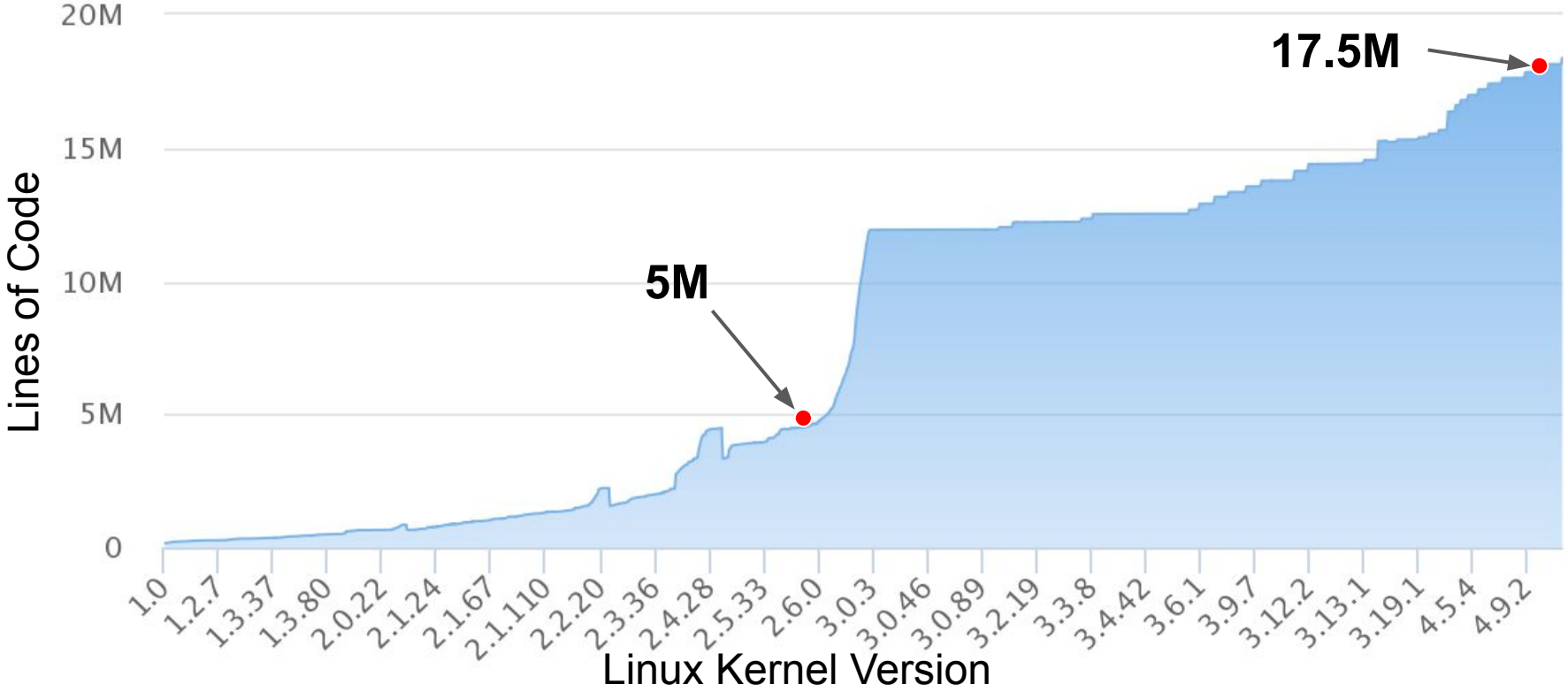
Razor: A Framework for Post-deployment Software Debloating

Chenxiong Qian, Hong Hu, Mansour Alharthi,
Pak Ho (Simon) Chung, Taesoo Kim, Wenke Lee

