

Анализ поверхности атаки Java-приложений

Довгалюк Павел

ИСП РАН

План доклада

- Как и зачем искать поверхность атаки
- Немного хардкора про JVM
- Поиск ошибок в Hello World (spring-petclinic)

Natch

- **Автоматизированный анализ поверхности атаки**
 - Интерфейсы приложений: адреса, файлы, порты
 - Компоненты и функции программ: библиотеки и собственный код
- **x86_64 и немножко AArch64**
- **C/C++, Go, Python, Java, JavaScript**
- **Регрессионное тестирование**
- **Отчёты для аналитиков и тестировщиков**

Зачем искать поверхность атаки (с помощью Natch)

- **Повышение защищённости**
- **Выявление потенциально уязвимых компонентов**
- **Сокращение уязвимых мест**
- **Приоритизация тестирования**
- **Дополнение результатов статического анализа**

Статический и динамический анализ

- **Статический**

- Без запуска приложения
- Весь код доступен
- Неизвестно, как в реальности идёт выполнение
- Входные данные неизвестны
- Сложно анализировать взаимодействие программ

- **Динамический**

- Программа работает в обычном режиме
- На входе реальные данные
- Не все функции активируются
- Сценарии работы могут быть длительными
- Замедление работы из-за анализа

Статический анализ: SBOM

```
grep "\"name\"" application.cdx.json
```

```
"name" : "spring-boot-starter-web",  
"name" : "spring-boot-starter-json",  
"name" : "jackson-datatype-jdk8",  
"name" : "spring-boot-starter-tomcat",  
"name" : "tomcat-embed-core",  
"name" : "tomcat-embed-websocket",  
"name" : "spring-web",  
"name" : "spring-webmvc",  
"name" : "spring-expression",  
"name" : "spring-boot-starter-validation",  
"name" : "tomcat-embed-el",  
"name" : "hibernate-validator",  
"name" : "jakarta.validation-api",  
"name" : "spring-boot-starter-thymeleaf",  
"name" : "thymeleaf-spring6",  
"name" : "thymeleaf",  
"name" : "attoparser",  
"name" : "unbescape",
```

...

```
grep "\"type\" : \"library\"" ./application.cdx.json | wc -l
```

```
92
```

Статический анализ: OWASP Noir

POST /owners/new

body: address=&city=&telephone=

file: ./src/main/java/org/springframework/samples/petclinic/owner/**OwnerController.java (line 75)**

POST /owners/{ownerId}/edit

path: ownerId

body: address=&city=&telephone=

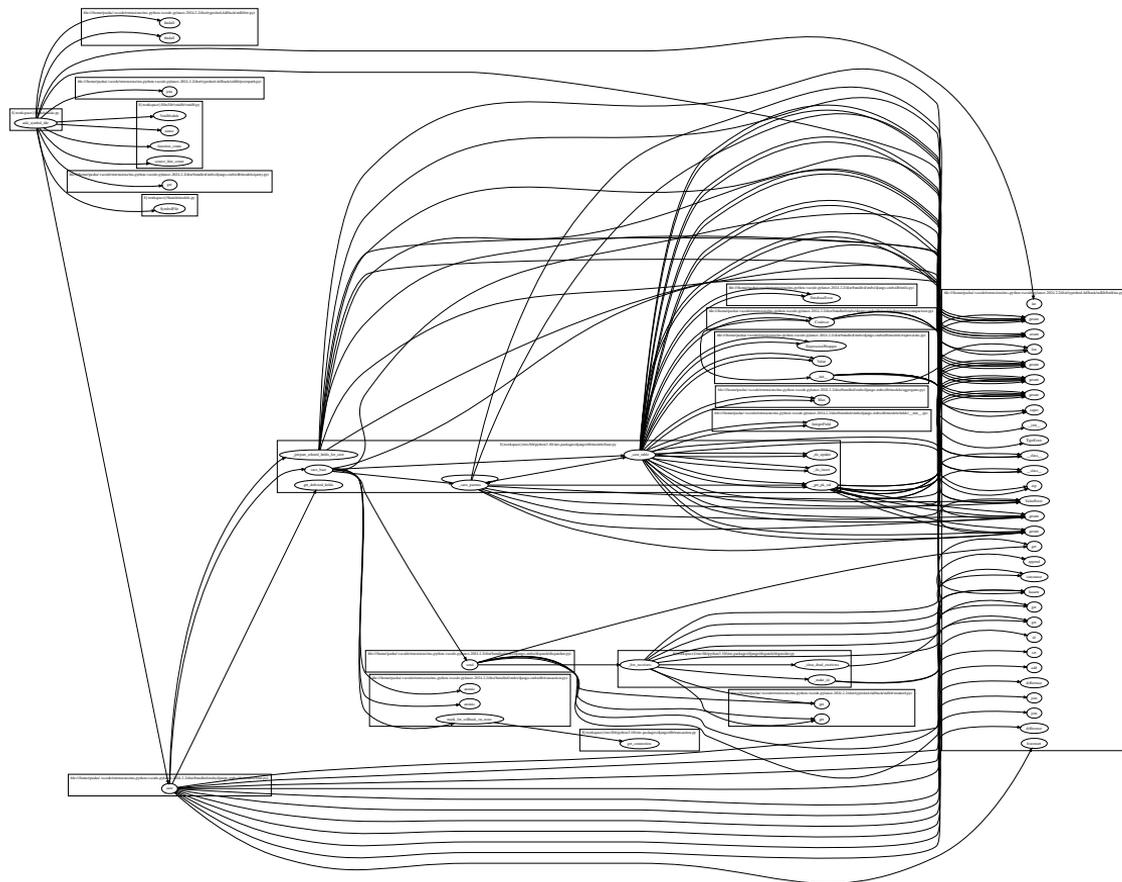
file: ./src/main/java/org/springframework/samples/petclinic/owner/**OwnerController.java (line 138)**

