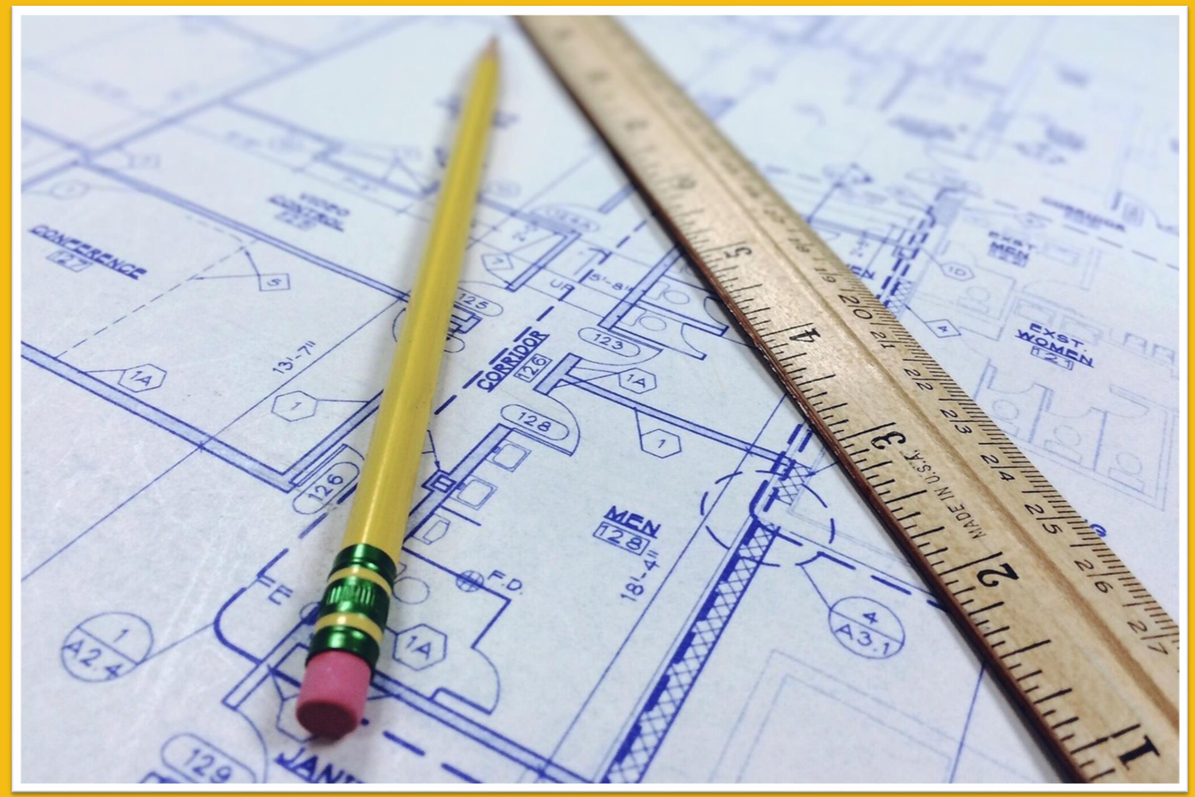
КОД
ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ

ПРИНЦИПЫ БЕЗОПАСНОГО ПРОЕКТИРОВАНИЯ ПРИЛОЖЕНИЙ И ИНФРАСТРУКТУР

10 принципов, которые помогут вам решить большинство проблем безопасности до их появления



ОСНОВНЫЕ АСПЕКТЫ ПРАКТИЧЕСКОЙ БЕЗОПАСНОСТИ

Ключевые принципы безопасного проектирования

- Минимальные привилегии
- Разделение обязанностей
- Осторожное доверие
- Проще - лучше
- Аудит чувствительных событий
- Найдите самое слабое звено
- Безопасные сбои и безопасность по умолчанию
- Безопасность через неизвестность не работает
- Внедряйте безопасность вглубь
- Никогда не изобретайте технологии безопасности

МИНИМАЛЬНЫЕ ПРИВИЛЕГИИ

Почему?	Широкие привилегии разрешают вредоносный или случайный доступ к защищаемым ресурсам
Принцип	Ограничьте права до необходимого минимума
Минусы	Менее удобный, менее эффективный, более сложный
Пример	Запускайте процессы на сервере под своими собственными учетными записями со строго необходимым набором привилегий

РАЗДЕЛЕНИЕ ОБЯЗАННОСТЕЙ

Почему?	Обеспечьте контроль и отчетность, ограничьте влияние успешных атак, сделайте атаки менее привлекательными
Принцип	Разделите и раздробите обязанности и привилегии
Минусы	Увеличение затрат на разработку и тестирование, сложность эксплуатации, сложнее устранение недостатков
Пример	Администратор модуля «Платежи» не имеет доступа к контролю функций модуля «Заказы»

