Организация процессов безопасной разработки и типовые роли участников

Дмитрий Шмойлов,
Руководитель отдела безопасности
программных продуктов

kaspersky

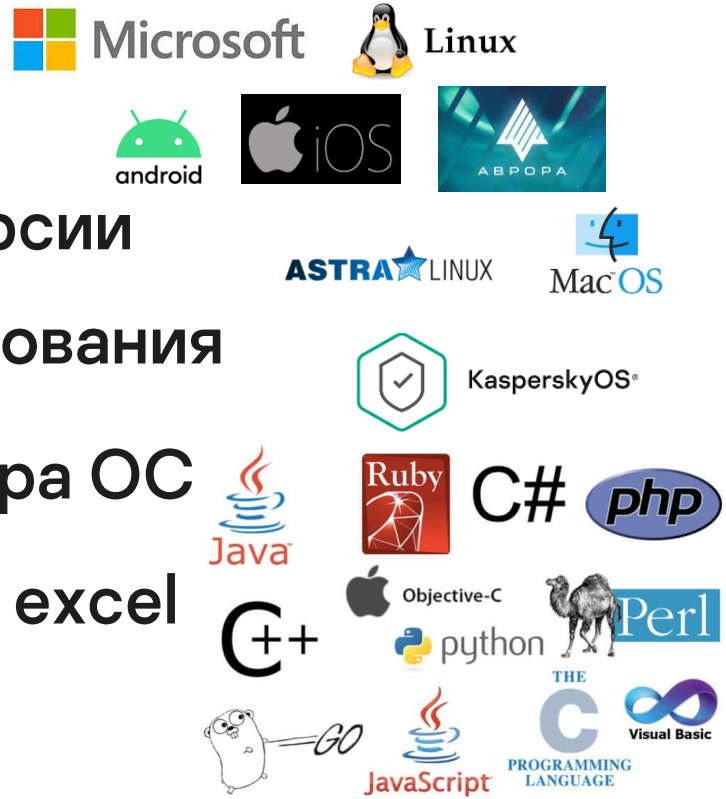
Agenda

- Что есть безопасная разработка (SDLC)
- Роли в рамках SDLC
- Процессы и практики
- Инструментарий

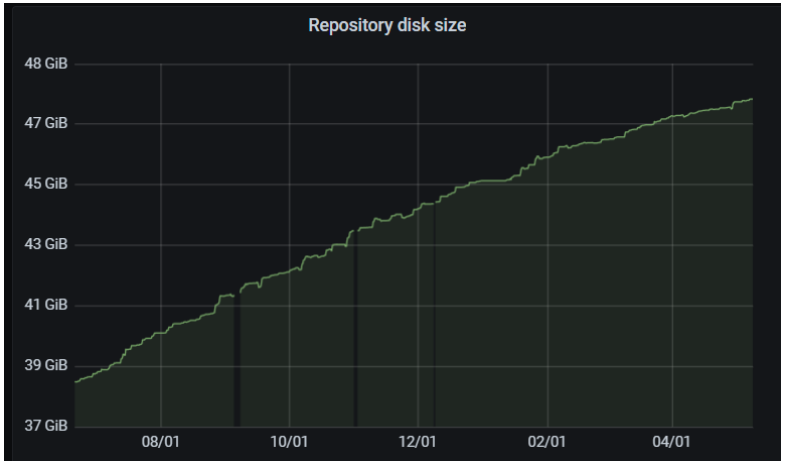
Kaspersky: разработка в цифрах

kaspersky

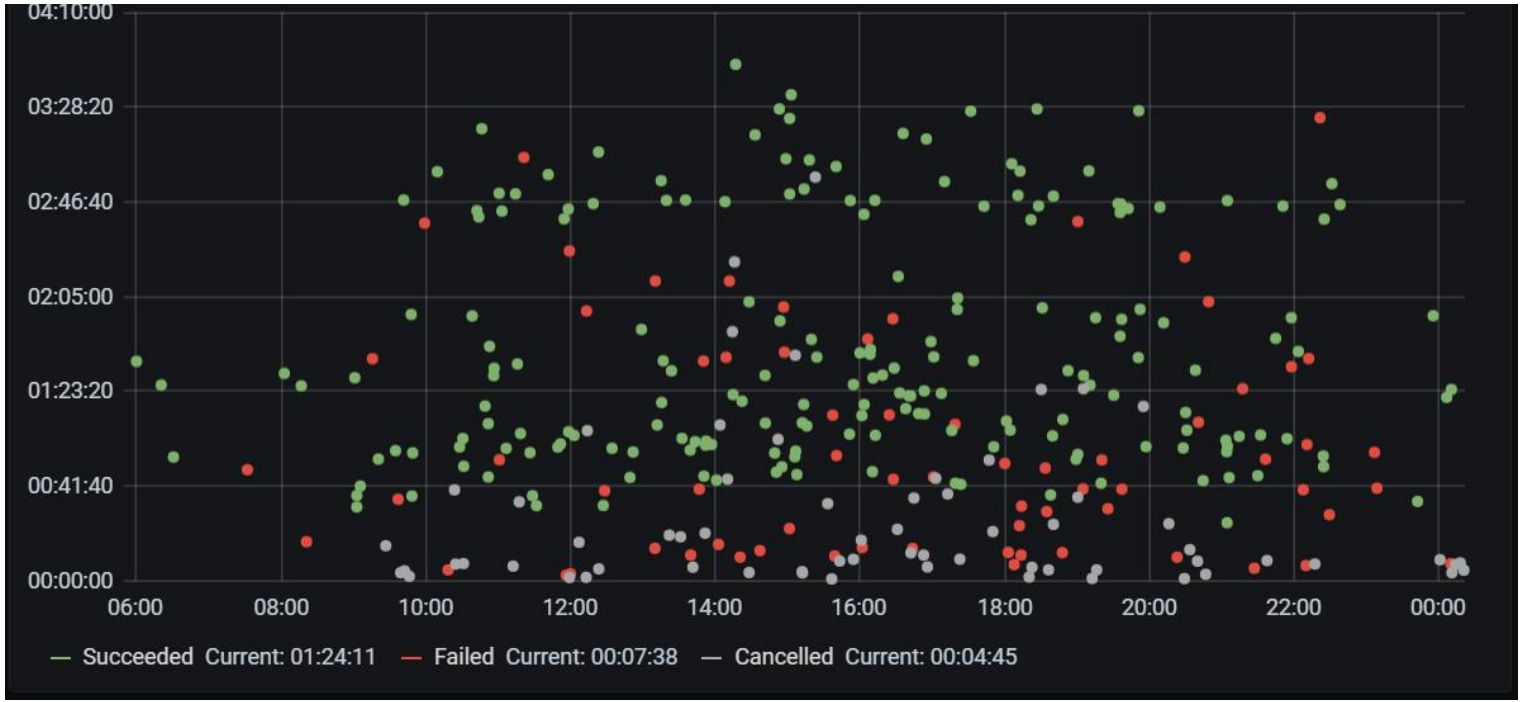
- Все платформы ОС (и даже специфические) и разные их версии
- Множество языков программирования
- Исполняемый код начиная от ядра ОС заканчивая Web или макросами excel



