

# Определение поверхности атаки сложных систем с помощью Natch

Довгалюк Павел

ИСП РАН

# Что такое Natch

- **Автоматизированный анализ поверхности атаки**
  - Интерфейсы приложений
  - Компоненты и функции программ
- **x86\_64 и немножко AArch64**
- **C/C++, Go, Python, Java, JavaScript\***
- **Регрессионное тестирование**
- **Отчёты для аналитиков и тестировщиков**

# Динамический анализ

- Можно увидеть подробности о конкретном сценарии работы
- Более сложная настройка, чем при статическом анализе
- Нужно много времени на анализ

# Динамический анализ – это сложно

- Полносистемный анализ
- Отслеживание потока управления и потоков данных
- Нужна тщательная настройка сценариев
- Не обойтись без отладочных символов

# Почему анализ может быть медленным?

- Медленный многоядерный процессор вместо одного быстрого ядра
- Используется графический интерфейс
- Есть лишние программы
- Чрезмерно большой сценарий
- Непроизводительные паузы в сценарии
- Natch не слишком быстр

# Лишние программы и GUI

- Тратят процессорное время
- Попадают в лог записи/воспроизведения
- Попадают в результаты анализа
- GUI усложняет автоматизацию тестирования
- `unattended-upgrades` в Ubuntu

# Сокращение сценария

- Чем короче сценарий, тем лучше
- Нужно как можно быстрее отправить входные данные и завершить VM
- Лучше стартовать анализ для уже запущенных программ, подведённых к нужному состоянию
- Но для Java-приложений старт должен попадать в сценарий

# Поиск поверхности атаки

- **Какие данные важны?**
- **Куда они не должны проникать?**
- **Как они должны обрабатываться?**
  
- **Все потоки данных сразу отследить невозможно**



# Последовательность работы

- **Настроить VM, удалить лишние сервисы**
- **Определить содержание сценария**
- **Определить входные данные**
- **Записать сценарий, чётко выделив временные границы**
- **Пометить выбранные входные данные**
- **Воспроизвести сценарий**

# Примеры

- Поиск поверхности атаки в тестовом приложении на Java – `spring-petclinic`
- Анализ утечек данных в настоящих СУБД

