Безопасность цепочки поставки Open Source компонентов

Алексей Смирнов

whoami & whoweare

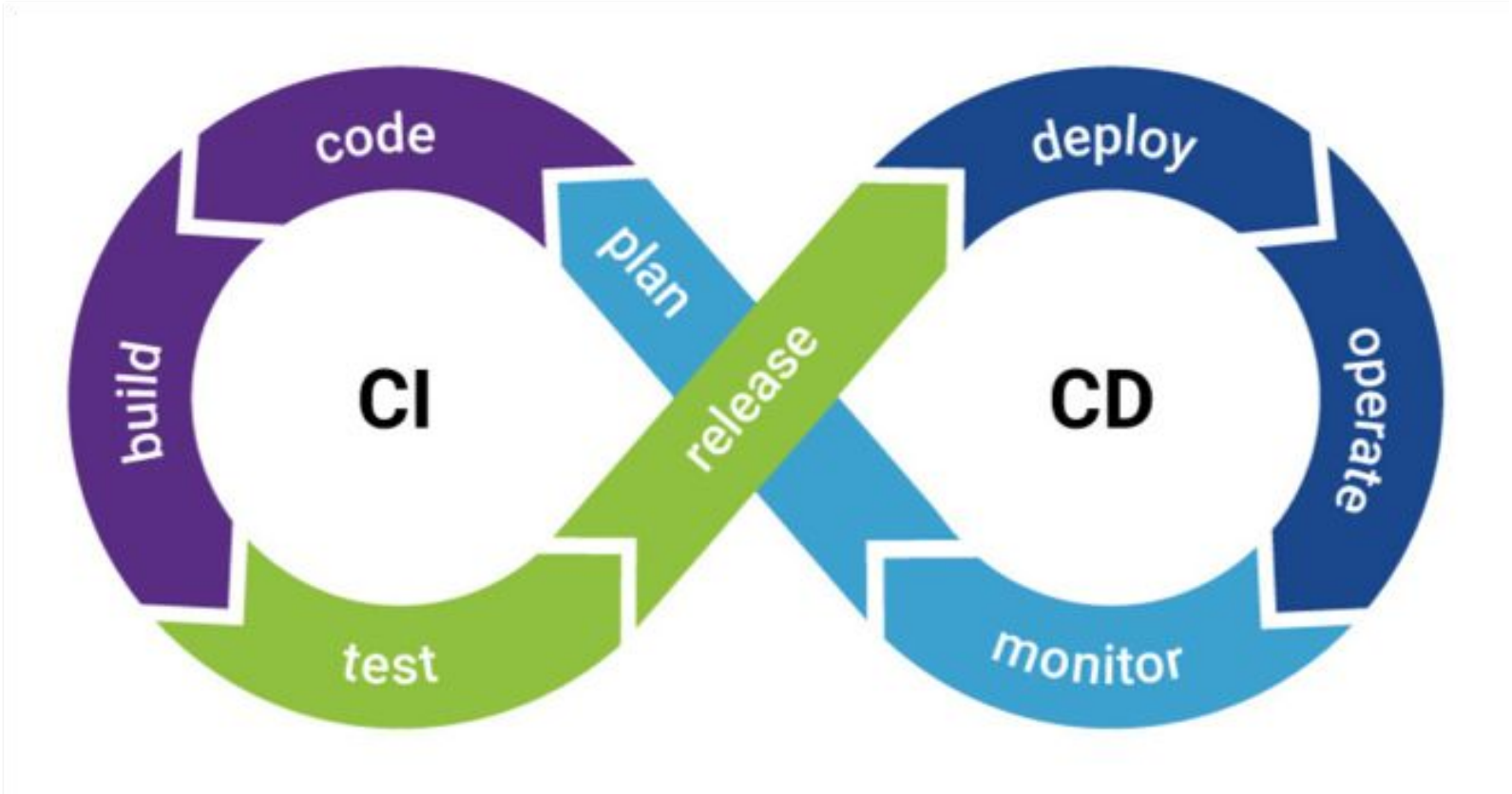
Алексей Смирнов, основал profiscope.io

- проводим технический аудит
- выпускаем CodeScoring — российский SCA
- просвещаем в Code Mining @ods.ai, tg: @codemining
  - 3 года ведем конференцию про анализ кода
  - делаем бесплатные курсы
  - публикуемся и выступаем

DevOps

- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate
- Monitor

2008



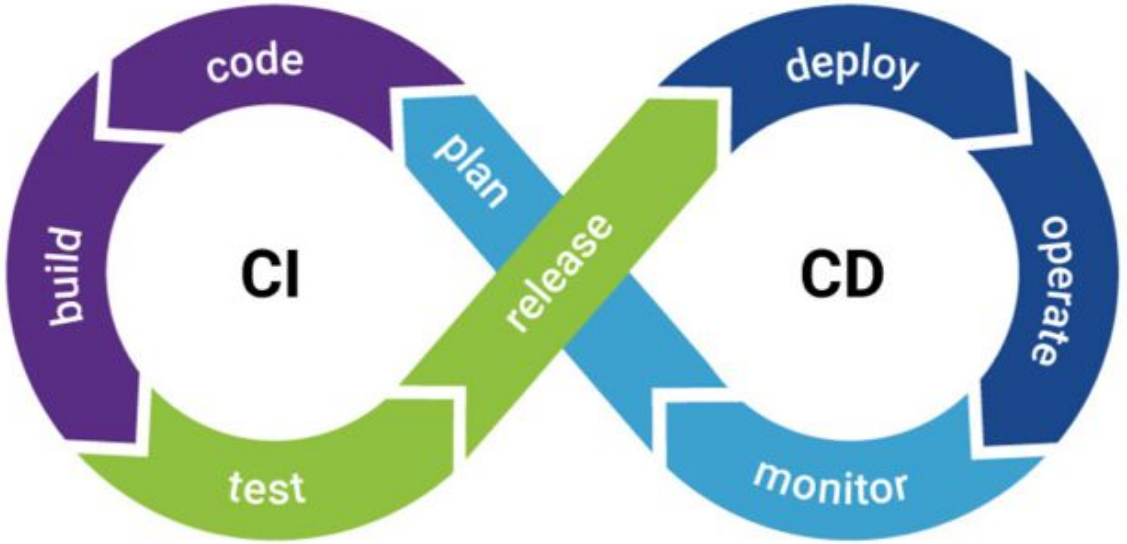
Microsoft Security Development Lifecycle