- ~800 разработчиков
- >200k строк кода меняется ежегодно
- ~70 продуктов (приложений) на поддержке
- 100+ крупных релизов продуктов ежегодно



~ 100 изменений в день

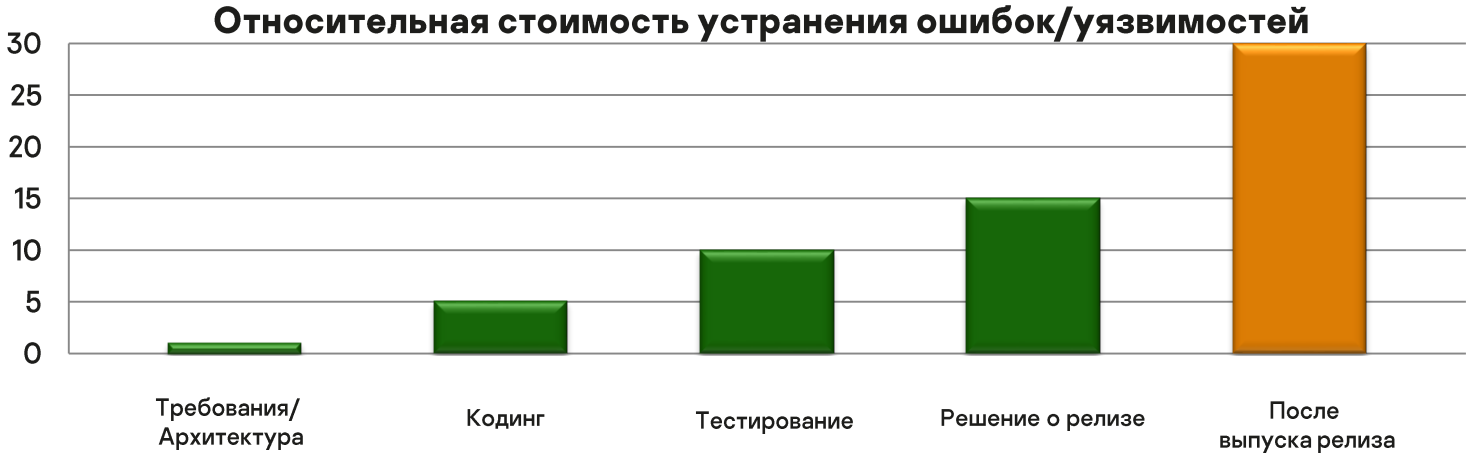


SDLC внутри

kaspersky

SDLC – цикл безопасной разработки

1. TRAINING	2. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	Execute Incident Response Plan
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modeling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	

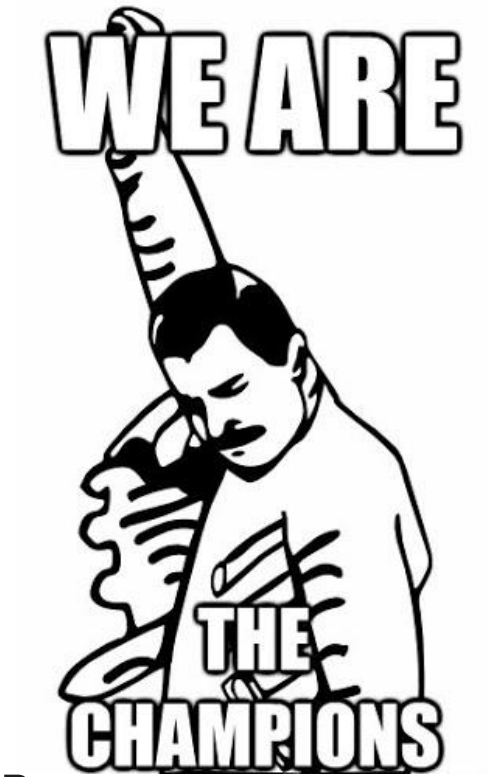


Люди и роли

kaspersky

- Команда разработки
  - Менеджер проекта
  - Архитекторы
  - Разработчики
  - Security champion
- Менеджеры продуктов
- QA (контроль качества)
- Отдел безопасности продуктов

- Security champion – кто он?
  - Находится внутри команды
    - Как правило архитектор или ведущий разработчик
  - Знает об изменениях
  - Проводит предварительный анализ безопасности
  - Привлекает отдел безопасности продуктов



- Серьёзно? Нам разработчиков то не хватает…
  - Это не вакансия. Это роль.
- Где же его взять? ВУЗы их не выпускают.
  - ~~НАЗНАЧИТЬ приказом!~~
  - Выращивать внутри. Вовлекать.
- Как мотивировать?
  - Хороший код = безопасный код.

