирования угроз? Нет, но позволяет уменьшить объем изменений архитектуры после технического моделирования, в идеальном случае – до нуля. Кроме того, подобный подход обеспечивает более высокую устойчивость системы к угрозам, сценариям и способам атак, которые не были известны на момент моделирования угроз.

Верификация качества кода

Верификация кода, подтверждение корректности работы и качества с точки зрения безопасности – важнейший этап создания безопасной системы. Возможно, изменятся способы верификации и инструментарий, но не отменится сама процедура.

Особенность кибериммунного подхода заключается в том, что строгой верификации подвергается не весь код, а сравнительно небольшая его часть – только так называемый доверенный код, критические компоненты, непосредственно влияющие на достижение целей безопасности, а остальной код верифицируется наиболее экономичными способами.

Требования к архитектуре и дизайну кибериммунной системы

Кибериммунный подход предъявляет требования к организации базовой архитектуры системы. Мы предлагаем использовать проверенные архитектурные шаблоны безопасности, которые значительно упрощают разработку по-настоящему безопасных и надежных систем. Такая разработка ведется быстрее и эффективнее, а результат вызывает больше доверия.

Базовых требований к архитектуре и дизайну системы несколько:

- строгая изоляция компонентов друг от друга, исключающая их неконтролируемое взаимодействие;
- обязательный контроль разрешенных коммуникаций между изолированными компонентами на основе формализованных политик безопасности;
- выделение подмножества доверенного кода из всего множества кода системы (TCB, trusted computing base) и минимизация этого подмножества;
- соответствие схемы разрешенных коммуникаций между доверенными и недоверенными компонентами кибериммунной модели целостности, о которой мы поговорим далее.

Изоляция и контроль

Базовые архитектурные паттерны, которым предлагает следовать кибериммунный подход, представлены в архитектурной концепции MILS [5] и архитектуре FLASK [3]. В российских стандартах MILS-системы соответствуют системам с разделением доменов, и в дальнейшем мы будем использовать этот термин [6, 7].

Система с разделением доменов – это система, разделенная на строго изолированные модули, называемые доменами безопасности, коммуникации между которыми определяются архитектурой политики и контролируются ядром разделения системы. Все остальные функции безопасности реализуются внутри изолированных доменов безопасности [6, 7].

Такой архитектурный подход основан на выводе исследователей в области ИБ о том, что безопасность системы зависит от двух факторов: насколько хорошо изолированы компоненты системы друг от друга и от качества собственных функций безопасности этих компонентов [8, 9].

Преимущество подхода в том, что в системе с разделением доменов можно сделать доказательные выводы об уровне защищенности всей системы в целом за конечное время. В тех случаях, когда система нетривиальна и состоит из гетерогенных компонентов, разработанных, возможно, разными поставщиками, это свойство становится не просто желательным, а необходимым.

В системе с разделением доменов каждый домен безопасности полагается на собственные функции безопасности и не доверяет никому, кроме ядра разделения – самого доверенного компонента всей системы. Эта концепция идейно соответствует

Zero Trust-архитектуре (архитектуре нулевого доверия), доказавшей свою эффективность в обеспечении безопасности корпоративных сетей [10, 11]. Концепция MILS также соответствует известному шаблону безопасности Distrustful Decomposition [12].

У разработчиков архитектуры FLASK в ее изначальном варианте была цель, близкая той, которую преследовали разработчики концепции систем с разделением доменов. Обе концепции отвергают идею монолитного ядра безопасности, ответственного за всю безопасность системы. Разработчики FLASK тоже стремились к надежному контролю взаимодействий модулей системы, но главное внимание уделили гибкости политик безопасности [3]. Для этого важны возможность управлять уже выданными разрешениями на доступ и отсутствие централизованных шаблонных правил, заданных ядром безопасности системы.

Понятие безопасности расширилось, ни одно общее для всех случаев определение не будет неизменным и достаточным. Но любое понятие выражается в политике безопасности, поэтому общеприемлемая система определения и вычисления политик должна допускать произвольное расширение как множества политик, так и их типов [3].

Задача управления разрешениями на доступ включает контроль за гранулированностью прав, контроль распространения и отзыв прав. Все эти функции не являются тривиальными. Например, UNIX-подобные операционные системы закрепляют права доступа за дескриптором каждого открытого файла, поэтому отзыв прав не подействует на уже открытые файлы. В случае использования популярного механизма контроля доступа Object Capabilities серьезной проблемой является контроль распространения прав. Как и в любой другой системе дискреционного типа, субъект может передавать права (Capabilities) другому субъекту, и система «миграционного контроля» способна быть чрезвычайно изощренной.