# Spring-petclinic

## 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

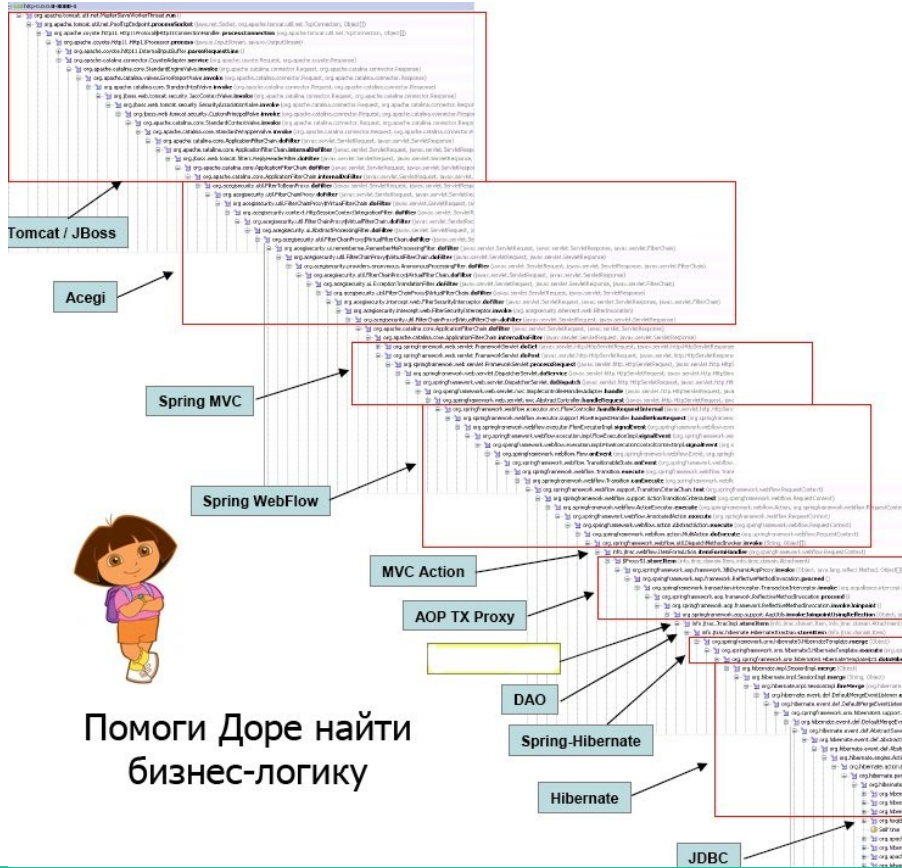
Add New Pet

## Pets and Visits

# Анализ поверхности атаки petclinic

- Поверхность атаки сложно найти вручную
- В sbom внутри jar-файла описаны 92 зависимости
- Много абстракций
- Глубина стека 200+ функций

# На что похож backtrace / call graph



# 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: apache

```
⊖ 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
  L ⊖ java/util/Optional org/hibernate/validator/internal/engine/constraintvalidation/ConstraintTree::validateSingleConstraint(org/hibernate/va
    L ⊖ 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
        L ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/NotBlankValidator::isValid(java/lang/CharSequence, jakarta/va
          L java/lang/String java/lang/String::trim() java/lang/String
        L ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/PatternValidator::isValid(java/lang/Object, jakarta/validation/Con
          L ⊖ boolean org/hibernate/validator/internal/constraintvalidators/bv/PatternValidator::isValid(java/lang/CharSequence, jakarta/vali
            L ⊖ boolean java/util/regex/Matcher::matches() java/util/regex/Matcher
              L ⊖ boolean java/util/regex/Pattern$BmpCharProperty::match(java/util/regex/Matcher, int, java/lang/CharSequence) java/u
                L ⊖ boolean java/util/regex/CharPredicates::lambda$ASCII_DIGIT$18(int) java/util/regex/CharPredicates
                  L 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
- Возможности для багов: `escape` и `unescape` существуют в трёх (!) почти одинаковых экземплярах (одна для строк, другая для потоков, третья для массивов)



# unescape()

1:268,269c

3:252,253c

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

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

2:295,297c

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

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

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



# Результат анализа petclinic

- **Библиотеки, входящие в поверхность атаки**
  - unescape
  - thymeleaf
  - hibernate
  - h2
- **В коде в явном виде фигурируют не все**
- **Заимствованный код тоже нужно тестировать**
- **Собственные тесты в библиотеках есть, но без фаззинга**
- **Возможно фаззинг делается разработчиками, но тесты не публикуются**

# Natch для поиска утечек

1. Выделить чувствительные данные
2. Запустить анализ сценария работы системы
3. Получить поверхность атаки
  - легальное использование данных – надо протестировать
  - утечка – нужно предотвратить

# Кейс – удаление данных в СУБД

- Если удалить данные с помощью DELETE, исчезнут ли они из БД?
- А если удалить таблицу?
- А если даже всю базу?

# Кейс – удаление данных в СУБД

- **Добавляем данные в таблицу**
- **Удаляем данные**
  - DELETE FROM <table>
  - DROP DATABASE <db>
- **Проверяем, где они остались**

# Кейс – удаление данных в СУБД

- MySQL
- MariaDB
- PostgreSQL

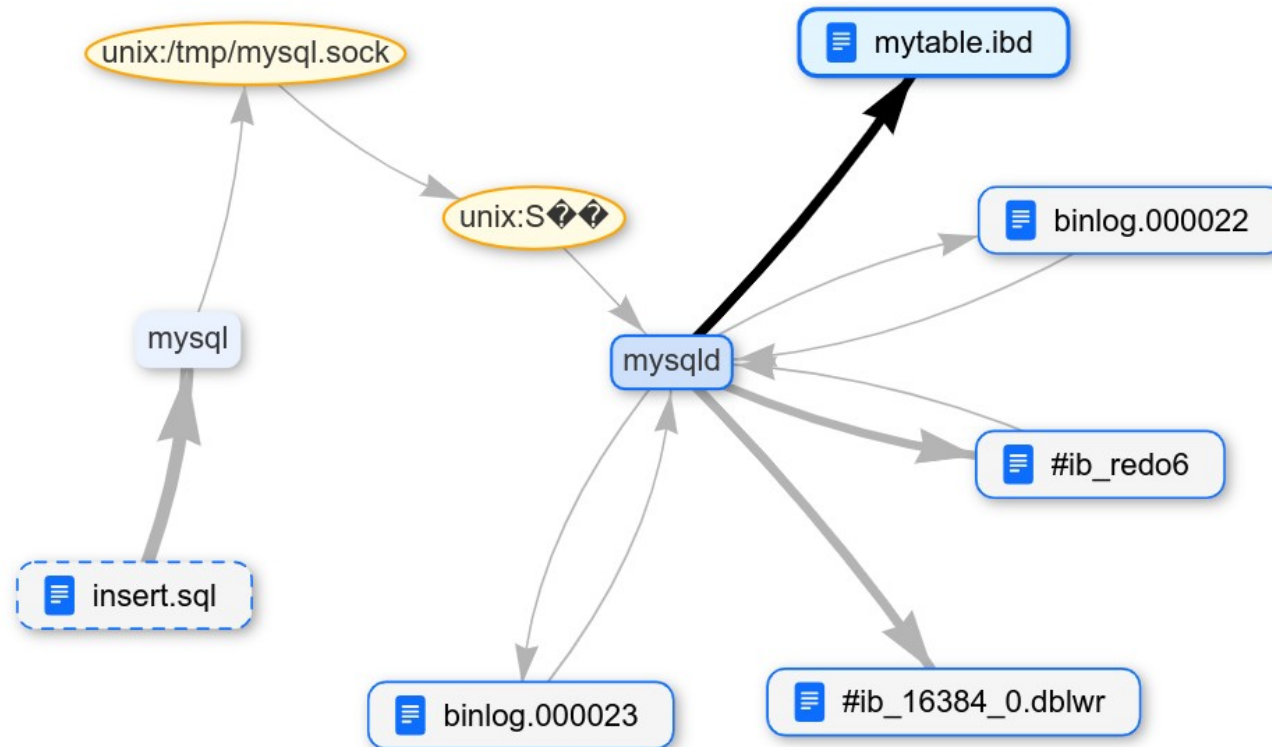
# MySQL. Куда попадают данные?