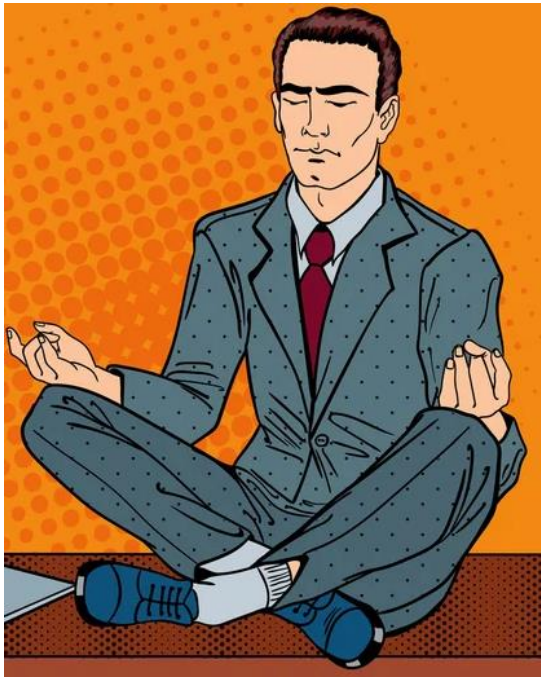
Дать возможность сделать код хорошим.

- Идентифицировать все известные риски (команды их знают)
- Описать архитектуру – найти новые риски
- Не думать о стоимости исправления на этом этапе, приоритезировать
- Постараться описывать риски без привязки к реализации – просто суть проблемы
- Создать security backlog

Владелец продукта:

- Выделить квоту на исправления безопасности

  Если нужно – запросить у руководства.
- Оценить стоимость конкретных исправлений
- Запланировать исправления в ближайшие релизы
- Иметь план Б (срочные обновления, например)

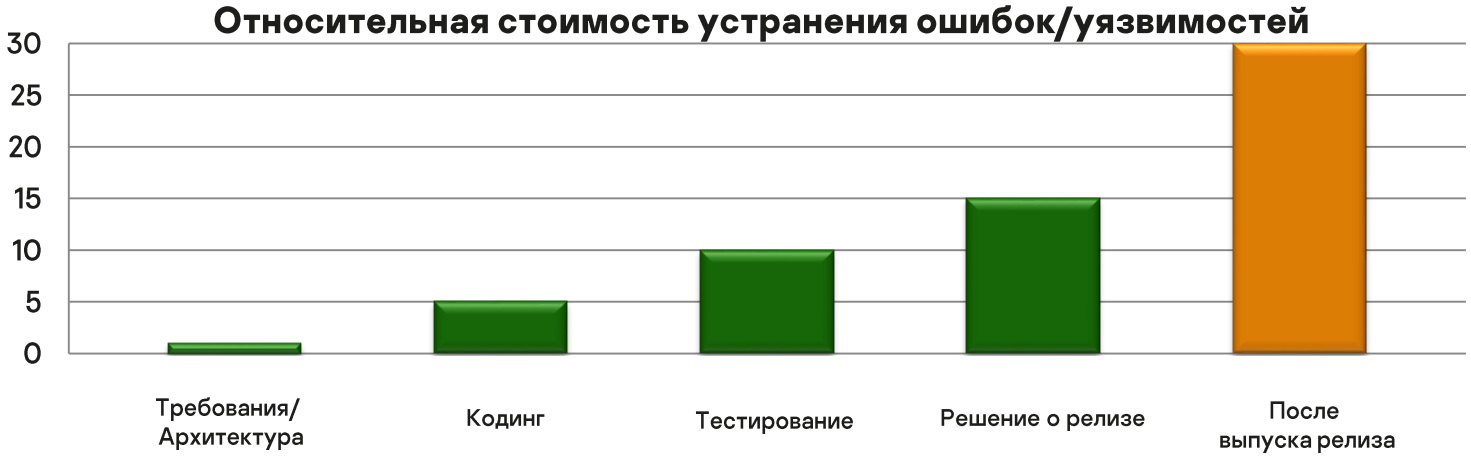


SDLC практики

kaspersky

SDLC – цикл безопасной разработки

1. TRAINING	2. REQUIREMENTS	3. DESIGN	4. IMPLEMENTATION	5. VERIFICATION	6. RELEASE	7. RESPONSE
1. Core Security Training	2. Establish Security Requirements	5. Establish Design Requirements	8. Use Approved Tools	11. Perform Dynamic Analysis	14. Create an Incident Response Plan	Execute Incident Response Plan
	3. Create Quality Gates/Bug Bars	6. Perform Attack Surface Analysis/ Reduction	9. Deprecate Unsafe Functions	12. Perform Fuzz Testing	15. Conduct Final Security Review	
	4. Perform Security and Privacy Risk Assessments	7. Use Threat Modeling	10. Perform Static Analysis	13. Conduct Attack Surface Review	16. Certify Release and Archive	



SDLC. Документы

Название документа
Руководство по разработке безопасного программного обеспечения
Процесс Secure code review
Secure coding guideline (checklist)
Процесс моделирования угроз
Инструкция для Security Champion по процессу моделирования угроз
Процесс статического анализа кода
Процесс динамического анализа кода
Процесс использования сторонних библиотек
Процесс фаззинг тестирования
Процесс управления уязвимостями
Правила проведения тестирования на проникновение

SDLC. Обучение

Наименование курса	Участники
SDL. Вводный курс	Все
Безопасная разработка C/C++	Разработчики, Руководители команд, Архитекторы
Моделирование угроз	Руководители команд, Архитекторы, Security champions
Fuzzing	Руководители команд, Архитекторы, Security champions
Безопасность Web приложений	Все
Безопасность мобильных приложений	Разработчики, Руководители команд, Архитекторы
Основы криптографии	Разработчики, Руководители команд, Архитекторы

SDLC. Обучение

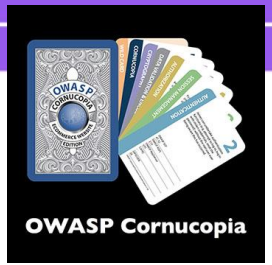
Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

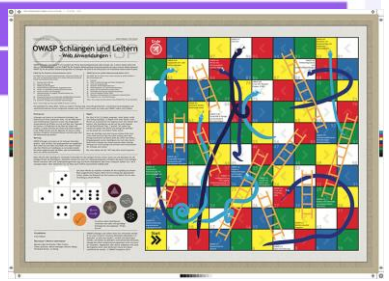
https://itsecgames.blogspot.com/

https://owasp.org/www-project-snakes-and-ladders/

https://owasp.org/www-project-security-shepherd/

https://owasp.org/www-project-cornucopia/





bWAPP
an extremely buggy web app !

