Finding Realtime Safety Bugs Through Static Analysis

Mark McCurry

May 19, 2017
A bit of history

- ZynAddSubFX had issues with low latency
- Stoat was used for refactoring
- Performance was greatly improved via Stoat
A bit of history

- ZynAddSubFX had issues with low latency
A bit of history

- ZynAddSubFX had issues with low latency
- Stoat was used for refactoring
A bit of history

- ZynAddSubFX had issues with low latency
- Stoat was used for refactoring
- Performance was greatly improved via Stoat
Real time & xruns

- xrun - excessive run time
Real time & xruns

- xrun - excessive run time
- realtime - code with a real time based timing constraint
Real time & xruns

- xrun - excessive run time
- realtime - code with a real time based timing constraint
- realtime hazard - code which may violate the real time constraint often through non-determinism
Realtime hazard example

```c
void function(...)
{
    fwrite(...);
}
```
Realtime hazard example

```c
void function(...) {
    fwrite(...);
}
```

1. start write to file
2. is IO buffer full?
2b. flush file buffer
3. return
Unsafe functions

- dynamic memory
- inter-process communication
- file IO
- threading locks
# Realtime safety within Linux audio

| 6PM       | Add64     | Alsa Modular | amSynth |
| Borderlands | Bristol   | Calf *       | Cellular Auto. |
| Dexed     | DX-10     | Helm         | Hexter   |
| JX-10     | LB-302    | LMMS         | Monstro  |
| Mr. Alias 2 | Mx44      | Nekobee      | Newtonator |
| OBXD      | Organic   | Oxe FM Synth | Peggy2000 |
| Petri-Foo | Phasex    | Samplev1     | SetBFree |
| Sineshaper | Sorcer    | Synthv1      | Triceratops |
| Triple Oscillator | Tunefish 4 | Vex         | Watsyn   |
| WhySynth | Wolpertinger | Xsynth     | ZynAddSubFX |

---
Realtime safety within Linux audio

<table>
<thead>
<tr>
<th>6PM</th>
<th>Add64</th>
<th>Alsa Modular</th>
<th>amSynth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Borderlands</td>
<td>Bristol</td>
<td>Calf *</td>
<td>Cellular Auto.</td>
</tr>
<tr>
<td>Dexed</td>
<td>DX-10</td>
<td>Helm</td>
<td>Hexter</td>
</tr>
<tr>
<td>JX-10</td>
<td>LB-302</td>
<td>LMMS</td>
<td>Monstro</td>
</tr>
<tr>
<td>Mr. Alias 2</td>
<td>Mx44</td>
<td>Nekobee</td>
<td>Newtonator</td>
</tr>
<tr>
<td>OBXD</td>
<td>Organic</td>
<td>Oxe FM Synth</td>
<td>Peggy2000</td>
</tr>
<tr>
<td>Petri-Foo</td>
<td>Phasex</td>
<td>Samplev1</td>
<td>SetBFree</td>
</tr>
<tr>
<td>Sineshaper</td>
<td>Sorcer</td>
<td>Synthv1</td>
<td>Triceratops</td>
</tr>
<tr>
<td>Triple Oscillator</td>
<td>Tunefish 4</td>
<td>Vex</td>
<td>Watsyn</td>
</tr>
<tr>
<td>WhySynth</td>
<td>Wolpertinger</td>
<td>Xsynth</td>
<td>ZynAddSubFX</td>
</tr>
</tbody>
</table>
Realtime safety within Linux audio

- hard realtime software is comparatively rare/niche
Realtime safety within Linux audio

- hard realtime software is comparatively rare/niche
- Linux Audio attracts many individuals who are learning
Realtime safety within Linux audio

- hard realtime software is comparatively rare/niche
- Linux Audio attracts many individuals who are learning
- tools for identifying realtime bugs are sparse
Finding realtime hazards?

- manual code review
- runtime analysis → jack-interposers
Why are existing options insufficient?

- code review takes time
- runtime analysis need exhaustive testing
Why are existing options insufficient?

- code review takes time
- runtime analysis need exhaustive testing
- bugs can be missed easily
Finding realtime hazards?

