

TACE

Rapit Taint Assisted Concolic Execution



Ridhi Jain

(ridhi.jain@tii.ae)

(Presenter)



Norbert Tihanyi

(norbert.tihanyi@tii.ae)



Mthandazo Ndhlovu

(mthandazo.ndhlovu@tii.ae)



Mohamed Amine Ferrag

(mohamed.ferrag@tii.ae)



Lucas Carvalho Cordeiro

(lucas.cordeiro@manchester.uk)

Why Software Testing?

Why Software Testing?

Dec' 21

WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS MORE

A Year Later, That Brutal Log4j Vulnerability Is Still Lurking

Sept' 22

CNBC MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV INVESTING CLUB PRO

AUTOS

GM's Cruise recalls and updates self-driving software in cars following crash

PUBLISHED THU, SEP 1 2022-12:40 PM EDT | UPDATED THU, SEP 1 2022-1:24 PM EDT

Jan' 23

T-MOBILE / MOBILE / TECH

T-Mobile announces another data breach, impacting 37 million accounts

Feb' 23

CNBC MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV INVESTING CLUB PRO

TECH

Tesla recalls 362,758 vehicles, says Full Self-Driving Beta software may cause crashes

PUBLISHED THU, FEB 16 2023-12:49 PM EST | UPDATED THU, FEB 16 2023-6:56 PM EST

May' 15

INSIDER Newsletters Login Subscribe

HOME > FINANCE

A software problem caused a brand-new Airbus military plane to crash

July' 13

CNBC BUSINESS Markets Tech Media Success Perspectives Videos

PayPal accidentally credits man \$92 quadrillion

June' 22

HEALTH NEWS JUNE 1, 2022 / 4:01 AM

Pacemakers, insulin pumps can be hacked, experts say

June' 22

Healthcare IT News

CISA warns of Medtronic cardiac device security vulnerability

Software crashes may not only cause heavy **monetary losses** but can also be **life threatening**.

Motivation

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.



Symbolic execution can improve improve input quality.

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.



Symbolic execution can improve input quality.



Interpretation



No. of constraints

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.



Symbolic execution can improve input quality.



Interpretation



No. of constraints



SymCC [1],
SymQEMU[2]



Don't Interpret, Compile!

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.



Symbolic execution can improve improve input quality.



Interpretation



No. of constraints



SymCC [1],
SymQEMU[2]



Don't Interpret, Compile!



No. of constraints

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.



Symbolic execution can improve input quality.



Interpretation



No. of constraints



SymCC [1],
SymQEMU[2]



Don't Interpret, Compile!



No. of constraints



LSym[3]



Constraint Debloating

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

Motivation



Fuzzing can be helpful.



Depends on Quality of input.



Symbolic execution can improve input quality.



Interpretation



No. of constraints



SymCC [1],
SymQEMU[2]



Don't Interpret, Compile!



No. of constraints



LSym[3]