SDLC. Общие требования

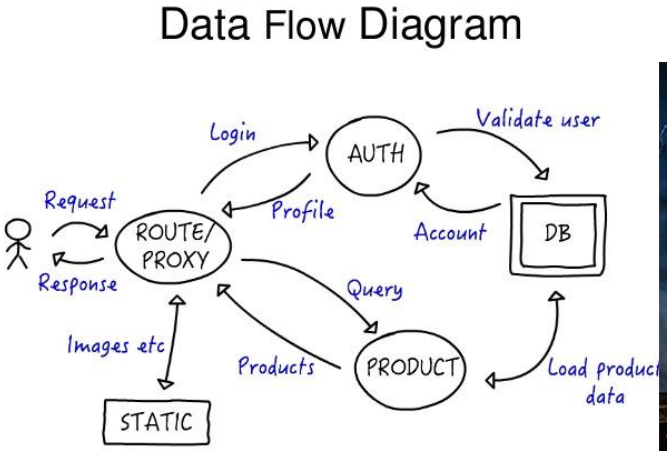
Требования
Basic security requirements
Secure communication
Prevent local privilege escalation
Mitigate risks of sensitive data disclosure
Up-to-date third party software
Perform static analysis
Perform dynamic analysis
Mitigate network vulnerabilities (incl OWASP's Top 10)
SDL procedures completeness

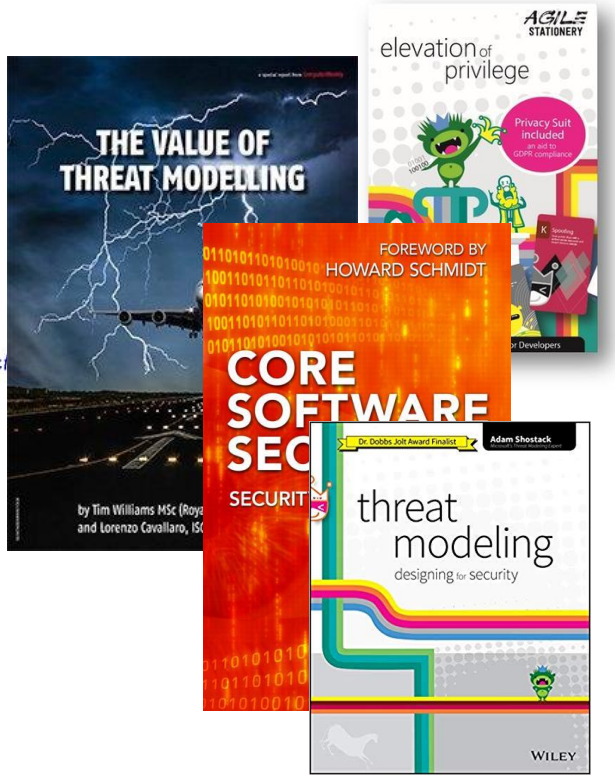
SDLC. Моделирование угроз

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

Методологии:

1. STRIDE
2. DREAD
3. PASTA
4. Kill Chain
5. OWASP
6. ..... mixed?





Рекомендации гуру:

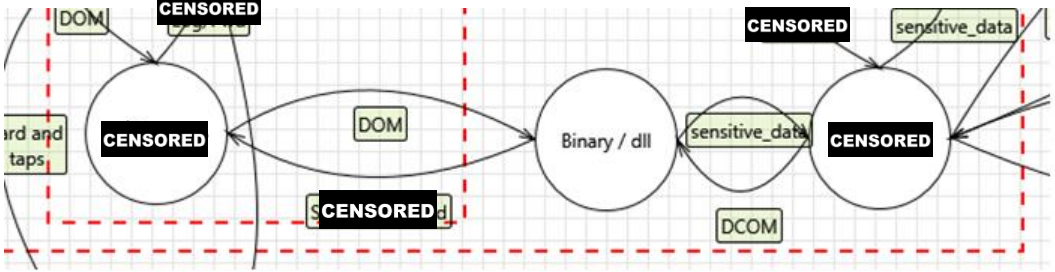
1. Концентрироваться на важных продуктах/модулях
2. Описывать и обсуждать изменения
3. Рисовать диаграммы, схемы, идеально – архитектуру (на доске, бумаге, где удобнее)
4. Задавать вопросы «а что если»
5. Идентифицировать риски без привязки к реализации

SDLC. Моделирование угроз

Training | Require-ments | **Design** | Implemen-tation | Verification | Release | Response

Основные этапы:

1. **Создание диаграммы (DFD, sequence, etc.)**



   1. **Границы доверия**
   2. **Сущности**
   3. **Потоки данных**
   4. **Хранилища**

