AppMetrica

DevSecOps

СОДЕРЖАНИЕ

01 Наш подход к реализации проекта 3

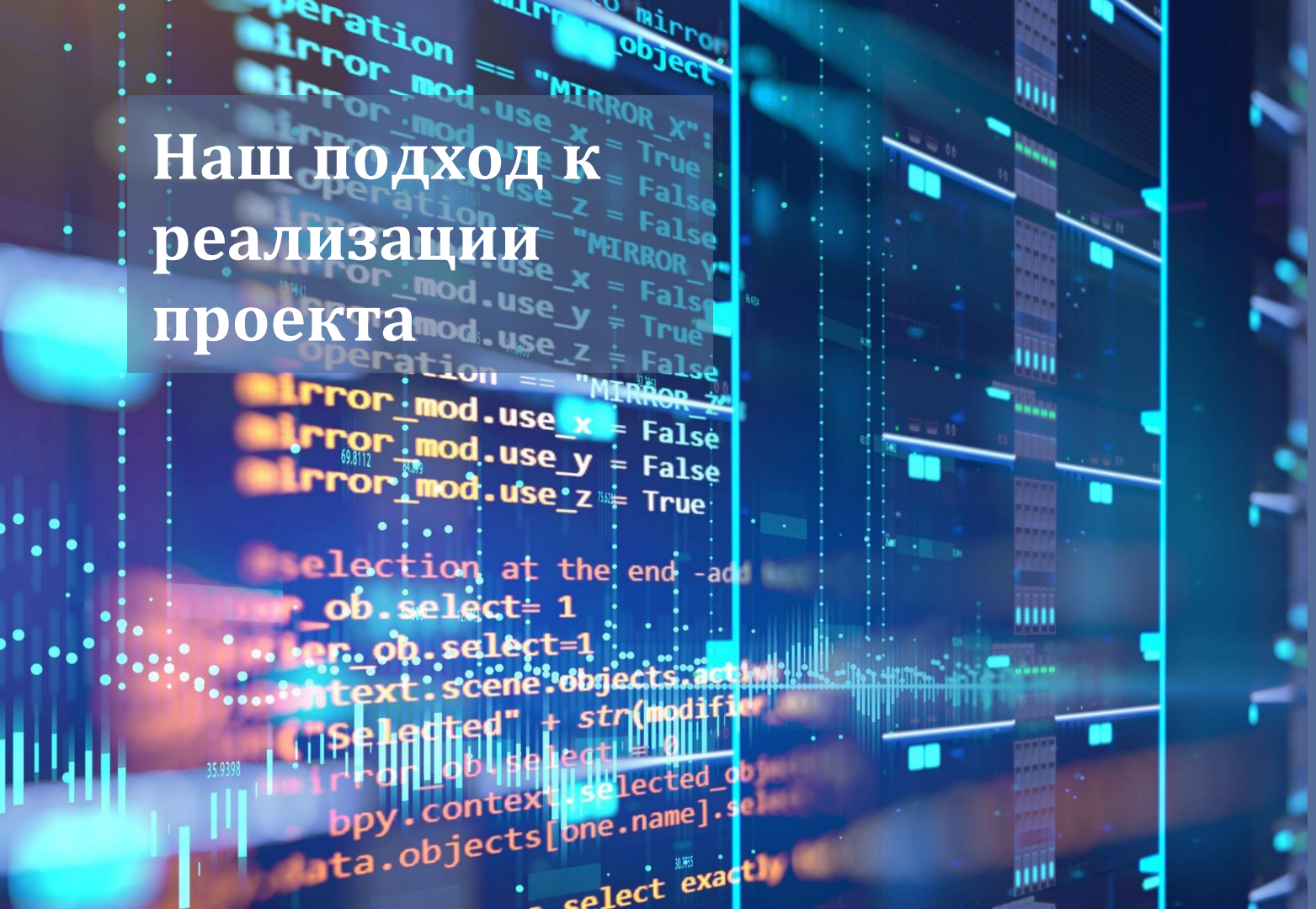
02 Инструменты безопасной разработки DevSecOps 19

03 Услуга SecOps 25

04 Наша команда 28

05 Проектный опыт 30

Наш подход к реализации проекта



01

Цель, предпосылки и задачи проекта

Цель проекта:

Провести комплексную оценку текущего состояния процессов безопасной разработки, спроектировать целевую модель на базе DevSecOps для осуществления перехода к целевым процессам и непрерывного улучшения уровня зрелости безопасной разработки.

Предпосылки проекта:

- Необходимость выявления текущих недостатков и областей для улучшения в процессах безопасной разработки.
- Потребность в формировании целевого видения и стратегии трансформации SDLC в направлении DevSecOps.
- Важность разработки комплекса организационно-распорядительной документации для регламентации процессов ИБ.
- Стремление к повышению уровня автоматизации безопасности и внедрению лучших инструментальных практик.
- Необходимость проектирования детальных воркфлоу безопасности для обеспечения прозрачности и контролируемости.
- Потребность в обучении и поддержке команд для эффективного перехода к целевым процессам DevSecOps.

Задачи проекта:

Аудит текущего состояния безопасной разработки
- Анализ документации, интервью, опросы команд
- Оценка зрелости процессов и уровня автоматизации ИБ

Формирование целевой модели безопасной разработки (DevSecOps)
- Определение принципов безопасности
- Проектирование референсной архитектуры и процессов SDLC+ИБ

Подготовка организационно-распорядительной документации
- Разработка политик, стандартов, регламентов ИБ
- Определение метрик и KPI для мониторинга прогресса

Проектирование детальных воркфлоу безопасности (DevSecOps)
- Интеграция инструментов ИБ в CI/CD конвейер
- Разработка процедур и чек-листов для ручных проверок

Внедрение инструментальных средств безопасности (SAST, DAST, SCA)
- Выбор и интеграция утилит ИБ в конвейер разработки
- Настройка политик и гейтов качества кода

Обучение и поддержка команд при переходе на DevSecOps
- Тренинги, менторинг, обмен знаниями
- Непрерывная коммуникация между разработкой, ИБ, эксплуатацией

Анализ результатов и подготовка отчета с рекомендациями
- Формализация текущего и целевого состояния
- План перехода к целевой модели (этапы, метрики, ответственные)

Технологическая практика

Ключевые практики

Смежные практики

Наше позиционирование:

Мы является отличным кандидатом для выполнения проектов по аудиту кибербезопасности бизнес-приложений в составе КИИ.

Мы обладаем уникальным опытом в сфере анализа защищенности отраслевых информационных систем, выявления угроз и выработки рекомендаций по повышению уровня ИБ. Наши эксперты глубоко понимают специфику промышленной автоматизации и соответствующие киберриски.

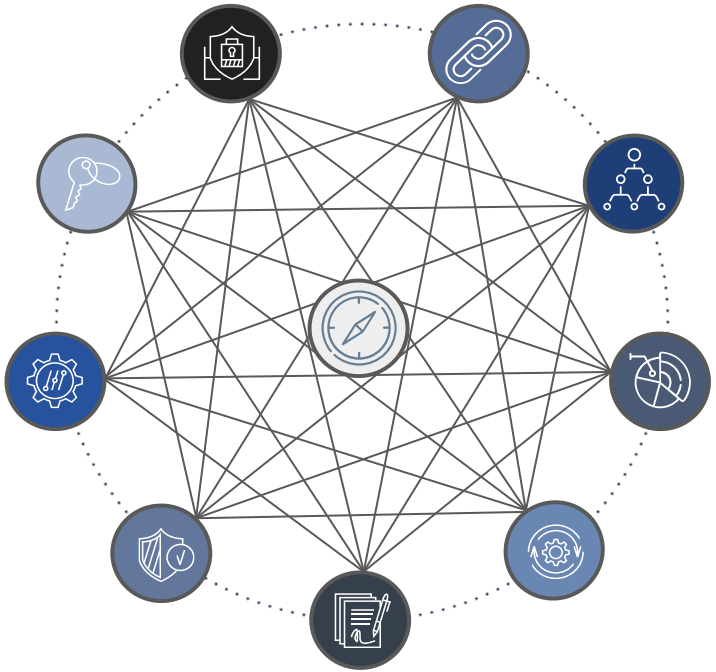
Ключевыми преимуществами нашей компании являются:

- Богатый опыт аудитов ИБ аналогичных КИИ;
- Владение передовыми практиками оценки киберзащищенности;
- Наличие специалистов по безопасности ИТ/ОТ систем;
- Комплексный подход, охватывающий технические и организационные аспекты ИБ;
- Предоставление end2end услуг и выполнение роли доверенного партнёра в технологиях и digital за счёт наличия широкого спектра компетенций;
- Обеспечение полного охвата потребностей заказчика на протяжении всего жизненного цикла технологических решений.

Выбрав нас, вы получите **максимально полную и объективную оценку** текущего состояния защищенности вашего бизнес-приложения **и рекомендации** по ее повышению от лидеров рынка кибербезопасности.