Training	Requirements	Design	Impelementation	Verification	Release	Response
Core Security Trainig	Establish Security Requirements	Establish Design Requirements	Use Approved Tools	Dynamic Analysis	Incident Response Plan	Execute Incident Responce Plan
	Create Quality Gates / Bug Bars	Analyze Attack Surface	Deprecate Unsafe Function	Fuzz Testing	Final Security Review	
	Security & Privacy Risk Assessment	Threat Modeling	Static Analysis	Attack Surface Review	Realese Archive	

2008

DevOps -> DevSecOps

- **SAST**
  Static Application Security Testing
- **DAST/IAST**
  Dynamic/Interactive Application Security Testing
- **OSA**
  Open Source Analysis
- **SCA**
  Software Composition Analysis



- **BAST**
  Behavioral Application Security Testing
- **BCA**
  Bytecode & Container Analysis
- **WAF**
  Web Application Firewall

DevOps -> DevSecOps

- OSA

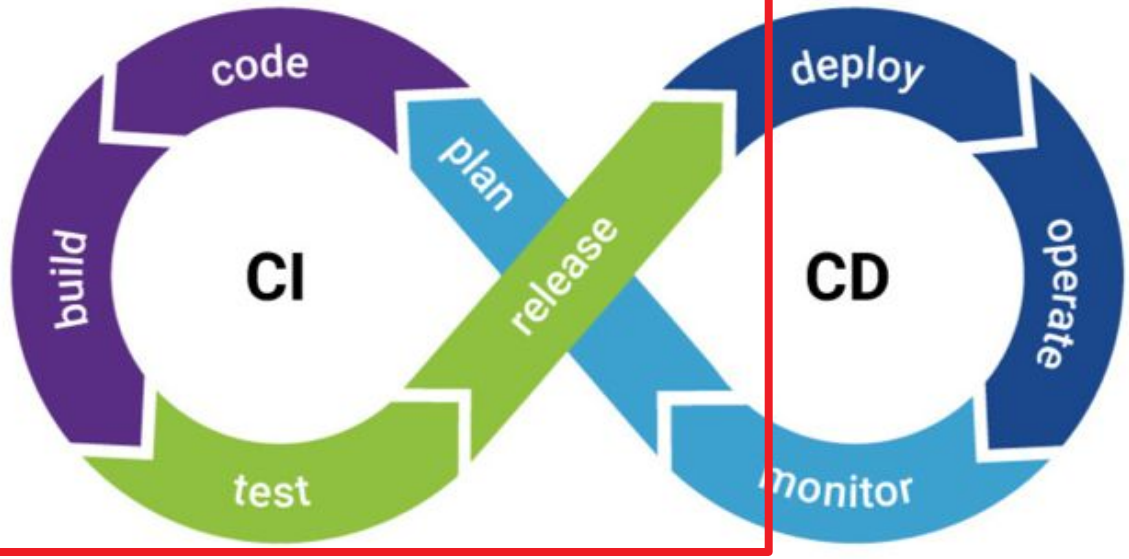
Open Source Analysis

- SCA

Software Composition Analysis

- SAST

Static Application Security Testing



- DAST/IAST

Dynamic/Interactive Application Security Testing

- BAST

Behavioral Application Security Testing

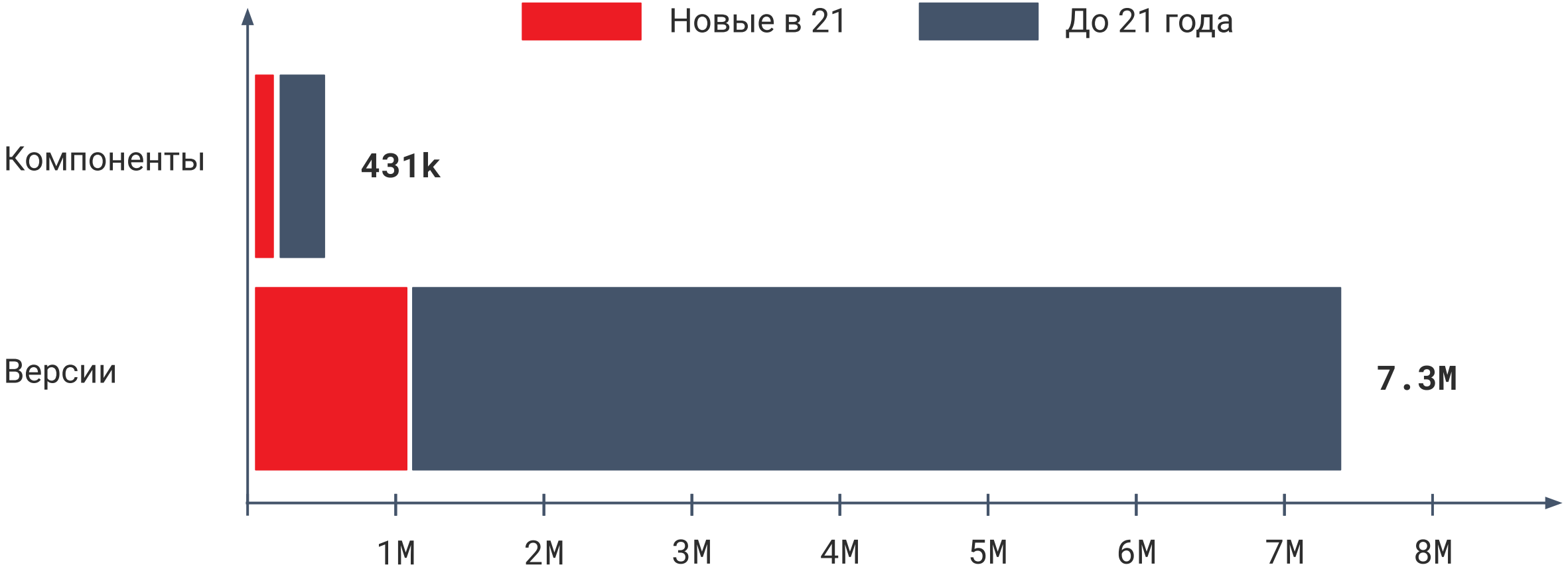
- BCA

Bytecode & Container Analysis

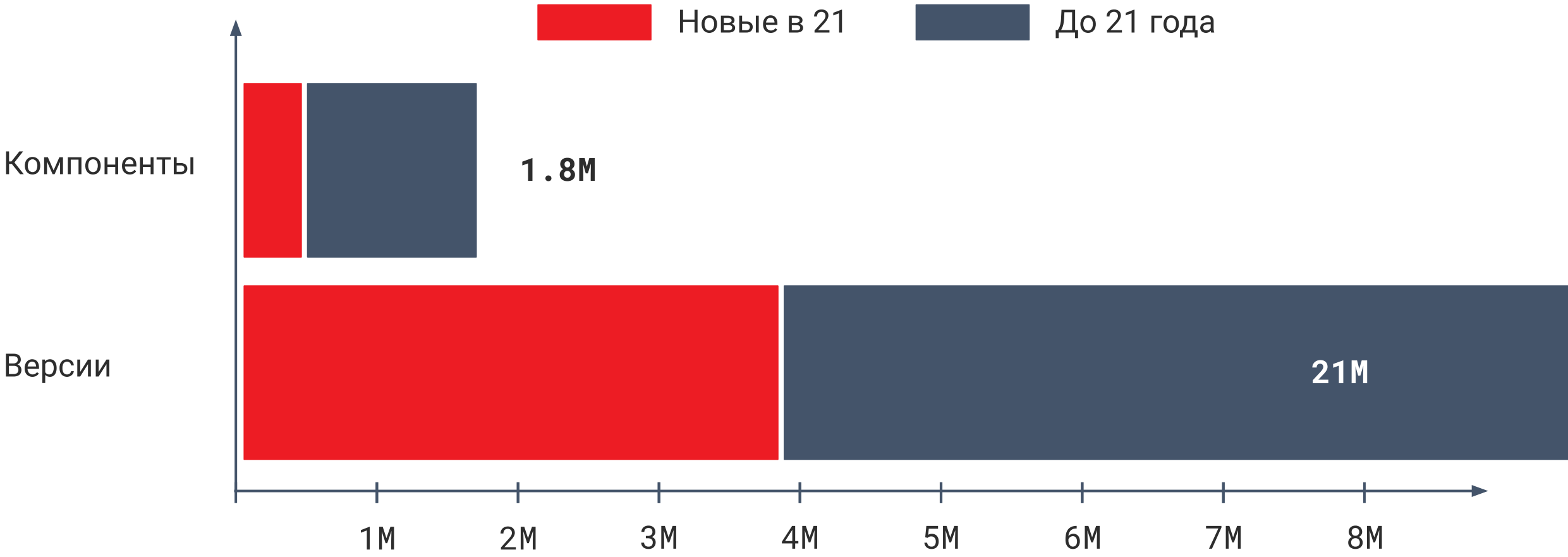
- WAF

Web Application Firewall