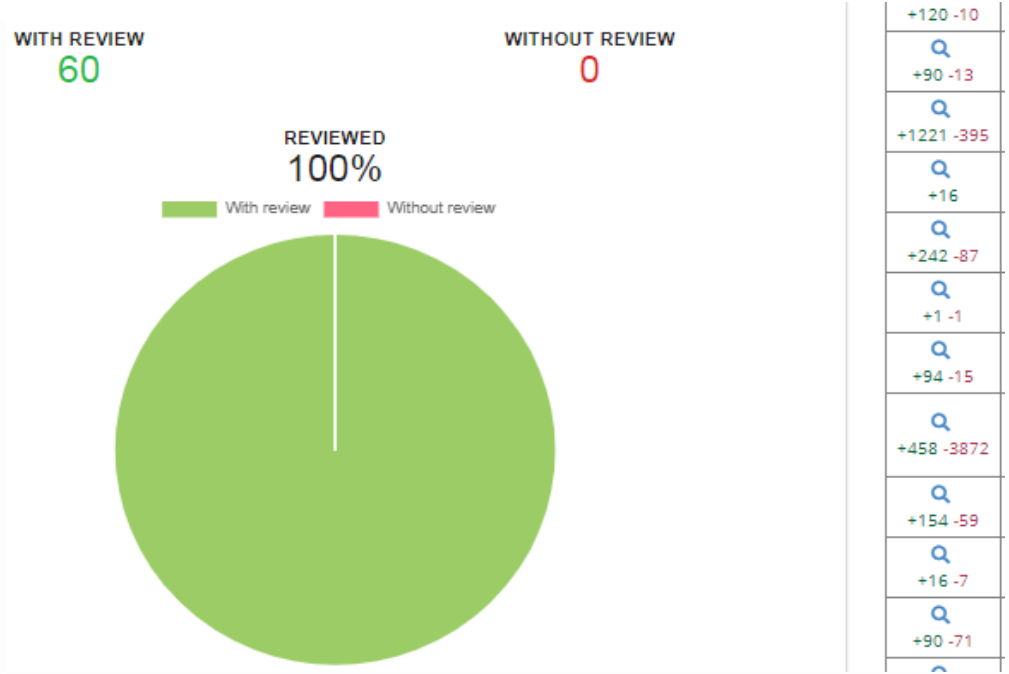
2. **Описание сущностей:**

Сущность	Потенциальный риск
Интерфейсы	избыточная поверхность атаки
Парсинг данных	выполнение злонамеренного кода (в т.ч. удаленное)
Сетевое взаимодействие	раскрытие данных, перехват управления
Изменение окружения	локальное повышение привилегий
Хранение данных	раскрытие данных

SDLC. Анализ исходного кода

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

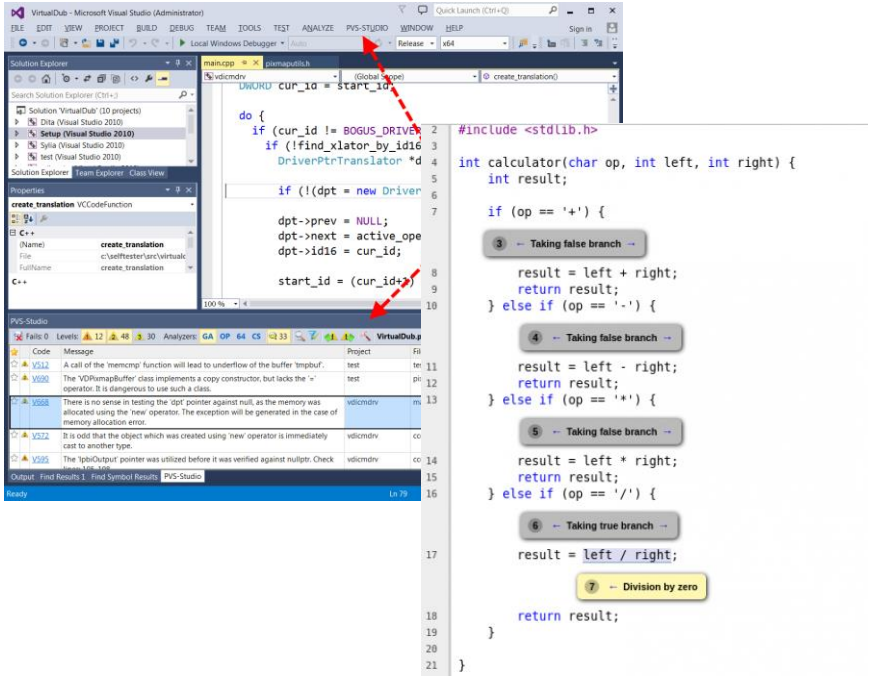
Сбор артефактов ревью и построение отчетов о покрытии



Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

Инструменты:

- «Free SAST», google it
  - Clang
  - GitHub code scanning
  - Lint
  - etc...
- Платные:
  - Coverity (synopsys)
  - SVACE (ИСП РАН)
  - etc...



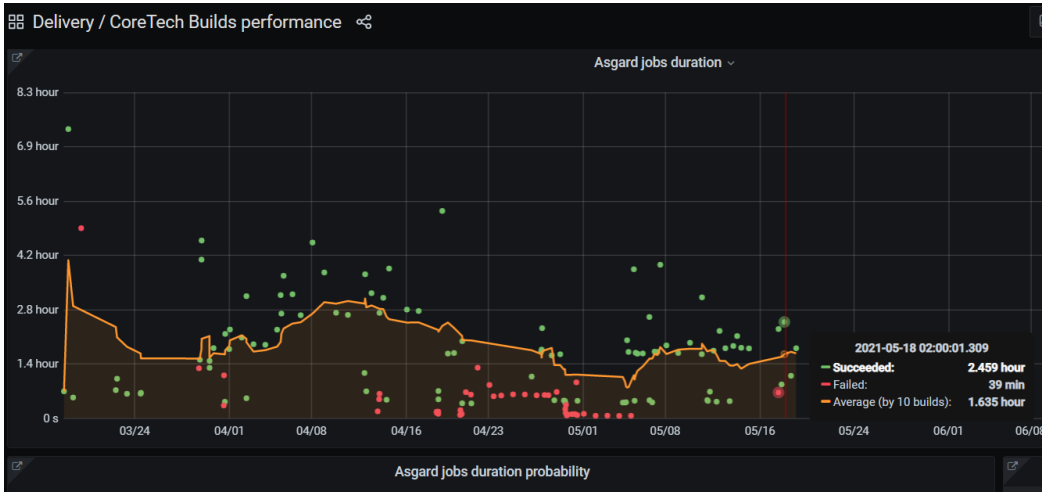
Статический анализатор должен работать на каждом коммите