```
mysql -u user -p -D my < insert.sql
```

- **Пометка insert.sql в Natch**
  - можно проследить путь байтов из строки



# MySQL. Куда попадают данные?



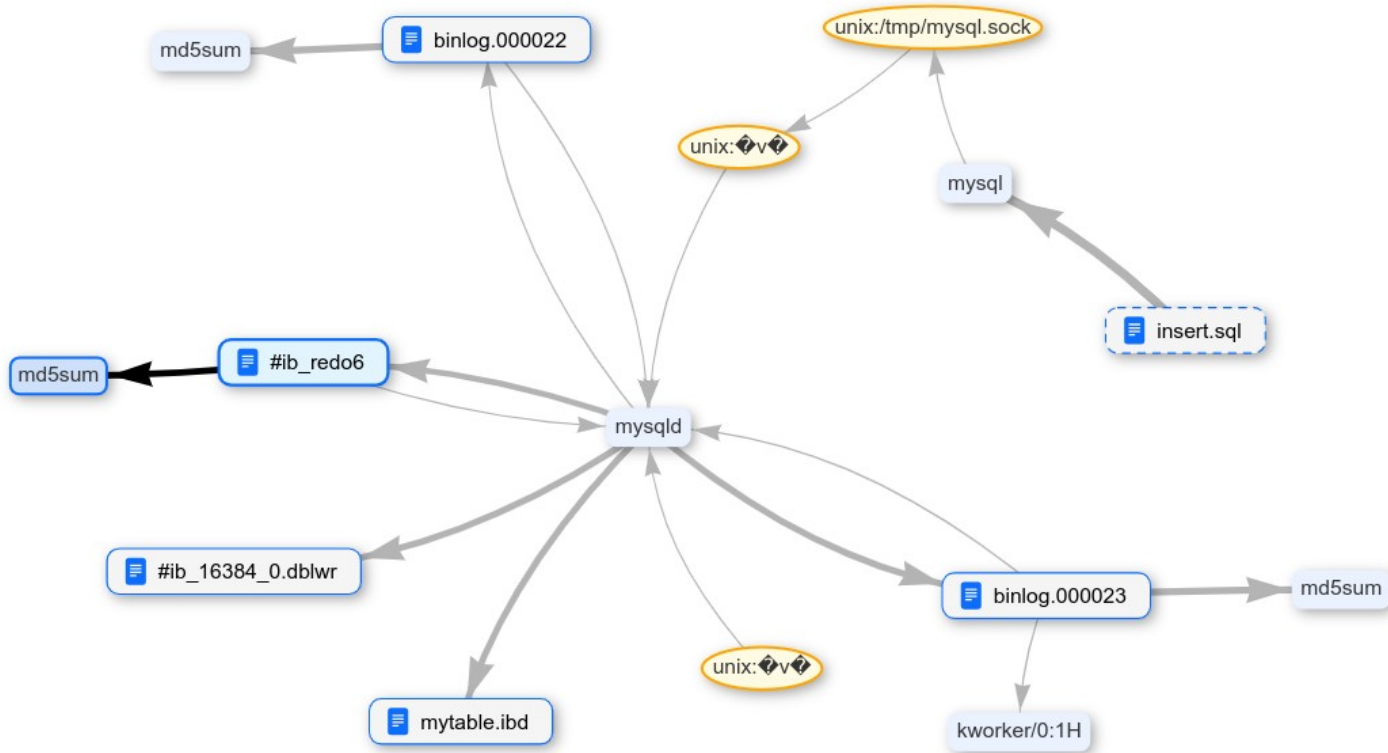
# MySQL. Где остались данные после удаления?