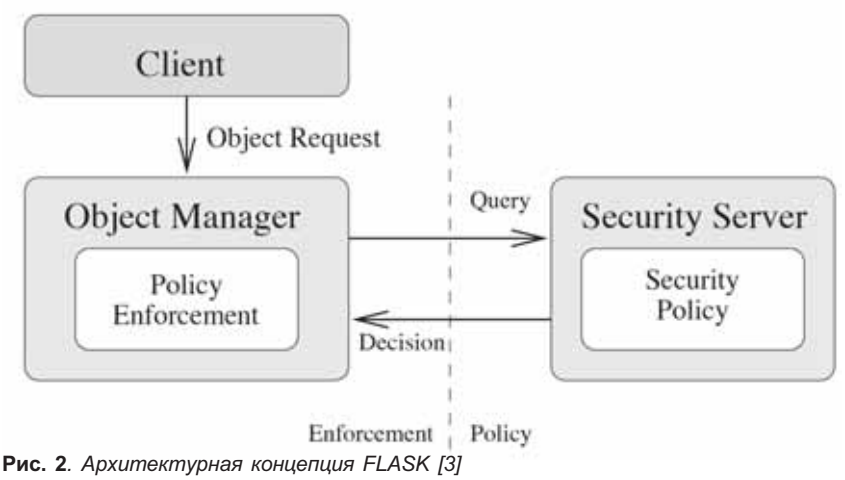

Рис. 2. *Архитектурная концепция FLASK [3]*

Разработчики FLASK предложили архитектуру, которая стала еще одним широко известным шаблоном безопасности [12]: разделение компонентов, ответственных за предоставление доступа на основании готовых разрешений доступа (Policy Enforcement Point), и компонентов, ответственных за вычисление этих разрешений (Policy Decision Point, он же монитор или сервер безопасности) – рис. 2.

Кибериммунный подход в части требований к архитектуре системы обязывает на системном уровне следовать шаблонам безопасности «декомпозиция с утратой доверия» (Distrustful Decomposition – DD) и «разделение точек принятия и исполнения решений безопасности» (Policy Decision Point & Policy Enforcement Point – PDP/PEP), в соответствии с двумя архитектурными концепциями – системами с разделением на домены и FLASK (в изначальном микроядерном варианте).

Кибериммунная модель целостности

Одно из центральных понятий системы с разделением доменов – архитектура политики, т. е. схема допустимых направленных взаимодействий между изолированными компонентами системы.

Кибериммунный подход предъявляет дополнительное требование: система связей, заданная архитектурой политики, должна отвечать кибериммунной модели целостности, которая является модификацией модели целостности Биба и по смыслу близка модели целостности Кларка – Вилсона (рис. 3).

Для этого все домены безопасности разделяются на три класса: недоверенные, доверенные и те, которые повышают доверие к проходящим через них данным. Классифицируются они на основании влияния на достижение заданных целей безопасности (напомним, что задача

