Ein Kontrollflussüberwachungsdienst für KESO-Anwendungen

Simon Schuster

03.07.2015

Department of Computer Science 4
Friedrich-Alexander-Universität
Erlangen-Nürnberg

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

System Software Group

Zuverlässigkeit

Zuverlässigkeit und Fehlertoleranz

Kontrollflussfehler | Datenfehler

Kontrollflussüberwachung

„We present a[n] [...] version [...] (ECCA) [...] targeted for the **detection of control flow-errors** with **low overhead** and **low detection latency** [...]“

- Alkhalifa et al.: Design and Evaluation of System-Level Checks for On-Line Control Flow Error Detection

Zuverlässigkeit

Zuverlässigkeit und Fehlertoleranz

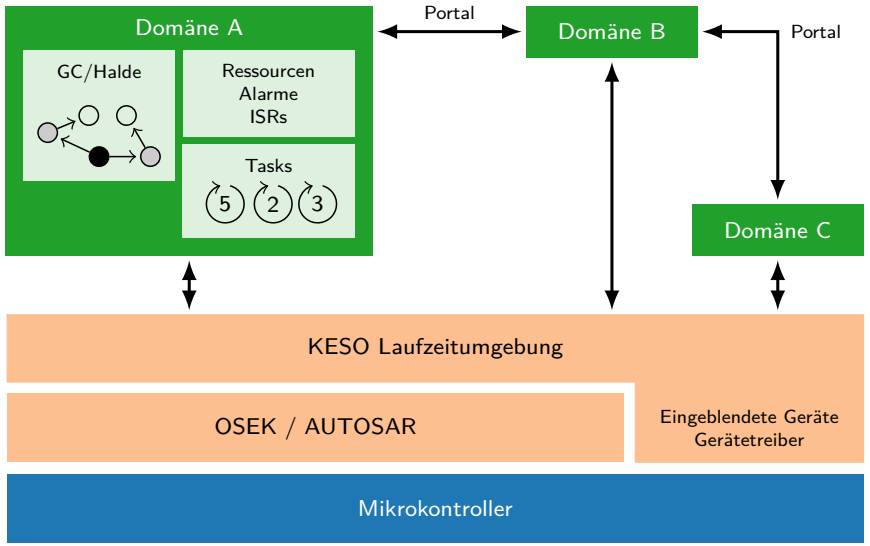
Kontrollflussfehler | Datenfehler

Kontrollflussüberwachung

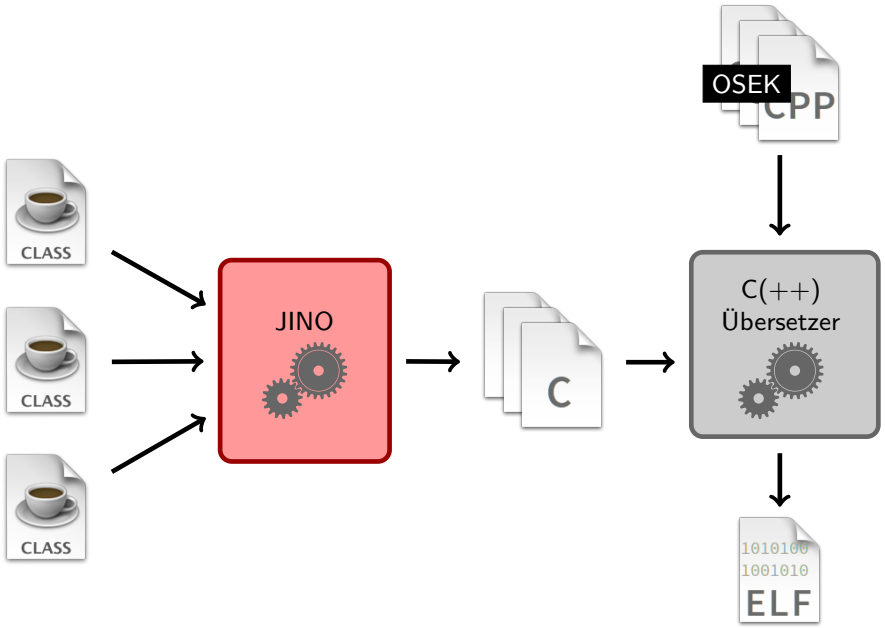
„We present a[n] [...] version [...] (ECCA) [...] targeted for the **detection of control flow-errors** with **low overhead** and **low detection latency** [...]“

- Alkhalifa et al.: Design and Evaluation of System-Level Checks for On-Line Control Flow Error Detection

