Finding Realtime Safety Bugs Through Static Analysis

Mark McCurry

May 19, 2017

A bit of history

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

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

Realtime hazard example

```
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

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

- 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 $\rightarrow$ jack-interposer

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

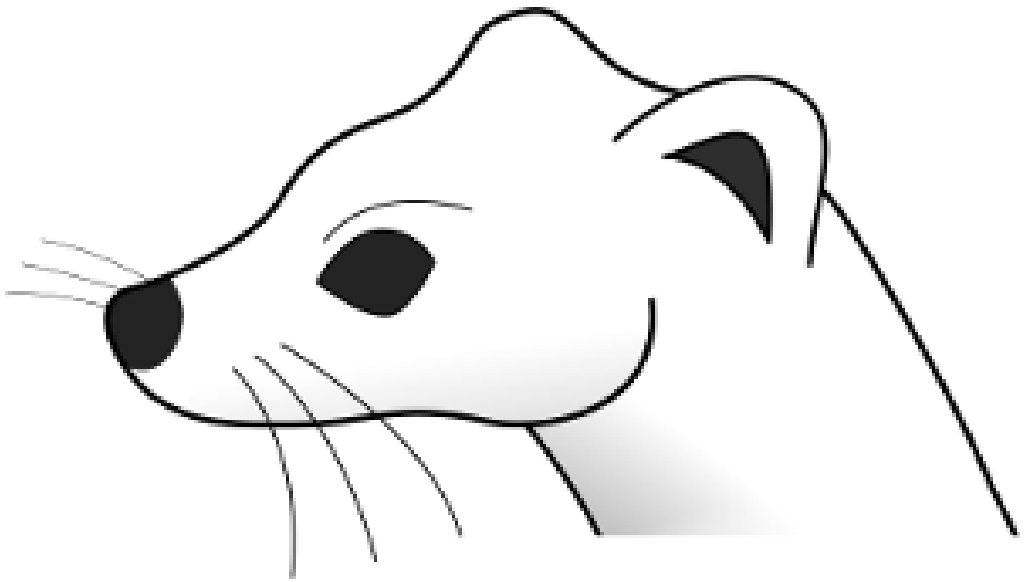
Finding realtime hazards?

- Manual code review
- Runtime analysis → jack-interposer

Finding realtime hazards?

- Manual code review
- Runtime analysis → jack-interposer
- Static analysis → stoat

Stoat



- **ST**atic (LLVM)
- **O**bject
- **A**nalysis
- **T**ool

What does stoat do?

1. hooks onto LLVM (.c/.cpp → .bc)

What does stoat do?

1. hooks onto LLVM (.c/.cpp $\rightarrow$ .bc)
2. extracts call graph (.bc $\rightarrow$ metadata)

What does stoat do?

1. hooks onto LLVM (.c/.cpp $\rightarrow$ .bc)
2. extracts call graph (.bc $\rightarrow$ metadata)
3. identifies realtime hazards (metadata $\rightarrow$ errors)

Translations

```
int REALTIME main()
{
    int barbar;
    foo(barbar);
    baz();
    return 0;
}
```

Translations

```
; ModuleID = '<stdin>'
target datalayout = "e-m:e-i64:64-f80:128-n8:16:32:6
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
}
```

Translations

```
; ModuleID = '<stdin>'
target datalayout = "e-m:e-i64:64-f80:128-n8:16:32:6
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
}
```

Translations

```
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 -f Makefile.binary CONFIG=Release DEBCXXFLAGS=""
    DEBLDFLAGS=""
make -f Makefile.ttl_generator CONFIG=Release
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/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
Parsing './standalone/builds/linux/build/intermediate/Release/
```