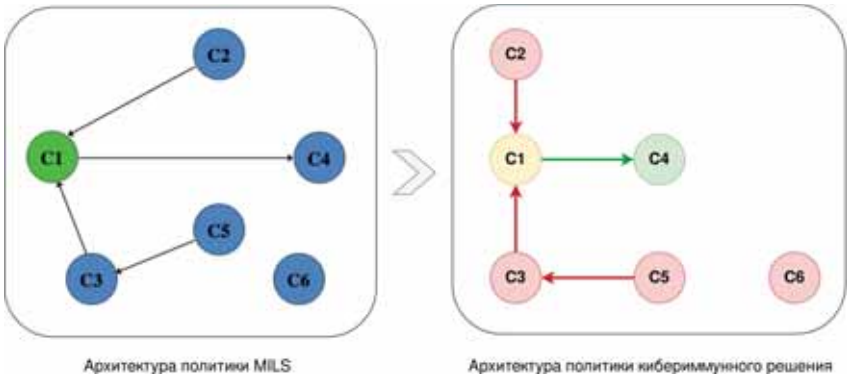

Рис. 3. *Архитектурные политики*

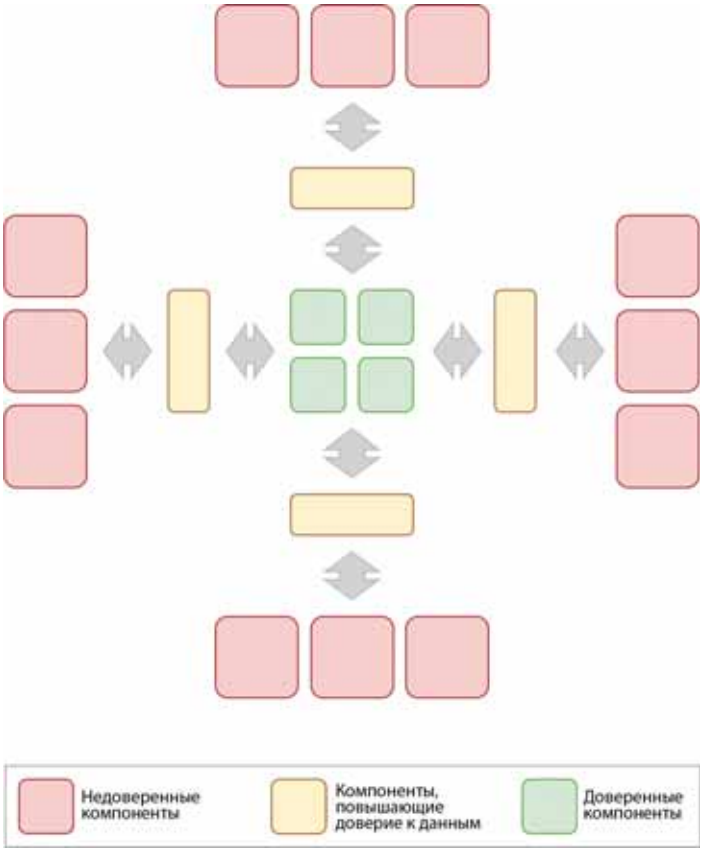

Рис. 4. *Политики кибериммунного решения*

кибериммунной системы – обеспечить достижение этих целей при любых обстоятельствах).

Кибериммунная модель целостности отличается от модели Биба (как и модель Кларка – Вилсона) наличием дополнительного класса элементов, повышающих целостность. Его необходимость обоснована практикой: одновременно соблюсти модель целостности Биба и реализовать весь необходимый прикладной функционал – не всегда решаемая задача. От модели Кларка – Вилсона кибериммунная модель целостности отличается меньшим объемом требований, также исходя из практических соображений.

Исследования показывают, что в системе с такой моделью безопасности злоумышленник не сможет распространить атаку на критически важные части системы [13, 14].

Архитектура политики кибериммунного решения, с некоторыми упрощениями, соответствует следующей схеме (рис. 4).

Почему только целостность? А как насчет конфиденциальности или других аспектов безопасности? Дело в том, что контроль функциональной целостности и целостности данных – необходимые условия обеспечения любых других видов безопасности. В самом деле, если злоумышленник может нарушить функциональную целостность модуля, отвечающего за конфиденциальность, о какой конфиденциальности может идти речь? Поэтому поддержание целостности функций и данных на системном уровне – необходимое условие, тогда обеспечение остальных аспектов безопасности возможно на прикладном или промежуточном уровне.

Доверенный и недоверенный код, минимизация ТСВ

Все модули кибериммунной системы классифицируются как недоверенные, доверенные и повышающие уровень доверия к данным. Модули системы, относящиеся к двум последним классам, входят в ТСВ наряду с доверенными системными компонентами, такими как ядро разделения и монитор безопасности.

Кибериммунный подход подчеркивает важность доказательства надежности программного кода. Однако доказывать надежность каждого компонента системы чрезвычайно дорого и не всегда возможно как из-за огромного объема кода, так и из-за использования заимствованного кода, который может часто обновляться.

Если системные компоненты верифицируются поставщиком, то верификация прикладных доверенных компонентов – обязанность прикладного разработчика. Поэтому код прикладных доверенных компонентов должен быть минимизирован, иначе верификация займет слишком много времени или будет коммерчески нецелесообразной. Недоверенные компоненты также верифицируются, но наиболее простыми и малозатратными методами.

К доверенным компонентам предъявляются определенные требования: они должны быть функционально просты, иметь небольшое количество интерфейсов со строгой типизацией параметров, быть небольшими по объему кода и функционально однородными. Эти требования направлены на минимизацию поверхности атаки на доверенные компоненты, а также на оптимизацию затрат на верификацию.

Подобно концепции Zero Trust, кибериммунная система сводит контроль поверхности атаки на систему к контролю относительно небольшой поверхности защиты, образованной доверенными компонентами. Гарантировать качество поверхности защиты технически проще, за счет чего киберсистема будет более надежной с точки зрения безопасности (рис. 5).