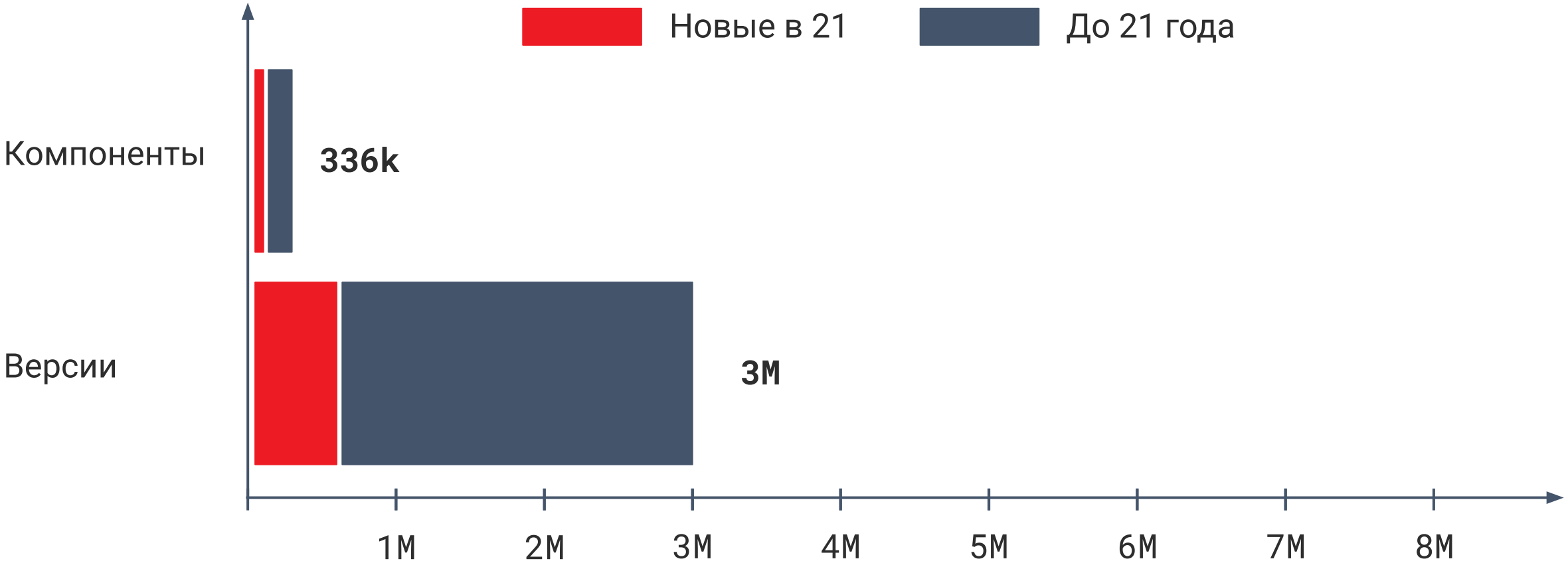
Open Source растёт / Java



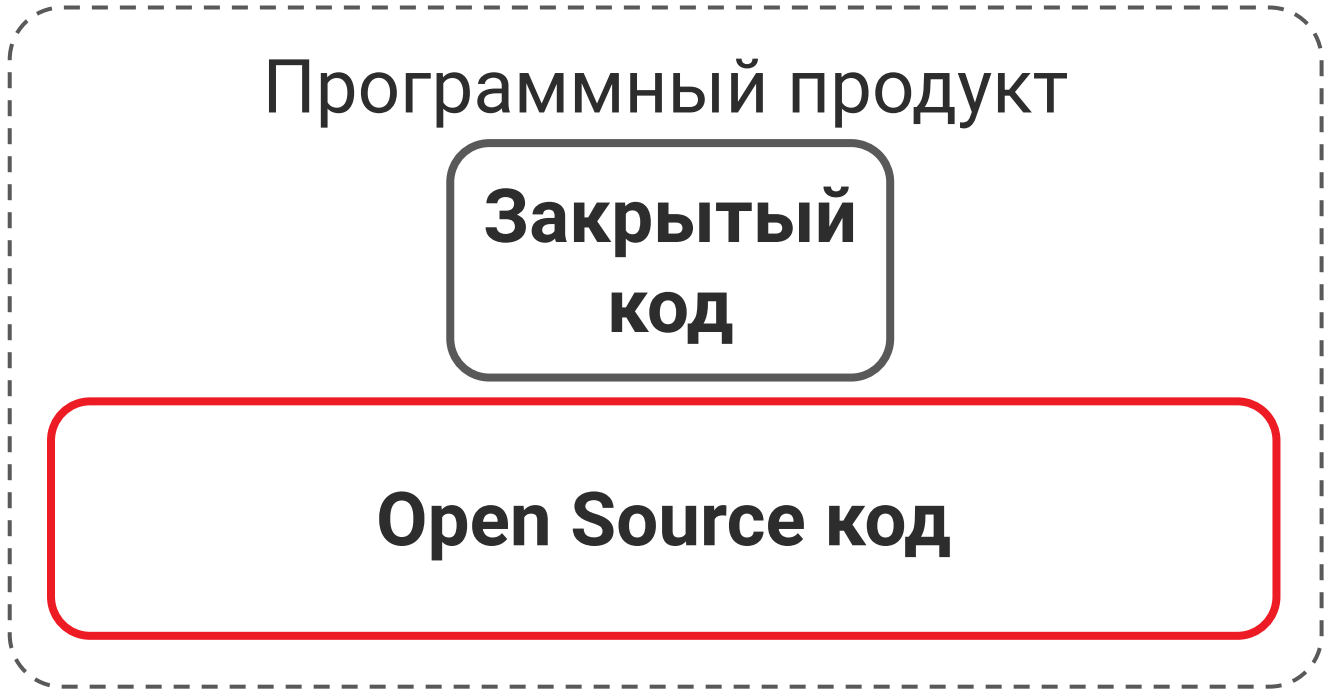
Open Source растёт / JavaScript



Open Source растёт /Python



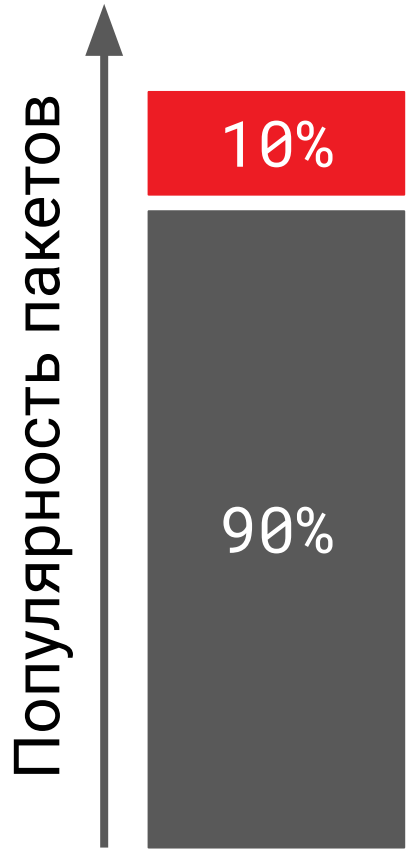
Open Source > Proprietary



Неизвестен
Нужно сканировать
Сложно исследовать

Известен
Идентифицируем
Исследуем

Популярное лучше изучено



в 36% проектов выявлены уязвимости

в 8% проектов выявлены уязвимости

Open Source небезопасен 💔

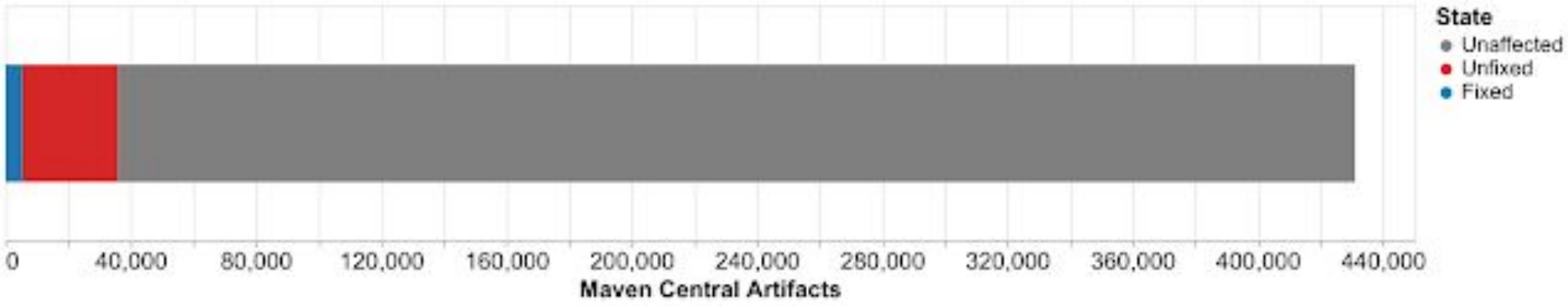
- **Уязвимости**
  Появляются каждый день. На многие уязвимости есть эксплоит на GitHub. Время подготовки атаки существенно сократилось.