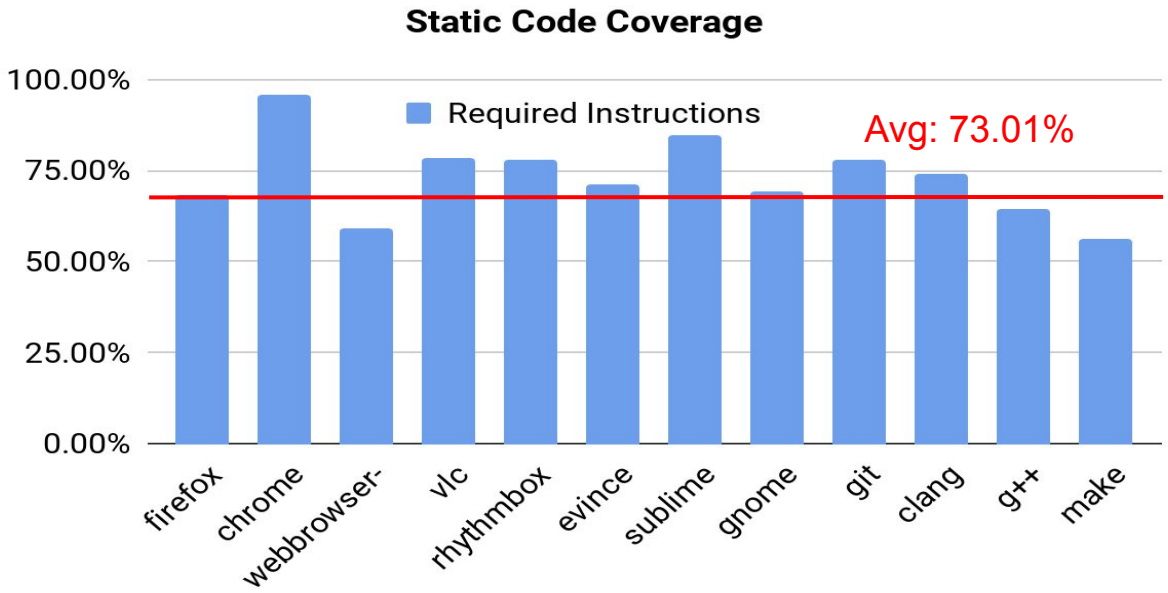
CREATING THE NEXT®

Software Is Getting Bigger



Software Is Bloated

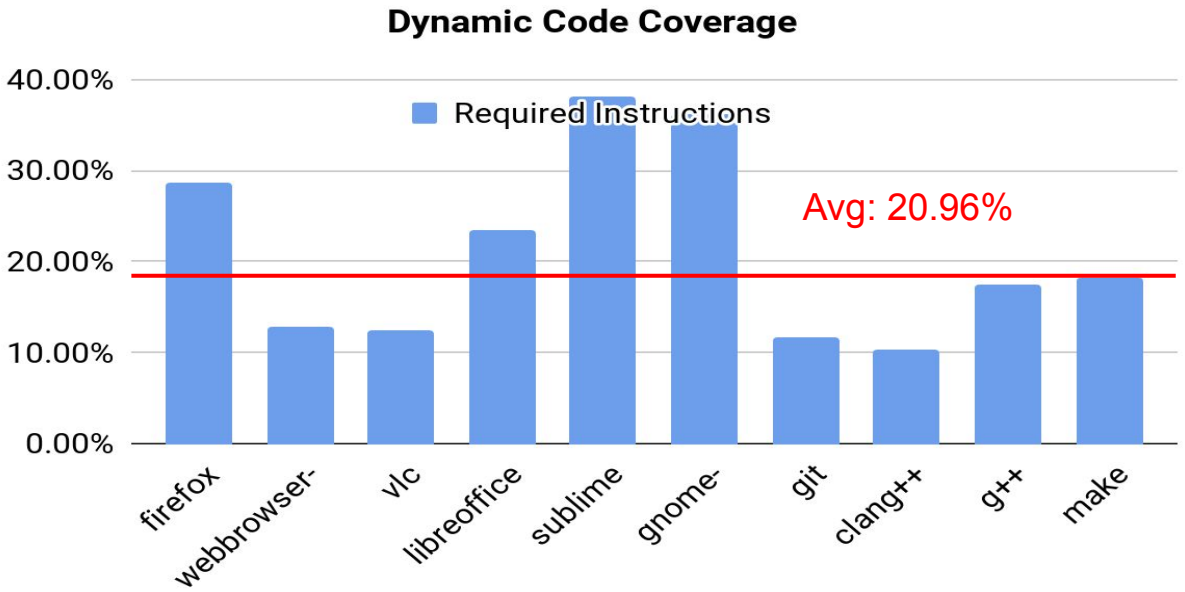
➢ Software contains dead code.