Community t.me/sdl_static

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

SVACE в цифрах:

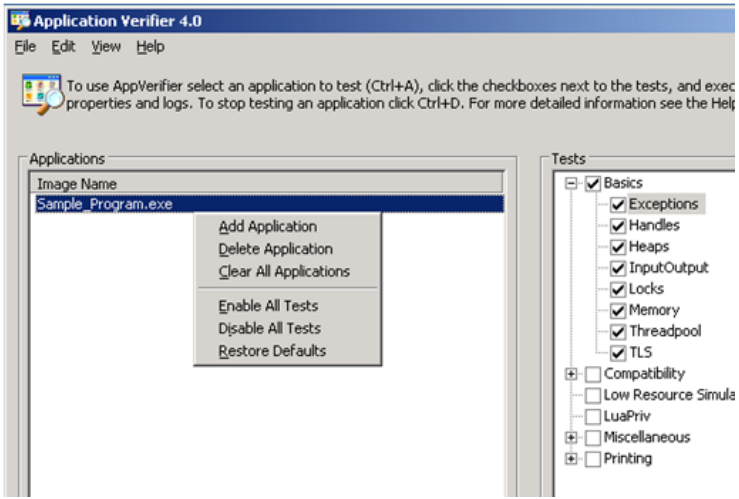
- Продуктовый код ~**1.5 M** SLOC
- Компоненты ~**10 M** SLOC
- 3rd party код ~**40 M** SLOC
- **~5K** срабатываний анализатора
- ~ 4 часов на сборку продукта
- 12 + issue на развитие функционала / устранение ошибок
- Предложения по борьбе с false positive



Training | Require-ments | Design | Implemen-tation | **Verification** | Release | Response

Инструменты:

- Application Verifier
- Driver Verifier
- * - sanitizer (address-, memory-, etc.)



SDLC. Инструменты. Сообщество

t.me/sdl_community	Общие вопросы
t.me/sdl_dynamic	Динамический анализ, фазинг
t.me/sdl_static	Статический анализ

SDLC. Снижение рисков использования OpenSource

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

Процессы:

1. Проверка публичных уязвимостей
2. Приоритезация
3. Передача запросов на исправления в продукты и компоненты
    1. SLA = 1 день

open source

Common Vulnerabilities and Exposures

NVD

Продукты, компоненты, сервисы:

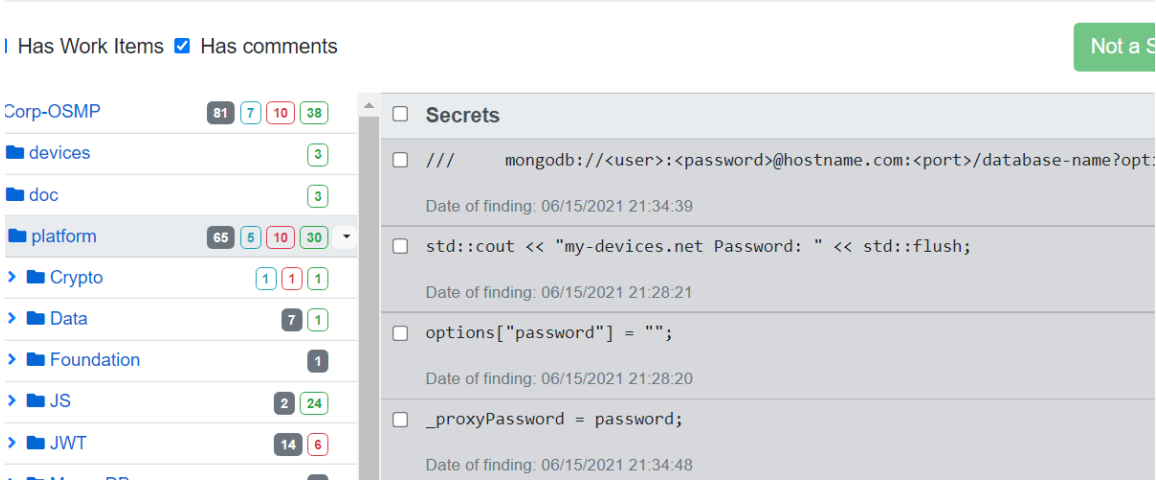
1. Устранение уязвимостей, обновление библиотек

SDLC. Инструменты поиска аутентификаторов в коде

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

- Код содержит аутентификаторы и пароли для интеграции с внешними системами
- В вашем организации нет никаких паролей в коде? – вы ошибаетесь

- Система поиска по коду
- Machine learning движок
- Интеграция с CI/CD
- Обработка дефектов



Training | Require-ments | Design | Implemen-tation | **Verification** | Release | Response

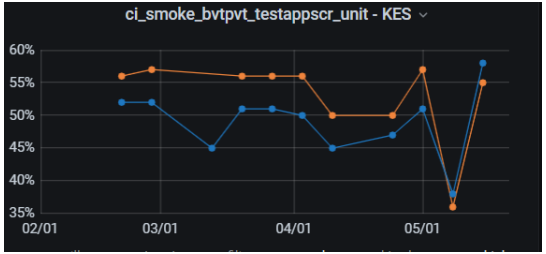
Продуктовые тесты

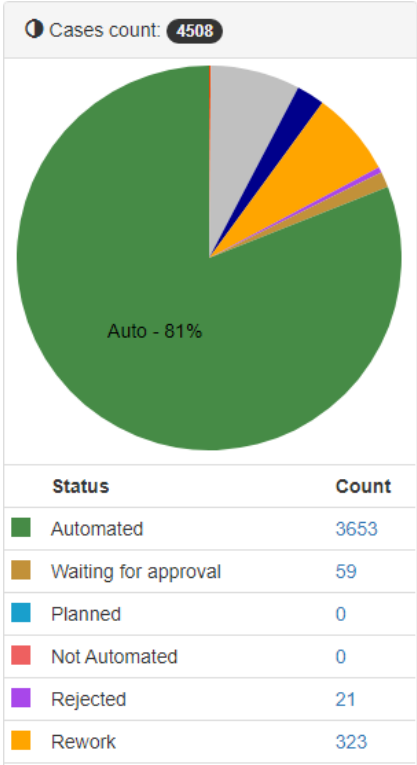
Более 10 000 тест кейсов
Более 8 500 тестов проверяется за релиз
До 14 дней - прогон с фиксированием результатов