GET /owners/{ownerId}

path: ownerId

file: ./src/main/java/org/springframework/samples/petclinic/owner/**OwnerController.java (line 163)**

Статический анализ: call graph



Динамический анализ

- Можно увидеть подробности о конкретном сценарии работы
- Более сложная настройка, чем при статическом анализе
- Нужно много времени на анализ
- Мало доступных инструментов
 - Natch – инструмент анализа поверхности атаки
 - PANDA – фреймворк для полносистемного анализа

Ограничения / требования

- **Системы из разнородных компонентов**
 - mysql + nginx + django + keycloak + ...
- **Полносистемное выполнение**
 - эмулятор QEMU

Динамический анализ – это сложно

- **Полносистемный анализ**
 - Много процессов
 - Низкоуровневая информация о выполнении
- **Отслеживание потока управления и потоков данных**
 - Необходимо динамическое инструментирование кода
- **Большой оверхед**
 - Анализ занимает время

Поиск поверхности атаки

- **Какие данные важны?**
- **Куда они не должны проникать?**
- **Как они должны обрабатываться?**

- **Все потоки данных сразу отследить невозможно**

Анализ потоков данных

- Инструментируются все машинные инструкции
- Добавляется теневая память для пометок данных
- Помечаются для отслеживания входные данные
- Копирование данных приводит к копированию пометок

Теневая память: помеченные данные

Память

H	e	l	l	o	,	W	o
r	l	d	!				