- **Лицензии**
  Их отсутствие, лицензионная (не)совместимость, риск смены лицензии на более строгую, риск введения экспортных ограничений.

- **Закладки**
  Такие закладки могут быть активированы по разным признакам: locale, geoip, geoloc, etc.

Пример. log4j (log4shell)

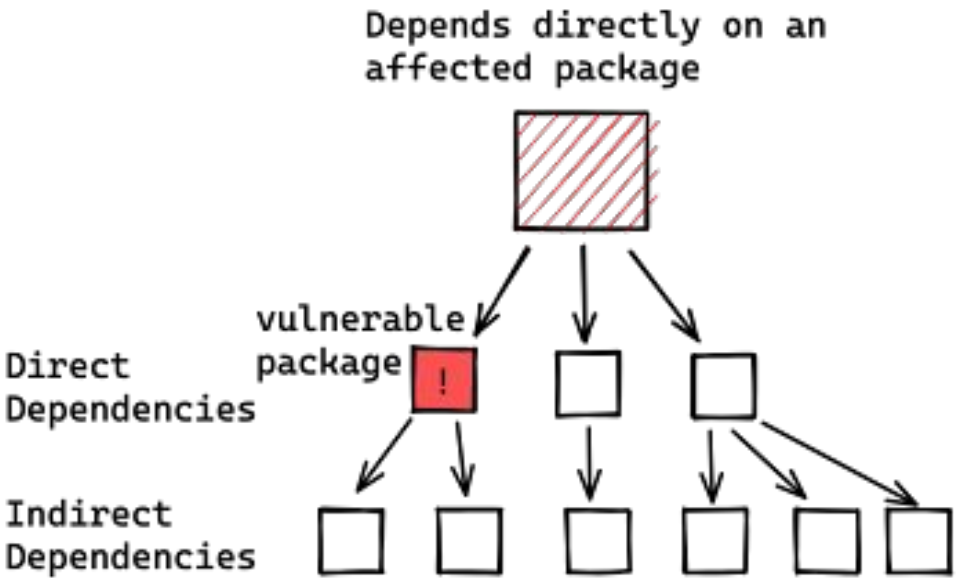
CVE-2021-44228, 10.12.2021 и CVE-2021-45046, 14.12.2021.
Выполнение произвольного кода на сервере
(Arbitrary Code Execution, ACE).

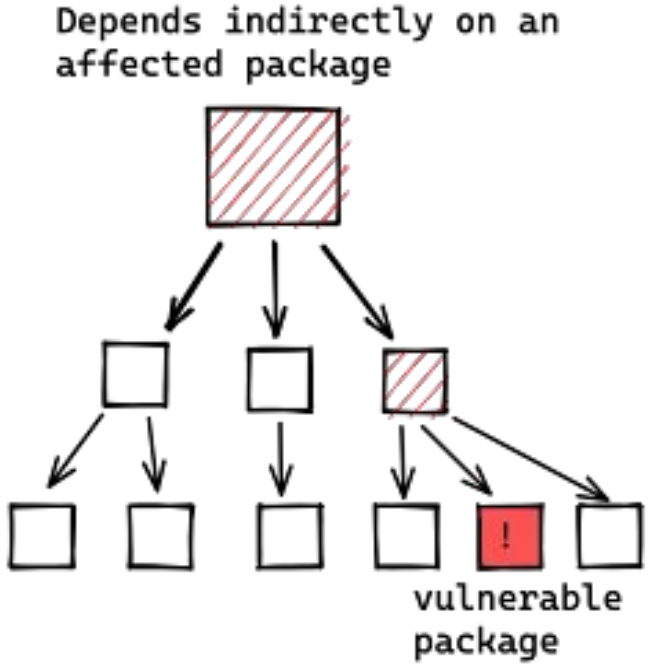


December 16, 2021. Google Security

log4shell. Глубже!

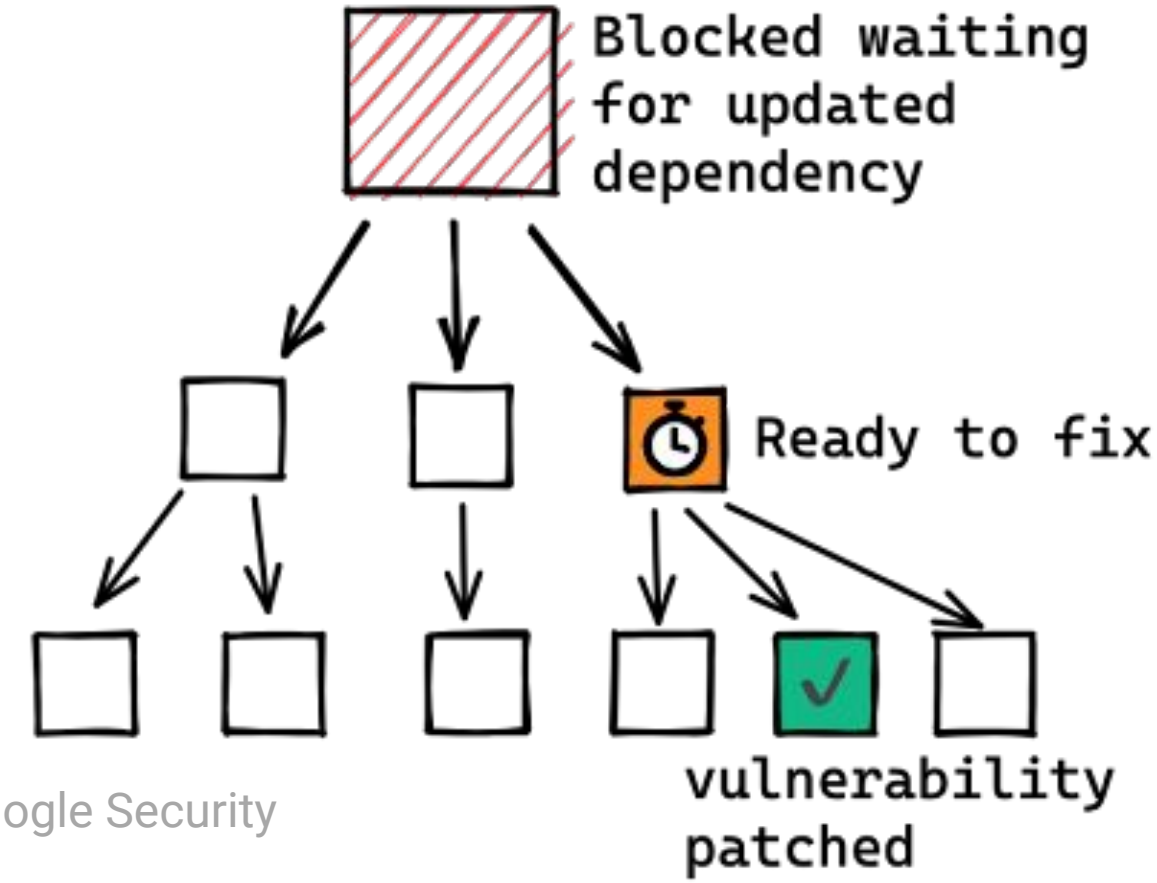
Уязвимый компонент может быть в транзитивных зависимостях.