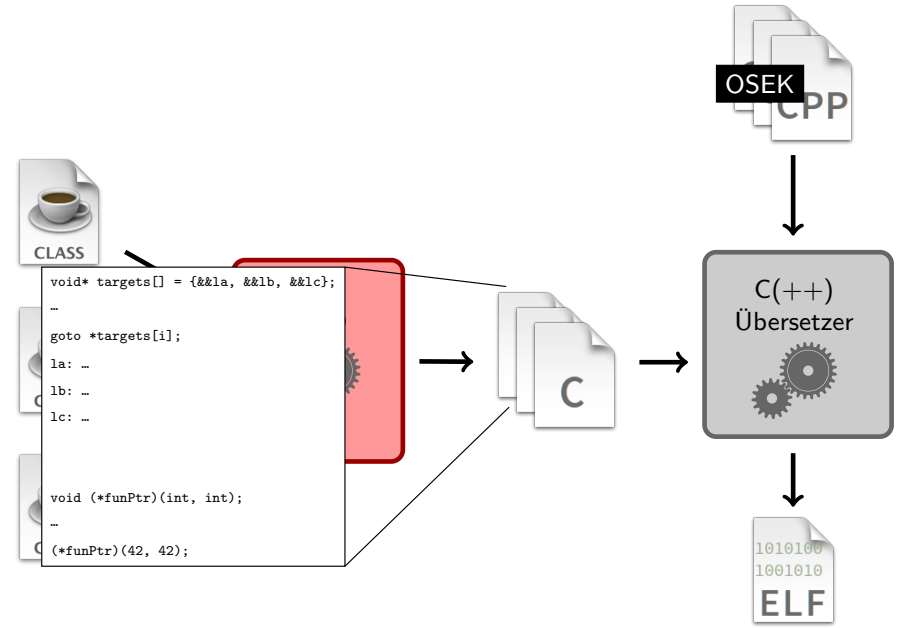
KESO-Aufbau



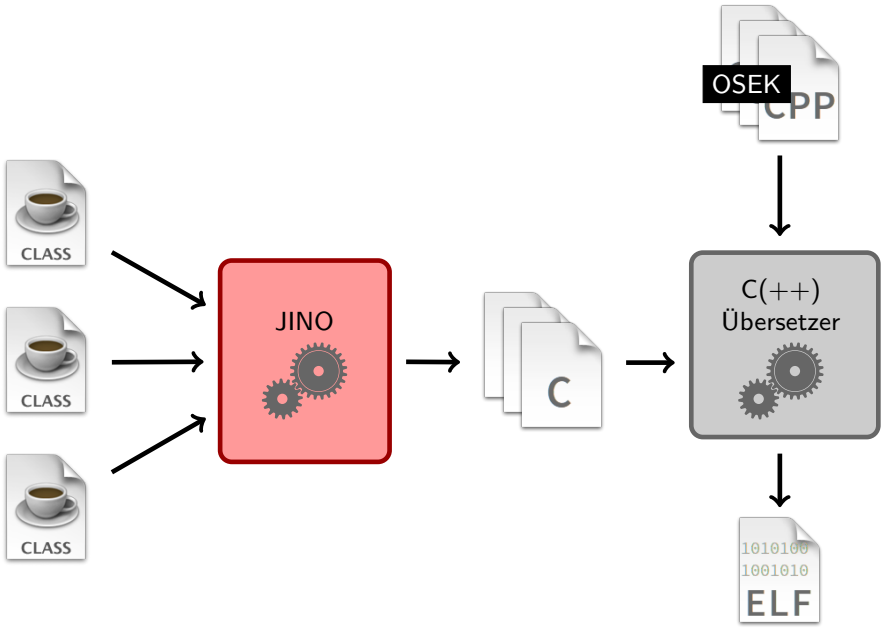
KESO-Bauprozess



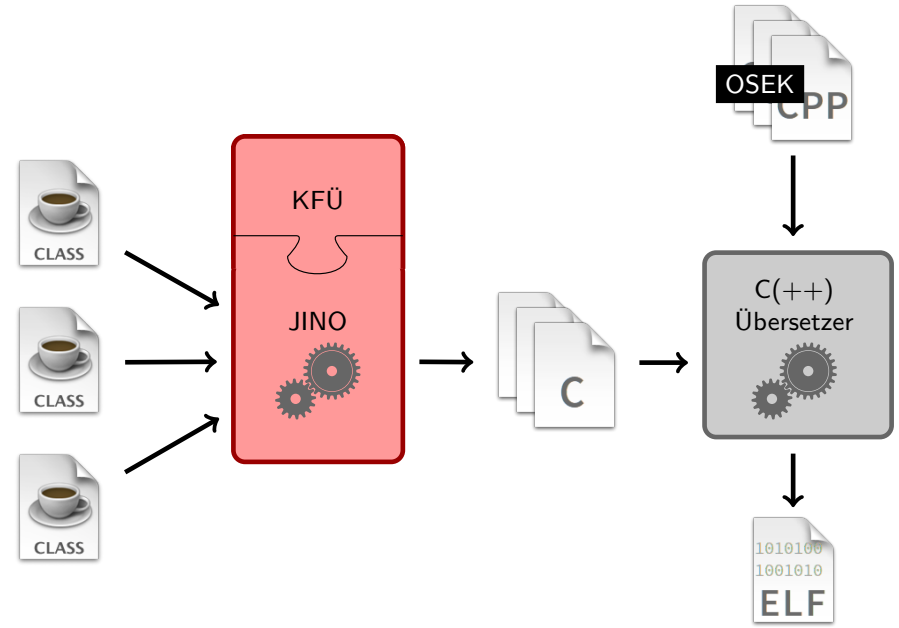
KESO-Bauprozess



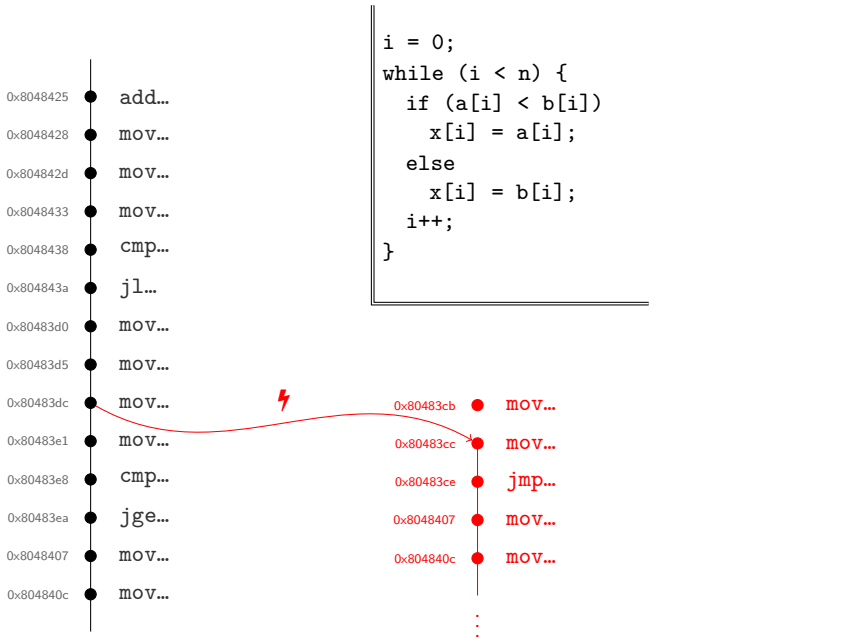
KESO-Bauprozess



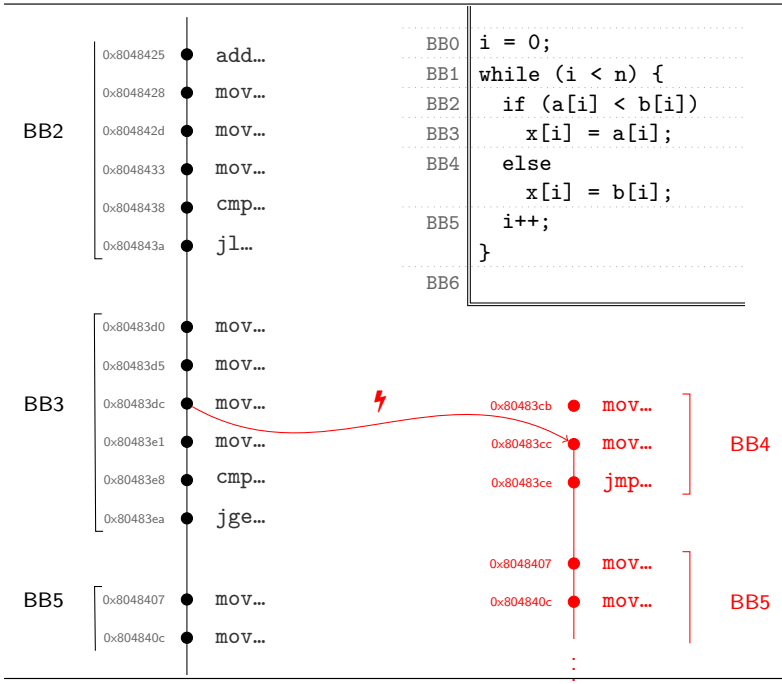
KESO-Bauprozess



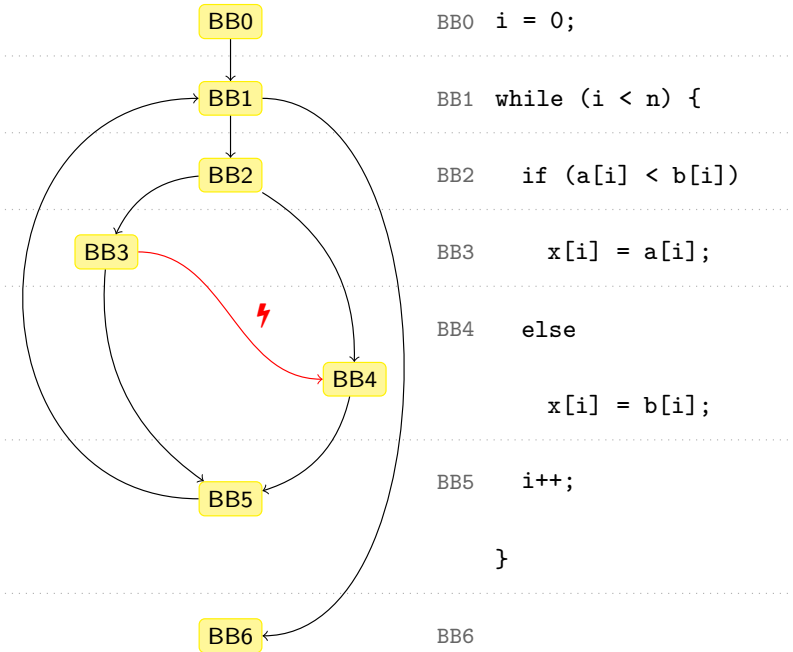
Kontrollflussüberwachung