Фреймворк информационной безопасности

Собственная методология XimiLab предполагает разделение функции ИБ на **10 процессов**, каждый из которых объединяет под собой связанные подпроцессы. Данный фреймворк является симбиозом лучших мировых практик: ISO 27001, NIST CSF, CIS18



1 Управление угрозами и инцидентами ИБ

- Мониторинг событий ИБ
- Управление источниками событий ИБ
- Управление исключениями ИБ
- Управление инцидентами ИБ
- Мониторинг информации о внешних угрозах

2 Управление доступом

- Управление доступом пользователей
- Управление доступом привилегированных пользователей (администраторов)
- Управление сетевыми доступами
- Разграничение полномочий (SoD)
- Управление доступом на уровне данных (большие данные)

3 Контроль защищенности ИТ-инфраструктуры

- Управление техническими уязвимостями ИБ
- Безопасная конфигурация компонентов ИТ-инфраструктуры
- Управление обновлениями безопасности

4 Защита конфиденциальной информации

- Инвентаризация и классификация конфиденциальной информации
- Противодействие утечкам конфиденциальной информации
- Управление цифровыми правами (DRM)
- Защита информации при использовании портативных устройств хранения информации

5 Управление электронными подписями и криптография

- Управление деятельностью УЦ
- Управление криптографическими ключами и инфраструктурой открытых ключей

6 Обеспечение ИБ на стадиях жизненного цикла ИС

- Определение и контроль требований ИБ на стадиях жизненного цикла ИС
- Анализ защищенности ИС на стадии разработки ПО

7 Управление техническими мерами обеспечения ИБ

- Разработка архитектуры ИБ
- Эксплуатация систем ИБ
- Безопасность конечных узлов
- Безопасность корпоративной сети и защита сетевого взаимодействия
- Антивирусная защита
- Безопасность при использовании средств электронных коммуникаций
- Безопасность при осуществлении удаленного доступа
- Защита web- и мобильных приложений

8 Обеспечение ИБ при работе с персоналом

- ИБ при работе с персоналом
- Обучение и повышение осведомленности в области ИБ

9 Управление рисками ИБ

- Оценка воздействия на бизнес (BIRT)
- Идентификация и оценка рисков ИБ
- Выбор инициатив по управлению рисками ИБ
- Инвентаризация и учет физических активов

10 Управление ИБ

- Управление стратегией и операционной моделью ИБ
- Оценка эффективности процессов ИБ
- Формирование отчетности по ИБ
- Управление нормативной документацией ИБ
- Информационная безопасность при работе с контрагентами
- Внутренний и внешний аудит ИБ
- Управление проектами ИБ
- Выполнение нормативных требований в области ИБ
- Управление ИБ в дочерних компаниях

DevSecOps

Подход к разработке ПО, который **автоматизирует процессы** разработки, тестирования, развертывания и поддержки приложений, **обеспечивая** при этом **выполнение требований по безопасности и соответствию стандартам**

Планирование

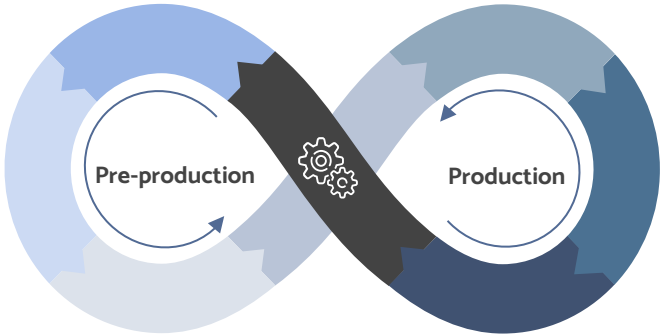
Обсуждение, проверка и разработка стратегии анализа безопасности

Развертывание

Управление доступом и конфигурациями, pen testing

Сборка

Проверка кода перед публикацией, анализ состава ПО, SAST, статический анализ безопасности приложений, анализ кода, безопасность контейнеров, сканирование на уязвимости, DAST



Эксплуатация

Сбор логов, RASP (Самозащита приложений во время выполнения),установка исправлений, WAF

Тестирование

DAST

Мониторинг

SIEM, мониторинг уязвимостей, контроль доступа

Прежде всего **необходимо провести комплексный аудит** текущих процессов разработки, тестирования и эксплуатации ПО, позволяющий выявить узкие места, недостатки и угрозы в существующих ИТ-процессах, определить приоритетные направления для внедрения практик DevSecOps

Преимущества для организаций:

- ✓ **Сокращение расходов** на обеспечение безопасности приложений за счёт автоматизации;
- ✓ Более **раннее выявление уязвимостей** и устранение дефектов безопасности;
- ✓ **Ускорение вывода приложений на рынок** благодаря непрерывной интеграции и доставке;
- ✓ **Повышение качества приложений** как с точки зрения бизнес-функций, так и безопасности;
- ✓ **Формирование культуры DevSecOps** в компании, где каждый разработчик несёт ответственность за безопасность;
- ✓ **Повышение уровня зрелости процессов разработки ПО** и управления информационной безопасностью;
- ✓ Возможность **непрерывного мониторинга и аудита защищённости** приложений

Подход к внедрению SDLC

- Формирование команды по внедрению SDLC и назначение ответственных лиц
- Анализ текущего состояния процессов разработки ПО в организации с выявлением разрывов
- Определение целей и ключевых показателей эффективности (KPI) внедрения SDLC
- Разработка плана проекта внедрения SDLC с учетом особенностей организации
- Утверждение бюджета и сроков реализации проекта.

- Разработка Политики безопасной разработки ПО в соответствии с требованиями
- Создание стандартов и процедур SDLC, включая:
  - Процедуры безопасного проектирования архитектуры.
  - Правила безопасного кодирования.
  - Процедуры тестирования безопасности (SAST, DAST, SCA).
  - Управление уязвимостями и инцидентами безопасности.
  - Защита сред разработки, тестирования и продуктива.
- Разработка сценариев реагирования на инциденты и восстановления систем
- Утверждение ЛНА и их распространение среди сотрудников

- Определение потребностей в инструментах с учетом используемого стека технологий
- Внедрение инструментов SAST/DAST (например, SonarQube, OWASP ZAP) в CI/CD пайплайн
- Внедрение инструментов SCA (например, Dependency Check, Snyk) для проверки библиотек
- Настройка инструментов для фиксации и анализа событий информационной безопасности
- Обучение разработчиков принципам SDLC и работе с инструментами
- Обучение специалистов по качеству (QA) методам тест-дизайна и тестированию

- Выбор пилотных проектов для проверки внедренного SDLC.
- Аудит безопасности пилотных проектов на всех этапах SDLC.
- Анализ результатов пилотирования с выявлением недостатков.
- Корректировка локальных нормативных актов, процедур и настроек инструментов на основе результатов пилотирования

- Утверждение и закрепление ролей и обязанностей участников SDLC.
- Запуск полномасштабного внедрения SDLC на всех проектах разработки.
- Организация периодического аудита процессов SDLC и их эффективности.
- Проведение регулярного обучения и повышения осведомленности персонала.
- Организация процесса непрерывного совершенствования SDLC на основе результатов

Подход к проведению бенчмаркинга (1/2)

Этапность работ

1 этап — **Диагностика и бенчмаркинг** уровня зрелости Компании в области ИБ в сравнении с лучшими практиками

2 этап — **Анализ** защищенности периметра и оценка используемых средств защиты

3 этап — **Определение** направлений развития функции ИБ

Перечень работ

- Сбор и обработка информации о текущем состоянии процессов, политик и процедур ИБ, о существующих подходах к управлению ИБ и ИТ
- Анализ требований законодательства в области ИБ
- Оценка зрелости процессов ИБ в ИТ/ОТ областях
- Изучение планов и текущих подходов к обеспечению непрерывности бизнеса
- Изучение точек взаимодействия, коммуникаций и со структурами и организациями из группы компаний Заказчика по направлению ИБ
- Анализ текущей организационной структуры ИБ
- Проведение бенчмаркинга численности подразделения ИБ

- Сбор и обработка информации о текущем состоянии архитектуры ИБ
- Проведение бенчмаркинга ландшафта технологий ИБ
- Проведение OSINT-сканирования и анализа защищенности периметра сети Компании, доступного из сети Интернет, и проведение оценки уязвимостей оборудования ИТ / ОТ в целях определения критических уязвимостей / устаревших патчей и т.д., требующих оперативного исправления

- Формирование предложений о направлениях развития функции информационной безопасности
- Определение быстрых улучшений в процессах ИБ, операционной модели, архитектуре ИБ
- Формирование дорожной карты по улучшениям и инициативам ИБ
- Проведение семинара с заинтересованными сторонами по результатам проекта

Ключевые результаты

Отчетные материалы, содержащие:

- Описание текущего состояния процессов и организационно-функциональной структуры компании
- Результаты бенчмаркинга уровня зрелости с лучшими практиками
- Рекомендации по потенциальным направлениям развития процессов ИБ

Отчетные материалы, содержащие:

- Текущая карта технологий ИБ
- Отчёт по результатам тестирования на проникновение с рекомендациями по повышению уровня защищённости (описание сценариев компрометации, описание выявленных уязвимостей, заключение об уровне защищенности внешнего периметра сети)

Отчетные материалы, содержащие:

- Предложения по быстрым улучшениям
- Дорожная карта инициатив по реализации рекомендаций с расчетом финансовых расходов и требуемых FTE

Подход к проведению бенчмаркинга(2/2)

Команда проекта будет фокусироваться на компаниях из интересующего сектора

BSIMM

BSIMM создан на **основе анализа конкретных практик безопасности**, применяемых в различных компаниях, предоставляя фактическую модель для бенчмаркинга, а не просто теоретические рекомендации.

