SafeCode

Могут ли программисты делать безопасность

и является ли DevSecOps панацеей

Сергей Соболев
Старший архитектор по информационной безопасности, KasperskyOS Community Development, «Лаборатория Касперского»

Роберт Альдини
Cyber Immunity Education manager, KasperskyOS Community Development, «Лаборатория Касперского»

KasperskyOS

Примеры основных стереотипов

1. Информационная безопасность это - ответственность специалистов по ИБ

Примеры основных стереотипов

1. Информационная безопасность это - ответственность специалистов по ИБ

2. Шифрование это – лучшее средство для обеспечения ИБ

Примеры основных стереотипов

1. Информационная безопасность это - ответственность специалистов по ИБ

2. Шифрование это – лучшее средство для обеспечения ИБ

3. Собственный код это – самый безопасный код

Распространённые **вредные** особенности разработки

1. Непонимание уровня критичности кода в системе - потому что никто человеческим языком не объяснил, почему и что в системе критично

Распространённые **вредные** особенности разработки

1. Непонимание уровня критичности кода в системе - потому что никто человеческим языком не объяснил, почему и что в системе критично

2. Размазывание критичного кода по подсистемам – автоматически повышает уровень критичности всей подсистемы

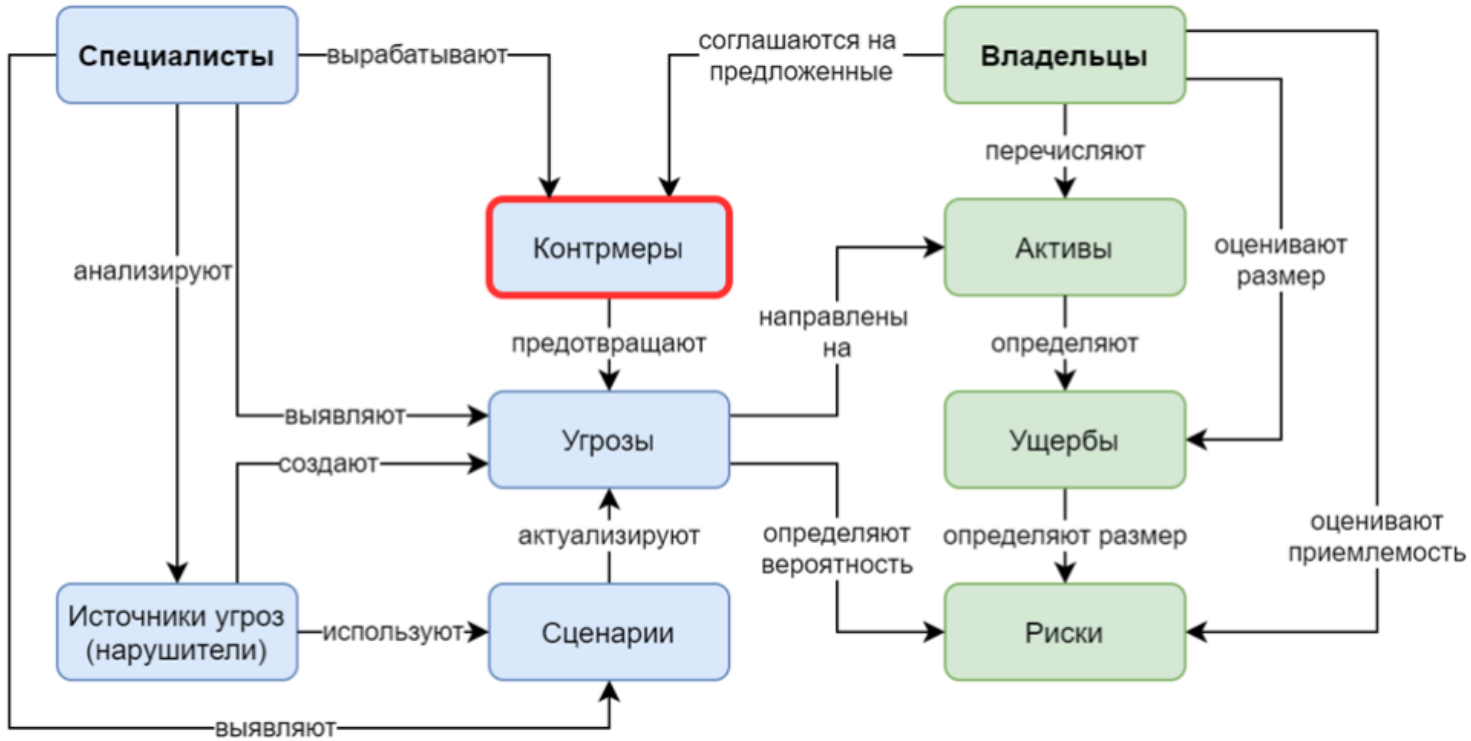
Распространённые **вредные** особенности разработки