- Manual code review
- Runtime analysis → jack-interposers
Finding realtime hazards?

- Manual code review
- Runtime analysis → jack-interposer
- Static analysis → stoat
Finding Realtime Safety Bugs Through Static Analysis

Stoat

▶ **S**Tatic (LLVM)
▶ **O**bject
▶ **A**nalysis
▶ **T**ool
What does stoat do?

1. hooks onto LLVM (.c/.cpp → .bc)
Finding Realtime Safety Bugs Through Static Analysis

What does stoat do?

1. hooks onto LLVM (.c/.cpp → .bc)
2. extracts call graph (.bc → metadata)
What does stoat do?

1. hooks onto LLVM (.c/.cpp → .bc)
2. extracts call graph (.bc → metadata)
3. identifies realtime hazards (metadata → errors)
int REALTIME main()
{
    int barbar;
    foo(barbar);
    baz();
    return 0;
}
Translators

; ModuleID = '<stdin>'
target datalayout = "e-m:e-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-unknown-linux-gnu"

; Function Attrs: nounwind uwtable
define i32 @main() #0 {
    %1 = alloca i32, align 4
    %barbar = alloca i32, align 4
    store i32 0, i32* %1
    %2 = load i32* %barbar, align 4
    call void @foo(i32 %2)
call void @baz()
    ret i32 0
}
Translators

; ModuleID = '<stdin>'
target datalayout = "e-m:e-i64:64-f80:128-n8:16:32:64-S128"
target triple = "x86_64-unknown-linux-gnu"

; Function Attrs: nounwind uwtable
declare i32 @main() #0 {
    %1 = alloca i32, align 4
    %barbar = alloca i32, align 4
    store i32 0, i32* %1
    %2 = load i32* %barbar, align 4
    call void @foo(i32 %2)
    call void @baz()
    ret i32 0
}
Translations

```java
function 'main'
calls 'foo'
calls 'baz'
```
A practical example - Helm
A practical example - setup/compilation

- stoat-compile/stoat-compile++ are used as a proxy for the compiler
- this builds the project and the metadata that stoat needs
A practical example - setup/compilation

mark@cvar:helm$ CC=stoat-compile CXX=stoat-compile++ make
A practical example

mark@cvar:helm$ CC=stoat-compile CXX=stoat-compile++ make
make -C standalone/builds/linux CONFIG=Release DEBCXXFLAGS="" DEBLDFLAGS=""
make -C builds/linux/LV2 CONFIG=Release DEBCXXFLAGS="" DEBLDFLAGS=""
make[1]: Entering directory ‘helm/builds/linux/LV2’
make[2]: Entering directory ‘helm/builds/linux/LV2’
stoat-compile++ lv2_ttl_generator.c -o lv2_ttl_generator -ldl
make[1]: Entering directory ‘helm/standalone/builds/linux’
Compiling alias.cpp
make[2]: Entering directory ‘helm/builds/linux/LV2’
Compiling alias.cpp
make[2]: Leaving directory ‘helm/builds/linux/LV2’
Compiling arpeggiator.cpp
Compiling bit_crush.cpp
Compiling arpeggiator.cpp
Compiling bypass_router.cpp
A practical example

- stoat analysis typically starts with a root function

Setting stoat’s target
mark@cvar:helm$ echo HelmPlugin::processBlock > whitelist.txt
mark@cvar:helm$ echo HelmPlugin::processBlock > whitelist.txt
mark@cvar:helm$ stoat -r . -w whitelist.txt
mark@cvar:helm$ echo HelmPlugin::processBlock > whitelist.txt
mark@cvar:helm$ stoat -r . -w whitelist.txt
Parsing './standalone/builds/linux/build/intermediate/Release/load_save_2c95b2e1.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/memory_5ee1bbc0.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/filter_8bfdb6b7.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/juce_gui_basics_3e4cc581.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/BinaryData_ce4232d4.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/synth_gui_interface_6337839d.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/step_sequencer_section_13843dc2.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/midi_keyboard_ec3d63f9.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/detune_lookup_ea628520.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/formant_section_51494d55.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/bit_crush_6b16ce74.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/lfo_section_f85fa167.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/patch_selector_fe7c390c.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/helm_standalone_editor_c8ee03cb.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/helm_voice_handler_35395fa6.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/startup_52cb2a28.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/smooth_value_7af0775f.o.bc'
Parsing './standalone/builds/linux/build/intermediate/Release/juce_audio_basics_181b4cb.o.bc'
Finding Realtime Safety Bugs Through Static Analysis