Constraint Debloating



Manual Summary Creation



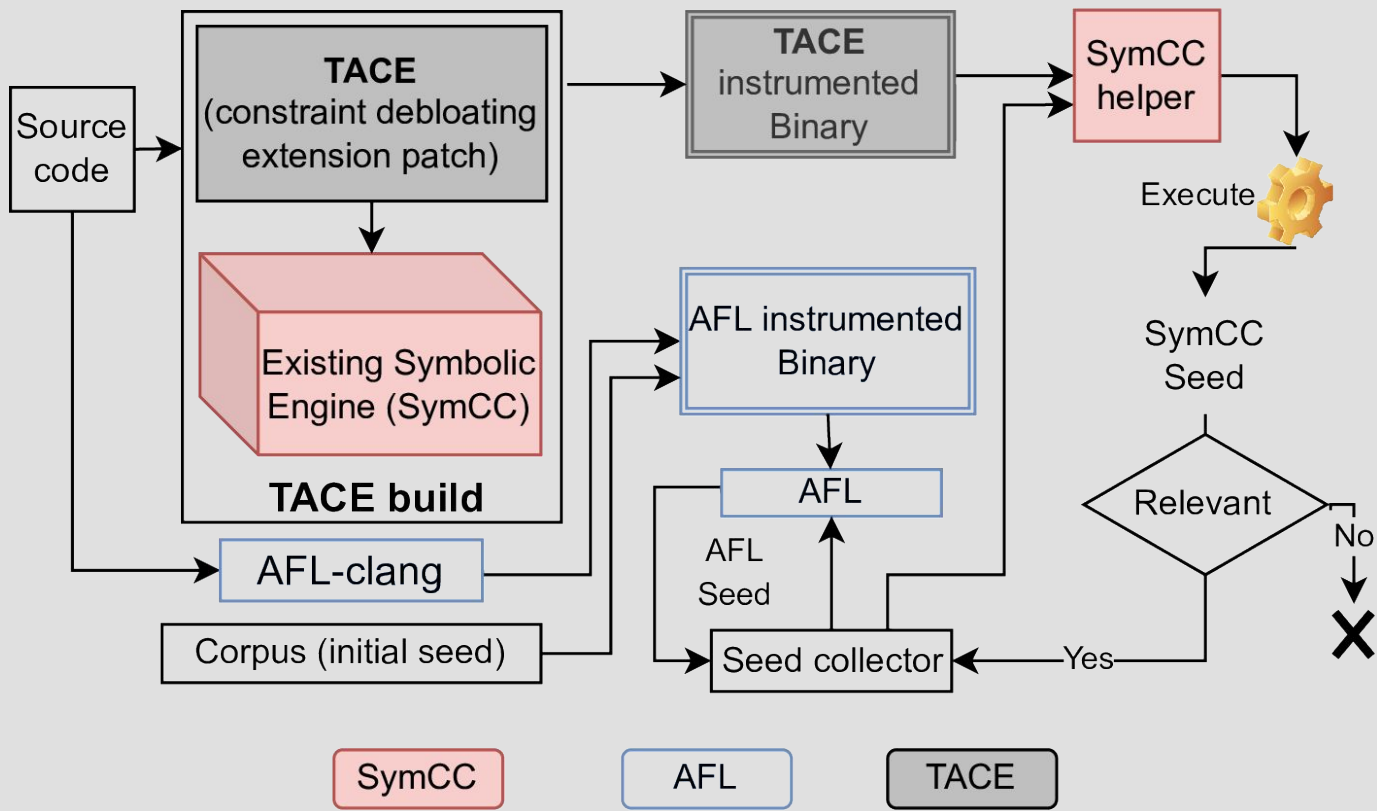
Unavailable

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!" *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

TACE - Hybrid Fuzzing Architecture



Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$

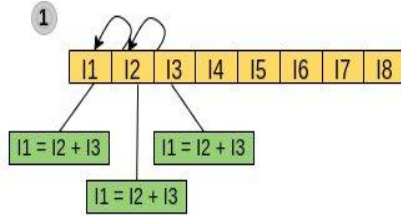
Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$



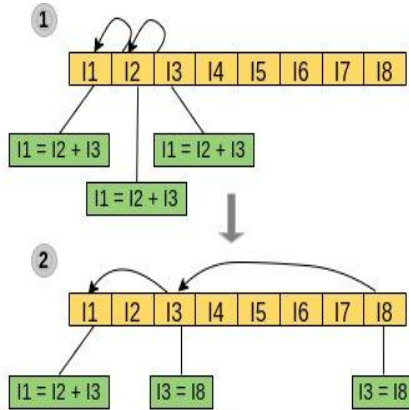
Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$



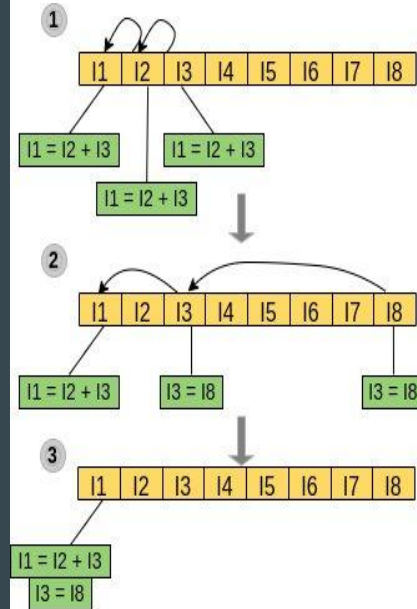
Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$