Kontrollflussüberwachung



Kontrollflussüberwachung



Umgesetzte Verfahren

Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Umgesetzte Verfahren

Basisblockebene:

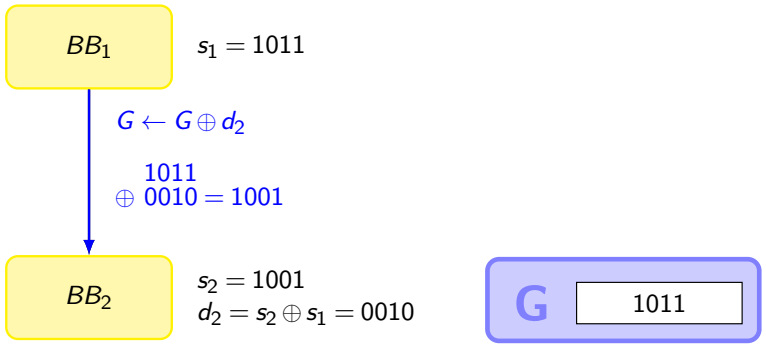
- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

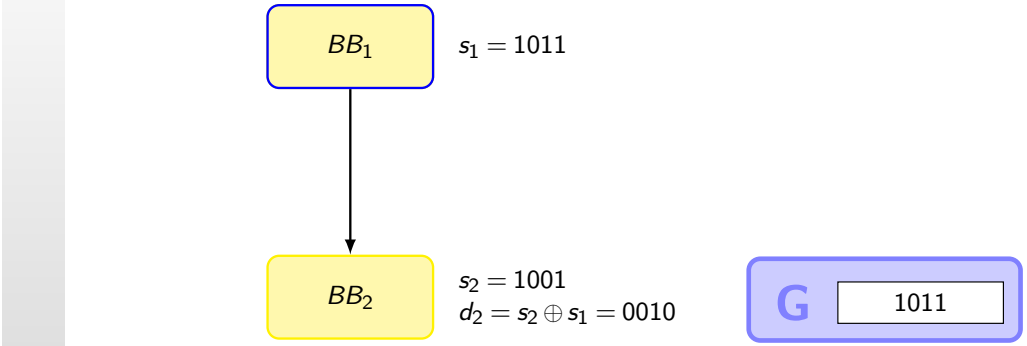
Control-Flow Checking by Software Signatures