Quach *et al*. (FEAST'17)

Software Is Bloated

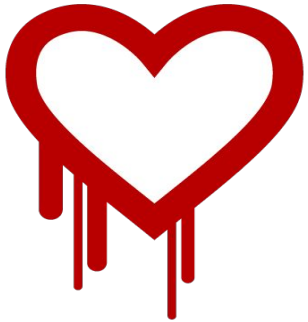
➢ Software contains code that is never used by users.



Quach *et al*. (FEAST'17)

Bloated Code Increases Attack Surface

- Example1: HeartBleed



  - TLS heartbeat extension.
  - Not used by most users.
  - Enabled in default.

- Example2: CVE-2014-0038
  - *compat_sys_recvmmsg* handles recvmmsg system call for x32 ABI.
  - x32 ABI takes advantage of the 64-bit environment while using 32-bit pointers for less overhead.
  - **No such programs exist in real world!**
  - **X32 is enabled by default in all major distributions like Ubuntu!**

Software Debloating

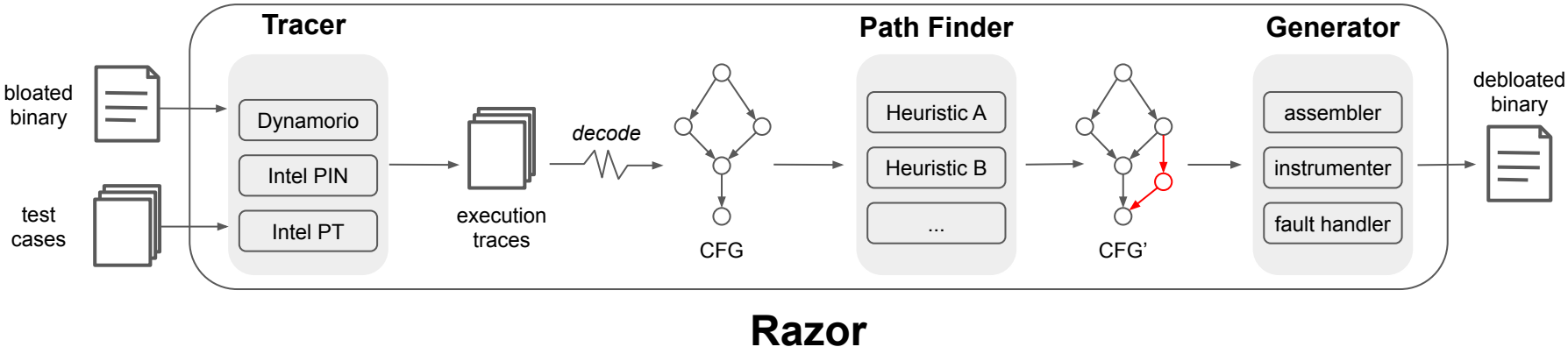
- All existing software debloating systems have the following limitations:
  - Require source code.
    - Source code is not always accessible to users.
    - It’s challenging and time-consuming to recompile source code.
  - Assume test cases are complete.
    - This assumption mostly fails in real world.
    - Impossible to provide complete test cases for a particular functionality.

Razor

- Performs code reduction for deployed **binaries**.

- Uses **heuristics** to infer related code for given test cases.

Overview


Tracer
Path Finder
Generator
bloated binary
test cases
Dynamorio
Intel PIN
Intel PT
execution traces
decode
CFG
Heuristic A
Heuristic B
...
CFG'
assembler
instrumenter
fault handler
debloated binary
Razor

Tracer

- Multiple tracers
  - Software-based tracers (Dynamorio, Intel PIN)
    - Complete trace
    - Significant overhead
  - Hardware-based tracer (Intel PT)
    - Small overhead
    - Incomplete trace
  - Programs under different tracing environments show divergent paths.
- The collected trace contains three parts:

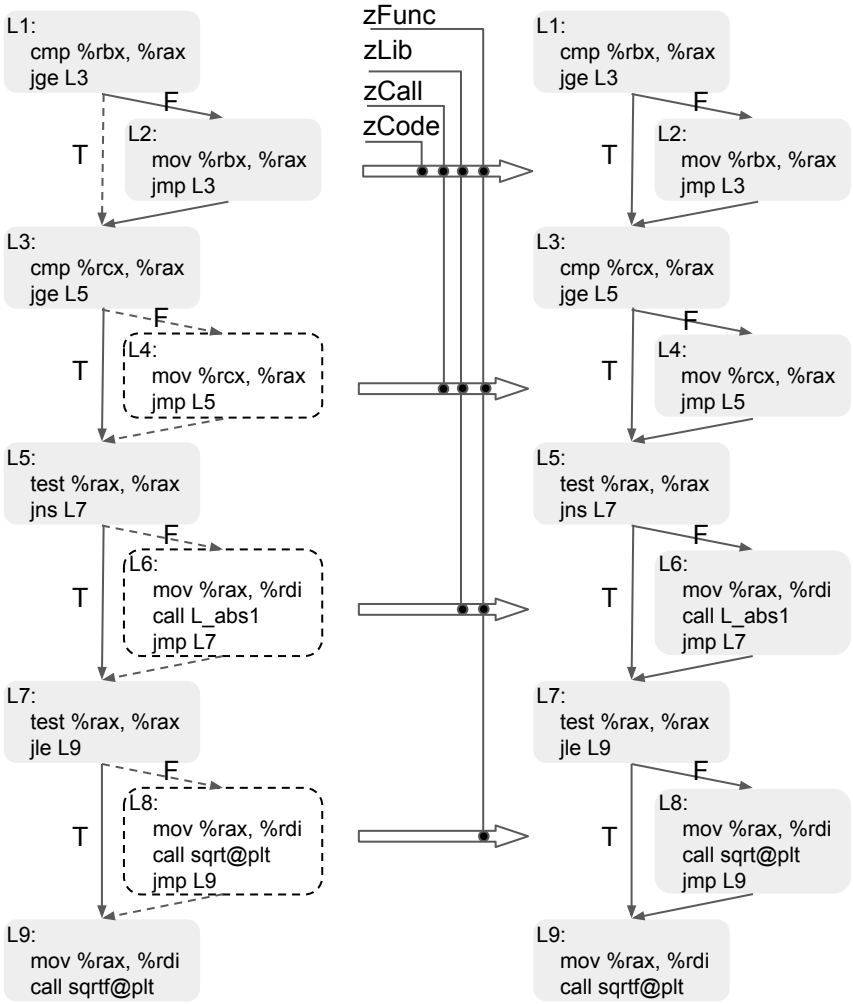
Executed Blocks
[0x4005c0, 0x4005f2]
[0x400596,0x4005ae]
...

Conditional Branches
[0x4004e3: true]
[0x4004ee: false]
[0x400614: true, false]
...

Indirect Calls/Jumps
[0x400677, 0x4005e6#18, 0x4005f6#6
...

Path Finder

- Four Heuristics
  - zCode (zero code)
    - Only adds edges.
  - zCall (zero call)
    - Call instructions are disallowed.
  - zLib (zero library call)
    - Non-executed library calls are disallowed.
  - zFun (zero functionality)
    - Library calls with different functionalities are disallowed.



Generator

- **Assembler**
  - **Disassembles the binary based on the expanded CFG.**
  - **Symbolizes basic blocks.**
- Instrumenter
  - Concretizes targets of indirect calls/jumps.
  - Fixes callback function pointers.
  - Enforce allowed control-flows.
- Fault handler
  - Dumps call stacks and exits the execution.
- Rewriter
  - Compiles the instrumented assembly code to an object file.
  - Copies the code section into original binary.
  - Fixes exception handlers' addresses in `.gcc_except_table` section.

Generator

- Assembler
  - Disassembles the binary based on the expanded CFG.
  - Symbolizes basic blocks.
- **Instrumenter**
  - **Concretizes targets of indirect calls/jumps.**
  - **Fixes callback function pointers.**
  - **Enforces allowed control-flows.**
- Fault handler
  - Dumps call stacks and exits the execution.
- Rewriter
  - Compiles the instrumented assembly code to an object file.
  - Copies the code section into original binary.
  - Fixes exception handlers' addresses in `.gcc_except_table` section.