Результаты прохождения тестов передаются в команду сертификации

Основные типы тестов:

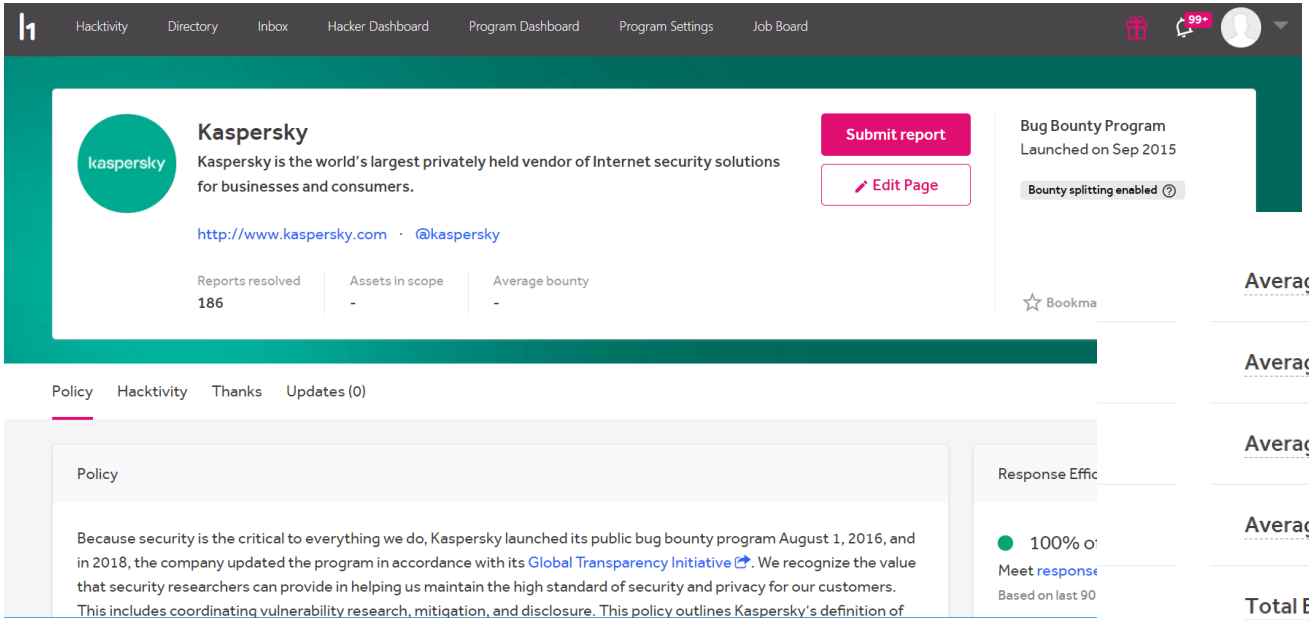
- End To End тесты
- Юнит тесты
- Интеграционные тесты





SDLC. Публичные уязвимости

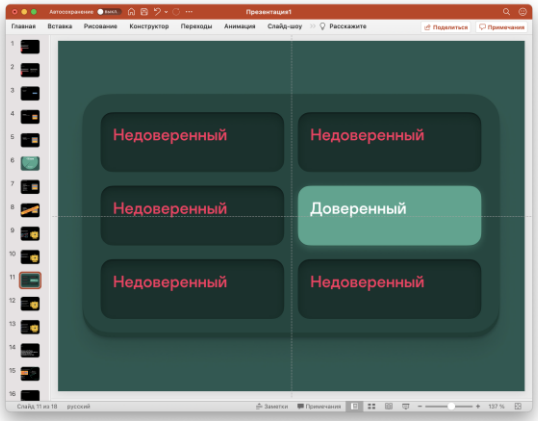
Training | Require-ments | Design | Implemen-tation | Verification | Release | Response



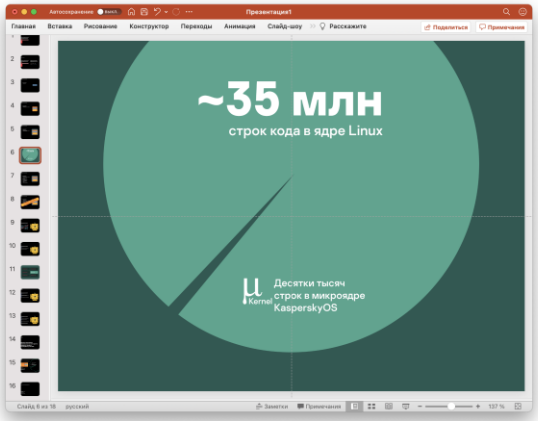
- BugBounty программа
- Платформа Hackerone
- До $100 000 вознаграждение

Average Response Time	5 days
Average Triage Time	a month
Average Bounty Time	23 days
Average Resolution Time	1 month
Total Bounties	$79,550
Average Bounty	$1,170

