

**Испытания статических анализаторов по требованиям
ГОСТ Р 71207-2024
АК-ВС 3. Опыт АО «НПО «Эшелон»**

AK-BC 3

«**AK-BC 3**» — это инструмент, который позволяет автоматизировать процесс:

- проведения **испытаний при сертификации** по требованиям **различных регуляторов**
- **проведения периодических проверок** в соответствии с требованиями **разработки безопасного ПО**



ТРЕБОВАНИЯ ФСТЭК России

Приказ ФСТЭК России от 2 июня 2020 г. N 76

- п. 19 «Испытания по выявлению уязвимостей и недеklarированных возможностей средства»

Методика ВУ и НДВ

- п. 4.2 «Статический анализ объекта оценки (САО)»
- п. 4.3.2 «Фаззинг-тестирование объекта оценки (ДАО.2)»
- п. 4.3.3 «Системное тестирование объекта оценки (ДАО.3)»

ГОСТ Р 56939-2024 «Разработка безопасного ПО»

- п. 5.10 «Статический анализ исходного кода»
- п. 5.11 «Динамический анализ исходного кода»

ГОСТ Р 71207-2024 «Статический анализ программного обеспечения»

- п. 8 «Требования к инструментам статического анализа»

Материалы по АК-ВС 3



Сравнительный анализ и выбор
статических анализаторов
безопасности кода



Анализатор кода вычислительных
систем АК-ВС 3. Комплексное решение
для обеспечения безопасности кода в
современных разработках



Как работают инструменты
статического и динамического
анализа кода. Опыт использования
АК-ВС 3

Испытания стат анализаторов. Плюсы

- Первая попытка провести испытания, чтобы оценить уровень анализаторов
- Возможность на этапе домашнего задания оценить недостатки
- Возможность доработать анализатор по результатам домашнего задания
- Возможность у каждого вендора предоставить свои тесты
- Получение обратной связи от жюри

Испытания стат анализаторов. Минусы

- Сжатые сроки
- Отсутствие унификации в тестах, некорректные тесты
- Сложный маппинг детекторов на требования ГОСТ
- Необходимость вручную анализировать результаты тестов
- Отсутствие унифицированного формата SARIF

SARIF

На этапе домашнего задания у жюри **возникли трудности оценивать результаты в формате sarif**. В итоге было принято решение **отказаться от оценки sarif** и в будущем **сделать подробную спецификацию**.

Пример нашей спецификации:

Line ↓	Message
2751	Игнорирование ошибок. Игнорирование исключений может позволить злоумышленнику оставить неожиданное поведение незамеченным.
3281	Игнорирование ошибок. Игнорирование исключений может позволить злоумышленнику оставить неожиданное поведение незамеченным.

INFO

ANALYSIS STEPS 0

STACKS 0

Игнорирование ошибок. Игнорирование исключений может позволить злоумышленнику оставить неожиданное поведение незамеченным.

Rule Id	4a4fb16e-c588-4c24-8a9c-4d2047dbe402
Rule Name	—
Rule Description	Игнорирование ошибок
Level	warning
Kind	—
Baseline State	new
Locations	ma.4.cs
Log	defects%20%282%29.sarif
Suppression	not suppressed

Примеры некорректных тестов

Большая часть тестов была разработана при помощи автогенерации, поэтому в результате появлялись **дополнительные ошибки**, или **достижимое срабатывание**, которое **необходимо было пропустить**

```
test_10_cwe15_External-control-of-system-or-configuration-  
settings_diamond-and.c"
```

```
void func(int param) {  
    char name[1024];  
  
    if (param & 0x1D) {  
        scanf("%s", name); //Возможно переполнение буфера  
    }  
  
    if (param & 0x1D) {  
        sethostname(name, sizeof(name)); // FLAW  
    }  
}
```

```
test_6cfcf636b179__cwe89__sqlite3-execute-  
f__while__ispras
```

```
import sqlite3  
  
def func(username):  
    connect = sqlite3.connect("users.db")  
    cursor = connect.cursor()  
    error_flag = False  
  
    query = f"SELECT * FROM users WHERE username = '{username}'"  
    error_flag = True  
  
    while (error_flag):  
        error_flag = False  
        cursor.execute(query)  
  
    return cursor.fetchall()
```


СПАСИБО ЗА ВНИМАНИЕ!

Арустамян Сас Сергеевич,
Директор Центра оценки соответствия и тестирования
АО «НПО «Эшелон»
sa@cnpo.ru