Generator

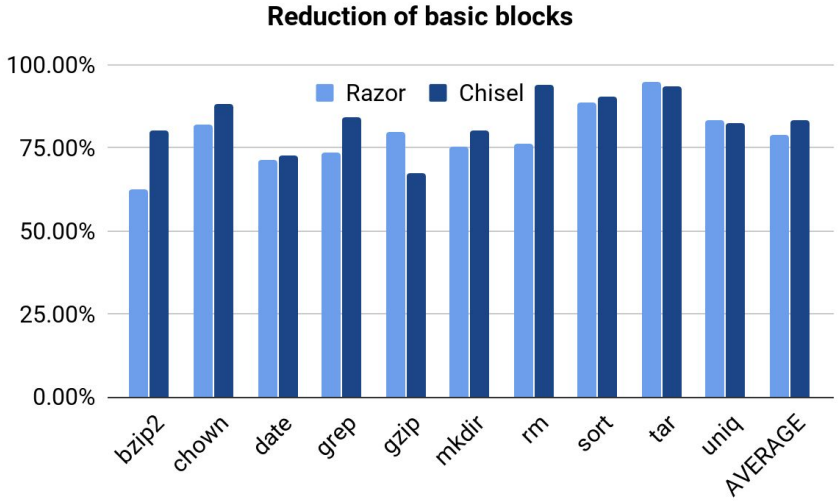
- Assembler
  - Disassembles the binary based on the expanded CFG.
  - Symbolizes basic blocks.
- Instrumenter
  - Concretizes targets of indirect calls/jumps.
  - Fixes callback function pointers.
  - Enforce allowed control-flows.
- **Fault handler**
  - **Dumps call stacks and exits the execution.**
- Rewriter
  - Compiles the instrumented assembly code to an object file.
  - Copies the code section into original binary.
  - Fixes exception handlers' addresses in `.gcc_except_table` section.

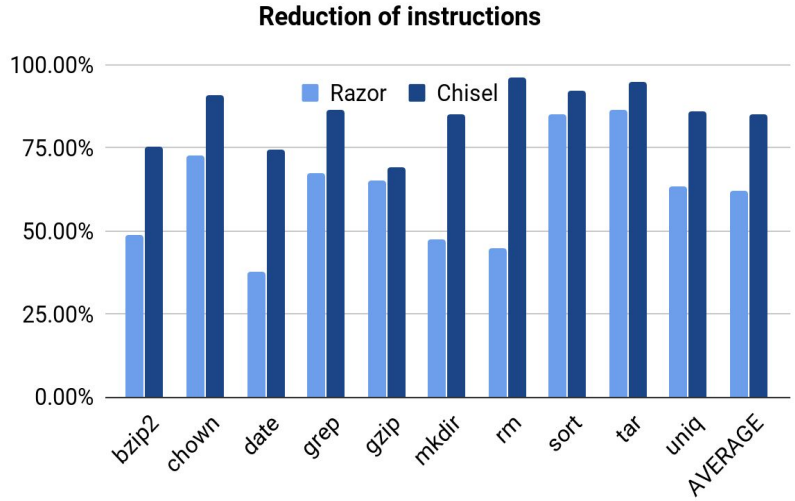
Generator

- Assembler
  - Disassembles the binary based on the expanded CFG.
  - Symbolizes basic blocks.
- Instrumenter
  - Concretizes targets of indirect calls/jumps.
  - Fixes callback function pointers.
  - Enforce allowed control-flows.
- Fault handler
  - Dumps call stacks and exits the execution.
- **Rewriter**
  - **Compiles the instrumented assembly code to an object file.**
  - **Copies the code section into original binary.**
  - **Fixes exception handlers' addresses in `.gcc_except_table` section.**

Code Reduction

- Comparing with Chisel
  - Basic blocks
    - Razor -- 78.8%, Chisel -- 83.4%
  - Instructions
    - Razor -- 61.9%, Chisel -- 85.1%