```
 - juce::AudioProcessorGraph::Node::~Node() _ZN4juce19AudioPr
 - juce::AudioProcessorGraph::Node::~Node() _ZN4juce19AudioPr
 - juce::AudioProcessorGraph::Node$vtable1 ?@? : Deduced Real
 - juce::ReferenceCountedObject$vtable1 ../../../JUCE/modules/
 - juce::ReferenceCountedObject::decReferenceCount() _ZN4juce
 - juce::var::VariantType_Object::cleanUp(juce::var::ValueUni
 - juce::var::VariantType_Object$vtable22 ?@? : Deduced Realt
 - juce::var::VariantType$vtable22 ../../../JUCE/modules/juce_
 - juce::var::~var() _ZN4juce3varD2Ev anonymous@? : Deduced R
 - juce::var::~var() _ZN4juce3varD1Ev ../../../src/common/loa
 - LoadSave::loadPatch(int, int, int, SynthBase*, std::map<st
 - MidiManager::processMidiMessage(juce::MidiMessage const&,
 - SynthBase::processMidi(juce::MidiBuffer&, int, int) _ZN9Sy
 - HelmPlugin::processBlock(juce::AudioBuffer<float>&, juce::
##The Contradiction Reasons:
 - operator delete(void*) _ZdlPv ../../../JUCE/modules/juce_a


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

```
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

```
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$vtable26
##The Deduction Chain:
 - mopo::ProcessorRouter$vtable26 : Deduced Realtime
 - mopo::ProcessorRouter::setBufferSize(int) _ZN4mopo15Proces
 - mopo::ProcessorRouter$vtable6 : Deduced Realtime
 - SynthBase::processAudio(juce::AudioBuffer<float>*, int, int
 - HelmPlugin::processBlock(juce::AudioBuffer<float>&, juce::
##The Contradiction Reasons:
 - mopo::ProcessorRouter::updateAllProcessors() _ZN4mopo15Pro


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) _ZN4mopo
##The Deduction Chain:
 - mopo::HelmVoiceHandler::noteOn(double, double, int, int) _
 - mopo::HelmVoiceHandler$vtable4 : Deduced Realtime
 - mopo::HelmEngine::noteOn(double, double, int, int) _ZN4mop
 - mopo::HelmEngine$vtable8 : Deduced Realtime
 - MidiManager::processMidiMessage(juce::MidiMessage const&,
 - SynthBase::processMidi(juce::MidiBuffer&, int, int) _ZN9Sy
 - HelmPlugin::processBlock(juce::AudioBuffer<float>&, juce::
##The Contradiction Reasons:
 - std::list<double, ...>::push_front(double const&) _ZNSt4li
 - std::list<mopo::Voice*, ...>::push_back(mopo::Voice* const
```

A practical example

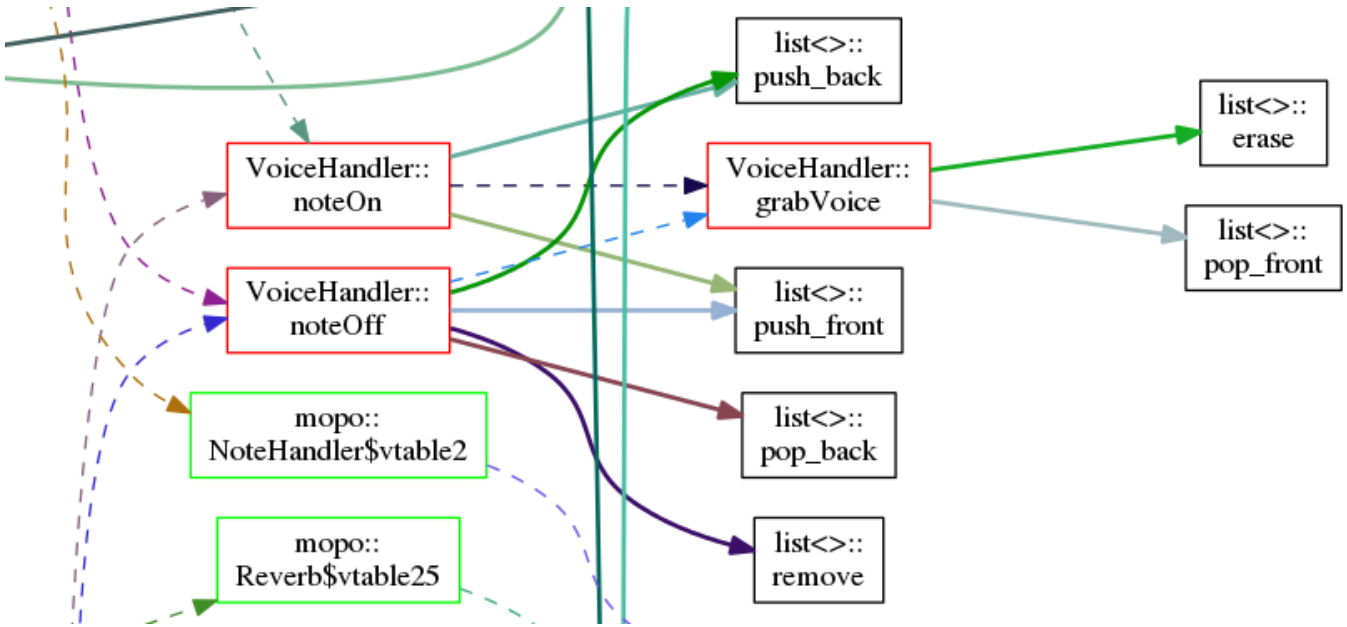
```
void VoiceHandler::noteOn(mopo_float note,
    mopo_float velocity, int sample, int channel) {