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response



Изоляция доверенного кода



Уменьшение доверенного кода

Использование шаблонов с доказанной безопасностью

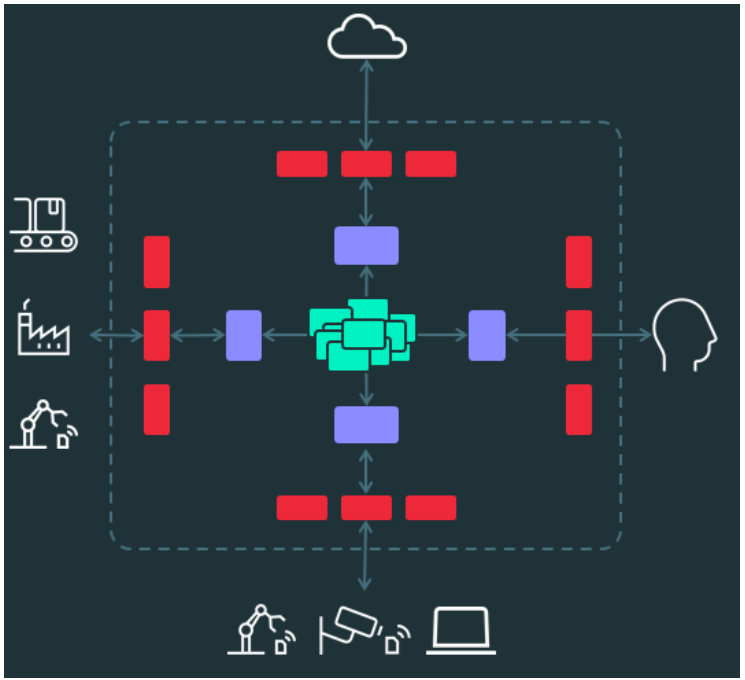
SDLC. Secure by Design

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

Недоверенные компоненты изолированы, они не влияют на безопасность

Высоко-доверенные компоненты, повышают целостность данных

Доверенные компоненты изолированы и теперь работают только с высоко-целостными данными



SDLC. Secure by Design для KOS

Training | Require-ments | Design | Implemen-tation | Verification | Release | Response

В KasperskyOS модель мандатного контроля целостности на основе расширенной модели Биба

Безопасность расширенной модели доказана математически

Secure by design можно применить ко всей системе

4.5 Свойства безопасности, обеспечиваемые моделью

Операция *upgrade*, которая должна выполняться в реальной системе только доверенными сущностями, подразумевает отсутствие информационных потоков к объекту, уровень целостности которого предполагается повысить. В противном случае источник такого информационного потока может стать менее целостным, чем приемник. В реальной системе для гарантии отсутствия таких потоков используются специальные средства (например, криптографические). Такие средства не рассматриваются в рамках модели. Поэтому при анализе модели на предмет свойств безопасности, которые она обеспечивает, мы рассматриваем только такие вычисления системы переходов, на которых доверенные сущности не выполняют операции, которые могут нарушить целостность системы.

Назовем *вычислениями без кооперации доверенных и недоверенных сущностей для реализации информационных потоков* такие вычисления системы переходов TS, на которых доверенные сущности не инициируют применение правила *upgrade*.

Теорема 1 показывает, что либо возникающие информационные потоки направлены от не менее целостных сущностей или объектов, либо расширение зоны захвата контроля носит строго ограниченный характер. Ограничение заключается в том, что в этом случае в множестве захваченных компонентов всегда уже имеется сущность с уровнем целостности большим либо равным уровню целостности компонента, к которому реализован информационный поток (рис. 10).

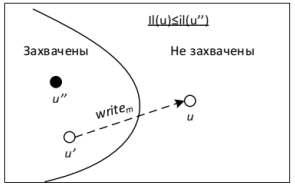


Рис. 10. Ограничение расширения зоны захвата контроля
Fig. 10. Restriction of the area of compromised components

https://ispranproceedings.elpub.ru/jour/article/view/1239/1032

SDL. Сертификация безопасности

СЕРТИФИКАТ

TÜV AUSTRIA

ZERTIFIKAT | CERTIFICATE | CERTIFICAT | CERTIFICADO | СЕРТИФИКАТ | شهادة | 证书 | 인증서

соответствия системы менеджмента требованиям стандарта ISO/IEC 27001:2013

В соответствии с процедурами TÜV AUSTRIA Deutschland настоящим подтверждается, что

АО «Лаборатория Касперского»
Ленинградское шоссе, д.39А/2
125212, г. Москва
Российская Федерация

Применяет систему менеджмента, соответствующую вышеназванному стандарту в следующих областях:

Доставка вредоносных и подозрительных файлов с помощью инфраструктуры Kaspersky Security Network (KSN), их безопасное хранение в Kaspersky Lab Distributed File System (KLDFS) и обеспечение доступа.

Действующее заявление о применимости: V2.0, 16.09.2019

Регистрационный номер сертификата: TAD ISMS 19924

Действительно с 2019-12-13
Действителен до 2022-12-12

Орган по сертификации
TÜV AUSTRIA Deutschland GmbH

Фильдерштадт, 2019-12-13

Данная сертификация была проведена в соответствии с процедурами аудитирования и сертификации TÜV AUSTRIA Deutschland GmbH и подлежит регулярным надзорным аудитам.
TÜV AUSTRIA Deutschland GmbH Kurze Straße 40 D-70794 Filderstadt www.tuv-ad.de

DAkkS

TÜV AUSTRIA

ZERTIFIKAT | CERTIFICATE | CERTIFICAT | CERTIFICADO | СЕРТИФИКАТ | شهادة | 证书 | 인증서

CERTIFICATE

Certification according to IEC 62443
Security for industrial automation and control systems
Part 4-1: Secure product development lifecycle requirements (IEC 62443-4-1:2018)

In accordance with TÜV AUSTRIA SERVICES procedures, it is hereby certified that

Kaspersky Lab JSC
Leningradskoe sh. 39A bld.2
125212 Moscow
Russian Federation

applies a secure product development lifecycle in line with the above standard **at a maturity level of 3** for the following scope

Kaspersky Industrial Cybersecurity for Networks

Based on inspection report: AU-20042706
Initial certification 2020-09-01
Valid until 2023-08-31

Certification Body
at TÜV AUSTRIA SERVICES GMBH

Vienna, 2020-09-01

This certification was conducted in accordance with TÜV AUSTRIA SERVICES auditing and certification procedures and is subject to regular surveillance audits.

TÜV AUSTRIA SERVICES GMBH Deutschstraße 10 A-1230 Wien www.tuv.at

Вопросы?

Дмитрий Шмойлов,
Dmitry.Shmoylov@kaspersky.com

Руководитель отдела безопасности
программных продуктов

kaspersky