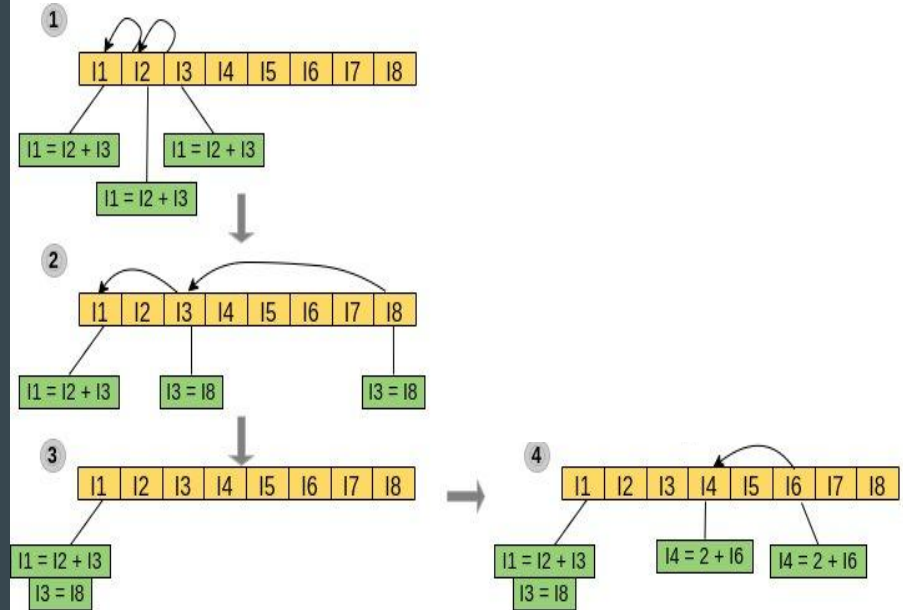
Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$



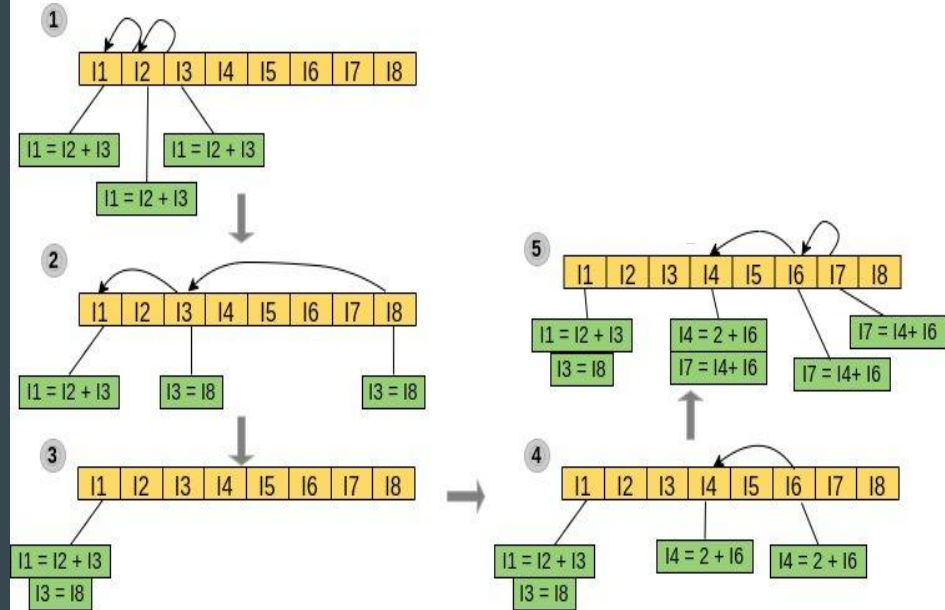
Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$