Practice	Level_number	Earth	Isv	Fin	Tech	Cloud
SM	1_1	91	29	26	23	20
SM	1_3					
SM	1_4					

ИССЛЕДОВАНИЕ					37,1	15,5
Активность	Наименование практики	Описание практики	Ретейл	% Доля фирм применяющих практику от 7	Общий процент по зрелости(Bsim)	Процент зрелости по практикам
Модели атак					21,0	20,0
AM1.2	Создайте схему классификации данных и инвентаризацию.	Заинтересованные	6	86		
AM1.3	Выявляйте потенциальных злоумышленников.	SSG выявляет потен	2	29		
AM1.5	Собирайте и используйте разведданные об атаке.	SSG гарантирует, чт	3	43	52	50

#	Направление ИТ	Текущая оценка	Целевая оценка	Отклонение
1	Стратегия и метрики	2.8	4.1	-1.39
2	Соответствие регуляторам	6.2	4.0	2.19
3	Обучение	3.1	3.3	-0.20
4	Модели атак	2.0	2.1	-0.10
5	Механизмы безопасности и дизайн	2.2	4.5	-2.30
6	Стандарты и требования	0.4	4.5	-4.11
7	Архитектура	0.0	1.9	-1.94
8	Анализ кода	0.3	2.5	-2.17
9	Тестирование защищенности	2.2	1.9	0.32

Исходя из уникальности регуляторных требований, специфики стран, где проводится исследования, **не все** стандартные **практики могут быть применимы** в рамках конкретного бизнеса. Опираясь на наш **релевантный опыт** проведения аналогичных исследований на территории России мы применяем гибридный подход к оценке. Данный подход позволяет нам гибко подходить к анализу, учитывая уникальные особенности бизнеса и российского рынка.

Сопоставление практики BSIMM и требований ГОСТ 57580.1, ЦБ 683-П, 757-П (1/2)

В управляющих документах ГОСТ 57580.1, ЦБ 683-П, 757-П содержатся следующие основные положения, относящиеся к безопасной разработке:

1

ГОСТ 57580.1-2017 «Безопасность финансовых (банковских) операций. Защита информации финансовых организаций. Базовый состав организационных и технических мер»

- П. 7.11.3-7.11.5 - о безопасной разработке прикладного ПО, включая требования к контролю исходного кода на наличие недекларированных возможностей, соблюдение процедур тестирования и внесения изменений.
- П. 7.11.6 - о разработке безопасного и отказоустойчивого ПО для систем управления базами данных.

2

Положение Банка России 683-П "О требованиях к системе управления операционным риском в кредитной организации и банковской группе"

- П. 8.6-8.9 декларируют требования к разработке, приобретению и сопровождению ПО, включая анализ уязвимостей и их устранение, регулярное тестирование безопасности, контроль доступа.

3

Положение Банка России 757-П "О требованиях к кредитным организациям по обеспечению защиты информации при осуществлении банковской деятельности..."

- П 2.6.5 устанавливает требования безопасной разработки, включая анализ архитектуры, моделирование угроз, статический анализ кода, тестирование проникновения
- П. 2.6.6 требует сбора информации об уязвимостях используемого ПО и их своевременного устранения
- П. 2.9.1 касается разработки планов восстановления работоспособности ПО и тестирования этих планов.

Указанные документы устанавливают базовые принципы безопасной разработки, такие как контроль кода, регулярное тестирование на уязвимости, управление изменениями, обеспечение отказоустойчивости и планирование восстановления работы ПО, что призвано снизить риски для финансовых организаций, связанные с использованием небезопасного или некачественного программного обеспечения.

Сопоставление практики BSIMM и требований ГОСТ 57580.1, ЦБ 683-П, 757-П (2/2)

Стратегия (Strategy & Metrics):
Соответствует требованиям к управлению рисками П. 8.6 **(683-П)**.
Compliance & Policy:
Отражено в требованиях к документированию Процедур в П. 7.11.4 **(ГОСТ 57580.1)**, П. 8.7 **(683-П)**
Обучение (Training):
Прямо не упомянуто в управляющих документах

Анализ архитектуры (Architecture Analysis):
требуется в П. 8.6 **(683-П)**, П 2.6.5 **(757-П)**
Анализ кода (Code Review):
упомянут в П. 7.11.3 **(ГОСТ 57580.1)**, П. 8.6 **(683-П)**, П 2.6.5 **(757-П)**
Тестирование безопасности (Security Testing):
требуется в П. 7.11.5 **(ГОСТ 57580.1)**, П. 8.6 **(683-П)**, П 2.6.5 **(757-П)**

Моделирование угроз (Attack Models):
требуется в П 2.6.5 **(757-П)**.
Анализ безопасности (Security Features & Design):
упоминается в П. 8.6 **(683-П)**, П 2.6.5 **(757-П)**.
Стандарты и требования (Standards & Requirements):
Отражены в общих требованиях документов к безопасной разработке

Управление уязвимостями (Vulnerability Management):
отражено в П. 8.8 **(683-П)**, П. 2.6.6 **(757-П)**
Конфигурация и гигиена (Configuration & Hygiene):
частично отражено в П. 2.9.1 **(757-П** (планы восстановления)
Управление средой (Environment Management):
Прямо не упомянуто в управляющих документах

Многие практики BSIMM находят отражение в рассматриваемых управляющих документах - в первую очередь, это практики по моделированию угроз, анализу архитектуры и кода, тестированию безопасности и управлению уязвимостями. В то же время некоторые практики BSIMM, такие как обучение разработчиков или управление средой разработки, явно в них не упомянуты

ГОСТ 57580.1, 683-П и 757-П **устанавливают базовые требования** в области безопасной разработки, в то время как **BSIMM предлагает более широкий и детальный набор практик**

Методология

BSIMM

Для каждой из 122 видов активностей BSIMM **осуществляется проверка**, выполняет ли организация аналогичную инициативу.

Доказательства собираются путем **изучения документации, опроса специалистов и изучения артефактов**. Описания, процедуры, инструменты, задействованный персонал и результаты документируются

Аудитор присваивает оценку модели зрелости возможностей **от 0 до 2** на основе полученных доказательств

GOVERNANCE			INTELLIGENCE			SSDL TOUCHPOINTS			DEPLOYMENT		
Активности	BSIMM Бенчмак (Ретейл)	Компания	Активности	BSIMM Бенчмак (Ретейл)	Компания	Активности	BSIMM Бенчмак (Ретейл)	Компания	Активности	BSIMM Бенчмак (Ретейл)	Компания
СТРАТЕГИЯ И МЕТРИКИ			МОДЕЛИ АТАК			АРХИТЕКТУРА			ТЕСТИРОВАНИЕ НА ПРОНИКНОВЕНИЕ		
SM1.1	71	1	AM1.2	86	1	AA1.1	86	0	PT1.1	86	2
SM1.3	57	1	AM1.3	29	0	AA1.2	14	0	PT1.2	100	1
SM1.4	86	1	AM1.5	43	2	AA1.3	14	0	PT1.3	100	1
SM2.1	57	0	AM2.1	14	0	AA1.4	100	-	PT2.2	14	-
SM2.2	29	-	AM2.2	0	0	AA2.1	0	-	PT2.3	29	0
SM2.3	43	1	AM2.5	14	0	AA2.2	0	0	PT3.1	29	-
SM2.6	43	0	AM2.6	0	1	AA3.1	0	-	PT3.2	0	-
SM2.7	57	0	AM2.7	0	0	AA3.2	14	-			
SM3.1	14	0	AM3.1	14	-	AA3.3	0	-			
SM3.2	14	-	AM3.2	0	-						
SM3.3	0	2	AM3.3	0	0						
SM3.4	0	0									

СООТВЕТСТВИЕ РЕГУЛЯТОРАМ			МЕХАНИЗМЫ БЕЗОПАСНОСТИ И ДИЗАЙН			АНАЛИЗ КОДА			СРЕДА ЭКСПЛУАТАЦИИ		
CP1.1	43	2	SFD1.1	100	-	CR1.2	57	0	SE1.1	57	2
CP1.2	100	2	SFD1.2	71	2	CR1.4	86	0	SE1.2	86	2
CP1.3	71	2	SFD2.1	14	0	CR1.5	29	0	SE2.2	14	0
CP2.1	57	0	SFD2.2	57	-	CR1.6	57	0	SE2.4	14	0
CP2.2	29	-	SFD3.1	14	1	CR1.7	57	1	SE2.5	43	1
CP2.3	29	0	SFD3.2	14	-	CR2.6	14	0	SE2.6	14	1
CP2.4	57	1	SFD3.3	14	0	CR2.7	14	0	SE2.7	14	2
CP2.5	29	1				CR3.2	14	-	SE3.2	14	0
CP3.1	14	2				CR3.3	0	-	SE3.3	14	2
CP3.2	14	1				CR3.4	0	0	SE3.6	0	0
CP3.3	0	1				CR3.5	0	0			