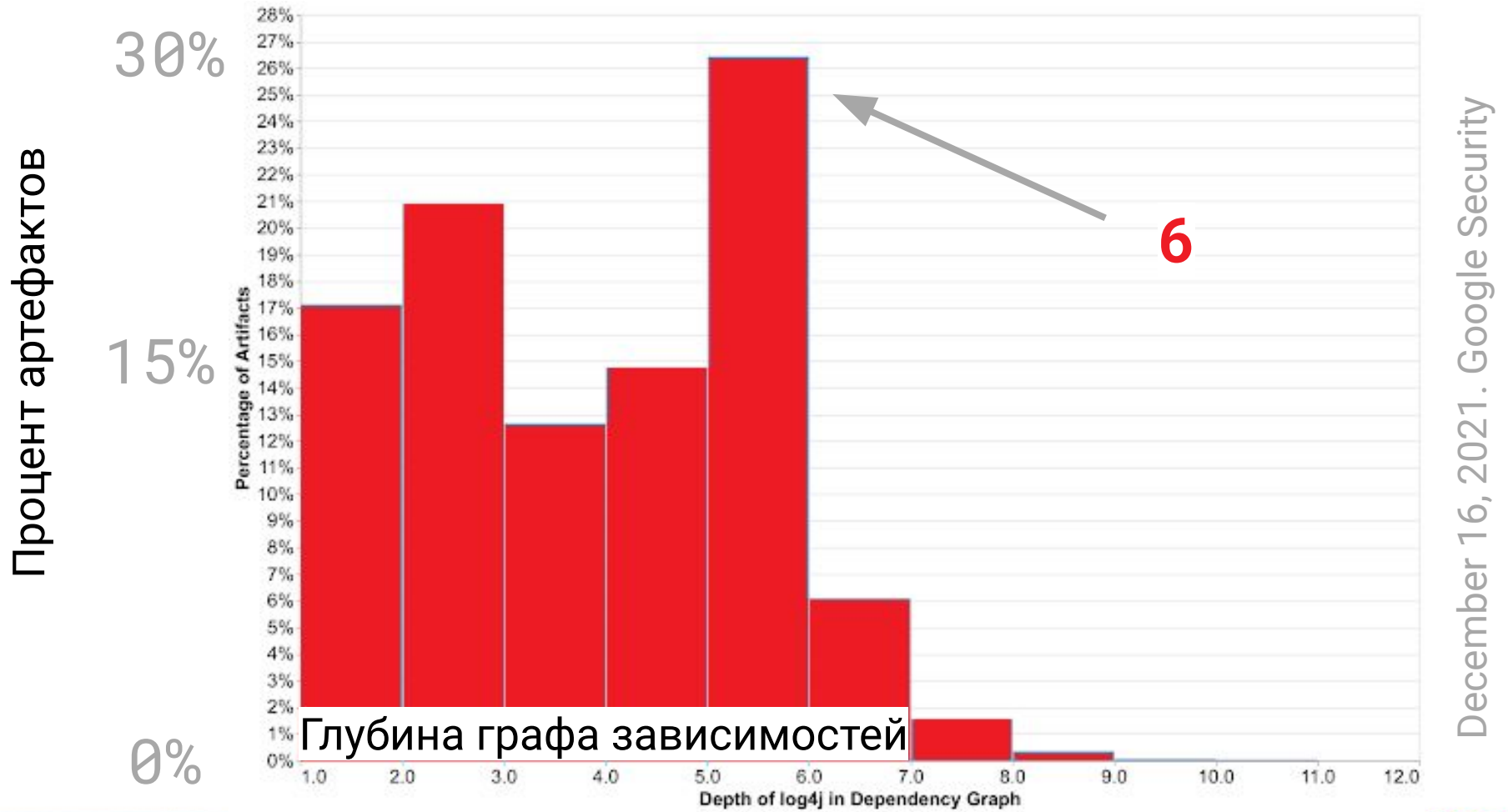
December 16, 2021. Google Security

log4shell. А кто починит транзит?