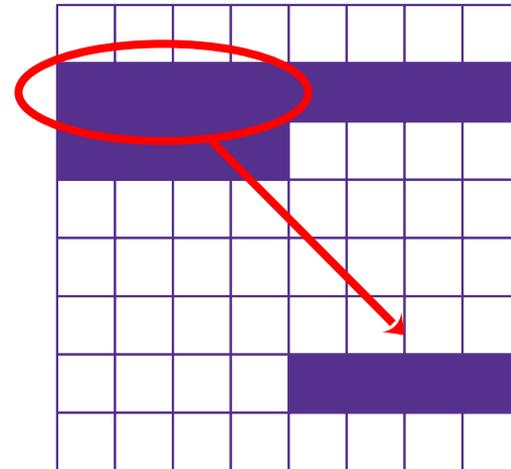
Теневая память

Теневая память: копирование данных

Память



Теневая память



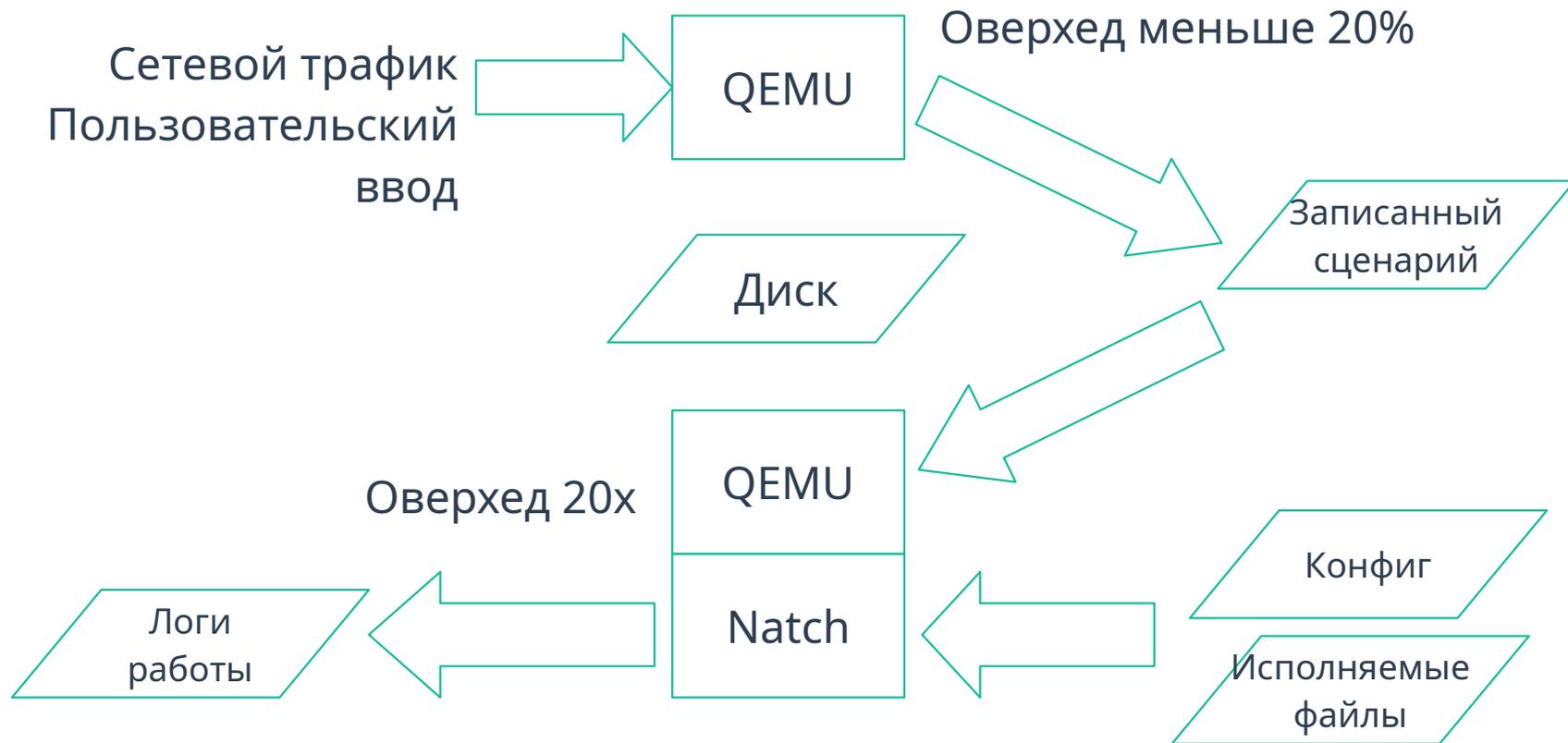
Оверхед при анализе

- **Без оверхеда никак**
- **Valgrind memcheck – инструмент для поиска ошибок работы с памятью**
 - “adds code to check every memory access and every value computed, making it run 10-50 times slower than natively”
- **Оверхед может помешать правильной работе программы**
 - пользователю неудобно работать
 - возникают сетевые задержки

Борьба с оверхедом: запись и воспроизведение

- **Сценарий работы пользователя в ВМ записывается**
- **Для анализа сценарий воспроизводится**
- **Полностью восстанавливается последовательность инструкций**

Запись и воспроизведение



Динамический анализ – всё ещё сложно

- **Неизвестно, что происходит внутри VM**
 - Нужны адреса загруженных исполняемых файлов
- **Не обойтись без отладочных символов**
- **Для анализа Java-приложений даже отладочных символов мало**
 - Символы не описывают то, что происходит в jit-коде

Java

- **Виртуальная машина Hotspot**
- **Байткод компилируется в машинные инструкции**
- **Даже трамплины в JIT-код “собираются” в рантайме**
- **Нужно парсить внутреннее представление класса, чтобы извлечь информацию о вызываемом методе**

Точки входа в методы

- **Трамплины**

- `jump_from_interpreted`
- `generate_math_entry`
- ...

- **Компиляторы**

- `new_nmethod`
- `generate_call_stub`
- `generate_ic_miss_helper`
- ...

jump_from_interpreted

```
void InterpreterMacroAssembler::jump_from_interpreted(Register method, Register temp) {  
    prepare_to_jump_from_interpreted();  
  
    if (JvmtiExport::can_post_interpreter_events()) {  
        ...  
        jmp(Address(method, Method::interpreter_entry_offset()));  
        bind(run_compiled_code);  
    }  
  
    jmp(Address(method, Method::from_interpreted_offset()));  
}
```

Получение названия метода (1)

```
class Method {  
    ...  
    ConstMethod* _constMethod; // Method read-only data.  
    ...  
};
```

```
class ConstMethod {  
    ...  
    ConstantPool* _constants;  
    u2 _name_index;  
    u2 _signature_index;  
    ...  
};
```

ConstantPool

`javap -v FuzzerHtml.class`

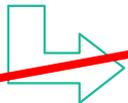
Constant pool:

```
#1 = Methodref      #2.#3      // java/lang/Object.<init>():V
#2 = Class          #4          // java/lang/Object
#3 = NameAndType    #5:#6      // "<init>":()V
#4 = Utf8           java/lang/Object
#5 = Utf8           <init>
#6 = Utf8           ()V
#7 = InterfaceMethodref #8.#9      // com/code_intelligence/jazzer/api/FuzzedDataProvider.consumeRemainingAsString():Ljava/lang/String;
#8 = Class          #10         // com/code_intelligence/jazzer/api/FuzzedDataProvider
#9 = NameAndType    #11:#12     // consumeRemainingAsString():Ljava/lang/String;
#10 = Utf8          com/code_intelligence/jazzer/api/FuzzedDataProvider
#11 = Utf8          consumeRemainingAsString
#12 = Utf8          ()Ljava/lang/String;
```