ОБУЧЕНИЕ			СТАНДАРТЫ И ТРЕБОВАНИЯ			ТЕСТИРОВАНИЕ ЗАЩИЩЕННОСТИ			УПРАВЛЕНИЕ КОНФИГУРАЦИЕЙ И УЯЗВИМОСТЯМИ		
T1.1	71	1	SR1.1	86	0	ST1.1	57	2	CMVM1.1	100	-
T1.7	57	1	SR1.2	71	0	ST1.3	43	2	CMVM1.2	100	1
T1.8	29	0	SR1.3	86	-	ST1.4	43	0	CMVM2.1	100	2
T2.5	29	-	SR2.2	57	-	ST2.4	14	0	CMVM2.2	86	-
T2.8	29	0	SR2.4	29	0	ST2.5	14	0	CMVM2.3	43	1
T2.9	57	1	SR2.5	57	-	ST2.6	0	0	CMVM3.1	0	-
T3.1	0	2	SR3.1	14	0	ST3.3	0	0	CMVM3.2	0	0
T3.2	14	-	SR3.2	0	0	ST3.4	0	-	CMVM3.3	14	0
T3.3	0	1	SR3.3	0	1	ST3.5	0	0	CMVM3.4	14	-
T3.4	14	0	SR3.4	14	0	ST3.6	0	-	CMVM3.5	0	-
T3.5	14	2							CMVM3.6	0	-
T3.6	0	-							CMVM3.7	0	-

BENCH MARKING

Результаты анализа визуализируются с использованием **10-бальной системы** оценки. Эта система предоставляет ясное представление о текущем состоянии безопасности разработки, где 10 баллов означают, что практики безопасности полностью внедрены и эффективно работают, а 0 баллов указывают на полное отсутствие или необходимость во внедрении практик или их доработку

Для оценки соответствия текущего уровня зрелости практик SSDLC* в компании применяется отраслевой стандарт BSIMM

**SSDLC (Secure Software Development Life Cycle) - это процесс, который включает в себя интегрированные меры безопасности на каждом этапе разработки программного обеспечения.*

Сбор данных

достигается путем проведения интервью с заинтересованными сторонами, такими как специалисты по информационной безопасности, разработчики, менеджеры и другие. Также анализируется документация и артефакты, относящиеся к практикам безопасной разработки ПО в компании.

Сопоставление инициатив с BSIMM

собранные данные сопоставляются со 122 мероприятиями по обеспечению безопасности ПО, определенными в рамках BSIMM, в контексте 12 практик безопасной разработки ПО. Каждая активность оценивается по уровню зрелости в соответствии со специальной матрицей BSIMM

Определение профиля зрелости

на основе полученных баллов определяется профиль зрелости практик безопасности ПО в организации. Это позволяет определить текущую стадию зрелости компании в данной области.

Сравнение с аналогами

профиль зрелости организации сопоставляется с профилями других организаций из базы данных BSIMM. Это позволяет определить текущий уровень развития практик безопасности ПО в компании относительно среднеотраслевых показателей.

Выявление недостаточно развитых практик

путем сравнения профиля зрелости с отраслевыми данными выявляются пробелы в текущих практиках безопасной разработки ПО.

Предоставление рекомендаций

даются конкретные рекомендации по мероприятиям, которые организация могла бы реализовать для повышения зрелости своих практик безопасности ПО и устранения выявленных пробелов

Комплексный подход к обеспечению безопасности в DevOps

01

Аудит процессов DevSecOps

- Анализ текущих практик и процессов безопасности в рамках DevOps
- Выявление потенциальных рисков и областей для улучшения
- Разработка рекомендаций по внедрению лучших практик DevSecOps

02

Внедрение/разработка/поддержка средств автоматизации DevSecOps

- SAST (Static Application Security Testing) –тестирование приложения на наличие ошибок и уязвимостей в исходном коде с применением статического анализа
- SCA (Software Composition Analysis) - анализ сторонних компонентов ПО
- DAST (Dynamic Application Security Testing) – динамическое тестирование безопасности приложений. DAST-сканеры работают автоматически, имитируя внешние атаки на приложение
- VMP (Vulnerability Management) - управление уязвимостями

03

Предоставление сервиса XimiSecOps

- Интеграция безопасности в CI/CD: внедрение автоматизированных проверок безопасности в конвейер разработки
- Непрерывный мониторинг безопасности: регулярные проверки на уязвимости, соответствие требованиям после релиза
- Экспертиза и поддержка: команда подрядчика обеспечивает внедрение и сопровождение безопасной разработки

Примеры результатов работ

Результат анализа сравнения текущих совокупных показателей SSDLC с целевыми

ПРИМЕР

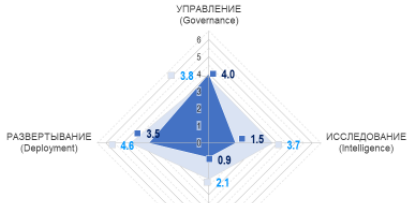
Применение специфического бенчмарка BSIMM для области ритейла позволило выявить аспекты безопасности разработки, в которых есть потребность в улучшении или модернизации.

УПРАВЛЕНИЕ (Governance)
Показатели безопасности **превосходят** средний уровень по отрасли, что свидетельствует **о наличии устойчивых и эффективных** механизмов управления безопасностью информации

ИССЛЕДОВАНИЕ (Intelligence)
Показатель **существенно уступает** отраслевому среднему, что указывает **на пробелы** в текущих системах исследования, мониторинга и аналитики потенциальных угроз

SSDL ВЗАИМОДЕЙСТВИЕ (SSDL Touchpoints)
Показатель **уступает** отраслевому среднему, что указывая **проблемы взаимодействия и интеграции практик безоп** различных этапах разработки программного обеспечени

РАЗВЕРТЫВАНИЕ (Deployment)
Показатель **уступает** отраслевому среднему, что указыва **необходимость доработки и усовершенствования** текущ



Результат анализа уровня зрелости показателей активностей SSDLC

ПРИМЕР

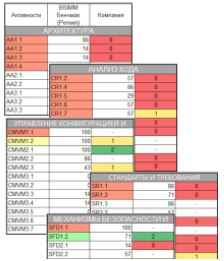
На основе проведенного анализа обнаружено, что некоторые ключевые аспекты в сфере безопасной разработки не достигают полного охвата первого уровня зрелости

Архитектура
Несмотря на важность обеспечения безопасности на этапе проектирования, текущие архитектурные решения требуют пересмотра и интеграции современных методов и практик SSDLC

Анализ кода
Эффективный анализ кода является ключевым элементом для выявления и устранения угроз на ранних этапах разработки. Необходимо внедрять передовые инструменты покрытия задач анализа OSA/SCA, DAST и методологии для этого процесса

Управление конфигурацией и уязвимостями
Четкое управление конфигурациями и устойчивое отслеживание уязвимостей требуют системного подхода, интегрированного с действующими рабочими процессами

Стандарты и требования
Для достижения первого уровня зрелости важно устанавливать и обновлять стандарты [illegible] вования и лучшие практики отрасли



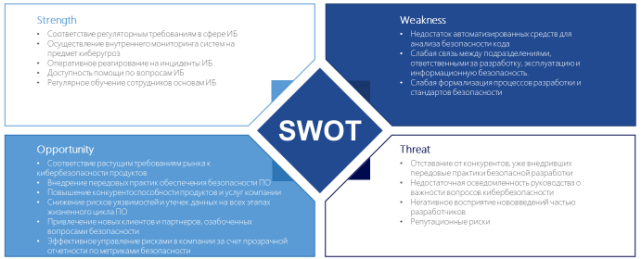
Результат анализа текущих показателей практик с целевыми

ПРИМЕР

Несмотря на определенные успехи в некоторых категориях, существуют **серьезные проблемы в области безопасности**, которые требуют **немедленного внимания и коррекции**. Для достижения уровня лучших практик рекомендуется провести **пересмотр** текущих подходов и внести **необходимые изменения** в процессы и методики.