December 16, 2021. Google Security

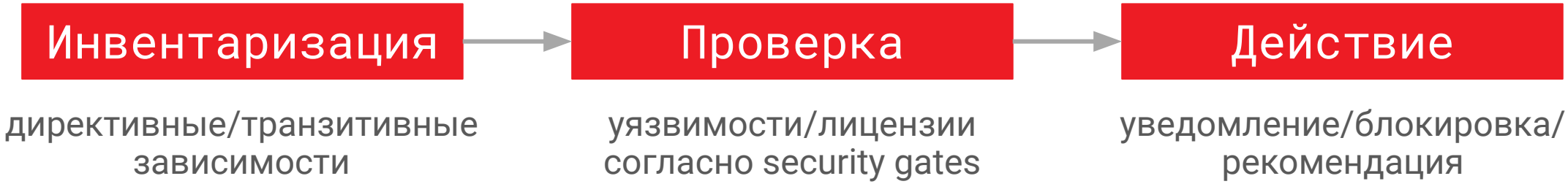
log4shell. Глубина проникновения



Software Composition Analysis, SCA

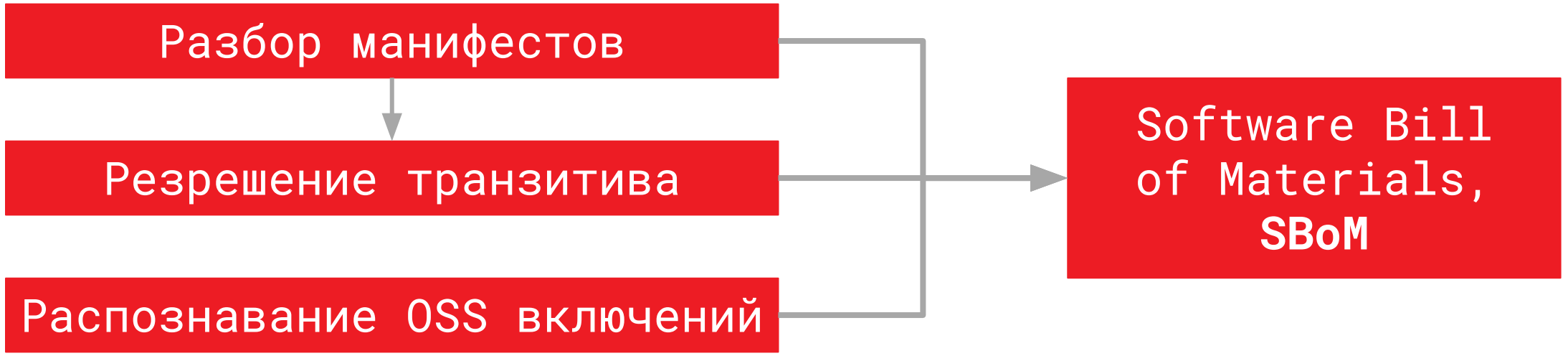
SCA — процесс определения компонентов, составляющих программное обеспечение.

SCA инструменты помогают в управлении рисками, связанными с безопасной разработкой.



Инвентаризация ПО, это как?

Найти манифесты, разобрать их, идентифицировать компоненты. А ещё бывают «включения» Open Source.



shazam для кода

Определение уязвимостей, это как?

Построить для компонента идентификатор и найти в какой-нибудь базе уязвимостей (CVE/GHSA/etc).

Основные используемые сегодня:

- Common Platform Enumeration, CPE
- Package URL, PURL

Common Platform Enumeration, CPE

CPE — идентификатор всех видов программного обеспечения (софта) и устройств. Атрибуты: part, vendor, product, version, update, edition, language, sw edition, target sw, target hw, etc. Матч уязвимостей идет **по словарю** :/.

```
Microsoft Internet Explorer 8.0.6001 Beta
cpe:2.3:a:microsoft:internet_explorer:8.0.6001:beta:*:*:*:*:*:*

Фреймворк Django 2.2.9
cpe:2.3:a:djangoproject:django:2.2.9:*:*:*:*:*:*:*

Фреймворк React 16.0.0-beta5
cpe:2.3:a:facebook:react:16.0.0:beta5:*:*:*:*:*:*
```

Package URL, PURL

Идентификатор составляют семь значений, объединённых в строку:

`scheme:type/namespace/name@version?qualifiers#subpath`

Примеры:

```
pkg:deb/debian/curl@7.50.3-1?arch=i386&distro=jessie
pkg:gem/ruby-advisory-db-check@0.12.4
pkg:github/package-url/purl-spec@244fd47e07d1004f0aed9c
```

Проблемы идентификации

PURL не без проблем, но ок и хорош.

Проблематика CPE Dictionary:

- словари неполные
- часто названия не соответствуют
- в большинстве словарных значений не указан язык программирования
- могут использоваться разные значения vendor и product для одного и того же софта разных версий

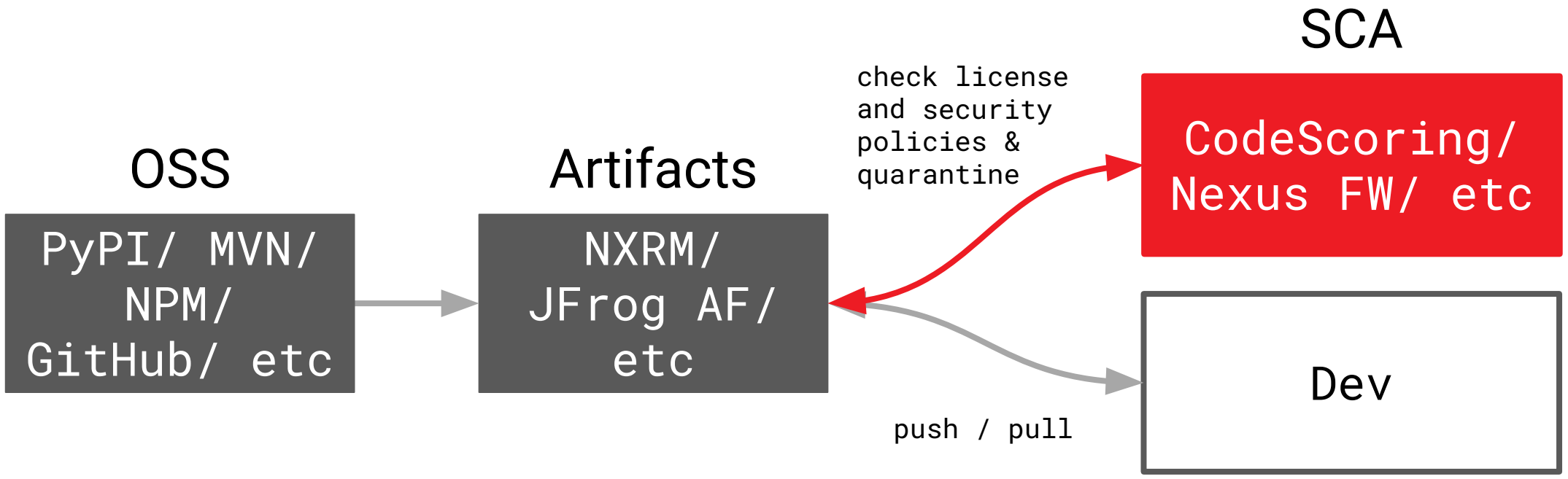
Для лучшей точности нужно применять NLP, ML и шлифовать rule-based.