ОСТОРОЖНОЕ ДОВЕРИЕ

Почему?	Множество проблем с безопасностью вызваны внедрением злонамеренных посредников в пути связи
Принцип	Предположим, что неизвестные объекты не являются доверенными, имеют четкий процесс установления доверия, проверяют, кто подключается
Минусы	Сложность эксплуатации (особенно восстановление после сбоев), надежность, дополнительные расходы на разработку
Пример	Не принимайте ненадежные соединения удаленных интерфейсов, используйте сертификаты клиентов, учетные данные или сетевые контроли

ПРОЩЕ - ЛУЧШЕ

Почему?	Безопасность требует понимания архитектуры. Сложный дизайн труден для понимания. Простота позволяет анализировать
Принцип	Избегайте ненужных или неявных функций, избегайте сложные сценарии отказов …
Минусы	Требуется больше усилий на этапе проектирования. Сложное принятие решений, относительно функций и их сложности
Пример	Действительно ли системе требуется динамическая конфигурация среды выполнения через настраиваемый DSL?

АУДИТ ЧУВСТВИТЕЛЬНЫХ СОБЫТИЙ

Почему?	Обеспечьте запись действий, сохраняйте архив журналов для восстановления цепочки событий, установите единую точку мониторинга
Принцип	Записывайте все важные события, связанные с безопасностью, в защищенном от несанкционированного доступа хранилище
Минусы	Производительность, сложность эксплуатации, стоимость разработки
Пример	Записывайте все изменения в "основных" бизнес-объектах в хранилище только для добавления с помощью (пользователь, IP-адрес, временная метка, объект, событие и тд)

НАЙДИТЕ САМОЕ СЛАБОЕ ЗВЕНО

Почему?	Проблема «бумажной стены» - обычное явление, когда основное внимание уделяется технологиям, а не угрозам.
Принцип	Найдите самое слабое звено в цепи безопасности и укрепите его - повторите! (Моделирование угроз)
Минусы	Требуются значительные усилия, часто проблемы обнаруживаются в самый неподходящий момент!
Пример	Угроза конфиденциальности данных с одной стороны минимизирована зашифрованными каналами связи, но в то же время хранится в незашифрованной базе данных с незашифрованными резервными копиями

БЕЗОПАСНЫЕ СБОИ И БЕЗОПАСНОСТЬ ПО УМОЛЧАНИЮ

Почему?	Пароли, порты и правила по умолчанию - открытые двери. Состояния сбоя и перезапуска часто по умолчанию слишком «небезопасны»
Принцип	Принудительно измените чувствительные к безопасности параметры. Подумайте о сбоях - они должны быть безопасными, но восстановимыми
Минусы	Удобство
Пример	Не разрешать доступы уровня «СИСТЕМА / МЕНЕДЖЕР» после установки. При сбое не отключайте и не сбрасывайте элементы управления безопасностью

БЕЗОПАСНОСТЬ ЧЕРЕЗ НЕИЗВЕСТНОСТЬ НЕ РАБОТАЕТ

Почему?	Скрывать вещи сложно – если кто-то захочет их найти, найдёт случайно или намеренно
Принцип	Предположим, что злоумышленник обладает совершенными знаниями, это требует безопасного проектирования системы
Минусы	Создание действительно безопасной системы требует времени и усилий
Пример	Предположим, что злоумышленник угадывает последовательность сетевого запроса «port knocking"

ВНЕДРЯЙТЕ БЕЗОПАСНОСТЬ ВГЛУБЬ

Почему?	Системы действительно подвергаются атакам, происходят взломы, совершаются ошибки - необходимо минимизировать воздействие
Принцип	Не полагайтесь на единую точку безопасности, защищайте каждый уровень, используйте разные механизмы, останавливайте распространение сбоев на одном уровне
Минусы	Избыточность политик, сложность решения задач и устранения неполадок может затруднить восстановление
Пример	Контроль доступа к интерфейсам, сервисам, базам данных, операционной системе

НИКОГДА НЕ ИЗОБРЕТАЙТЕ ТЕХНОЛОГИИ БЕЗОПАСНОСТИ

Почему?	Технологии безопасности сложно создать – это работа для специалистов. Самостоятельно избежать уязвимостей сложно
Принцип	Не создавайте собственные технологии безопасности, всегда используйте проверенные компоненты
Минусы	Время для оценки технологии безопасности, усилий по ее изучению, сложности
Пример	Не изобретайте собственный механизм SSO, хранилище секретов или криптографические библиотеки… выбирайте отраслевые стандарты

#CODEIB

СПАСИБО ЗА ВНИМАНИЕ

kazcrust@gmail.com

linkedin.com/in/konshvetsov/