## The Contradiction Reasons:

- `juce::AudioProcessorGraph::Node::~Node()` _ZN4juce19AudioProcessorGraph4NodeD2Ev_ `../../../JUCE/modules/juce_audio_processors/juce_audio_processors.cpp@60` : Deduced Realtime
- `juce::AudioProcessorGraph::Node::~Node()` _ZN4juce19AudioProcessorGraph4NodeD0Ev_ `?@? : Deduced Realtime`  
- `juce::ReferenceCountedObject::$vtable1` _ZN4juce22ReferenceCountedObject17decReferenceCountEv_ `../../../JUCE/modules/juce_core/juce_core.cpp@87` : Deduced Realtime
- `juce::ReferenceCountedObject::decReferenceCount()` _ZN4juce22ReferenceCountedObject17decReferenceCountEv_ `../../../JUCE/modules/juce_core/juce_core.cpp@267` : Deduced Realtime
- `juce::var::VariantType_Object::cleanUp(juce::var::ValueUnion)` _ZNK4juce3var18VariantType_Object7cleanUpERNS0_10ValueUnionE_ `?@? : Deduced Realtime`  
- `juce::var::VariantType_Object::$vtable22` _ZNK4juce3var18VariantType_Object7cleanUpERNS0_10ValueUnionE_ `?@? : Deduced Realtime`  
- `juce::var::VariantType::$vtable22` _ZNK4juce3var18VariantType_Object7cleanUpERNS0_10ValueUnionE_ `?@? : Deduced Realtime`  
- `juce::var::~var()` _ZN4juce3varD2Ev` `anonymous@ ?` : Deduced Realtime
- `juce::var::~var()` _ZN4juce3varD1Ev` `../../../src/common/load_save.cpp@628` : Deduced Realtime
- `LoadSave::loadPatch(int, int, int, SynthBase*, std::map<std::basic_string<char, std::char_traits<char>, std::allocator<char> >`) `../../../src/common/midi_manager.cpp@97` : Deduced Realtime
- `MidiManager::processMidiMessage(juce::MidiMessage const&, int)` _ZN11MidiManager18processMidiMessageERKN4juce11MidiMessageEi_ `../../../src/common/synth_base.cpp@187` : Deduced Realtime
- `SynthBase::processMidi(juce::MidiBuffer&, int, int)` _ZN9SynthBase11processMidiERN4juce10MidiBufferEii_ `../../../src/plugin/helm_plugin.cpp@177` : Deduced Realtime
- `HelmPlugin::processBlock(juce::AudioBuffer<float>&, juce::MidiBuffer&)` _ZN10HelmPlugin12processBlockERN4juce11AudioBufferIfEERNS0_10MidiBufferE_ `@` : Realtime (Whitelist)
- `operator delete(void*) _ZdlPv` `../../../JUCE/modules/juce_audio_processors/juce_audio_processors.cpp@97` : NonRealtime (Blacklist)

## Summary

Total of 519 error(s)
Total Functions: 49945
Total Realtime: 5478
Total NonRealtime: 1212
A practical example

- stoat finds many potential errors
- some are redundant
A practical example

```bash
mark@cvar:helm$ echo ".* == > juce::var::*" > suppression.txt
```

- undesired errors can be suppressed
- suppressions ignore parts of the callgraph
A practical example

LoadSave::varToState
LoadSave::loadPatch
LoadSave::saveMidiMapConfig
LoadSave::getPatchFile
LoadSave::saveMidiMapConfig
mopo::ProcessorRouter::updateAllProcessors

- blacklists can consolidate errors
A practical example

Stoat can be rerun without recompiling

```bash
mark@cvar:helm$ stoat -r . -w whitelist.txt -b blacklist.txt -s suppression.txt -SG error-graph.png
```
A practical example

