МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГОРОДА ХАБАРОВСКА «ЛИЦЕЙ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ»

Информационная Безопасность

ПРОТОТИП НАДСТРОЙКИ К АНАЛИЗАТОРУ КОДА С ПОДРОБНЫМ ОБЪЯСНЕНИЕМ УЯЗВИМОСТЕЙ

ТИП ПРОЕКТА:
ПРАКТИКО-ОРИЕНТИРОВАННАЯ ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА

ВЫПОЛНИЛ:
УЧЕНИК 9А КЛАССА ШАРИПОВ ЕГОР

РУКОВОДИТЕЛЬ:
ПЕДАГОГ ШЕСТОПАЛОВ ДМИТРИЙ ВАСИЛЬЕВИЧ

2025г.

Введение

Индустрия программного обеспечения развивается стремительно. Вместе с ростом числа разработчиков растёт и количество ошибок в коде, в том числе связанных с информационной безопасностью.

Уязвимости в приложениях становятся одной из главных причин утечек данных и нарушений работы информационных систем. Многие такие ошибки возникают из-за отсутствия опыта у начинающих программистов.

Анализаторы кода позволяют автоматически находить потенциально опасные конструкции, но их отчёты сложны для понимания новичками.

В рамках проекта разработана интерактивная надстройка, которая делает отчёты анализатора понятными, обучающими и удобными для новичков. Подобных прототипов и аналогов нет.

Проблема

уязвимости в программном обеспечении, создаваемом новичками

Актуальность

Рост числа разработчиков приводит к росту количества программ, создаваемых новичками. При этом, отчёты исследований показывают критическое значение ошибок безопасности.

Таким образом, обучение безопасному программированию необходимо осуществлять прямо в процессе разработки, а не постфактум.

По данным IBM Security (2024), 52% утечек данных происходят из-за уязвимостей в программном обеспечении. OWASP ежегодно публикует список наиболее опасных ошибок, многие из которых типичны именно для начинающих.

Актуальность

Для поиска уязвимостей используются автоматические анализаторы кода. Однако часто их отчёты — сухие, технические, написаны на английском, содержат абстрактные коды ошибок. Это затрудняет понимание для новичков. В результате: ошибки либо игнорируются, либо «закрываются» просто ради прохождения проверки без понимания причин.

В то же время наблюдается рост популярности языков программирования, используемых в современных проектах с большим объёмом данных, сетевой и облачной инфраструктурой. Один из таких языков — Go (Golang). Выбор Go в качестве демонстрационной платформы в проекте обоснован его актуальностью, доступностью для новых разработчиков и широкой областью применения.

Таким образом, существует реальная потребность — сделать инструмент, который не просто находит уязвимости, но объясняет их на понятном языке, помогает учиться на ошибках, повышает культуру безопасной разработки и снижает риск появления уязвимого кода.

Цель проекта

Создать прототип обучающей надстройки к анализатору, которая преобразует результаты проверки кода в расширенные понятные пояснения, включающие описание уязвимостей, их последствия, примеры и рекомендации.

Задачи проекта

- Проанализировать существующие подходы к статическому анализу кода и изучить основные возможности инструмента gosec
- Определить перечень наиболее распространённых и значимых уязвимостей в проектах на Go и сформировать структуру обучающей базы знаний
- Разработать архитектуру надстройки, обеспечивающую обработку JSON-отчётов gosec и формирование структурированного обучающего вывода
- Создать программный прототип (MVP) консольного приложения, преобразующего стандартный отчёт gosec в форматированный обучающий отчёт
- Подготовить набор тестовых файлов на языке Go, содержащих типичные уязвимости, для демонстрации работоспособности разработанного прототипа
- Провести тестирование прототипа, оценив корректность отображения уязвимостей и полноту пояснений
- Определить направления дальнейшего развития проекта.

Объект исследования:

анализаторы кода, которые ищут ошибки и дефекты безопасности, в частности для проектов на языке GO

Предмет исследования:

механизмы работы анализаторов

Продукт:

прототип надстройки

Теоретическая часть
Новизна проекта

Сочетание стандартного анализатора + обучающего интерфейса.
Проект делает анализ ошибок частью обучения, что отсутствует в большинстве SAST-решений:

- не просто выявляет ошибки — объясняет их простым и понятным языком.
- ориентирован на начинающих разработчиков, которые ещё не знакомы с тонкостями безопасности.
- превращает проверку кода в процесс обучения, а не просто «файловую» проверку.