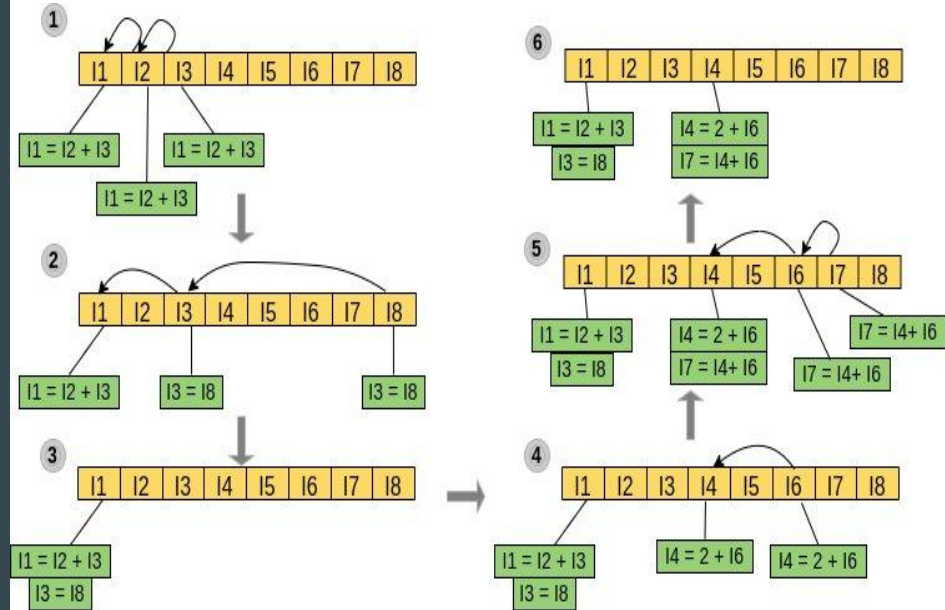
Taint Assisted Concolic Grouping

(Instruction 1) $I_1 = I_2 + I_3$

(Instruction 2) $I_3 = I_8$

(Instruction 3) $I_4 = 2 * I_6$

(Instruction 4) $I_7 = I_4 + I_6$



Research Questions

RQ1 What performance gains are offered by TACE compared to state-of-the-art tools?

RQ2 Does TACE report correct and reproducible bugs?

Results: Constraints Solving Time

Depth	SymQEMU (sec)	TACE (sec)	Symbolic Dep	Concrete Dep	Improvement (x Times)
1	4.28	4.30	3	2	0.99
2	7.44	5.69	4	3	1.30
3	9.76	6.26	4	4	1.55
4	15.02	7.84	4	6	1.91
5	26.11	9.35	4	10	2.79
6	122.65	7.81	4	18	15.7
7	145.50	4.38	4	34	0
8	280.48	5.59	4	66	33.2
					1
					50.1
					7

Results: Constraints Solving Time

Depth	SymQEMU (sec)	TACE (sec)	Symbolic Dep	Concrete Dep	Improvement (x Times)
1	4.28	4.30	3	2	0.99
2	7.44	5.69	4	3	1.30
3	9.76	6.26	4	4	1.55
4	15.02	7.84	4	6	1.91
5	26.11	9.35	4	10	2.79
6	122.65	7.81	4	18	15.7
7	145.50	4.38	4	34	0
8	280.48	5.59	4	66	33.2

1

RQ1 What performance gains are offered by TACE compared to state-of-the-art tools?

Hybrid Fuzzing Statistics from SymCC vs TACE

Test Details		SymCC					TACE					
Target	Project	Time (hh:mm:ss)	#Unique Hangs	#Unique tmouts	#Unique Crashes	#New Edges	#Cycles	#Unique Hangs	#Unique tmouts	#Unique Crashes	#New Edges	#Cycles
	minizip-ng	24:00:00	0	4	0	5	391	0	203	0	204	3.6k
	TCPDump	24:00:00	14	336	0	4148	223	36	1201	0	7	286
	GifLib	24:00:00	2	71	0	78	108k	13	62	7	72	181k
	OpenJpeg	24:00:00	0	1	0	3	89	10	894	0	71644	104k
	bzip2	24:00:00	0	2	0	4	76.2	0	2	0	4	229k

Hybrid Fuzzing Statistics from SymCC vs TACE

RQ2 Does TACE report correct and reproducible bugs?

Test Details		SymCC					TACE					
Target	Project	Time (hh:mm:ss)	#Unique Hangs	#Unique tmouts	#Unique Crashes	#New Edges	#Cycles	#Unique Hangs	#Unique tmouts	#Unique Crashes	#New Edges	#Cycles
minizip-ng		24:00:00	0	4	0	5	391	0	203	0	204	3.6k
TCPDump		24:00:00	14	336	0	4148	223	36	1201	0	7	286
GifLib		24:00:00	2	71	0	78	108k	13	62	7	72	181k
OpenJpeg		24:00:00	0	1	0	3	89	10	894	0	71644	104k
bzip2		24:00:00	0	2	0	4	76.2	0	2	0	4	229k

Metrics

CVSS Version 4.0

CVSS Version 3.x

CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:



NIST: NVD

Base Score: 7.1 HIGH

Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:H

Known Affected Software Configurations [Switch to CPE 2.2](#)

Configuration 1 [\(hide\)](#)

⚠️ cpe:2.3:a:giflib_project:giflib:5.2.1:*:*:*:*:*

[Show Matching CPE\(s\)](#)

🚨 CVE-2023-48161 Detail

Description

Buffer Overflow vulnerability in GifLib Project GifLib v.5.2.1 allows a local attacker to obtain sensitive information via the DumpScreen2RGB function in gif2rgb.c

Thank You



Questions?