Как обычно применяется OSA/SCA?

- **SCA Firewall**
  Блокирование нежелательных компонентов в прокси-репозиториях (хранилища артефактов)
- **CI/CD & security gates**
  Проверка и блокирование сборок в CI/CD pipelines (агент)
- **Continuous monitoring**
  Непрерывный мониторинг веток/тегов репозиториев (shift-left)

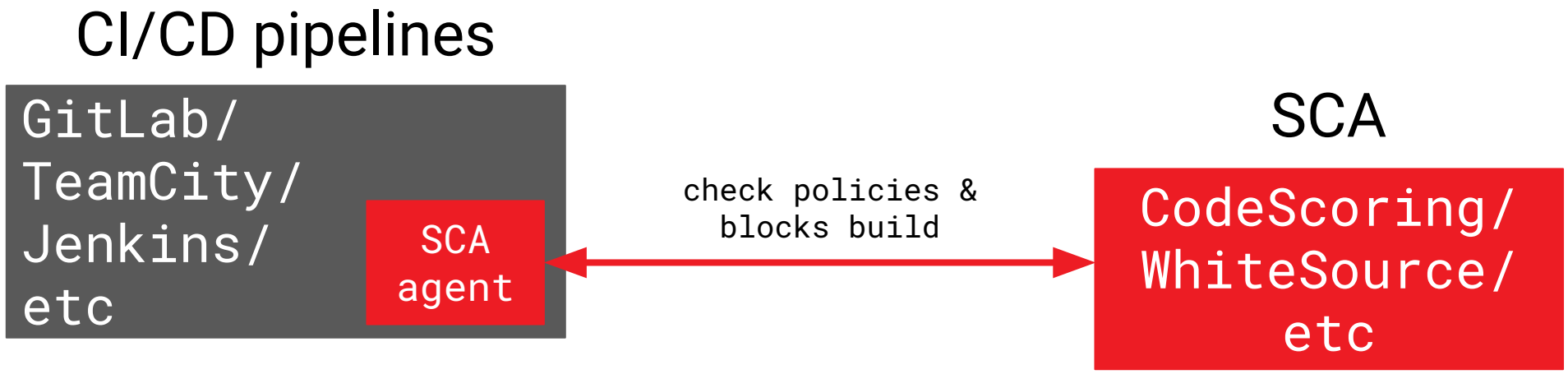
SCA.Firewall

Блокирование нежелательных компонентов в прокси-репозиториях (хранилища артефактов).



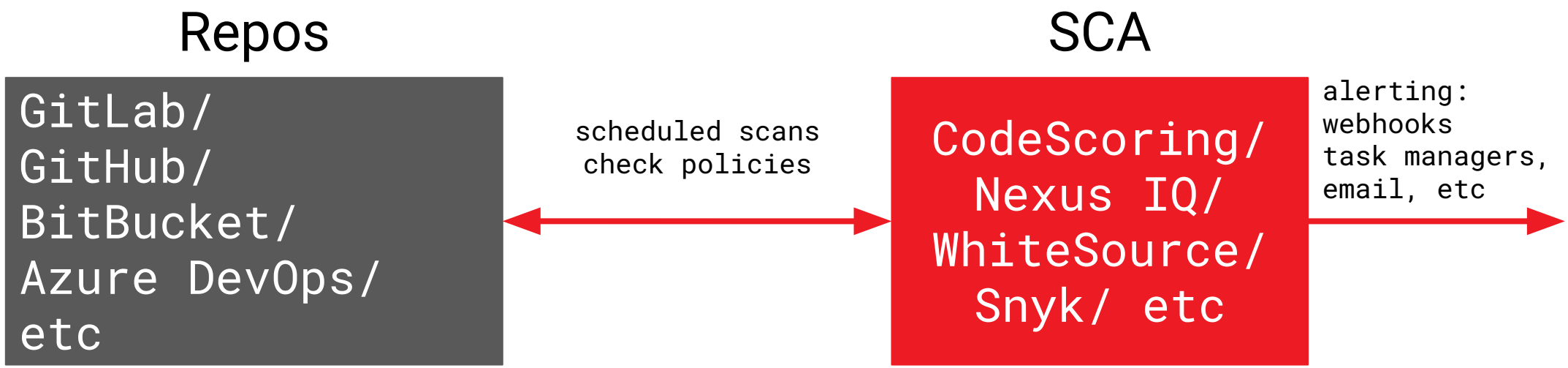
SCA.CI/CD

Проверка и блокирование сборок в CI/CD pipelines (агент).



SCA.Continuous monitoring

Непрерывный мониторинг веток/тегов репозиториев (shift-left).



Политики тоже бывают разные

IF This	Then That
• Найдена критическая уязвимость	Тогда Заблокировать сборку, поставить задачу на третью линию
• Появился пакет из стоп-листа	Тогда Поместить пакет в карантин
• Выпущена версия пакета после конкретной даты	Тогда Поместить в карантин и поставить задачу на SAST-ревью
• Нарушена лицензионная чистота или найдена неподходящая лицензия	Тогда Поставить задачу на замену библиотеки и уведомить тимлида и/или юристов

SCA — фундамент

Композиционный анализ дает нам основу и фундамент
для построения безопасных процессов разработки:

- безопасность компонентной базы, в первом приближении
- информированность об уязвимостях в фундаменте
- понимание лицензионной чистоты
- регулярный контроль ситуации
- автоматизация и понимание, что делать дальше

Спасибо за внимание

alexey@profiscope.io

profiscope.io - аудит
codescoring.ru - SCA

telegram:
@alsmirn - докладчик
@codemining - анализ кода