```
DROP DATABASE my;
```

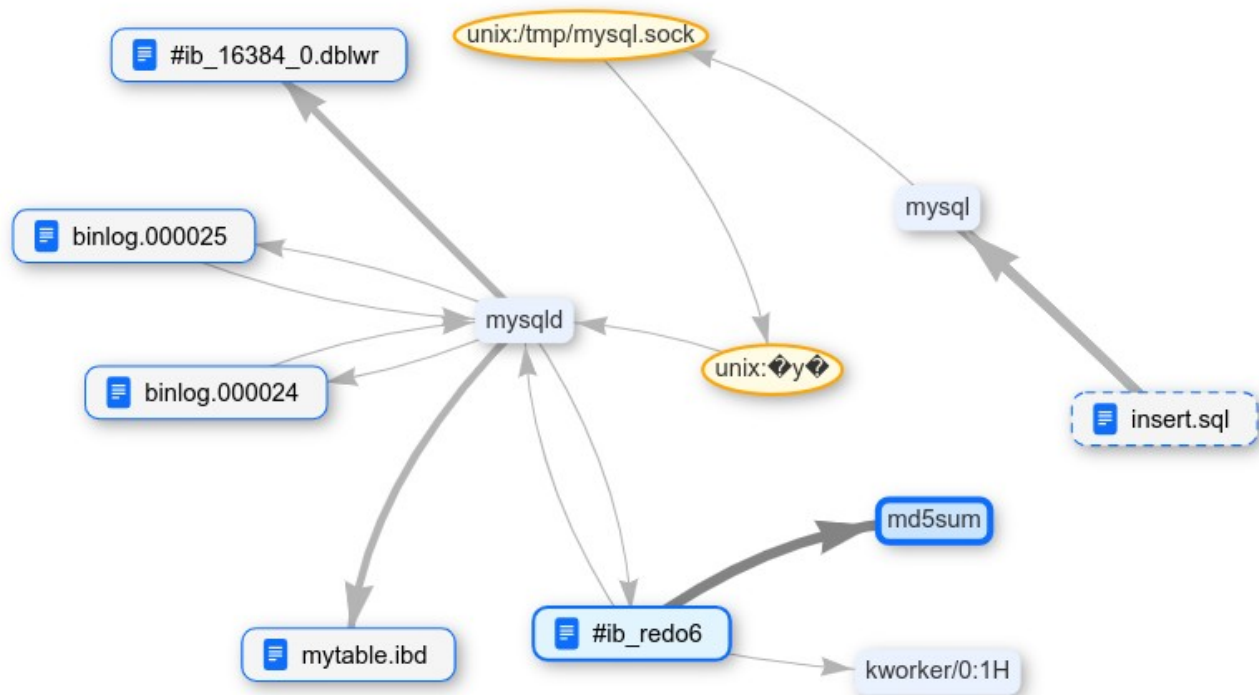
```
sudo md5sum <файлы с данными>
```

- md5sum
  - читает все байты
  - лучше, чем find, потому что заметит отдельные байты
  - **Natch** показывает все операции чтения
  - **Natch** определит чтение помеченных данных, если файл был кэширован

# MySQL. Где остались данные после удаления?



# MySQL. Удаление БД





# MySQL. Выводы

- **Данные не удаляются из внутренних логов**
- **И даже остаются в файловом кэше**

# Выводы по протестированным СУБД

✗ MySQL

✗ MariaDB

✗ PostgreSQL

- **Свободные СУБД не затирают все данные**
  - лучше всех MariaDB – хотя бы очищает память с закэшированными файлами
  - не стоит их использовать в сертифицируемых системах
- **Подробнее на хабре**
  - [https://habr.com/ru/companies/isp\\_ras/articles/827830/](https://habr.com/ru/companies/isp_ras/articles/827830/)

# Natch 3.3

- 3 марта 2025
- Работает быстрее
- Фильтрация пакетов для пометки
- Анализ JavaScript (V8)
- Помечается расшифрованный https-трафик, если помечены входящие пакеты



# Фильтрация пакетов

- **Выбор пакетов для пометки через скрипт, создаваемый пользователем**
- **Можно помечать фрагменты пакетов**
  - например, только вводимый пароль

# Анализ JavaScript

