Александр Дроздов

Инженер по SDL и информационной безопасности технологий
Axiom JDK

Безопасная разработка программной платформы Java

Java это

Версии: LTS 8, 11, 17

Аппаратные платформы: x86_32, x86_64, Aarch64, Байкал и другие

Операционные системы: Windows Server, Windows, Ubuntu Linux, Debian Linux, RHEL, CentOS, Astra Linux и другие

Функции безопасности:

1. Обеспечение независимости экземпляров виртуальных машин
2. Верификация class-файлов
3. Безопасное выполнение интерпретируемого кода
4. Управление доступом
5. Контроль целостности исполняемого кода (замкнутая программная среда)
6. Регистрация событий безопасности
7. Очистка памяти

Сертификация

- 4 УД, в соответствии с требованиями ФСТЭК России
- Верификация 12 миллионов строк
- Концепция процесса безопасной разработки ПО (SDL)
- Доработан сборщик мусора

Безопасность исходного кода

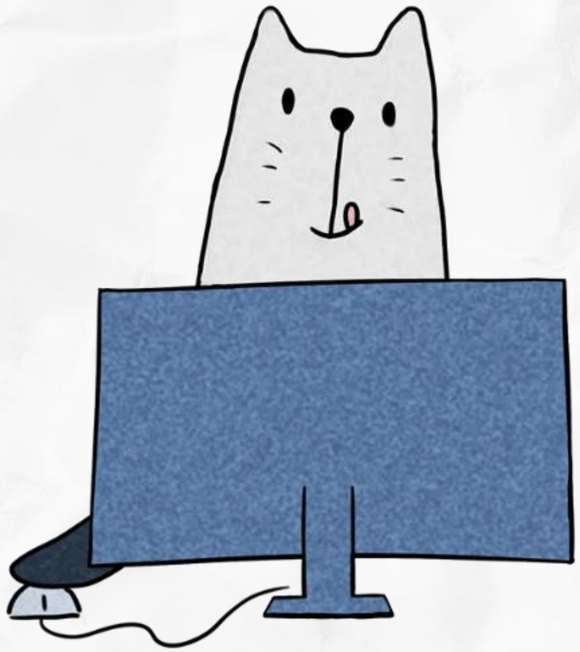
Svace - для анализа исходных текстов

Более 100 000 срабатываний

Стратегия обработки - Critical срабатывания

26 дефектов исправлены в JDK 8

22 дефекта исправлены в JDK 11



Фаззинг в JDK

libjavajpeg

JVM

```
int __attribute__((visibility("default")))
read_JPEG_file(char * filename) {

    struct jpeg_decompress_struct cinfo;
...
infile = fopen(filename, "rb")) == NULL)
...
jpeg_stdio_src(&cinfo, infile);
...
read_JPEG_file(argv[1]);
```

afl-clang-lto

ASAN UBSAN TSAN CFISAN MSAN

Необходимо учитывать работу кода в контексте JVM

Создана обертка с учётом JNI

Стало доступным использование afl-clang-lto

Удобно инструментировать

Хорошая скорость (~1000 зап/сек для каждого фаззера)

Фаззинг в JDK (продолжение)

afl-clang-lto

ASAN UBSAN TSAN CFISAN

libfontmanager + libfreetype

JVM

Необходимо учитывать работу кода в контексте JVM

Цель слишком сильно вплетена в JNI

Создаем JVM с помощью JNI, запускаем тестовый метод в persistent loop после создания JVM

afl-clang-lto + удобно инструментировать

Хорошая скорость (~1000 зап/сек)

```
long flag = JNI_CreateJavaVM(&javaVM, (void**)
      &jniEnv, &vmArgs);
...
methodId = jniEnv->GetStaticMethodID(jcls,
         "readFont", "(Ljava/lang/String;)V");
...
while (__AFL_LOOP(UINT_MAX)) {
...
       jniEnv->CallStaticVoidMethod(jcls,
methodId, jniEnv->NewStringUTF("./cur_input"));
```

Санитайзеры

Выполняем большие наборы jtreg тестов

Используем инструментированную санитайзерами виртуальную машину Java (JVM)

ASAN UBSAN

Настроили jtreg и санитайзеры для совместной работы

Готов ответить
на ваши вопросы

E-mail:
alexander.drozdov@axiomjdk.ru