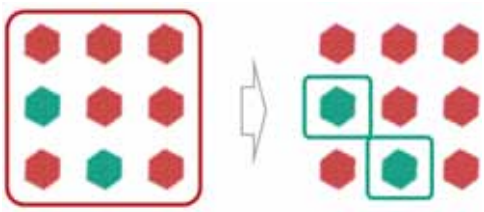

Рис. 5. *От поверхности атаки – к поверхности защиты*

Подведем итоги

Кибериммунный подход описывает методологию создания конструктивно безопасной киберсистемы методами, доступными большинству бизнес-заказчиков и разработчиков. В частности, подход помогает клиенту сформулировать задание по безопасности, не делегируя эту ответственность, а разработчику предоставляет критерии выполнения задачи.

Кибериммунные требования к архитектуре и дизайну системы построены на применении базовых шаблонов безопасности, что радикально повышает уровень безопасности системы, даже в отсутствие других методов защиты. Вместе с тем использование в прикладной разработке других шаблонов безопасности, мало известных большинству разработчиков, способствует повышению прозрачности системы и упрощает доказательство ее свойств безопасности.

Должным образом спроектированная и верифицированная кибериммунная система обеспечивает достижение заявленных целей безопасности даже под атакой, при условии выполнения предположений безопасности. Особо отметим, что кибериммунная система способна противостоять угрозам, неизвестным на момент разработки.

Конечно, разрабатывать безопасную систему дороже и дольше, чем систему, которая «просто работает» (когда «все хорошо»). Вопрос в том, какие риски и затраты готов понести владелец системы.

Практика подтверждает, что вопрос, будет ли атакована система, не стоит. Вопрос в том, когда она будет атакована. Каков тогда будет ущерб? Будет ли первый инцидент и последним? Сколько времени и ресурсов потребуется на восстановление системы и ее обновление? Можно ли будет восстановить репутацию и доверие заказчика, пользователей, владельцев системы? Систему надо обновлять не только после атаки, но и для ее предотвращения, если стали известны уязвимости в компонентах системы. В зависимости от особенностей системы обновление может быть как простой, так и крайне ресурсозатратной задачей, а иногда и вовсе невозможной.

В случае с кибериммунной системой дополнительные затраты на разработку с высокой вероятностью будут компенсированы экономией на поддержке. Вполне возможно, обновления не потребуются вовсе.

Вопрос безопасности стоит всегда и перед всеми системами. Необходимо только подсчитать, что экономически целесообразно: разработать кибериммунную систему или надеяться, что затраты на обновления системы будут невелики, а прямой и косвенный ущерб от атаки приемлем? ■

Литература и ссылки

1. *McKinsey & Company – Rethinking car software and electronics architecture, February 2018 https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture*
2. *Industrial Internet of Things Volume G4: Security Framework, Industrial Internet Consortium, Security Working Group, 2016.*
3. *The Flask Security Architecture: System Support for Diverse Security Policies, Ray Spencer (Secure Computing Corporation), Stephen Smalley, Peter Loscocco (National Security Agency), Mike Hibler, David Andersen, Jay Lepreau (University of Utah), Feb. 2004.*
4. *Kaspersky Global Corporate IT Security Risks Survey (ITSRS), 2021 https://www.kaspersky.com/blog/iot-report-2022/*
5. *MILS Architectural Approach Supporting Trustworthiness of the IIoT Solutions, whitepaper, Industrial Internet Consortium, Rance J. DeLong, Ekaterina Rudina, 2021.*
6. *ПНСТ 818-2023. Информационные технологии. Интернет вещей. Системы с разделением доменов. Базовые компоненты.*
7. *ПНСТ 819-2023. Информационные технологии. Интернет вещей. Системы с разделением доменов. Термины и определения.*
8. *J. M. Rushby. Design and Verification of Secure Systems, ACM SIGOPS Operating Systems Review, Volume 15, Issue 5, December 1981.*
9. *John Rushby. Partitioning in Avionics Architectures: Requirements, Mechanisms, and Assurance. U.S. Department of Transportation, Federal Aviation Administration, DOT/FAA/AR-99/58, National Aeronautics and Space Administration, NASA/CR-1999-209347, Final Report, March, 2000.*
10. *Zero Trust Architecture, NIST, Special Publication 800-207*
11. *Блог Касперского, Концепция Zero Trust: не доверяй — всегда проверяй, https://www.kaspersky.ru/blog/zero-trust-security/28780/*
12. *Secure Design Patterns, Dougherty et al, Software Engineering Institute, 2009*
13. *Модель мандатного контроля целостности в операционной системе KasperskyOS, В. С. Буренков, Д. А. Кулагин, АО «Лаборатория Касперского», DOI: 10.15514/ISPRAS-2020-32(1)-2*
14. *Формальная верификация модели мандатного контроля целостности в операционной системе KasperskyOS. В.С. Буренков, АО «Лаборатория Касперского», DOI: 10.15514/ISPRAS-2020-32(6)-3*