ecom.tech

Инвентаризация инструментов безопасности: как понять, что из текущего стека действительно нужно?

Нияз Кашапов

AppSec Team Lead

Строю appsec и немного devsecops

SSDLC на трех китах

Маршрутная карта внедрения

Первый этап

Внедрение инструментов и процессов

- Внедряем SAST
- Внедряем поиск секретов
- Внедряем SCA/OSA
- Внедрение процесса разбора уязвимостей

Второй этап

Закрепление и усиление практик

- Пилот Security Quality Gate
- Запуск обучения по безопасности кода
- Внедряем DAST
- Оптимизируем инструменты

Третий этап

Приведение метрик к целевым

- Распространение security Quality Gate на все сервисы
- Проведение аудита всех критичных сервисов
- Внедрение LLM для ускорения разбора найденных уязвимостей
- Внедрение автоматизированного сканирования на уязвимости

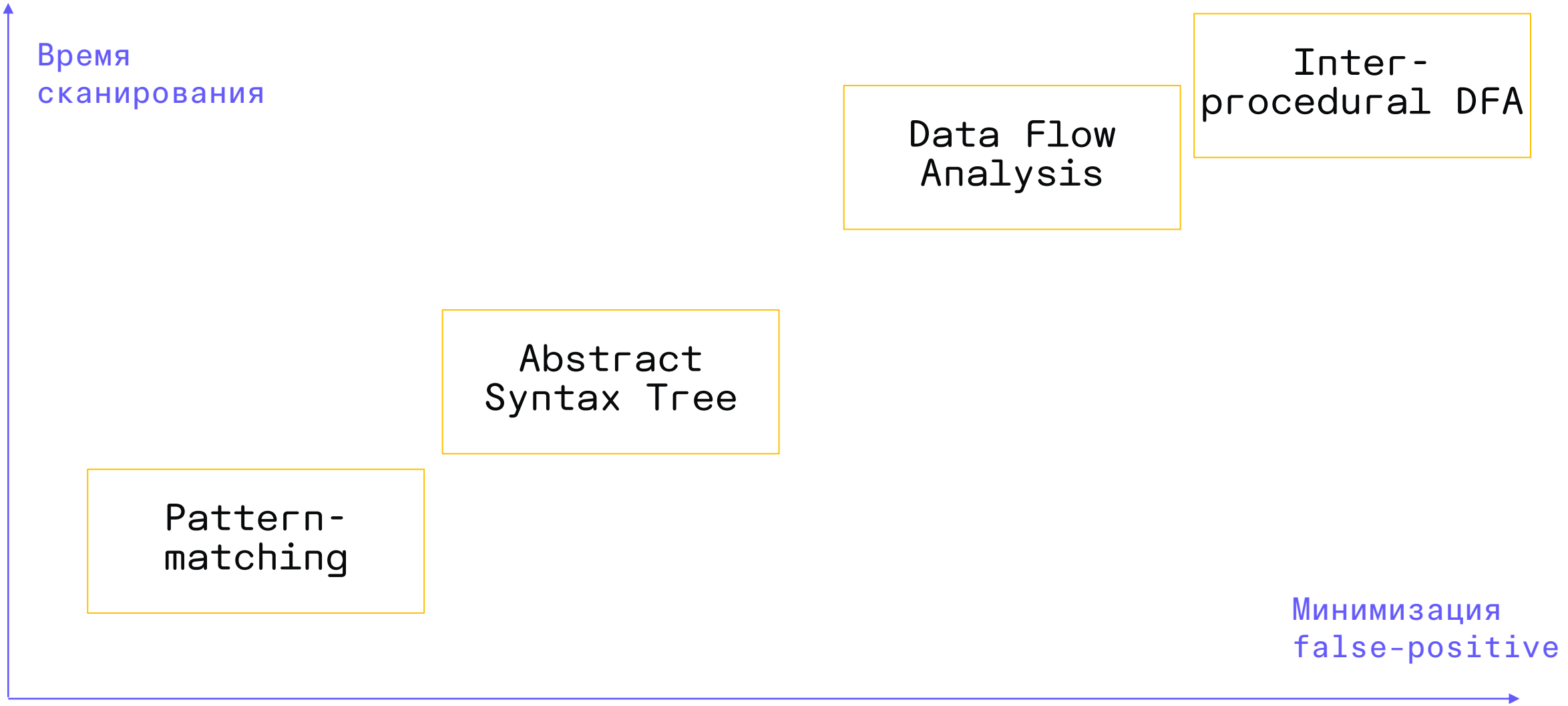
Четвёртый этап

Светлое будущее без уязвимостей

- Проверка кода сгенерированного LLM другой LLM
- Подсчет финансовых потерь от найденных уязвимостей
- Поиск уязвимого кода до пуша в ветку

Немного базы о SAST...