```
⊖ dest::ondata node:internal/streams/readable (libnode.so.109)
└─ ⊖ process::_write node:internal/streams/writable (libnode.so.109)
  └─ ⊖ process::writeOrBuffer node:internal/streams/writable (libnode.so.109)
    └─ ⊖ process::_write node:internal/streams/writable (libnode.so.109)
      └─ ⊖ process::writeOrBuffer node:internal/streams/writable (libnode.so.109)
        ├── jsmemcmp::jsmemcmp (libnode.so.109)
        ├── ⊖ process::emit node:events (libnode.so.109)
        │   └─ ⊖ self (libnode.so.109)
        │       ├── jsmemcmp::jsmemcmp (libnode.so.109)
        │       └─ ⊖ process::emit node:events (libnode.so.109)
        │           ├── slice node:buffer (libnode.so.109)
        │           └─ ⊖ process::emit node:events (libnode.so.109)
        │               └─ ⊖ process::emit node:events (libnode.so.109)
        │                   └─ contype (libnode.so.109)
        ├── jsmemcmp::jsmemcmp (libnode.so.109)
        ├── ⊖ copy node:buffer (libnode.so.109)
        │   └─ ⊖ _copy node:buffer (libnode.so.109)
        │       └─ _copyActual node:buffer (libnode.so.109)
        └─ ⊖ process::emit node:events (libnode.so.109)
            └─ self (libnode.so.109)
```

# Пометка https и сжатого трафика

tcp <-> 64.233.161.103:80 read 9Kb write 194b

Encoding:

ascii

Reads  Writes  Tainted only

read #1 from offset 0x0

```
ø0/≤00|gя000000?S400e0g0"0.0000Sv00l600|`0-000l000?30d00>w/.E"00403(z00&Nq0z0000'00S√  
/00$0,Q0/√`000H00n0,000I60~00wd00BH00%fK▶0x0}080  
"8I\xqa00G06↑005↵0/0}M00a  
0R♣:1Z-00000F0n0oa0↑0Q+:0600 00500D00080I0#0q|02go2100000000000P0490@0aK0a%00P0000Q0m05000  
D000BL"0000,H0 000>000M0♣0;)0P00*0hd00,H0{0000000(0u0{00000↑00G00(0A♣60w000n000$00000000  
000:000N&00n00TM09304KRK0h{0009m00003908000000000d0o0W000s0003000/c0!0wW$00000000000000-0  
G0}^90ws000?+0*000zM000p000cC00a000{♣0001T0000000000↑00c0♣0f0F3E0K)0~4*0H+0007♣000)00  
00J>{ }x00e`00G0h0W0k00↑0000o0&0200F0`0' C00(00↑=000@DW♣:@00>00♣e♣0*0♣00'0 "r0=000p40)Fo  
<+00{000 00 0z0↑0G0*c8000004100004♣430+00<0:0X0000X4♣&tm00qj♣0♣0400i00@00  
0D06♣T00>008'0qP0<0♣0♣(hk<♀ 0ΩP♣hV00 h0?0QD 0@uM00D♀ &0i00j000?00♣0↵t&t1h00Ecd←0♣E0☎  
←♣Q@0 Sø0?e00CA@/c 000y√000000♣00♣-F00z@000000<J3@♣00♣|♪000#00↑l♣<000a0oø0000 P♣0/√
```

# Пометка https и сжатого трафика

/home/user/index.html write 21Kb

Encoding:

ascii

Reads  Writes  Tainted only

write #1 from offset 0xd20