CFCSS



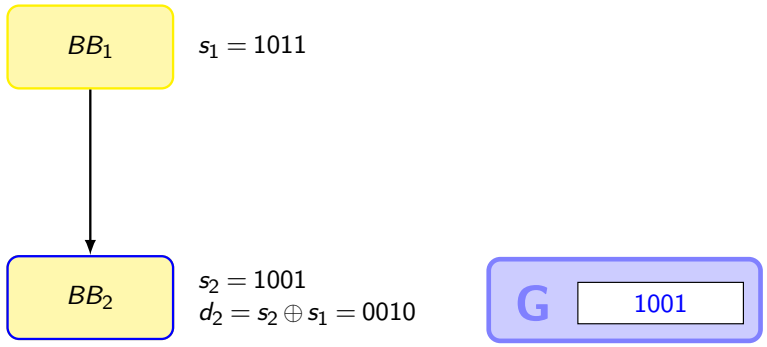
Control-Flow Checking by Software Signatures

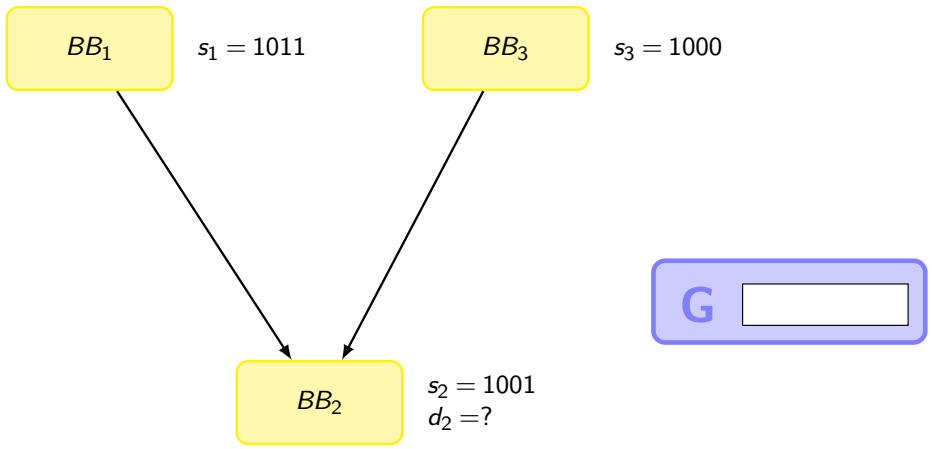
CFCSS

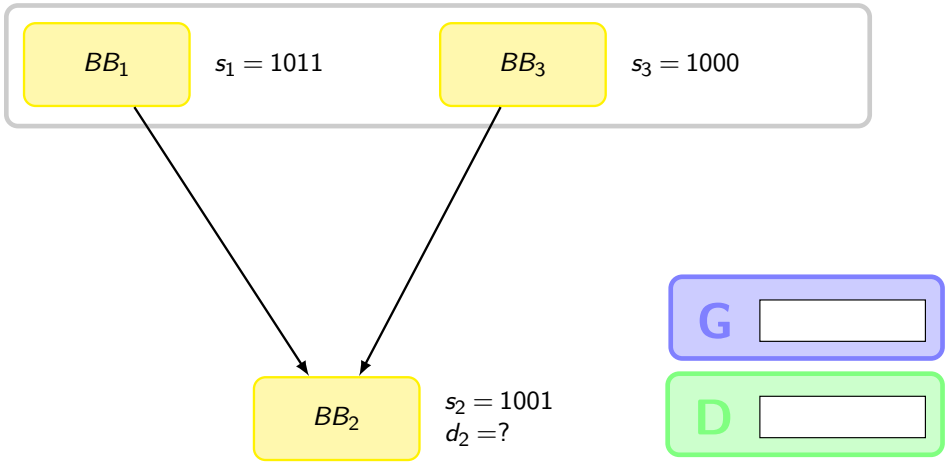


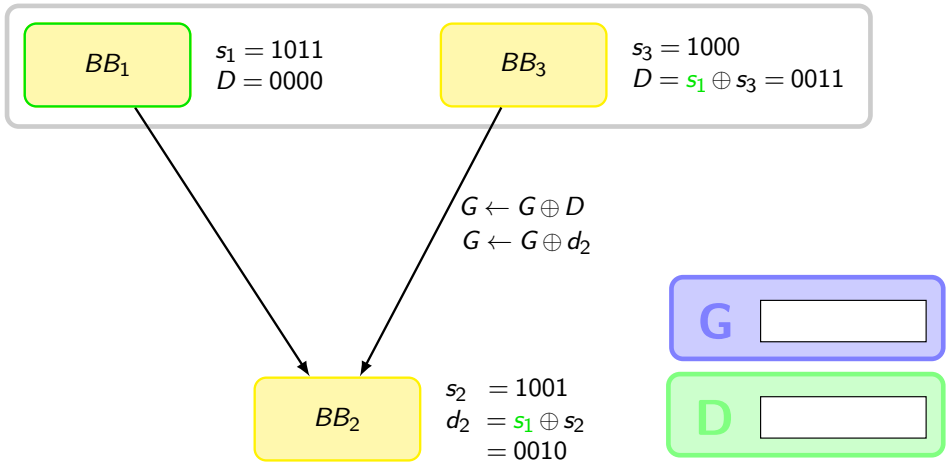
Control-Flow Checking by Software Signatures