Получение названия метода (2)

```
class ConstantPool : public Metadata {  
    ...  
    intptr_t* base() const { return (intptr_t*) (((char*) this) + sizeof(ConstantPool)); }  
    ...  
}
```

ConstantPool



Получение названия метода (3)

```
class Symbol : public MetaspaceObj {  
    ...  
    u2 _length;  
    u1 _body[2];  
    ...  
}
```

Получение названия метода (3)

```
class Symbol : public MetaspaceObj {
```

```
...
```

```
  u2 _length;
```

```
  u1 _body[2];
```

```
...
```

```
}
```

```
char char_at(int index) const {
```

```
  assert(index >=0 && index < length(), "symbol index overflow");
```

```
  return (char)base()[index];
```

```
}
```

Получение названия метода (3)

```
class Symbol : public MetaspaceObj {  
    ...  
    u2 _length;  
    u1 _body[2];  
    ...  
}
```

Symbol		Остаток строки										
length	body											
12	H e	l	l	o	,	w	o	r	l	d	!	

Примеры

- Поиск поверхности атаки в тестовом приложении на Java – `spring-petclinic`
- Фаззинг библиотек на поверхности атаки

Spring-petclinic

- **Web-сервис для клиники домашних животных**
 - владельцы
 - ПИТОМЦЫ
 - ВИЗИТЫ В КЛИНИКУ
- **Простейшее приложение на платформе Spring**
 - ~86 Kb Java-кода
 - есть юнит-тесты

Анализ spring-petclinic с помощью статического анализатора Svnace

[DEREF_OF_NULL]

DEREF_OF_NULL.RET: 1

[SPOTBUGS]

FB.EI_EXPOSE_REP: 4

FB.EI_EXPOSE_REP2: 1

[OTHER]

NO_CATCH: 1

Total warnings: 7

Deref of NULL

Reference 'pet', returned from function 'Owner.getPet(Integer)Pet' at VisitController.java:69, may be null and is dereferenced at VisitController.java:74.

```
@ModelAttribute("visit")
```

```
public Visit loadPetWithVisit(@PathVariable("ownerId") int ownerId, @PathVariable("petId") int petId,  
Map<String, Object> model) {
```

```
...
```

```
    Pet pet = owner.getPet(petId);
```

```
...
```

```
    pet.addVisit(visit);
```

```
...
```

```
}
```

Анализ поверхности атаки petclinic

- Поверхность атаки сложно найти вручную
- В sbom внутри jar-файла описана 101 зависимость
- Много абстракций
- Глубина стека 200+ функций

Create owner

Owner

First Name

myownerfirst

Last Name

myownerlast

Address

myowneraddr

City

myownercity

Telephone

1234567890

Add Owner

Spring-petclinic

Owner Information

Name	myownerfirst myownerlast
Address	myowneraddr
City	myownercity
Telephone	1234567890

Edit Owner

Add New Pet

Pets and Visits

Поиск поверхности атаки с OWASP Noir

POST /owners/new

body: address=&city=&telephone=

file: ./src/main/java/org/springframework/samples/petclinic/owner/**OwnerController.java (line 75)**

POST /owners/{ownerId}/edit

path: ownerId

body: address=&city=&telephone=

file: ./src/main/java/org/springframework/samples/petclinic/owner/**OwnerController.java (line 138)**

GET /owners/{ownerId}

path: ownerId

file: ./src/main/java/org/springframework/samples/petclinic/owner/**OwnerController.java (line 163)**

Создание записи

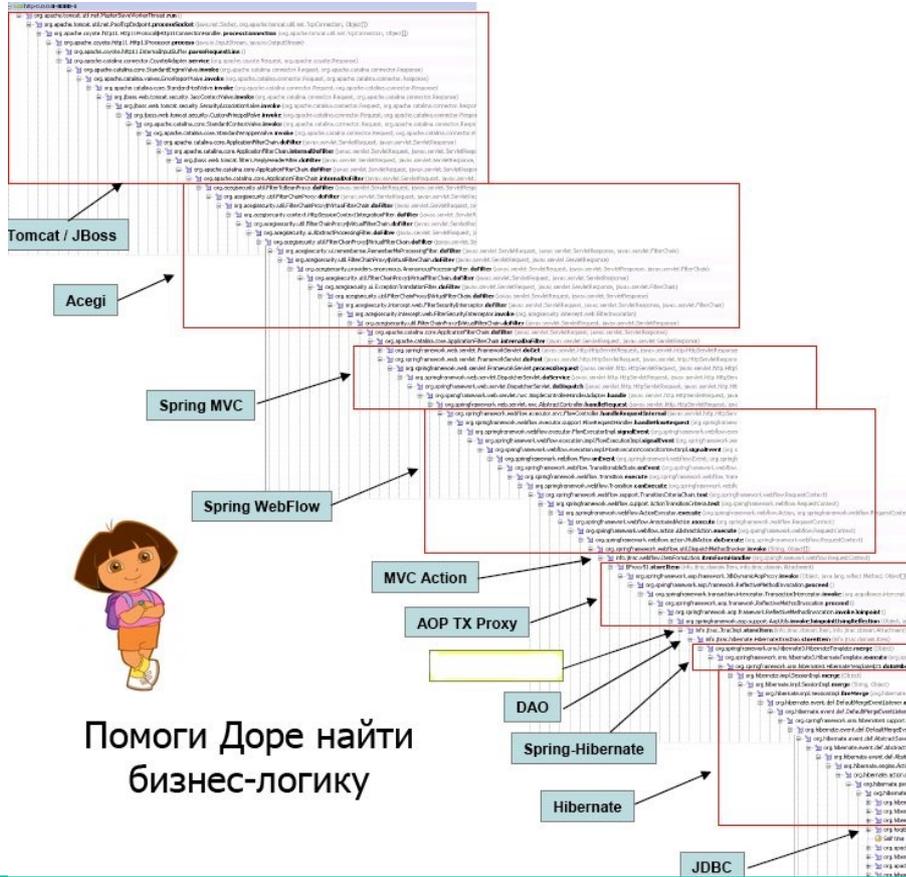
```
@PostMapping("/owners/new")
public String processCreationForm(@Valid Owner owner, BindingResult result, RedirectAttributes redirectAttributes) {
    if (result.hasErrors()) {
        redirectAttributes.addFlashAttribute("error", "There was an error in creating the owner.");
        return VIEWS_OWNER_CREATE_OR_UPDATE_FORM;
    }

    this.owners.save(owner);
    redirectAttributes.addFlashAttribute("message", "New Owner Created");
    return "redirect:/owners/" + owner.getId();
}
```