SWOT-анализ

ПРИМЕР



Пример скоупа работ по обследованию процессов DevSecOps

ПРИМЕР

Общее описание работ	Сроки выполнения
Этап 1. Отчёт о текущем состоянии безопасной разработки (презентация + блок схема процессов)	**20 рабочих дней**
Запрос и анализ информации о текущем состоянии безопасной разработки. Интервью с владельцами процесса и анкетирование пользователей	
Оценка текущего уровня зрелости процессов безопасной разработки (оценка полноты описания процессов и проектных документаций)	
Оценка текущей обеспеченности процессов безопасной разработки средствами автоматизации	
Интервьюирование 2-3 команд разработки на предмет соответствия зарегламентированным процессам	
Сканирование 2-3 сервисов инструментами SAST + SCA	
Анализ текущего состояния, формализация процессов и подготовка отчёта	
Этап 2. Целевая схема процесса и архитектуры, дорожная карта достижения целевого состояния для реализации (презентация + блок схема процессов)	**25 рабочих дней**
Разработка целевого процессного подхода безопасной разработки (в т.ч. схема)	
Разработка целевой обеспеченности процесса безопасной разработки средствами автоматизации SAST; SCA; DAST; VMP (в т.ч. схема)	
Подготовка рекомендаций по команде DevSecOps с описанием трудозатрат по каждой роли	
Разработка дорожный карты перехода к парадигме Quality Gate	
Итого	**45 рабочих дней**

ПРИМЕР

День 1

Проектирование

Проектирование целевой архитектуры

День 13

Установка и настройка SAST

- Установка сканера SAST на выделенный сервер
- Настройка правил сканирования под требования проекта
- Интеграция SAST с CI/CD через plugins

День 25

Интеграция SAST с CI/CD

- Настройка запуска скана при каждом коммите
- Настройка нотификаций об уязвимостях

День 35

Установка и настройка SCA

- Установка сканера зависимостей для языка проекта
- Интеграция SCA с репозиториями кода
- Настройка блокировки pull request при критических уязвимостях

День 39

Интеграция SCA с CI/CD

- Ежедневный запуск сканера зависимостей
- Блокировка merge при обнаружении уязвимостей
- Настройка отчётности о статусе зависимостей

День 47

Установка и настройка ASOC

- Установка и настройка сервера Defect Dojo
- Интеграция Defect Dojo Создание моделей данных о приложениях
- Ролевая модель доступа пользователей
- Интеграция с процессами разработки и эксплуатации

День 55

Тестирование и ввод в эксплуатацию

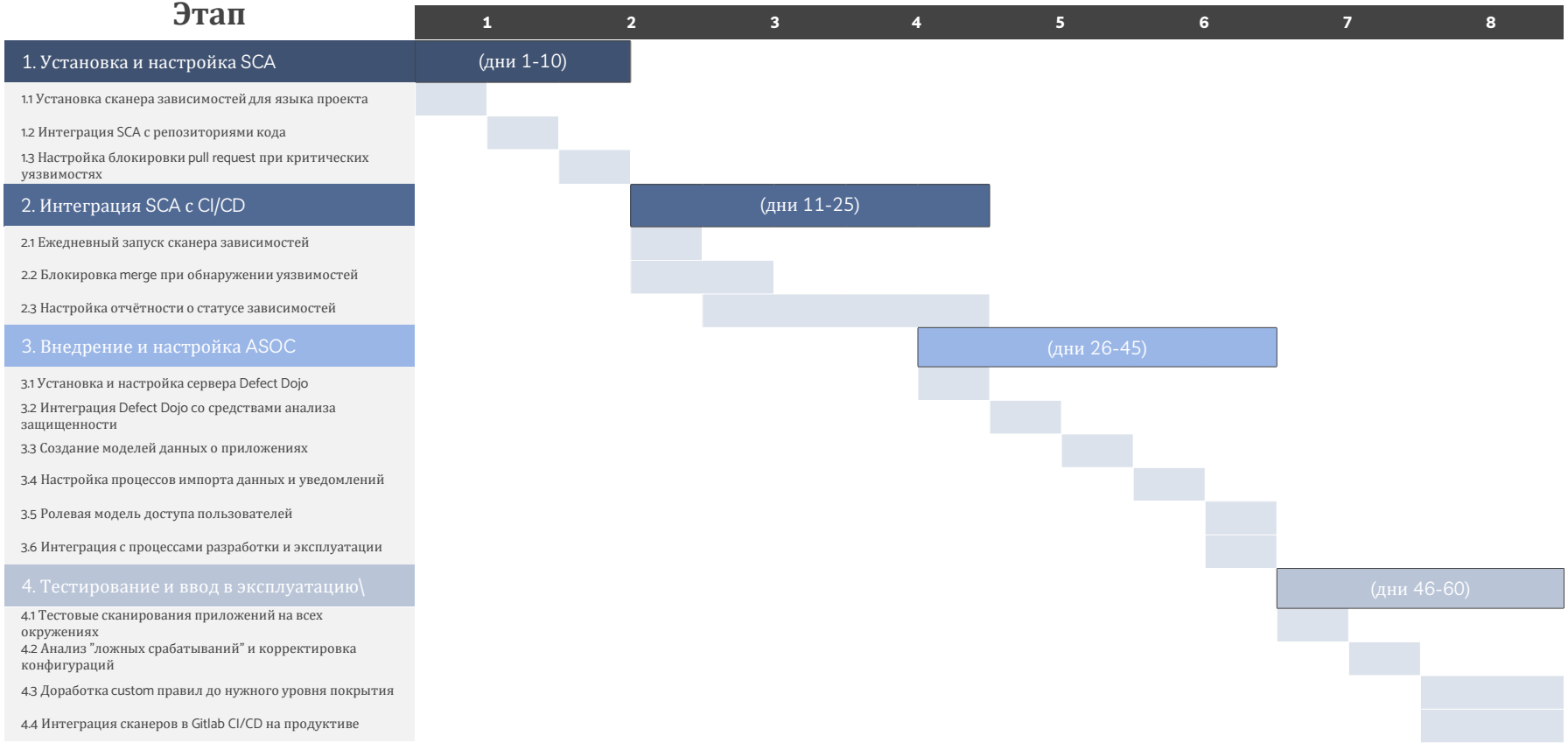
- Тестовые сканирования приложений на всех окружениях(ПМИ)
- Анализ ”ложных срабатываний” и корректировка конфигураций
- Доработка custom правил до нужного уровня покрытия
- Интеграция сканеров в Gitlab CI/CD на продуктиве

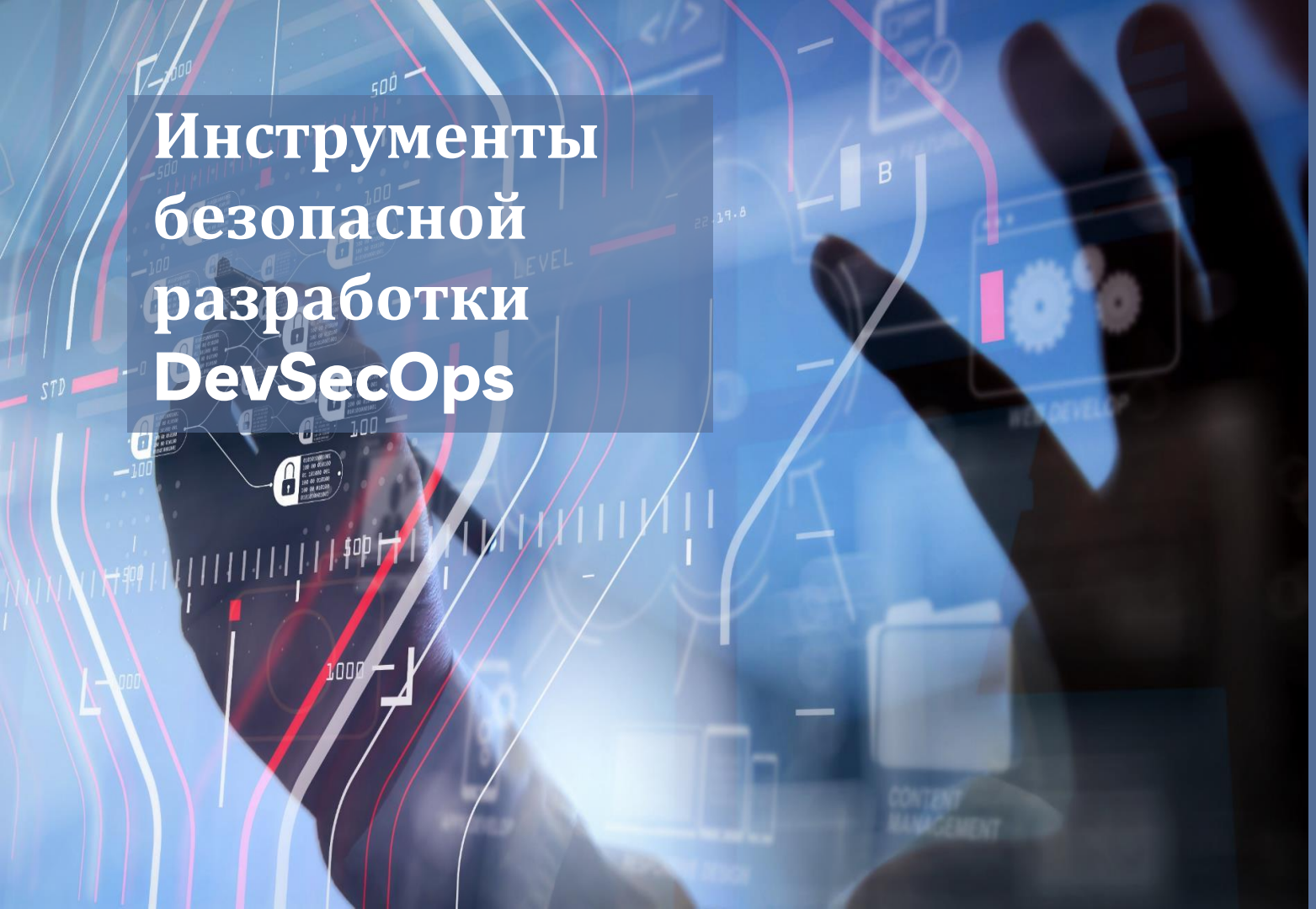
День 60

Сдача в продуктив

ПРИМЕР

Дорожная карта внедрения продуктов автоматизации





Инструменты безопасной разработки DevSecOps

02

Инструменты безопасной разработки

1. **Сканеры уязвимостей.** Анализируют работающие системы и приложения на предмет уязвимостей:
   - Tenable Nessus - один из самых популярных vulnerability scanner'ов;
   - OpenVAS - альтернатива Nessus со схожим функционалом;
   - Qualys VM - коммерческий сканер уязвимостей с большими возможностями настройки проверок;
   - XSpider - сканер безопасности веб-приложений от российской Positive technology;
   - Prisma Cloud - облачный WAF и сканер уязвимостей инфраструктуры.