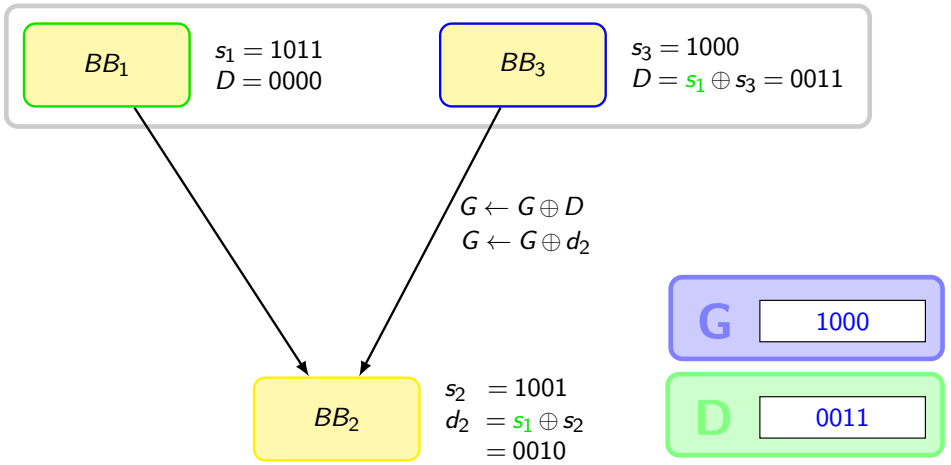
CFCSS

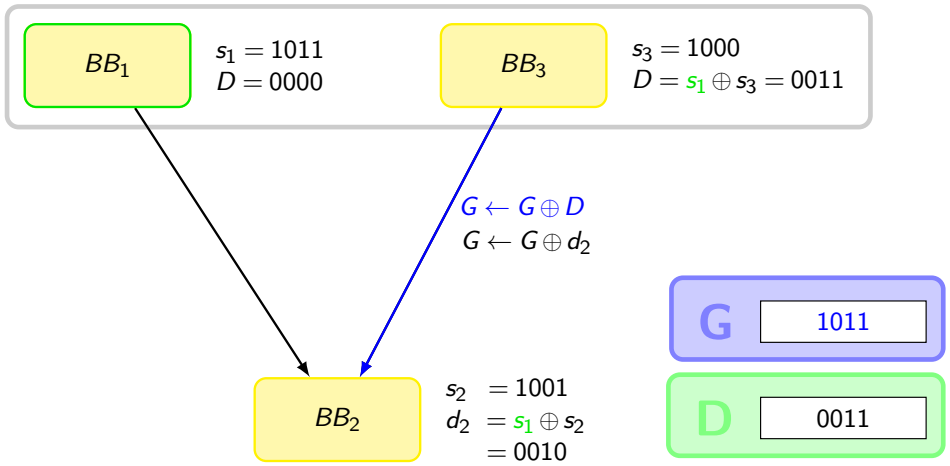


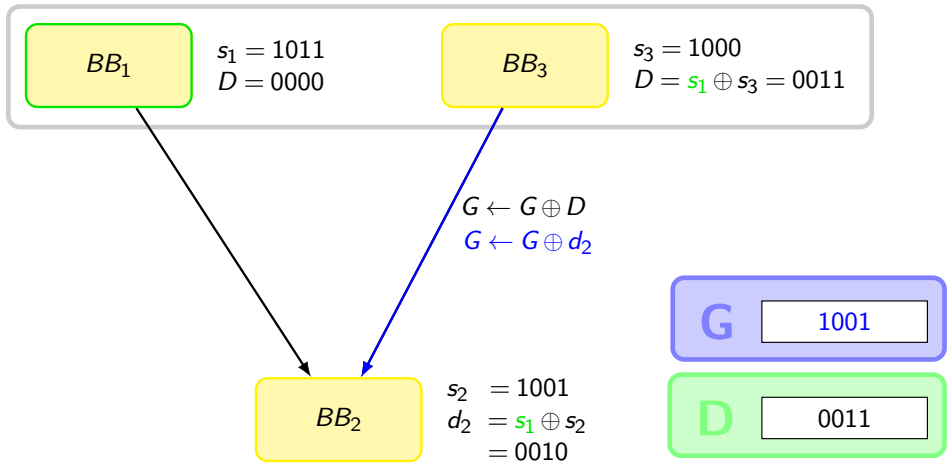

BB_1
$s_1 = 1011$
BB_3
$s_3 = 1000$
BB_2
$s_2 = 1001$
$d_2 = ?$
G

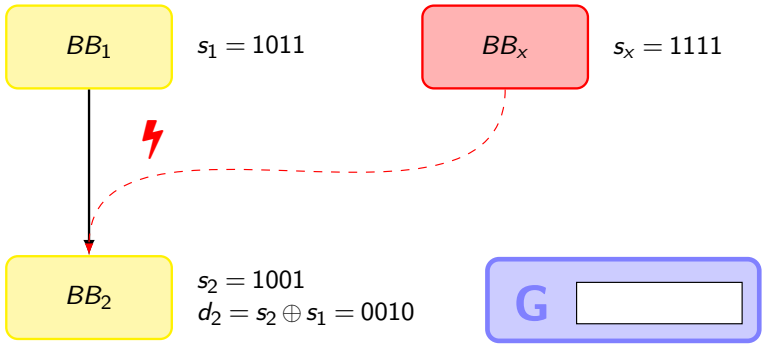

BB_1
$s_1 = 1011$
BB_3
$s_3 = 1000$
BB_2
$s_2 = 1001$
$d_2 = ?$
G
D

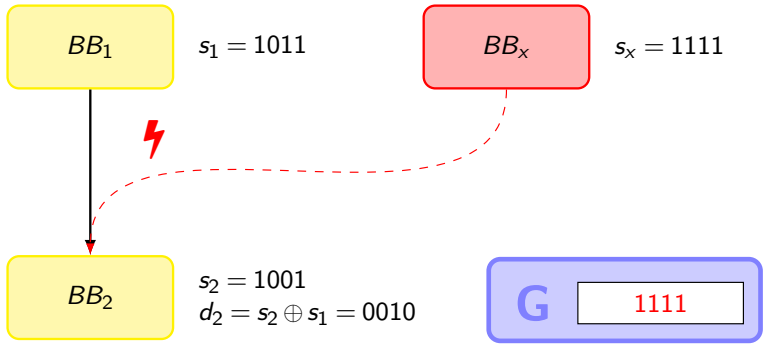

BB_1
$s_1 = 1011$
$D = 0000$
BB_3
$s_3 = 1000$
$D = s_1 \oplus s_3 = 0011$
$G \leftarrow G \oplus D$
$G \leftarrow G \oplus d_2$
BB_2
$s_2 = 1001$
$d_2 = s_1 \oplus s_2$
$= 0010$
G
D

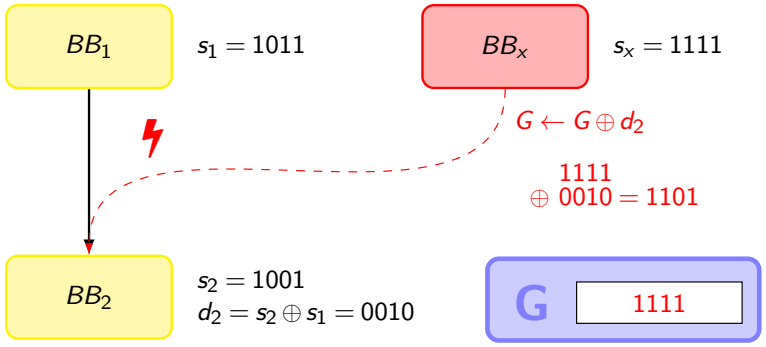

BB_1
$s_1 = 1011$
$D = 0000$
BB_3
$s_3 = 1000$
$D = s_1 \oplus s_3 = 0011$
$G \leftarrow G \oplus D$
$G \leftarrow G \oplus d_2$
BB_2
$s_2 = 1001$
$d_2 = s_1 \oplus s_2$
$= 0010$
G
1000
D
0011