Чтение записи

```
@GetMapping("/owners/{ownerId}")
public ModelAndView showOwner(@PathVariable("ownerId") int ownerId) {
    ModelAndView mav = new ModelAndView("owners/ownerDetails");
    Optional<Owner> optionalOwner = this.owners.findById(ownerId);
    Owner owner = optionalOwner.orElseThrow(() -> new IllegalArgumentException(
        "Owner not found with id: " + ownerId + ". Please ensure the ID is correct "));
    mav.addObject(owner);
    return mav;
}
```

На что похож backtrace / call graph



OwnerController



Атака на цепочку поставок

- **Ошибки могут быть в заимствованных компонентах**
- **Обновление**
- **Мониторинг**
- **Резервное копирование**
- **Безопасная разработка**
 - В том числе анализ поверхности атаки

План исследования приложения petclinic

- **В веб-интерфейсе**
 - Создать карточку владельца
 - Просмотреть карточку
- **В инструменте Natch**
 - Пометить полученные данные формы для отслеживания
 - Убедиться, что помеченные данные вернулись назад пользователю при просмотре формы
- **В интерфейсе анализа SMatch**
 - Выбрать интересные модули и функции
- **Фаззинг или поиск багов вручную**

Входные данные

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:49152/owners/new
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9,ru;q=0.8
Cookie: sidenav-state=pinned; csrftoken=v82VHdvacMs8EraSTY528xP49j9Efbs8; django_language=en-us; session=eyJsb2dnZWRfaW4iOmZhbHNlfQ.ZyHHTA.-IH3r05WSU8-MNKDjZzuowFfwcQ; JSESSIONID=D1625244AC34D07B3F8D54F79F77CC51
```

```
firstName=myownerfirst&lastName=myownerlast&address=myowneraddr&city=myownercity&telephone=1234567890
```

Страница с карточкой владельца с подсвеченными входными данными

```
<table class="table table-striped">
  <tr>
    <th>Name</th>
    <td><b>myownerfirst myownerlast</b></td>
  </tr>
  <tr>
    <th>Address</th>
    <td>myowneraddr</td>
  </tr>
  <tr>
    <th>City</th>
    <td>myownercity</td>
  </tr>
  <tr>
    <th>Telephone</th>
    <td>1234567890</td>
  </tr>
</table>
```

Изучение поверхности атаки

- Разбор запроса
- Валидация входных данных
- Работа с БД
- Вывод html

Разбор запроса: tomcat

```
⊖ void org/apache/tomcat/util/http/Parameters::processParameters(byte[], int, int) org/apache/tomcat/util/http/Para
└─ ⊖ void org/apache/tomcat/util/http/Parameters::processParameters(byte[], int, int, java/nio/charset/Charset) org
    └─ ⊖ java/lang/String org/apache/tomcat/util/buf/ByteChunk::toString() org/apache/tomcat/util/buf/ByteChunk
        └─ ⊖ java/lang/String org/apache/tomcat/util/buf/ByteChunk::toString(java/nio/charset/CodingErrorAction, jav
            └─ ⊖ boolean org/apache/tomcat/util/buf/AbstractChunk::isNull() org/apache/tomcat/util/buf/AbstractChu
                └─ ⊖ java/lang/String org/apache/tomcat/util/buf/StringCache::toString(org/apache/tomcat/util/buf/Byt
                    └─ ⊖ java/lang/String org/apache/tomcat/util/buf/ByteChunk::toStringInternal(java/nio/charset/Codi
                        └─ ⊖ java/nio/CharBuffer java/nio/charset/Charset::decode(java/nio/ByteBuffer) java/nio/charset
                            └─ ⊖ java/nio/charset/CharsetDecoder sun/nio/cs/ThreadLocalCoders::decoderFor(java/lang/C
                                └─ ⊖ java/nio/CharBuffer java/nio/charset/CharsetDecoder::decode(java/nio/ByteBuffer) j
                                    └─ java/nio/charset/CoderResult java/nio/charset/CharsetDecoder::decode(java/nio/Byt
```

Валидация входных данных: hibernate