Типы SAST


Время сканирования
Pattern-matching
Abstract Syntax Tree
Data Flow Analysis
Inter-procedural DFA
Минимизация false-positive

SAST

- semgrep
- codeql
- bandit
- security code scan
- flawfinder
- spotbugs
- ESLint
- NodeJsScan
- MobSF

- PVS-studio
- checkmarx
- PT Application Inspector
- Solar appScreener
- codeql
- Sonarqube
- Veracode
- Codacy
- HCL AppScan
- Klocwork
- Reshift

Как оптимизировать практику SAST?

Как понять что текущий SAST не очень вам подходит?

Покрываем ли текущий SAST весь стек технологий?

Как часто SAST выдает false-positive?

Поддерживаем ли SAST кастомные правила?

Умеет ли SAST проводить taint-анализ кода?

Удобна ли автоматизация?

Почему плохо иметь множество SAST?

Нужна агрегация в едином месте (коррелировать и убирать дубли)

Нужна своевременная поддержка и обновления

Увеличивается время сканирования и триажа

Когда действительно нужно несколько SAST?

Единый инструмент не покрывает все ЯП

Есть более эффективные инструменты для специфичных задач

Нужен инструмент для настраиваемых правил

Пример когда нужен второй сканер

Платный сканер

Умеет в taint-анализ.

Сканирует долго.

Не умеет в инкрементальный анализ.

Бесплатный сканер

Имеет инкрементальный анализ.

Можно писать свои правила.

Покрывает +2 ЯП.

Используется вне пайплайнов

Вывод

Два и более сканера оправданы когда они друг друга дополняют или выполняют разные задачи

Как протестировать SAST в домашних условиях?

Тестируем SAST

True Positive Rate

Берем несколько уязвимых приложений на вашем стеке

Запускаем сканы

Разбираем сработки

Сравниваем с заявленными ошибками кода

False Positive Rate

Берем самый большой репозиторий в компании

Запускаем сканы

Разбираем сработки

Считаем количество неверных сработок на количество строк кода

Пример тестирования

https://github.com/OWASP/Vulnerable-Web-Application

	Уязвимость	Описание	Путь к файлу	Номер строки	Вызываемый метод	Критичность	Сканер 1	Сканер 2	Сканер 3
1	CWE-78	User input is passed to...	/src/CommandExecution/CommandExec-3.php	39		MEDIUM	+		
2	CWE-78	Command Injection	CommandExecution/CommandExec-1.php	25	$_GET["username"]	High	+	+	+
3	CWE-78	Command Injection	CommandExecution/CommandExec-2.php	25	$_GET["typeBox"]	High	+	+	+
4	CWE-78	Command Injection	CommandExecution/CommandExec-4.php	44	$_GET["typeBox"]	High	+	+	+
5	CWE-78	Command Injection	File: /src/CommandExecution/CommandExec-3.php:39	39	39 echo shell_exec($target);	CRITICAL			+
6	CWE-79	Cross-Site Scripting	CommandExecution/CommandExec-1.php	25	$_GET["username"]	High		+	+
7	CWE-79	Cross-Site Scripting	CommandExecution/CommandExec-2.php	25	$_GET["typeBox"]	High		+	+
8	CWE-79	Cross-Site Scripting	CommandExecution/CommandExec-3.php	39	$_GET["typeBox"]	High		+	+

Какие репы можно использовать?

https://github.com/securityelixir/potion_shop

https://github.com/optiv/InsecureShop

https://github.com/TheHackerDev/damn-vulnerable-golang

https://github.com/cr0hn/vulnerable-node

https://github.com/SasanLabs/VulnerableApp

SCA/OSA - оптимизируемо или нет?

В чем отличие SCA и OSA?

SCA

это анализ всего развернутого приложения и его зависимостей.

OSA

это проверка компонентов на этапе их загрузки разработчиком

На что смотрим при выборе OSA?

Поддержка используемого реестра компонент

Поддержка различных источников информации - OSS/OSV/Private

Настраиваемые политики

Аналитика использования компонент

Управление false positive

На что смотрим при выборе SCA?