1. Непонимание уровня критичности кода в системе - потому что никто человеческим языком не объяснил, почему и что в системе критично

2. Размазывание критичного кода по подсистемам – автоматически повышает уровень критичности всей подсистемы

3. Отсутствие контроля взаимодействия подсистем – потому что, «а что плохого может случиться, если к подсистеме нет доступа снаружи»?

Концепция безопасности продукта



Заказчик или владелец ДОЛЖЕН описать **ценности** и какие **бизнес**-риски в отношении них для него неприемлемы.

Конструктивная киберзащита

Как построить решение,
которому можно доверять,
из компонентов, большинству
из которых доверять нельзя?

Конструктивная киберзащита

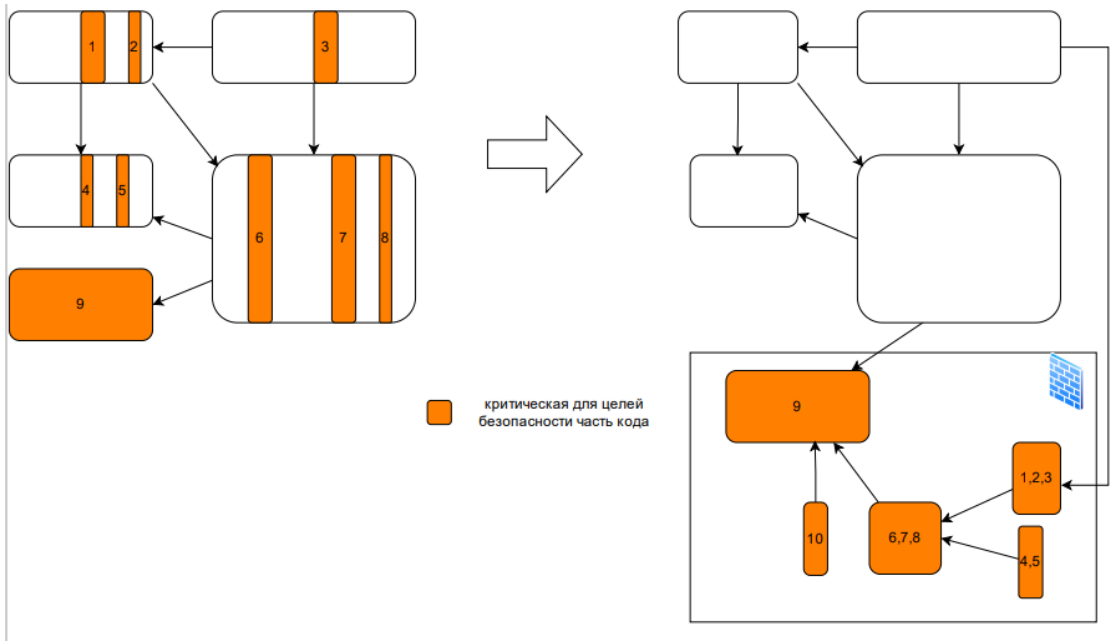
Secure by Design:

Система должна быть спроектирована так, чтобы быстро обосновать ее безопасность

Три фундаментальных принципа:

- Изоляция
- Контроль
- Минимизация доверенной кодовой базы

Кибериммунный подход – отделение кода безопасности / минимизация доверенной кодовой базы



Для выделения кода безопасности часто удобно создать новые «простые» компоненты системы:

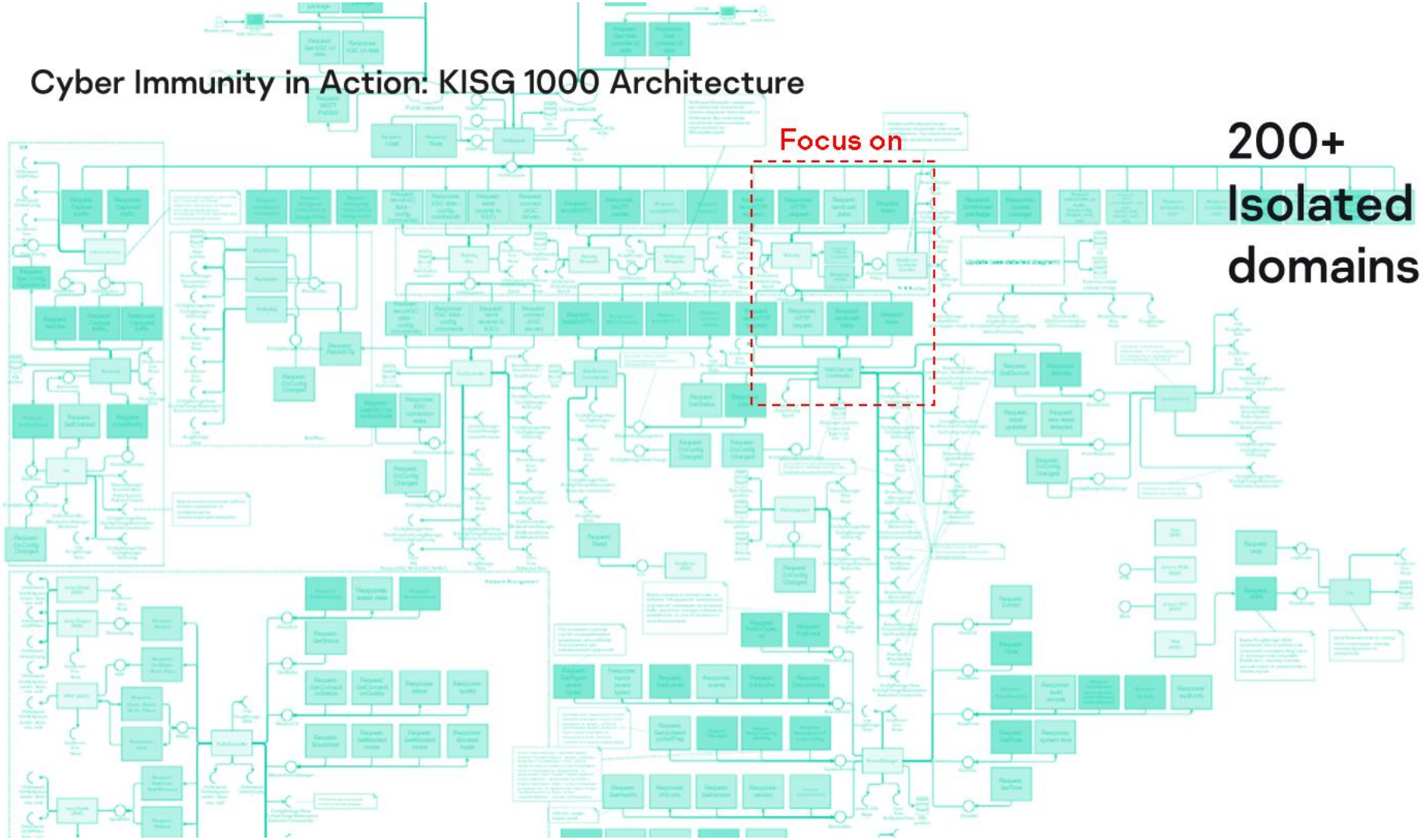
+ большие/сложные функциональные компоненты при этом изолируются