Functionality Validation

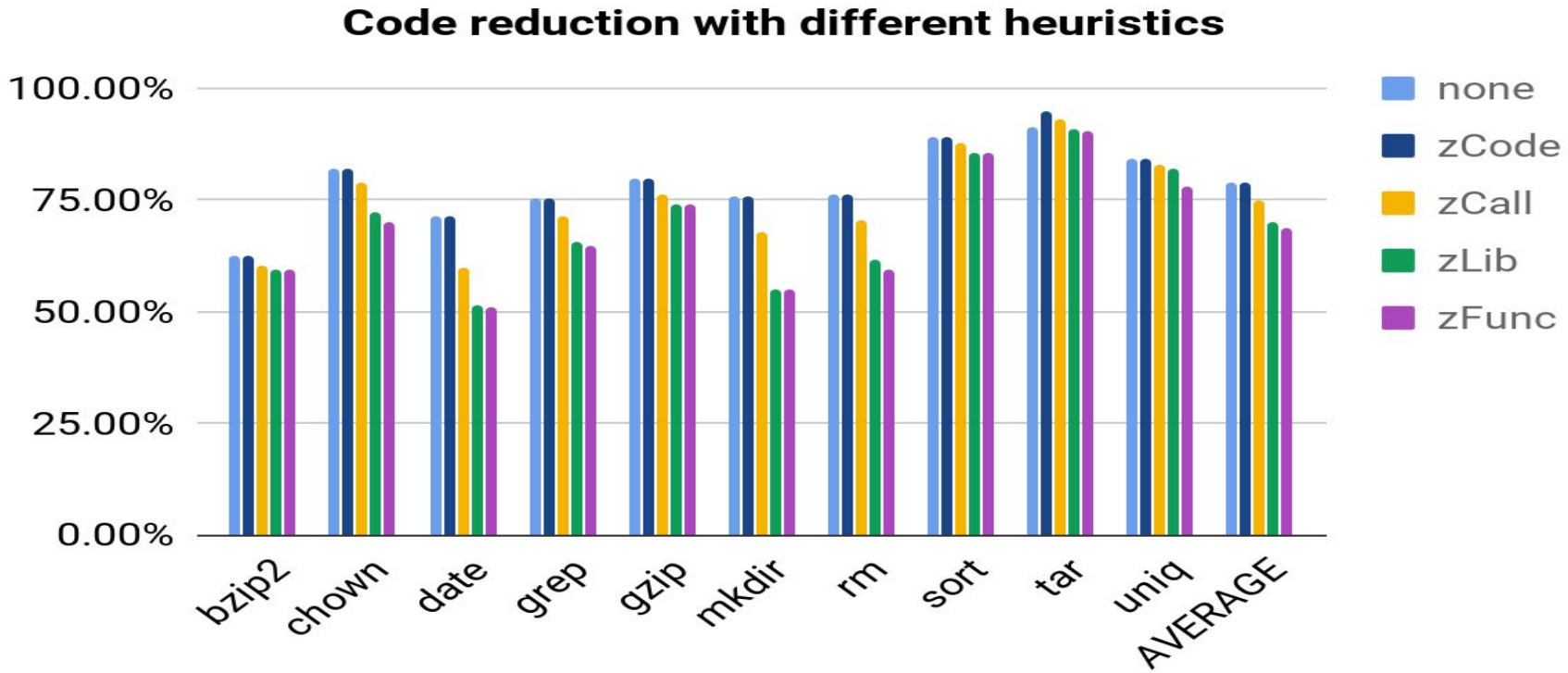
➢ Run the debloated binaries on the same test cases.

Program	# of Tests	Failed by Chisel				Failed by Razor
		W	I	C	M	
bzip2	6	2	--	2	--	-- (zLib)
chown	14	--	--	--	--	-- (zFunc)
date	50	5	--	3	--	-- (zLib)
grep	26	--	--	--	6	-- (zLib)
gzip	5	--	1	--	--	-- (zLib)
mkdir	13	--	--	--	1	-- (zLib)
rm	4	2	--	--	--	-- (zFunc)
sort	112	--	--	--	--	-- (zCall)
tar	26	3	--	--	4	-- (zCall)
uniq	16	--	--	--	--	-- (zCall)

W : Wrong operation
I : Infinite loop
C : Crash
M : Missing output

Effectiveness of Heuristics

➢ Run the debloated binaries on the different test cases.