```
⊖ org/hibernate/validator/internal/engine/constraintvalidation/ConstraintValidatorContextImpl org/hibernate/validator/internal/engine/valida
├─ ⊖ java/util/Optional org/hibernate/validator/internal/engine/constraintvalidation/ConstraintTree::validateSingleConstraint(org/hibernate/va
├─ ⊖ java/lang/Object org/hibernate/validator/internal/engine/valuecontext/ValueContext::getCurrentValidatedValue() org/hibernate/valid
├─ ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/NotBlankValidator::isValid(java/lang/Object, jakarta/validation/Cc
├─ ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/NotBlankValidator::isValid(java/lang/CharSequence, jakarta/va
├─ ⊖ java/lang/String java/lang/String::trim() java/lang/String
├─ ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/PatternValidator::isValid(java/lang/Object, jakarta/validation/Con
├─ ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/PatternValidator::isValid(java/lang/CharSequence, jakarta/vali
├─ ⊖ boolean java/util/regex/Matcher::matches() java/util/regex/Matcher
├─ ⊖ boolean java/util/regex/Pattern$BmpCharProperty::match(java/util/regex/Matcher, int, java/lang/CharSequence) java/u
├─ ⊖ boolean java/util/regex/CharPredicates::lambda$ASCII_DIGIT$18(int) java/util/regex/CharPredicates
├─ ⊖ boolean java/util/regex/ASCII::isDigit(int) java/util/regex/ASCII
```

Вставка значений в БД: h2

```
⊖ void org/h2/mvstore/db/MVTable::addRow(org/h2/engine/SessionLocal, org/h2/result/Row) org/h2/mvstore/db/MVTable
  L ⊖ java/lang/Object java/util/ArrayList$Itr::next() java/util/ArrayList$Itr
    L ⊖ void org/h2/mvstore/db/MVSecondaryIndex::add(org/h2/engine/SessionLocal, org/h2/result/Row) org/h2/mvstore/db/MVSecondaryIndex
      L ⊖ boolean org/h2/index/Index::needsUniqueCheck(org/h2/result/SearchRow) org/h2/index/Index
        L ⊖ java/lang/Object org/h2/mvstore/tx/TransactionMap::put(java/lang/Object, java/lang/Object) org/h2/mvstore/tx/TransactionMap
          L ⊖ java/lang/Object org/h2/mvstore/tx/TransactionMap::set(java/lang/Object, java/lang/Object) org/h2/mvstore/tx/TransactionMap
            L ⊖ void org/h2/mvstore/tx/TxDecisionMaker::initialize(java/lang/Object, java/lang/Object) org/h2/mvstore/tx/TxDecisionMaker
              L ⊖ java/lang/Object org/h2/mvstore/tx/TransactionMap::set(java/lang/Object, org/h2/mvstore/tx/TxDecisionMaker, int) org/h2/mvstore/tx/TransactionMap
                L ⊖ java/lang/Object org/h2/mvstore/MVMap::operate(java/lang/Object, java/lang/Object, org/h2/mvstore/MVMap$DecisionMaker) org/h2/mvstore/MVMap
                  L ⊖ boolean org/h2/mvstore/RootReference::isLocked() org/h2/mvstore/RootReference
                    L ⊖ org/h2/mvstore/CursorPos org/h2/mvstore/CursorPos::traverseDown(org/h2/mvstore/Page, java/lang/Object) org/h2/mvstore/CursorPos
                      L ⊖ int org/h2/mvstore/db/RowDataType::binarySearch(org/h2/result/SearchRow, java/lang/Object, int, int) org/h2/mvstore/db/RowDataType
                        L ⊖ int org/h2/mvstore/db/RowDataType::binarySearch(org/h2/result/SearchRow, org/h2/result/SearchRow[], int, int) org/h2/mvstore/db/RowDataType
                          L ⊖ int org/h2/mvstore/db/RowDataType::compareSearchRows(org/h2/result/SearchRow, org/h2/result/SearchRow) org/h2/mvstore/db/RowDataType
                            L ⊖ int org/h2/mvstore/db/ValueDataType::compareValues(org/h2/value/Value, org/h2/value/Value, int) org/h2/mvstore/db/ValueDataType
```

Рендер страницы: thymeleaf