+ их влияние на безопасность заметно снижается

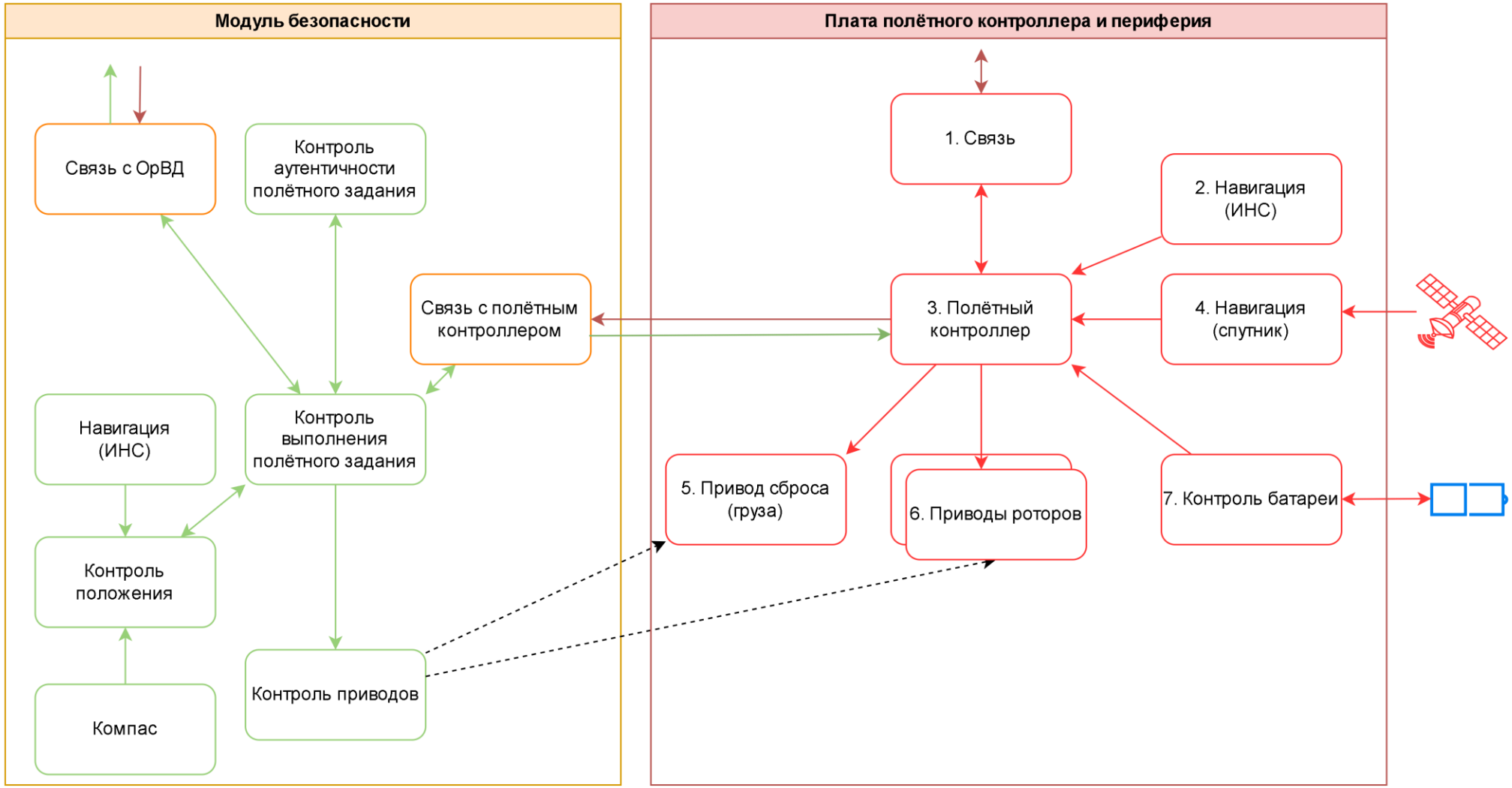
+ поиск/устранение уязвимостей в них из критичного становится просто желательным.

+ это позволит сфокусировать (возможно, и удешевить тоже) работу в контексте DevSecOps на сравнительно небольшом количестве сравнительно простого критического кода

Иллюстрация на примере архитектуры промышленного шлюза



Прототип конструктивно защищённого автономного квадрокоптера от ВУЗа-партнёра



Можно тестировать поведение системы под атаками, даже не зная самих атак

Результат

- «Проверяемая» безопасность
- «Встроенная» безопасность, без полагания только на «периметр»
- Значительное снижение затрат
- Под контролем разработчиков, а не просто под диктовку ИБ

Обратная сторона?

- Надо учиться
- Поначалу странно, как на велосипеде или коньках

Обучение кибериммунному подходу к разработке

https://t.me/learning_cyberimmunity

@LEARNING_CYBERIMMUNITY

Сергей Соболев	Старший архитектор по информационной безопасности, KasperskyOS Community Development	Sergey.P.Sobolev @kaspersky.com
Роберт Альдини	Cyber Immunity Education manager, KasperskyOS Community Development	Robert.Aldini @kaspersky.com

KasperskyOS