Популярность современных языков программирования и обоснование выбора технологической платформы

Go современный, популярный, быстрорастущий, активно используется в backend, облаках, инфраструктуре, имеет качественный анализатор безопасности (gosec- хороший пример «сложных для новичка» сообщений).

Таким образом, реализация проекта на примере Go даёт: практическую применимость (многие реальные проекты используют Go)
и хорошую демонстрацию того, зачем нужен подобный инструмент,
а так же возможность в будущем расширить поддержку на другие языки.

Практическая часть

Изучение принципов работы анализатора gosec и форматов его отчётов.

Защита программного обеспечения начинается с раннего выявления ошибок проектирования и реализации.
Одним из ключевых методов является статический анализ кода — “SAST” (Static Application Security Testing), позволяет обнаруживать уязвимости без запуска программы.

Именно он рассматривается в рамках проекта, так как предоставляет формализованные отчёты, которые можно автоматически преобразовывать в обучающий формат, что важно для создания учебной надстройки.

Проблемы использования gosec в учебных проектах и среди начинающих разработчиков

Несмотря на широкую распространённость, выводы gosec ориентированы на опытных специалистов и автоматизированные системы, а не на начинающих разработчиков. Это создаёт ряд существенных затруднений.

Он хорошо выполняет диагностическую функцию, но не способствует обучению и не помогает понять природу уязвимостей.

Это создаёт потребность в надстройке, которая переводит технический отчёт в понятный человеку формат — с примерами, объяснениями и рекомендациями.

JSON как основной формат данных в отчётах gosec

Такой формат делает отчёты gosec удобными для последующей автоматической обработки, что важно для интеграции с внешними инструментами.

Пример простого JSON-фрагмента:

```
{
  "rule_id": "G401",
  "severity": "MEDIUM",
  "line": 12
}
```

Структура JSON-отчёта gosec

Представляет собой список объектов, каждый из которых содержит сведения о конкретной найденной уязвимости.

Наиболее значимые поля включают:

"rule_id" — идентификатор правила gosec;
"details" — краткое описание обнаруженной проблемы;
"severity" — уровень серьёзности;
"confidence" — уровень уверенности инструмента;
"file" и "line" — местоположение уязвимости в исходном коде;
"code" — фрагмент кода, вызвавший предупреждение;
"cwe" — при наличии, ссылка на номер в базе CWE (Common Weakness Enumeration).

Выбор уязвимостей на основе исследований и его научно-статистическое обоснование

Для формирования обучающей надстройки важно опираться на объективные данные о том, какие уязвимости в программном обеспечении являются наиболее опасными и наиболее часто встречаются в реальных проектах.
В качестве основы выбора использованы международные рейтинги и аналитические отчёты. Это позволило определить набор уязвимостей, которые имеют наибольшую практическую значимость и актуальность.

Структура обучающих материалов и формат хранения данных

Для каждой выбранной уязвимости подготовлена отдельная обучающая запись, включающая:
идентификатор (CWE и/или правило gosec), описание сути уязвимости, объяснение угроз и возможных последствий, рекомендации по устранению. Такая структура обеспечивает системное и последовательное представление информации.

На этапе создания прототипа обучающие материалы хранятся в виде структурированных записей (словаря или JSON-объекта) в самой программе. Это обеспечивает удобство обработки и возможность дальнейшего расширения базы без изменения принципов работы приложения.

Тестирование

Проводилось на:
намеренно ошибочных примерах,
учебных проектах,
небольших функциях с уязвимостью.
Проверено:
корректность отображения ошибок,
соответствие объяснений,
удобство интерфейса.