```
⊖ void org/thymeleaf/spring6/view/ThymeleafView::render(java/util/Map, jakarta/servlet/http/HttpServletRequest, jakarta/servlet/http/HttpServletResponse)
└─ ⊖ void org/thymeleaf/spring6/view/ThymeleafView::renderFragment(java/util/Set, java/util/Map, jakarta/servlet/http/HttpServletRequest, jakarta/servlet/http/HttpServletResponse)
    └─ ⊖ java/io/PrintWriter jakarta/servlet/ServletResponseWrapper::getWriter() jakarta/servlet/ServletResponseWrapper
        └─ ⊖ void org/thymeleaf/TemplateEngine::process(java/lang/String, java/util/Set, org/thymeleaf/context/IContext, java/io/Writer) org/thymeleaf/TemplateEngine
            └─ ⊖ void org/thymeleaf/TemplateEngine::process(org/thymeleaf/TemplateSpec, org/thymeleaf/context/IContext, java/io/Writer) org/thymeleaf/TemplateEngine
                └─ ⊖ void org/apache/catalina/connector/CoyoteWriter::flush() org/apache/catalina/connector/CoyoteWriter
                    └─ ⊖ void org/apache/catalina/connector/OutputBuffer::flush() org/apache/catalina/connector/OutputBuffer
                        └─ ⊖ void org/apache/catalina/connector/OutputBuffer::doFlush(boolean) org/apache/catalina/connector/OutputBuffer
                            └─ ⊖ void org/apache/catalina/connector/OutputBuffer::flushCharBuffer() org/apache/catalina/connector/OutputBuffer
                                └─ ⊖ void org/apache/catalina/connector/OutputBuffer::realWriteChars(java/nio/CharBuffer) org/apache/catalina/connector/OutputBuffer
                                    └─ ⊖ void org/apache/tomcat/util/buf/C2BConverter::convert(java/nio/CharBuffer, java/nio/ByteBuffer) org/apache/tomcat/util/buf/C2BConverter
                                        └─ ⊖ java/nio/charset/CoderResult java/nio/charset/CharsetEncoder::encode(java/nio/CharBuffer, java/nio/ByteBuffer) org/apache/tomcat/util/buf/C2BConverter
                                            └─ ⊖ java/nio/charset/CoderResult sun/nio/cs/UTF_8$Encoder::encodeLoop(java/nio/CharBuffer, java/nio/ByteBuffer) org/apache/tomcat/util/buf/C2BConverter
                                                └─ ⊖ java/nio/charset/CoderResult sun/nio/cs/UTF_8$Encoder::encodeArrayLoop(java/nio/CharBuffer, java/nio/ByteBuffer) org/apache/tomcat/util/buf/C2BConverter
                                                    └─ ⊖ int java/lang/System$2::encodeASCII(char[], int, byte[], int, int) java/lang/System$2
                                                        └─ ⊖ int java/lang/StringCoding::implEncodeAsciiArray(char[], int, byte[], int, int) java/lang/StringCoding
```

Экранирование вывода: unescape

```
⊖ java/lang/String org/thymeleaf/util/EscapedAttributeUtils::unescapeAttribute(org/thymeleaf/templatemode/Templa
  L ⊖ void org/thymeleaf/standard/processor/AbstractStandardExpressionAttributeTagProcessor::doProcess(org/thyme
    L ⊖ java/lang/Object org/thymeleaf/standard/expression/Expression::execute(org/thymeleaf/context/IExpression
      ⊕ org/thymeleaf/standard/expression/IStandardVariableExpressionEvaluator org/thymeleaf/standard/expres
      L ⊖ void org/thymeleaf/standard/processor/StandardTextTagProcessor::doProcess(org/thymeleaf/context/ITer
        L ⊖ java/lang/String org/thymeleaf/standard/processor/StandardTextTagProcessor::produceEscapedOutput
          L ⊖ java/lang/String org/unescape/html/HtmlEscape::escapeHtml4Xml(java/lang/String) org/unescap
            L ⊖ java/lang/String org/unescape/html/HtmlEscape::escapeHtml(java/lang/String, org/unescape/
              L ⊖ java/lang/String org/unescape/html/HtmlEscapeUtil::escape(java/lang/String, org/unescap
                L char java/lang/String::charAt(int) java/lang/String
```

unescape

- **HtmlEscapeUtil**
- **Функция escape выглядит не очень сложной**
- **А вот в unescape много ветвлений и циклов**
- **Code smells**
 - escape и unescape существуют в трёх (!) почти одинаковых экземплярах (одна для строк, другая для потоков, третья для массивов)

diff для версий unescape()

1:268,269с

3:252,253с

```
if ((f - (i + 1)) <= 0) {
```

```
    // We weren't able to consume any alphanumeric
```

2:295,297с

```
if (escapei == 0) {
```

```
    // We weren't able to consume any decimal chars
```

```
    writer.write(c1);
```

Фаззинг escape + unescape

```
String result = HtmlEscapeUtil.escape(text,  
HtmlEscapeType.HTML5_NAMED_REFERENCES_DEFAULT_TO_DECIMAL,  
    HtmlEscapeLevel.LEVEL_2_ALL_NON_ASCII_PLUS_MARKUP_SIGNIFICANT);  
String result2 = HtmlEscapeUtil.unescape(result);  
if (!text.equals(result2)) {  
    System.out.println(" in: " + text);  
    System.out.println("esc: " + result);  
    System.out.println("out: " + result2);  
    throw new Exception();  
}
```

Фаззинг escape + unescape: не-ASCII символы