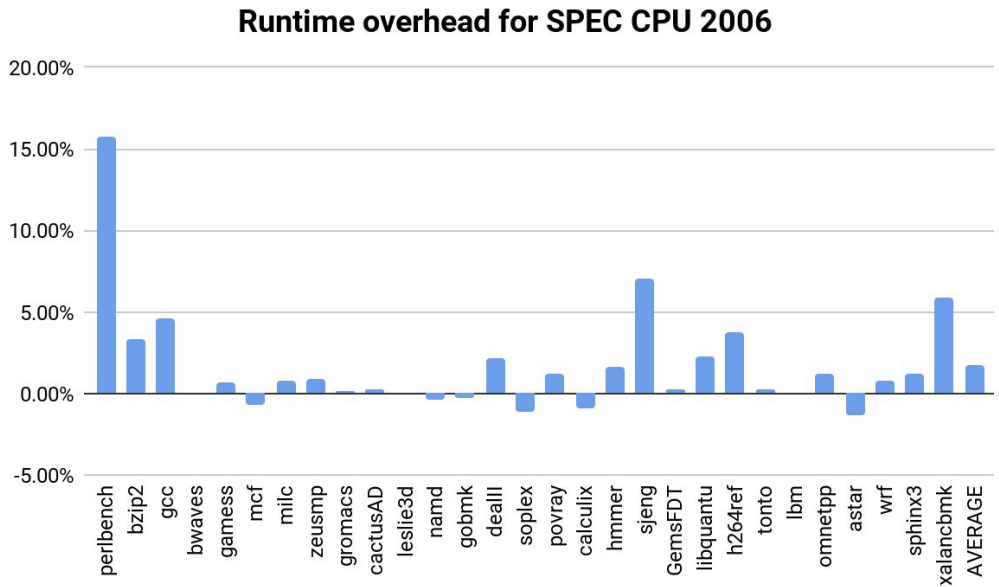
Security Benefits

Program	CVE	Orig	Chisel	Razor
bzip2	CVE-2010-0405	✓		
	CVE-2008-1372	✘	✓	
	CVE-2005-1260	✘	✓	
chown	CVE-2017-18018*	✓	✘	✘
date	CVE-2014-9471*	✓	✘	✓
grep	CVE-2015-1345*	✓	✘	✘
	CVE-2012-5667	✘	✓	
gzip	CVE-2005-1228*	✓	✘	✘
	CVE-2009-2624	✓		
	CVE-2010-0001	✓	✘	✘
mkdir	CVE-2005-1039*	✓		
rm	CVE-2015-1865*	✓		
tar	CVE-2016-6321*	✓	✘	✓

✓ binary is vulnerable to the CVE.
✘ binary is not vulnerable to the CVE.
* CVEs with * are evaluated by Chisel.

Runtime Overhead

- On average, Razor introduces 1.7% slowdown.
  - 15.8% overhead for *perlbench*



Real-world Software Debloating

- Firefox
  - Load top 50 Alexa websites.
  - Randomly pick 25 websites for debloating, and use the other 25 websites for testing.
- FoxitReader
  - Open and scroll 55 different PDF files.
  - Randomly pick 15 files for debloating, and use the other 40 files for testing.

Heuristic	Firefox		FoxitReader	
	crash-sites	code-reduction	crash-PDFs	code-reduction
none	13	67.6%	39	89.8%
zCode	13	68.0%	10	89.9%
zCall	2	63.1%	5	89.4%
zLib	0	60.1%	0	87.0%
zFunc	0	60.0%	0	87.0%

Real-world Software Debloating

- Use N-fold validation approach to apply zLib heuristic on Firefox.
  - Split Alexa's top 50 websites into five groups.
  - Select two groups (20 websites) for debloating and use the other 30 for testing.

Group ID	# of Failed Websites	Code Reduction	Failed Websites
G01	1	59.3%	wordpress.com
G02	0	59.3%	
G03	1	59.3%	wordpress.com
G04	1	59.3%	twitch.tv
G12	1	59.3%	wordpress.com
G13	1	59.5%	wordpress.com
G14	2	59.5%	twitch.tv, wordpress.com
G23	1	59.3%	twitch.tv
G24	1	59.3%	twitch.tv
G34	2	59.6%	twitch.tv, wordpress.com

Per-site Browser Isolation

➢ Create minimal versions of web browsers for particular websites.

Type	Website	Code Reduction	Heuristic	Benefits
Banking	bankofamedica.com	69.4%	zCall	6.3%
	chase.com	69.6%	zCall	6.5%
	wellsfargo.com	68.8%	zCall	5.7%
	all-3	68.1%	zCall	5.0%
E-commerce	amazon.com	71.4%	none	3.8%
	ebay.com	70.7%	none	3.1%
	ikea.com	70.6%	none	3.0%
	all-3	70.4%	none	2.8%
Social Media	facebook.com	70.8%	zCall	7.7%
	instagram.com	71.6%	zCall	8.5%
	twitter.com	74.0%	none	6.4%
	all-3	71.8%	none	4.2%

Summary

- Performs code reduction for deployed **binaries**.

- Uses **heuristics** to infer related code for given test cases.

Questions?