Области применения

- ✓ Учебные курсы
- ✓ IT-кружки
- ✓ Индивидуальная подготовка
- ✓ Малые команды разработчиков
- ✓ Проектная деятельность школьников и студентов

Заключение

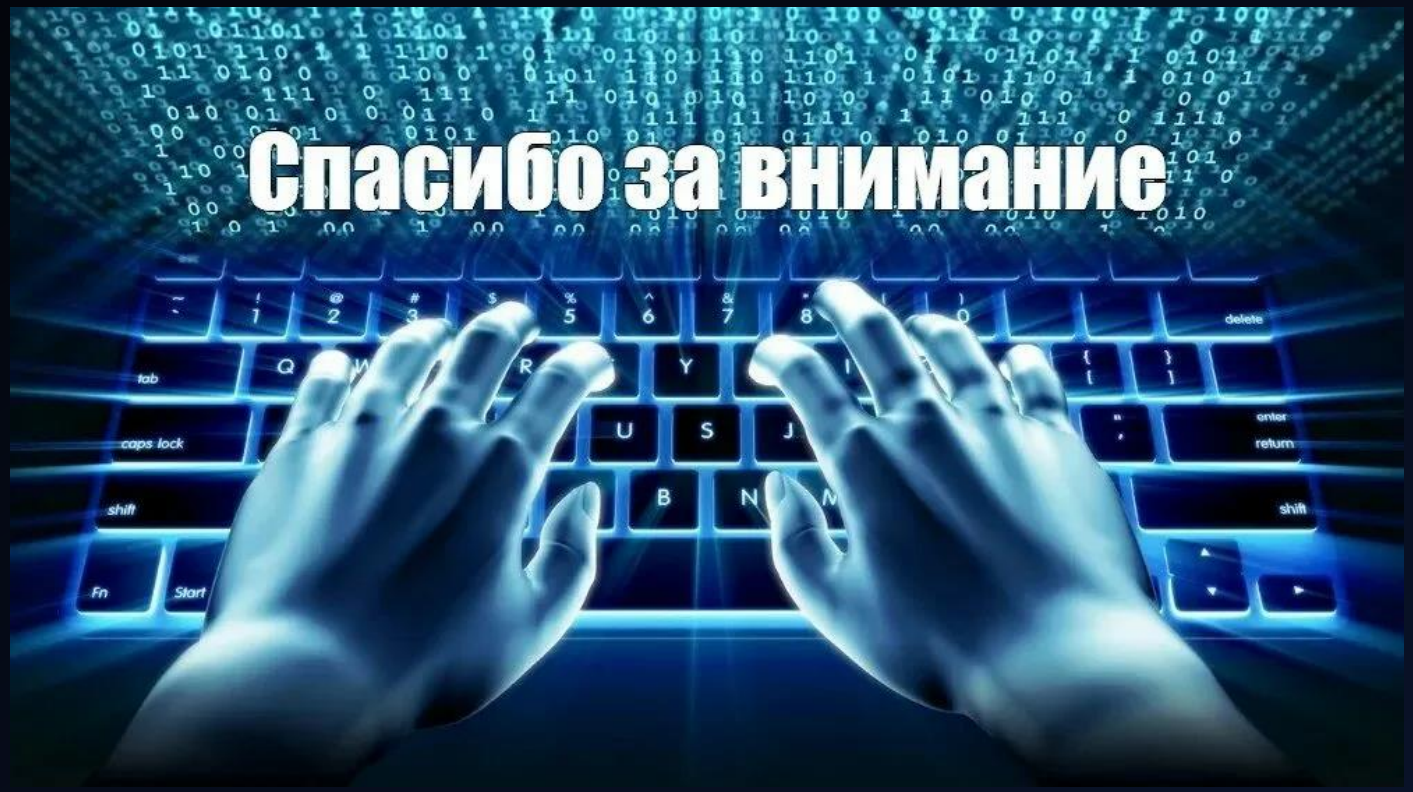
В ходе выполнения проекта была разработана обучающая надстройка к анализатору gosec, предназначенная для улучшения объяснения результатов статического анализа кода на языке Go.

Созданный прототип выполняет автоматический запуск gosec, обрабатывает отчёт в формате JSON, сопоставляет найденные уязвимости с базой знаний и отображает подробные обучающие пояснения. Тестирование подтвердило корректность работы программы: все тестовые уязвимости успешно определены, а пояснения отображены в полном и понятном виде.

Прототип демонстрирует возможность превращения анализа кода в образовательный процесс, что особенно важно для начинающих разработчиков.

В перспективе возможны такие улучшения, как расширение базы знаний, поддержка всех отчетов gosec, использование инструмента для других языков программирования и анализаторов.

Проект достиг своей цели: создана работоспособная обучающая надстройка, подтверждающая эффективность предложенного подхода и демонстрирующая практическую значимость в сфере информационной безопасности.


Спасибо за внимание