```
== libFuzzer crashing input ==
MS: 4 ChangeByte-Custom-ChangeByte-Custom-; base unit: e0fbd8cc7b1840b748dfcb6d5628d81739be64de
0x58,0x5c,0x5c,0x5c,0x5c,0xad,0x5c,0xdf,0x81,0x81,0x86,0x3b,0x81,0x5b,0x42,0x81,0x86,0x7e,0x81,0
X\\\\\\\\\\255\\\\337\\201\\201\\206;\\201[B\\201\\206~\\201\\201\\307\\201\\201\\201\\\\221%\\006\\\\D\\243\\006\\\\
artifact_prefix='./'; Test unit written to ./crash-c336249dfbd6f7770416e44b6a9d89265f16e4d3
Base64: WfxcXFytXN+BgYY7gVtCgYZ+gYHHgYGBXJElBlxEowZc
in: X\\\\\\\\\\1Æ;UBÆ~Á||Á\\€ \\DÆ\
esc: X\\\\\\\\\\&#860;&#1985;&AElig;8Ucirc;B&AElig~&Aacute;&#449;&Aacute;\\&#1125; \\D&AElig\
out: X\\\\\\\\\\1ÆÛBÆ~Á||Á\\€ \\DÆ\
reproducer_path='./'; Java reproducer written to ./Crash_c336249dfbd6f7770416e44b6a9d89265f16e4d3
```

AElig;	U+000C6	Æ
AElig	U+000C6	Æ
aelig;	U+000E6	æ
aelig	U+000E6	æ

Тест для `escape` + `unescape`

```
String text = "\306;\304;\305;";
```

```
System.out.println(text);
```

```
String result = HtmlEscapeUtil.escape(text,  
HtmlEscapeType.HTML5_NAMED_REFERENCES_DEFAULT_TO_DECIMAL,  
HtmlEscapeLevel.LEVEL_2_ALL_NON_ASCII_PLUS_MARKUP_SIGNIFICANT);
```

```
String result2 = HtmlEscapeUtil.unescape(result);
```

```
System.out.println(result);
```

```
System.out.println(result2);
```

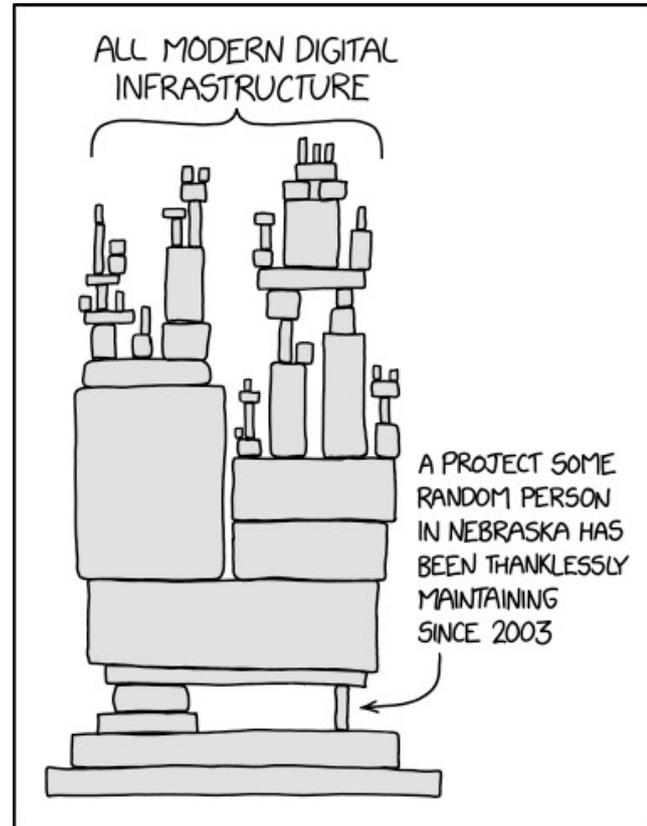
Æ; Ä; Å;

Æ Ä ; Å ;

ÆÄ; Å;

Недетерминированное поведение `escape` + `unescape`

- Преобразование разных символов работает по-разному
- Оказывается, уже есть `issue` и `pull request` от 2023 года на эту тему
- Коммитов, меняющих код, не было с 2018 года



Выводы

- **Нельзя оставлять без внимания заимствованный код**
- **Проблемные зависимости быстрее находятся, если знать поверхность атаки**
- **Фаззинг ещё не стал правилом для Java-библиотек**
 - По крайней мере, таких тестов нет в найденных зависимостях `spring-petclinic`

Telegram-канал Natch

- https://t.me/ispras_natch
- Ссылки на документацию и релизы
- Вопросы от пользователей
- Разборы кейсов
- Анонсы вебинаров
- Уведомления о новых релизах
- Собственный стикерпак с нарвалами :)