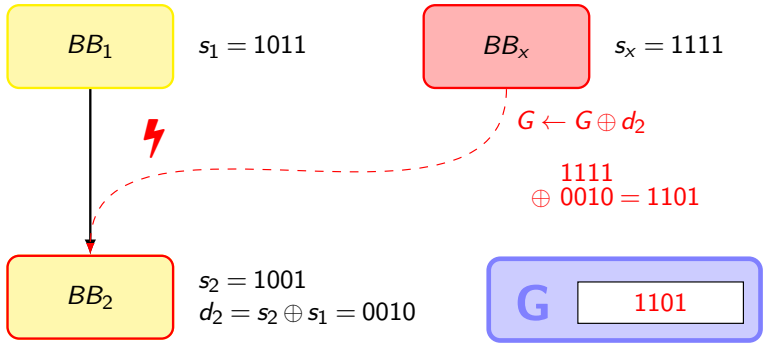

BB_1
$s_1 = 1011$
$D = 0000$
BB_3
$s_3 = 1000$
$D = s_1 \oplus s_3 = 0011$
$G \leftarrow G \oplus D$
$G \leftarrow G \oplus d_2$
BB_2
$s_2 = 1001$
$d_2 = s_1 \oplus s_2$
$= 0010$
G
1011
D
0011


BB_1
$s_1 = 1011$
$D = 0000$
BB_3
$s_3 = 1000$
$D = s_1 \oplus s_3 = 0011$
$G \leftarrow G \oplus D$
$G \leftarrow G \oplus d_2$
BB_2
$s_2 = 1001$
$d_2 = s_1 \oplus s_2$
$= 0010$
G
1001
D
0011


BB_1
$s_1 = 1011$
BB_x
$s_x = 1111$
BB_2
$s_2 = 1001$
$d_2 = s_2 \oplus s_1 = 0010$
G


BB_1
$s_1 = 1011$
BB_x
$s_x = 1111$
BB_2
$s_2 = 1001$
$d_2 = s_2 \oplus s_1 = 0010$
G
1111





Umgesetzte Verfahren

Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Umgesetzte Verfahren

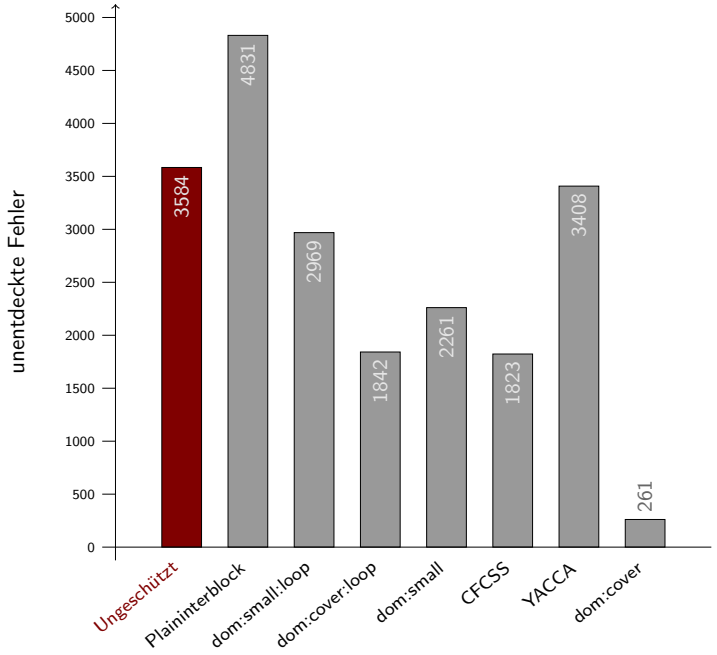
Basisblockebene:

- Plain Interblock Error Detection [1], [2]
- Enhanced Control-Flow Checking using Assertions (ECCA) [3]
- Control-Flow Checking by Software Signatures (CFCSS) [4]
- Yet Another Control-Flow Checking using Assertions (YACCA) [5], [6]

Dominanzregionsebene:

- Dominatorbasiertes Verfahren [7]
  - Vergabe-Strategien: „Kleinste“ und „Verteilung“
  - Schleifenoptimierung

Kontrollflussfehler



Unausgerichtete Sprünge

Ursprüngliche Instruktionsfolge

0x1014af	83	
0x1014b0	c2	add edx,0x1
0x1014b1	01	
0x1014b2	0f	
0x1014b3	af	
0x1014b4	4c	imul ecx,DWORD PTR
0x1014b5	ab	[ebx+ebp*4+0x8]
0x1014b6	08	
0x1014b7	01	
0x1014b8	ce	add esi,ecx
0x1014b9	0f	
0x1014ba	b7	movzx ecx,WORD PTR
0x1014bb	48	[eax+0xa]
0x1014bc	0a	
0x1014bd	83	
0x1014be	f1	xor ecx,0x7
0x1014bf	07	

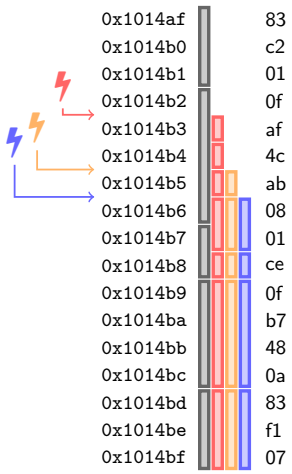
Unausgerichtete Sprünge

	Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
0x1014af 83		
0x1014b0 c2	add edx,0x1	
0x1014b1 01		
0x1014b2 0f		
0x1014b3 af		scas eax,DWORD PTR es:[edi]
0x1014b4 4c	imul ecx,DWORD PTR	dec esp
0x1014b5 ab	[ebx+ebp*4+0x8]	stos DWORD PTR es:[edi],eax
0x1014b6 08		or BYTE PTR [ecx],al
0x1014b7 01		
0x1014b8 ce	add esi,ecx	into
0x1014b9 0f		
0x1014ba b7	movzx ecx,WORD PTR	movzx ecx,WORD PTR
0x1014bb 48	[eax+0xa]	[eax+0xa]
0x1014bc 0a		
0x1014bd 83		
0x1014be f1	xor ecx,0x7	xor ecx,0x7
0x1014bf 07		

Unausgerichtete Sprünge

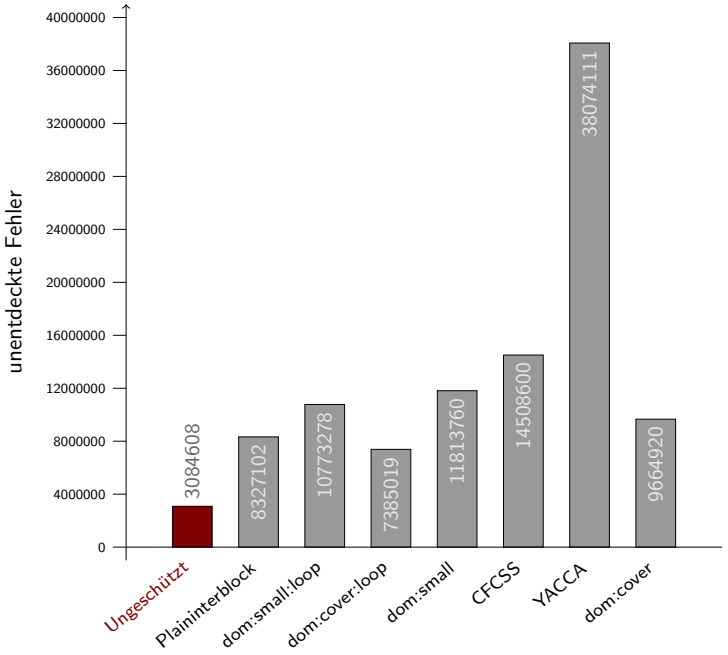
	Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
0x1014af 83		
0x1014b0 c2	add edx,0x1	
0x1014b1 01		
0x1014b2 0f		
0x1014b3 af		scas eax,DWORD PTR es:[edi]
0x1014b4 4c	imul ecx,DWORD PTR	dec esp
0x1014b5 ab	[ebx+ebp*4+0x8]	stos DWORD PTR es:[edi],eax
0x1014b6 08		or BYTE PTR [ecx],al
0x1014b7 01		
0x1014b8 ce	add esi,ecx	into
0x1014b9 0f		
0x1014ba b7	movzx ecx,WORD PTR	movzx ecx,WORD PTR
0x1014bb 48	[eax+0xa]	[eax+0xa]
0x1014bc 0a		
0x1014bd 83		
0x1014be f1	xor ecx,0x7	xor ecx,0x7
0x1014bf 07		