  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;
}
```

```
(gdb) bt
#0  0xb7fd8cc4 in malloc () from /work/mytmp/jack_interposer/
#1  0xb7bfcab5 in operator new(unsigned int) () from /usr/lib/
#2  0x080dadc7 in mopo::VoiceHandler::noteOn(double, double, i
#3  0x0814184b in mopo::HelmVoiceHandler::noteOn(double, doubl
#4  0x08135624 in mopo::HelmEngine::noteOn(double, double, int
#5  0x080e7ffc in MidiManager::processMidiMessage(juce::MidiMe
#6  0x080eb5a2 in SynthBase::processMidi(juce::MidiBuffer&, in
#7  0x0812eb01 in HelmStandaloneEditor::getNextAudioBlock(juce
#8  0x0812ebda in non-virtual thunk to HelmStandaloneEditor::g
#9  0x0816b808 in juce::AudioSourcePlayer::audioDeviceIOCallba
#10 0x08168e37 in juce::AudioDeviceManager::audioDeviceIOCallb
#11 0x08177add in juce::AudioDeviceManager::CallbackHandler::c
#12 0x08174d5c in juce::JackAudioIODevice::process(int) ()
#13 0x081748bb in juce::JackAudioIODevice::processCallback(uns
#14 0xb79b0a36 in jack_process_thread_work (arg=0x9b814e0) at
#15 0xb79b72dc in jack_thread_proxy (varg=0x9b6ce78) at threa
#16 0xb7caa955 in start_thread () from /lib/libpthread.so.0
#17 0xb7ab085e in clone () from /lib/libc.so.6
```

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

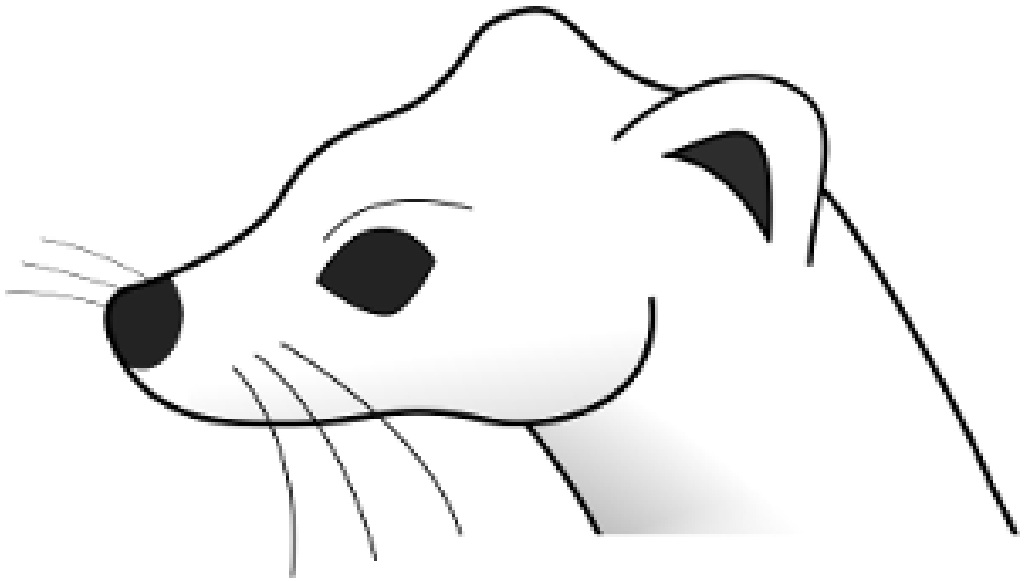
Stoat's use in the wild

- 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