2. **Фаззеры.** Генерируют неверные, неожиданные данные для поиска уязвимостей:
   - AFL - популярный фаззер с открытым исходным кодом;
   - OWASP ZAP - инструмент DAST с встроенным фаззером веб-приложений;
   - boofuzz - фреймворк фаззинга с открытым кодом.

3. **Системы DAST.** Автоматизируют аудит безопасности веб-приложений:
   - Burp Suite - интегрированная платформа для автоматизированного и ручного анализа безопасности;
   - OWASP ZAP - сканер уязвимостей веб-приложений с открытым кодом;
   - SQLmap - инструмент для обнаружения уязвимостей SQL инъекций;
   - PT BlackBox - on-premise-сканер DAST, ориентированный на поиск уязвимостей методом черного ящика

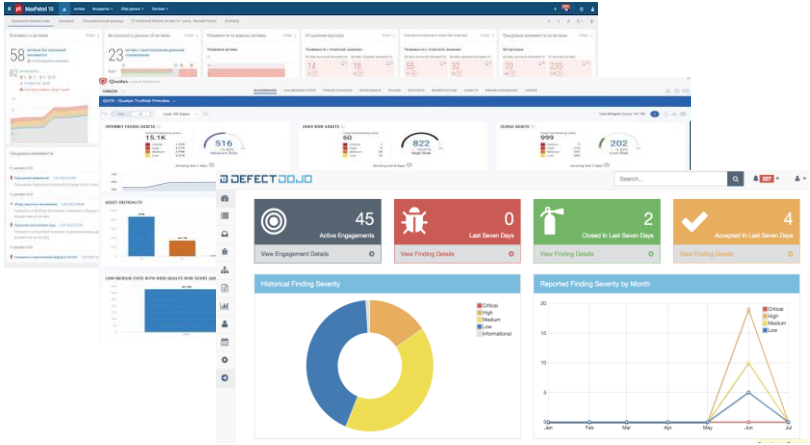
4. **Инструменты SAST.** Анализируют исходный код на уязвимости без запуска ПО.
   - Checkmarx - коммерческое решение SAST. Имеет точный и глубокий анализ уязвимостей;
   - Veracode - решение SAST в форме услуги. Имеет возможности глубокого сканирования без ложных срабатываний;
   - Synopsys - набор коммерческих инструментов анализа кода, в том числе SAST;
   - SonarQube - сканер уязвимостей исходного кода от российской С-Терра СиЭсПИ.

5. **SCA инструменты.** Проверяют соответствие исходного кода и зависимостей корпоративным политикам и лицензионным соглашениям.
   - Black Duck - коммерческое решение для аудита открытого ПО;
   - Veracode - предоставляет возможности SCA помимо SAST;
   - Synopsys - решения для различных видов анализа кода, включая SCA;
   - Линукс Просканер - сканер лицензий открытого ПО от российской Астерос.

VMP (Vulnerability Management) - управление уязвимостями

Комплексная автоматизированная платформа, объединяющая процессы сбора данных об уязвимостях, оценки рисков, приоритизации и контроля устранения угроз, формирования отчётности о состоянии защищённости систем с целью оптимизации расходов на кибербезопасность

1. Консолидирует информацию из анализаторов кода
2. Группирует проекты по сервисам
3. Гибкие варианты интеграции
4. Прозрачная отчётность по уязвимостям



Для эффективного управления киберрисками **требуется создание специализированных аналитических центров (АСОК)**, которые бы осуществляли сбор данных об уязвимостях из разрозненных источников, их агрегацию и анализ, разработку моделей киберугроз, оценку возможного ущерба, выработку оптимальных решений по приоритизации и митигации наиболее опасных рисков. Это **необходимо для регулярного информирования руководства** организаций о текущем состоянии защищённости критически важных информационных активов и возможных векторах кибератак, чтобы своевременно принимать стратегические решения в области управления ИБ и развития киберзащиты.

Appmetria – платформа управления уязвимостями

Комплексная платформа управления уязвимостями, разработанная собственными силами организации, позволяющая автоматизировать и объединить ключевые процессы обеспечения безопасности на всех этапах жизненного цикла разработки ПО

Консолидирует информацию из анализаторов кода

Группирует проекты по сервисам/ИС

Открытый API
Гибкие варианты интеграции

Прозрачная отчётность по уязвимостям

Гибкая настройка SLA и Quality Gate

Управление ЖЦ и визуализация уязвимостей

Высокая производительность

Мультиканальная система оповещений и совместной работы

Создание единого центра управления уязвимостями:

- Централизованное хранение и обработку данных об уязвимостях из различных источников
- Возможность категоризации и группировки уязвимостей по информационным системам (ИС) для эффективного анализа и приоритизации
- Единую точку принятия решений и оценки рисков, связанных с уязвимостями

Предоставление удобной и информативной отчетности для анализа уязвимостей:

- Отчеты по критичности уязвимостей для определения приоритетов устранения
- Группировка уязвимостей по типам и категориям для выявления системных проблем и трендов
- Динамика обнаружения и устранения уязвимостей для контроля прогресса и эффективности процессов безопасности

Гибкие возможности по построению Quality Gate на основе данных об уязвимостях:

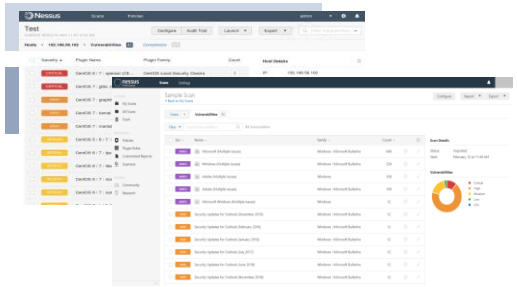
- Определение критериев качества и безопасности для продвижения релизов по конвейеру разработки
- Автоматическая проверка соответствия критериям качества на основе результатов сканирования уязвимостей
- Блокировка релизов, не удовлетворяющих требованиям безопасности, для предотвращения внедрения уязвимого кода

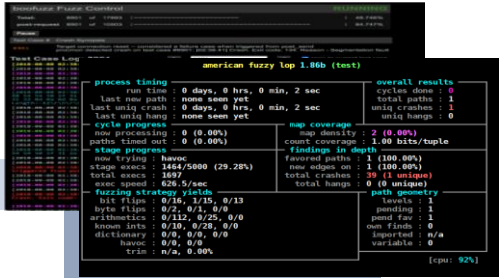
Платформа **Ametria**, позволяет организации выстроить эффективные процессы управления уязвимостями, обеспечить контроль безопасности на всех этапах разработки и получить наглядную картину состояния защищенности информационных систем.

Категории инструментов безопасной разработки (1/2)

1. Сканеры уязвимостей

Регулярное **сканирование на уязвимости** систем и приложений помогает **выявлять слабые места в корпоративной ИТ-инфраструктуре и контролировать ход работ по их устранению** для приведения в соответствие со стандартами информационной безопасности. Это, в свою очередь, значительно снижает вероятность успешных кибератак и несанкционированного доступа к конфиденциальным данным.



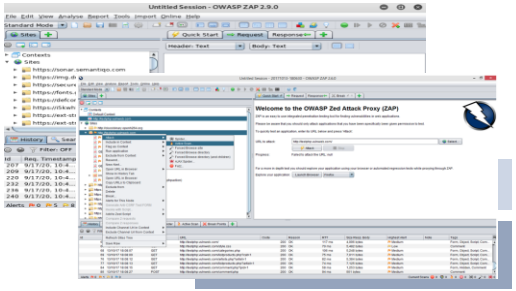


2. Фаззеры

Фаззеры позволяют находить неизвестные уязвимости в приложениях, **имитировать действия злоумышленников** для тестирования на проникновение, а также генерировать большие объемы некорректных данных **для проверки обработки приложениями непредвиденных ситуаций.**

3. Системы DAST

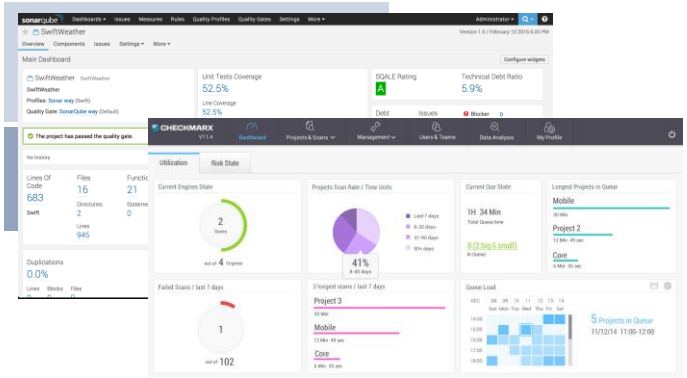
Системы DAST (Dynamic Application Security Testing) обеспечивают постоянный мониторинг безопасности веб-приложений в режиме реального времени с генерацией отчётов по найденным уязвимостям для их оперативного устранения и снижения рисков кибератак и утечек данных.