Error #61:
mopo::Reverb$vtake26

## The Deduction Chain:
- mopo::ProcessorRouter$vtake26 : Deduced Realtime
- mopo::ProcessorRouter::setBufferSize(int) _ZN4mopo15ProcessorRouter13setBufferSizeEi : Deduced Realtime
- mopo::ProcessorRouter$vtake6 : Deduced Realtime
- SynthBase::processAudio(juce::AudioBuffer<float>*, int, int, int) _ZN9SynthBase12processAudioEPN4juce11AudioBufferIfEEiii : Deduced Realtime
- HelmPlugin::processBlock(juce::AudioBuffer<float>&, juce::MidiBuffer&) _ZN10HelmPlugin12processBlockERN4juce11AudioBufferIfEERNS0_10MidiBufferE : Realtime (Whitelist)

## The Contradiction Reasons:
- mopo::ProcessorRouter::updateAllProcessors() _ZN4mopo15ProcessorRouter19updateAllProcessorsEv : NonRealtime (Blacklist)

Total of 61 error(s)
Total Functions: 49945
Total Realtime: 1570
Total NonRealtime: 1715
A practical example

Error #18:
mopo::VoiceHandler::noteOn(double, double, int, int) _ZN4mopo12VoiceHandler6noteOnEddii

## The Deduction Chain:
- mopo::HelmVoiceHandler::noteOn(double, double, int, int) _ZN4mopo16HelmVoiceHandler6noteOnEddii : Deduced Realtime
- mopo::HelmVoiceHandler$vtable4 : Deduced Realtime
- mopo::HelmEngine::noteOn(double, double, int, int) _ZN4mopo10HelmEngine6noteOnEddii : Deduced Realtime
- mopo::HelmEngine$vtable8 : Deduced Realtime
- MidiManager::processMidiMessage(juce::MidiMessage const&, int) _ZN11MidiManager18processMidiMessageERKN4juce11MidiMessageEi : Deduced Realtime
- SynthBase::processMidi(juce::MidiBuffer&, int, int) _ZN9SynthBase11processMidiERN4juce10MidiBufferEii : Deduced Realtime
- HelmPlugin::processBlock(juce::AudioBuffer<float>&, juce::MidiBuffer&): Realtime (Whitelist)

## The Contradiction Reasons:
- std::list<double, ...>::push_front(double const&) _ZNSt4listIdSaIdEE10push_frontERKd : NonRealtime (Blacklist)
- std::list<mopo::Voice*, ...>::push_back(mopo::Voice* const): NonRealtime (Blacklist)
A practical example

```cpp
void VoiceHandler::noteOn(mopo_float note,
                           mopo_float velocity, int sample, int channel) {

    Voice* voice = grabVoice();
    pressed_notes_.push_front(note);

    if (last_played_note_ < 0)
        last_played_note_ = note;
    voice->activate(...)
    active Voices_.push_back(voice);
    last_played_note_ = note;
}
```
A practical example
A practical example - Reaction

- realtime hazards exist
- stoat shows where they can occur
- identifying bugs is the first step to fixing them
Stoat’s use in the wild

- ZynAddSubFX - Synth
  - original target of stoat
  - number of hazards greatly reduced
Stoat’s use in the wild

- ZynAddSubFX - Synth
  - original target of stoat
  - number of hazards greatly reduced
- librtosc - OSC implementation
  - API built with stoat in mind
  - uses metadata to annotate numerous callbacks
Stoat’s use in the wild

- ZynAddSubFX - Synth
  - original target of stoat
  - number of hazards greatly reduced
- librtosc - OSC implementation
  - API built with stoat in mind
  - uses metadata to annotate numerous callbacks
- carla - Host
- ingen - Synth
- jalv - Host
Conclusions

- Realtime safety is frequently violated in FLOSS audio projects
- Static and dynamic analysis tools help fix these problems
- Stoat offers a solution
Conclusions

- Source Available at https://github.com/fundamental/stoat
- Try it out and let’s fix some bugs