Легкость интеграции с CI/CD

Совместимость с языками программирования и платформами

Скорость анализа и производительность

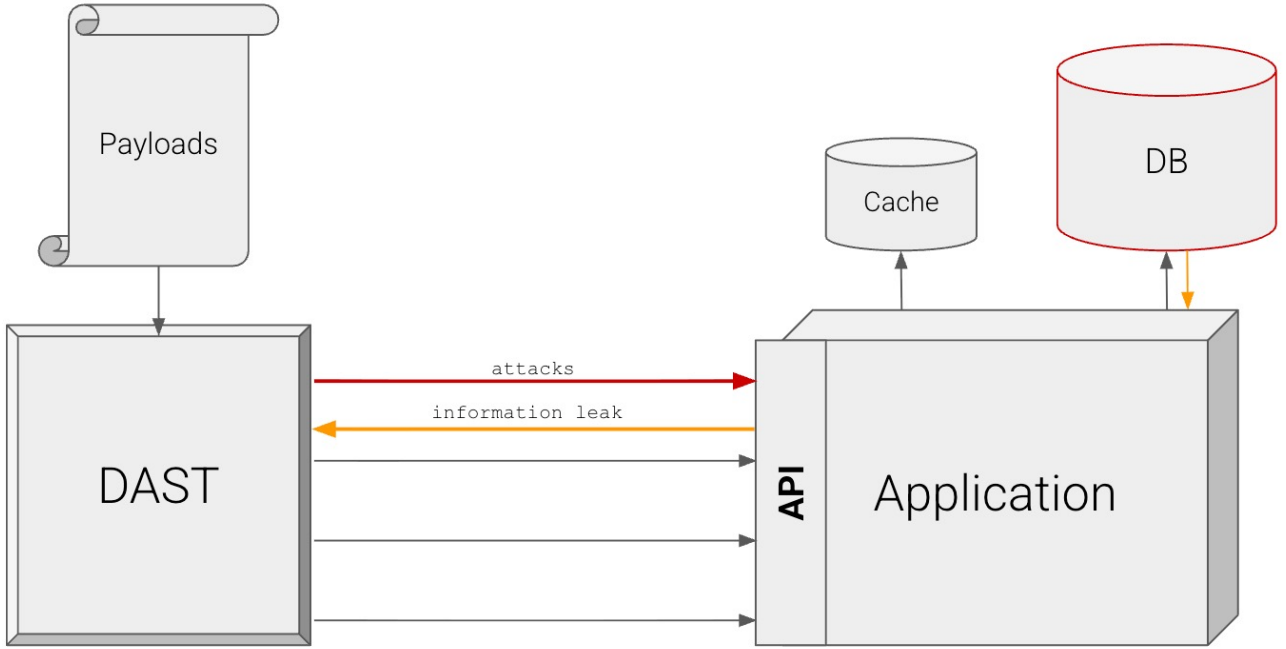
Поддержка license compliance

Цена и условия поддержки

DAST.
А надо ли оно вам?

DAST штука простая. Или нет?

Основа DAST - нагрузка. И анализ ответов для распознавания уязвимости



Тестируем DAST

True Positive Rate

Берем несколько уязвимых приложений на вашем стеке

Запускаем сканы

Разбираем сработки

Сравниваем с заявленными уязвимостями

False Positive Rate

Берем самый полный OpenAPI

Настраиваем аутентификацию

Запускаем сканы

Разбираем сработки

Пример тестирования

Уязвимость	Acunetix	Tenable.io	ZAP	NeuraLegion	Burp	StackHawk	Intruder
	4-	8-	9	4	8	7-	2
Tornado XSS в параметре https://<host>:7777/search.html	✗	✓	✓	✓	✓	✓	✓
Tornado Path Traversal https://<host>:7777/read/* Есть возможность смотреть системные файлы	✗	✓	✓	✗	✓	✓	✓
Tornado IDOR https://<host>:7777/static/* Есть возможность скачивать файлы не только с папки static	✗	✗	✗	✗	✗	?	✗
Tornado CSRF https://<host>:7777/upload	✓	✗	✓	✗	✗	✓	✗
Tornado IDOR в загрузке файлов https://<host>:7777/upload&file1=*	✗	✗	✗	✗	✗	✗	✗
Tornado Command Injection https://<host>:7777/server host	✗	✓	✓	✗	✓	✓	✗
Juice Shop Reflected XSS on search	✓	✗	✓	✗	✗	✗	✗

А что там с LLM?

Мы тут потестировали чуть-чуть…

Инструмент	Среда запуска	Найдено уязвимостей	Итог
Semgrep p/default	Локально	12	8/13
Semgrep p/ci	Локально	1	1/13
qwen2.5-coder-14b-instruct	LM Studio	6	5/13
qwen2.5-coder-3b-instruct	LM Studio	7	6/13
Mistral 8B	openrouter	9	9/13
lmstudio-community/deepseek-r1-distill-llama-8b	LM Studio	10	8/13
golang-ci-lint	Локально	8	8/13
deepseek/deepseek-r1-distill-qwen-32b (free)	openrouter	11	10/13
deepseek-r1:14b	ollama	9	7/13
deepseek-r1:7b	ollama	13	4/13
deepseek-r1:1.5b	ollama	2	2/13
deepseek-r1-distill-qwen-7b	LM Studio	8	2/13
DeepSeek R1 Distill Qwen 32B	openrouter	10	8/13

Вывод

Берем все инструменты, смотрим на них со *скепсисом*.

Оцениваем эффективность, тестируем со всех сторон.

Выбираем самый сбалансированный инструмент.
Дополняем его по необходимости.

ecom.tech

Вопросы

Нияз Кашапов @niazkashapov