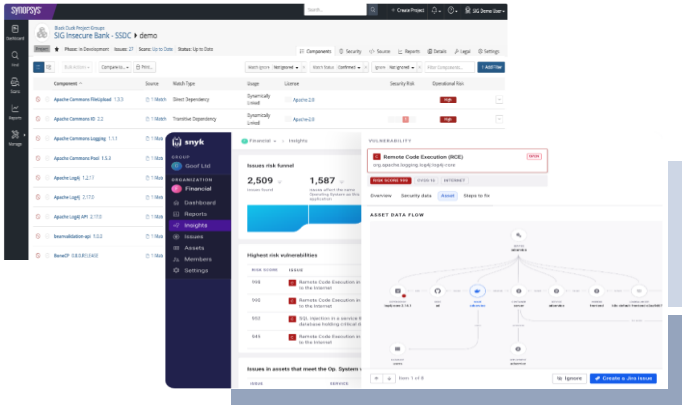


Категории инструментов безопасной разработки (2/2)

4. Инструменты SAST.

SAST (Static Application Security Testing) инструменты выявляют широкий спектр уязвимостей с высокой скоростью по сравнению с динамическим анализом, поддерживают основные языки программирования и позволяют снизить затраты на исправление дефектов ПО за счёт их обнаружения на ранних этапах разработки.





5. SCA инструменты

SCA (Software Composition Analysis) - это анализ сторонних компонентов программного обеспечения, направленный на выявление всех используемых библиотек, модулей и других зависимостей от третьих лиц. Основная цель SCA - обнаружение устаревших, неподдерживаемых или содержащих уязвимости элементов, что позволяет минимизировать риски для безопасности из-за известных уязвимостях в старых версиях библиотек, а также избежать проблем совместимости и вопросов лицензирования

SecOps

03

Сервис SecOps

При принятии решения о внедрении элементов DevSecOps важно **рационально подходить к вопросу и делать экономически взвешенный выбор** тех практик и инструментов, которые дадут наибольший эффект для конкретной организации.

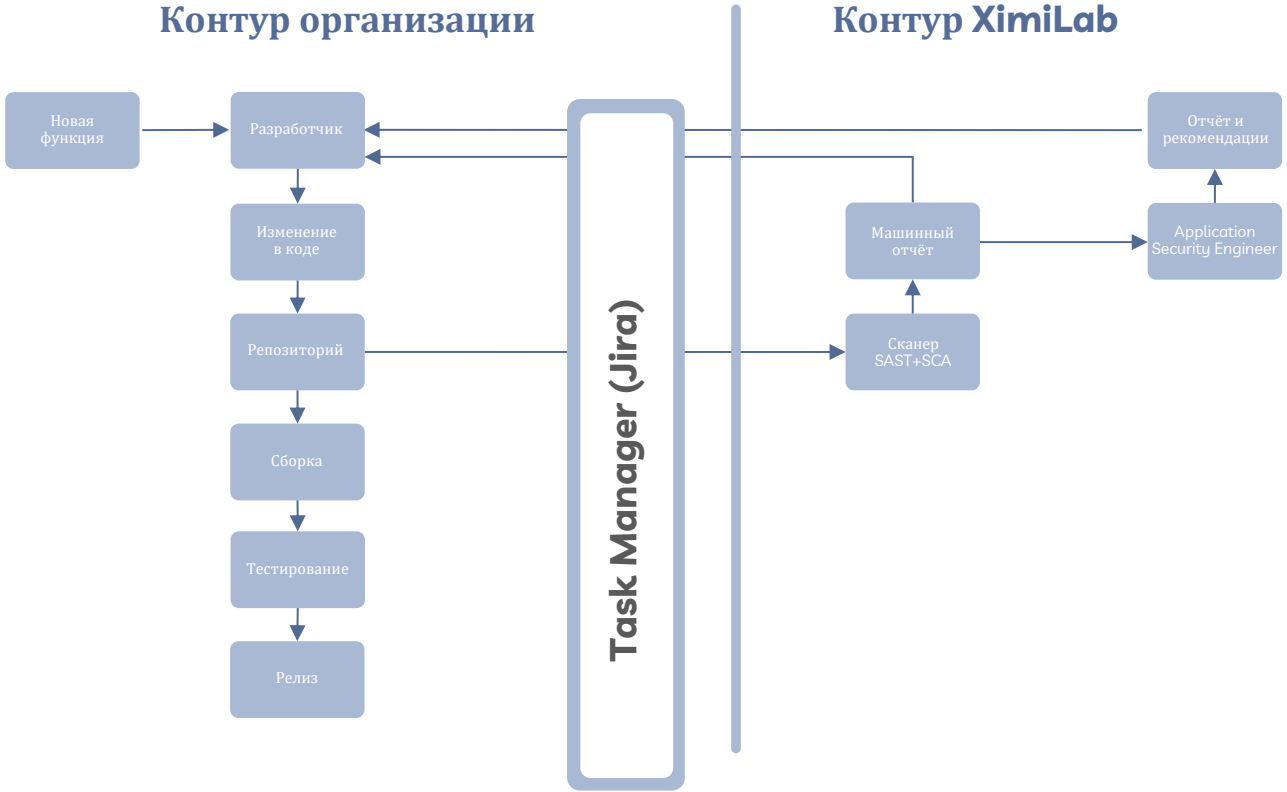
Целесообразность DevSecOps зависит от:

- **Критичности приложений** - потенциального ущерба в случае реализации ИБ угроз, уровня подлежащей защите конфиденциальности и ценностью данных
- **Требований комплаенса** - необходимости соответствия требованиям регуляторов, стандартов безопасности
- **Зрелости ИБ** и разработки текущих возможностей и эффективности процессов обеспечения ИБ

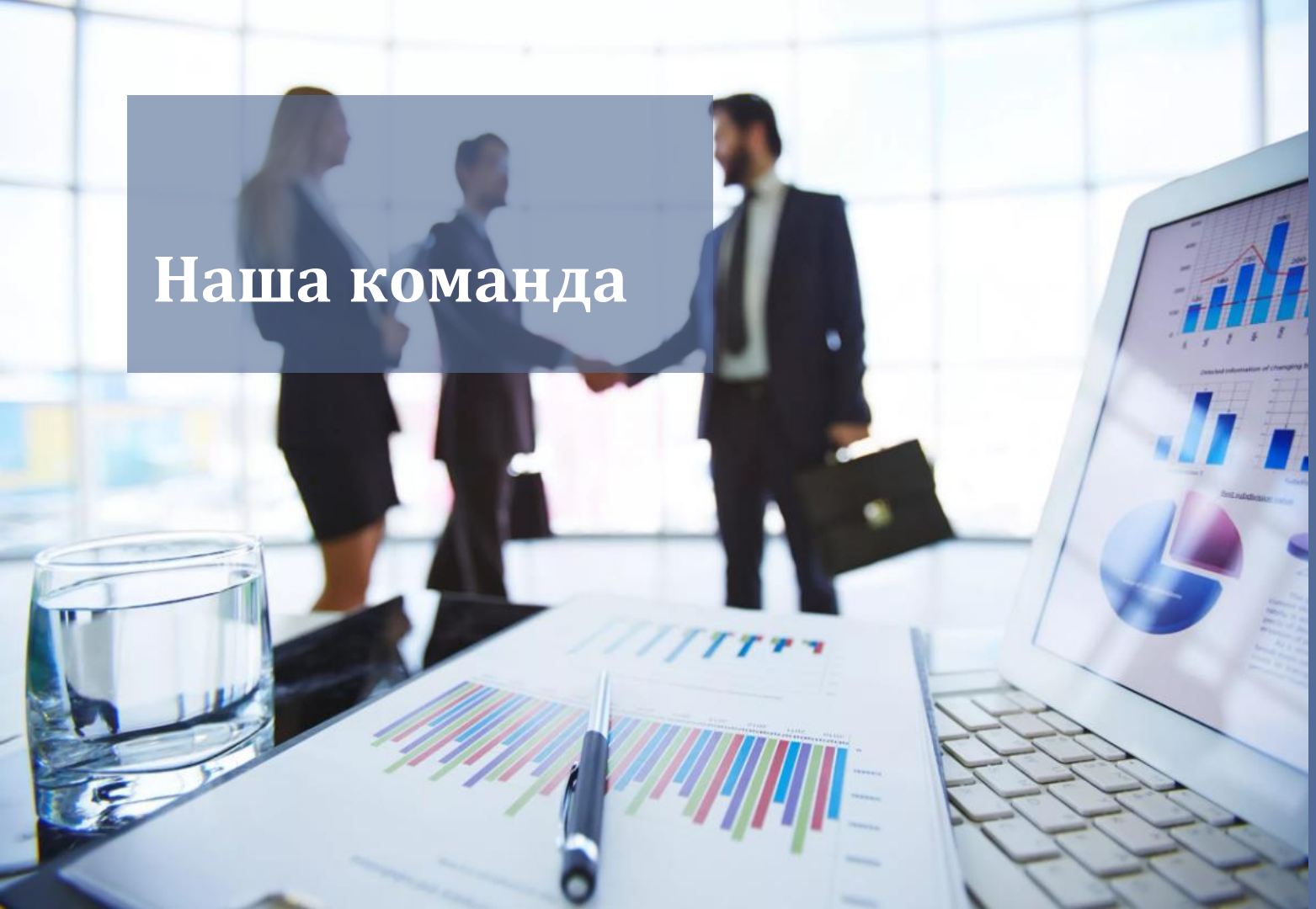
Коммерческие инструменты DevSecOps	Сервис XimiSecOps
Стоимость ПО от 5Р млн	Стоимость услуги **от 2Р млн. в год**
Стоимость годовой ТП от 1.5Р млн	Интеграция с вашими инструментами разработки **входит в стоимость**
Необходима команда для сопровождения ПО	**Не требует** выделения вычислительных ресурсов и сопровождения
Требуется выделение вычислительных ресурсов	**Не нагружает** FOT
Отсутствует возможность оперативно увеличивать производительность	**Опциональные услуги** наших Application Security специалистов
Требуется команда AppSec специалистов для устранения уязвимостей	

Наш подход - **гибкая ценовая и ресурсная политика** в зависимости от потребностей и возможностей заказчика.