```
ement)if(a.tagName==="A"){a=a.getAttribute("data-nohref")==="1";break a}a=!1}a&&b.preventDefault(),!0);}).call(this);</script><style>#gbar,#guser{font-size:13px;padding-top:1px !important;}#gbar{height:22px}#guser{padding-bottom:7px !important;text-align:right}.gbh,.gbd{border-top:1px solid #c9d7f1;font-size:1px}.gbh{height:0;position:absolute;top:24px;width:100%}@media all{.gb1{height:22px;margin-right:.5em;vertical-align:top}#gbar{float:left}a.gb1,a.gb4{text-decoration:underline !important}a.gb1,a.gb4{color:#00c !important}.gbi .gb4{color:#dd8e27 !important}.gbf .gb4{color:#900 !important}</style><style>body,td,a,p,.h{font-family:sans-serif}body{margin:0;overflow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-height:17px}form{margin-bottom:20px}.h{color:#1967d2}em{font-weight:bold;font-style:normal}.lst{height:25px;width:496px}.gsfi,.lst{font:18px sans-serif}.gsfs{font:17px sans-serif}.ds{display:inline-block;display:inline-
```

# libz

```
⊖ gzip_decompress /home/nat/wget2/libwget/decompressor.c:129 (libwget.so.1.0.0)
├─ inflate (libwget.so.1.0.0)
├─ ⊖ inflate ./inflate.c:625 (libz.so.1.2.11)
│   ├── crc32 (libz.so.1.2.11)
│   ├── crc32_z (libz.so.1.2.11)
│   ├── inflate_fast ./inffast.c:53 (libz.so.1.2.11)
│   └─ ⊖ updatewindow ./inflate.c:400 (libz.so.1.2.11)
│       └─ memcpy (libz.so.1.2.11)
├─ ⊖ get_body /home/nat/wget2/src/wget.c:3643 (/home/user/wget/wget2)
│   └─ ⊖ wget_buffer_memcat /home/nat/wget2/libwget/buffer.c:380 (libwget.so.1.0.0)
│       └─ memcpy@GLIBC_2.2.5 ./string/./sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:135 (libc-2.31.so)
```

# libcrypto

```
⊖ EVP_DecryptUpdate /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./crypto/evp/evp_enc.c:799 (libcrypto.so.3)
├─ ⊖ ossl_gcm_stream_update /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./providers/implementations/ciphers/ciphercommon_gcm.c
│   └─ ⊖ gcm_cipher_internal /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./providers/implementations/ciphers/ciphercommon_gcm.c:3
│       ├── ⊖ ossl_gcm_aad_update /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./providers/implementations/ciphers/ciphercommon_gcm.c:3
│       │   └─ CRYPTO_gcm128_aad /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./crypto/modes/gcm128.c:907 (libcrypto.so.3)
│       └─ ⊖ generic_aes_gcm_cipher_update /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./providers/implementations/ciphers/ciphercommon_gcm.c:3
│           └─ ⊖ CRYPTO_gcm128_decrypt /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./crypto/modes/gcm128.c:1201 (libcrypto.so.3)
│               └─ gcm_ghash_4bit /usr/src/openssl-3.0.13-0ubuntu3.4/build_shared/./crypto/modes/ghash-x86_64.s:110 (libcrypto.so.3)
```

# Telegram-канал Natch

- [https://t.me/ispras\\_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 :)