Пример схемы взаимодействия*



*Данная схема носит исключительно ознакомительный характер.



Наша команда

04

Компетенции нашей проектной команды

01 Комплексный аудит политик и процедур информационной безопасности, процессов разработки и эксплуатации информационных систем, механизмов защиты информации, моделей и наборов данных для ИИ на соответствие лучшим практикам и отраслевым стандартам

02 Разработка комплексной стратегии обеспечения информационной безопасности и плана реализации программы внедрения процессов безопасной разработки программного обеспечения в соответствии с лучшими мировыми практиками

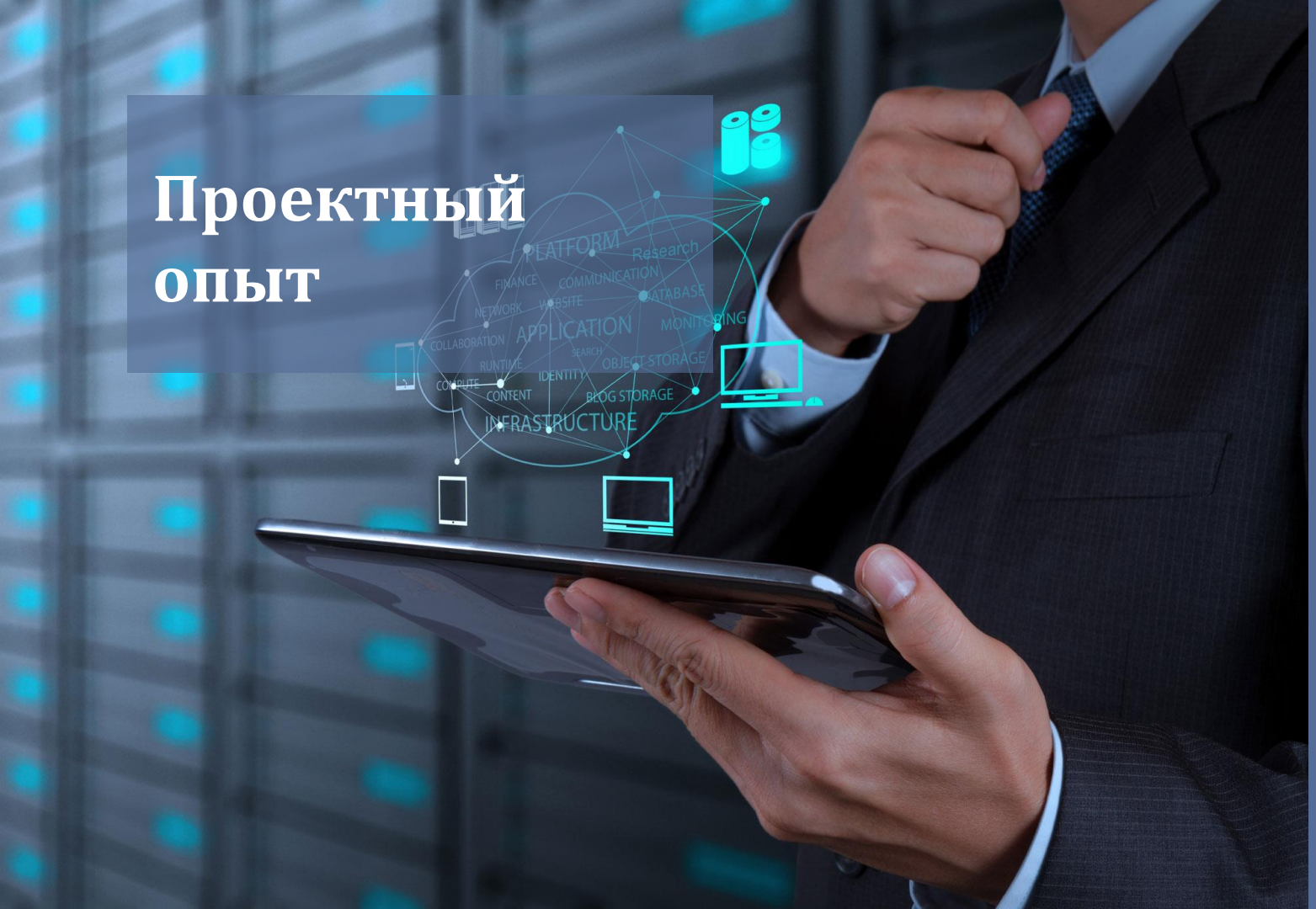
03 Внедрение и сопровождение комплексных систем автоматизированного мониторинга процессов обеспечения информационной безопасности, анализа угроз и управления инцидентами с целью реализации необходимых мер в корпоративной ИТ-инфраструктуре

04 Комплексная независимая оценка защищенности информационных систем от кибератак методом имитации действий злоумышленников, включая тестирование на проникновение, анализ возможных векторов атак и оценку эффективности средств защиты информации

05 Анализ рисков ИБ, оценка вероятности реализации кибератак и их потенциального ущерба, моделирование возможных сценариев нарушения ИБ для выработки эффективных мер противодействия и разработка рекомендаций по минимизации киберрисков

06 Разработка отчётной документации и планов мероприятий по актуализации системы управления ИБ

Проектный опыт



05

Примеры проектов нашей команды (1/2)

Российский ретейлер (2022 г.)

Оценка процессов безопасной разработки и подготовка стратегии по её развитию

- Провела оценку текущих подходов и методов безопасной разработки, проинтервьюированы 15 команд разработки;
- Разработала целевой процессный подход безопасной разработки и предложила варианты формирований команд SecOps;
- Предложила методы поэтапного развёртывания изменений в производственную среду и процедуры обеспечивающие высокое качество кода;
- Подготовила обоснование закупки инструментов тестирования, мониторинга транзакций, анализа кода на уязвимости, защиты контейнеров и обезличивания данных в БД.

Заказчик улучшил качество и защищённость продуктов без ущерба Time-to-Market

Крупный российский оператор связи (2021 г.)

Внедрение Container Security Platform

- Подготовила функциональное и финансовое сравнение различных систем CSP;
- Провела POC и обосновала закупку решения данного класса;
- Разработала механизмы интеграции CSP в текущие процессы разработки организации;
- Выполнила интеграцию платформа с внешними системами (AD, SIEM, Registry, CI/CD, SecretStore);
- Настроила компоненты защиты (RBAC Concept, Assurance Policy, Runtime Policy, Сканеры);
- Разработала документацию к системе и паспорт проекта;
- Провела обучение и инструктаж специалистов заказчика.

Обеспечена безопасность оркестратора, образов, контейнеров и реализована интеграция с SIEM

Примеры проектов нашей команды (2/2)

Российский банк (2019 г.)

Интеграция процессов и инструментов анализа кода на уязвимости в DevOps

- Проанализировала текущие процессы DevOps;
- Провела интервьюирование 10 команд разработки и согласовала требования к процессу и инструментам анализа кода;
- Интегрировала инструменты анализа кода с CI системой заказчика (GitLab CI) и с окружением заказчика (LDAP, Jira, IDE, Mail и др.);
- Задокументировала изменение в процессах разработки организации;
- Провела обучение и инструктаж специалистов заказчика по Code Research.

Заказчик улучшил безопасность продуктов, повысился уровень защиты критических приложений

Крупный российская страховая компания (2021 г.)

Оценка текущего уровня зрелости SOC и подготовка плана его развития

- Провела обследование текущего уровня зрелости направлений SOC и выявила возможности для быстрых улучшений;
- Разработала стратегию развития SOC на 3 года (описание и карту процессов, список источников событий для мониторинга и порядок их подключения, финансовую оценку реализации ключевых инициатив, техническую архитектуру SIEM-системы);
- Провела качественную оценку SIEM-решений;
- Выполнила финансовое сравнение собственного SOC и внешнего SOC на основе услуг MSSP-провайдера;
- Разработала план и детальные дорожные карты развития направлений SOC.

Задачи выполнены в срок, через год заказчик достиг целевое состояние системы

Спасибо за внимание