Unausgerichtete Sprünge

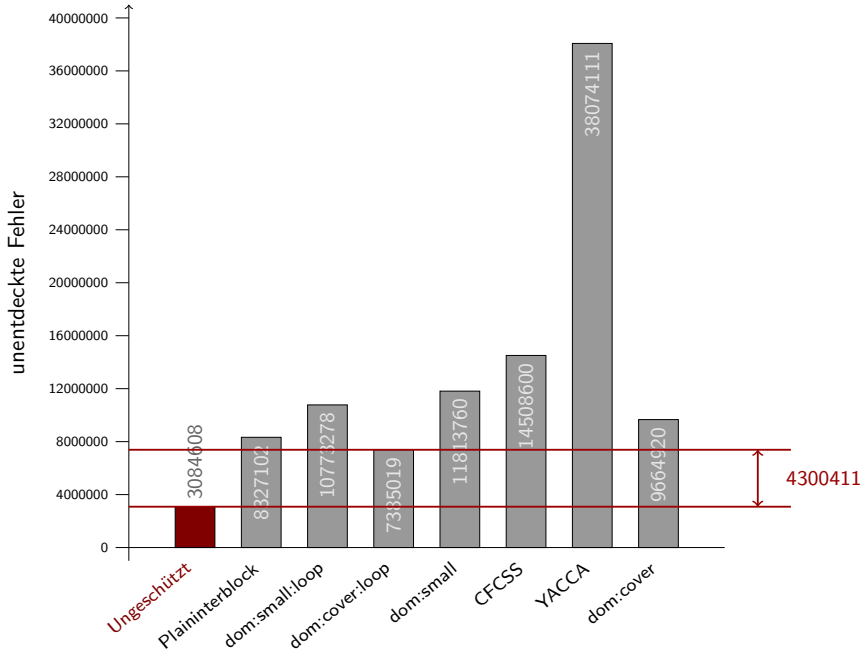


		Ursprüngliche Instruktionsfolge	Unausgerichtete Instruktionsfolge
0x1014af	83		
0x1014b0	c2	add edx,0x1	
0x1014b1	01		
0x1014b2	0f		
0x1014b3	af	imul ecx,DWORD PTR	scas eax,DWORD PTR es:[edi]
0x1014b4	4c	[ebx+ebp*4+0x8]	dec esp
0x1014b5	ab		stos DWORD PTR es:[edi],eax
0x1014b6	08		or BYTE PTR [ecx],al
0x1014b7	01	add esi,ecx	
0x1014b8	ce		into
0x1014b9	0f		
0x1014ba	b7	movzx ecx,WORD PTR	movzx ecx,WORD PTR
0x1014bb	48	[eax+0xa]	[eax+0xa]
0x1014bc	0a		
0x1014bd	83		
0x1014be	f1	xor ecx,0x7	xor ecx,0x7
0x1014bf	07		

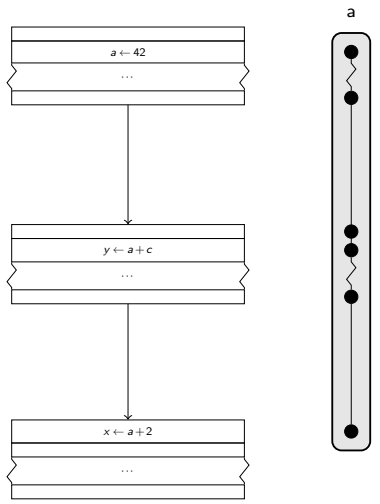
Allgemeine Einbitfehler



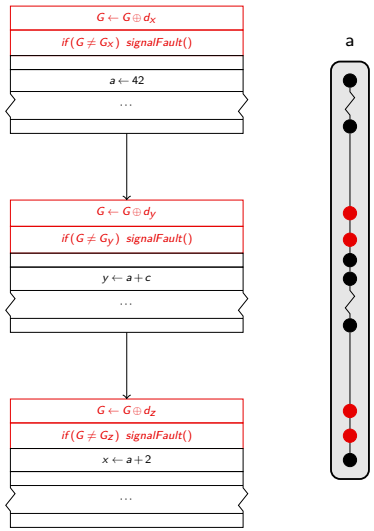
Allgemeine Einbitfehler



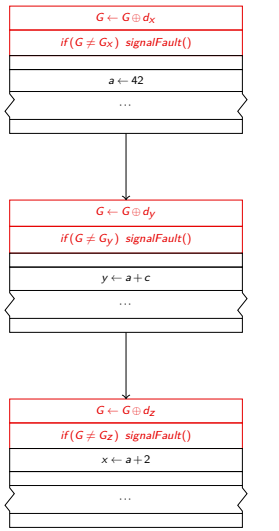
Datenfehler

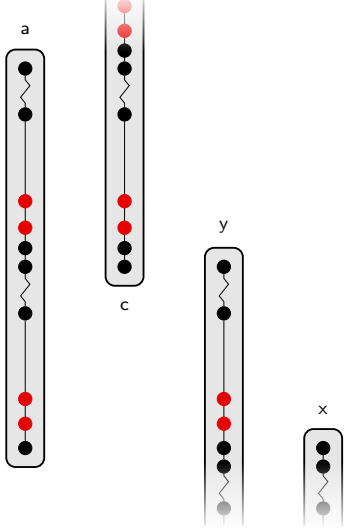


Datenfehler

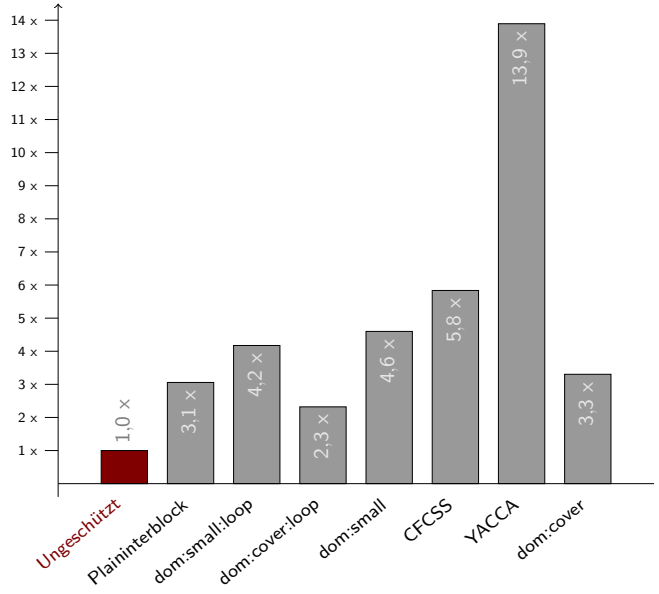


Datenfehler

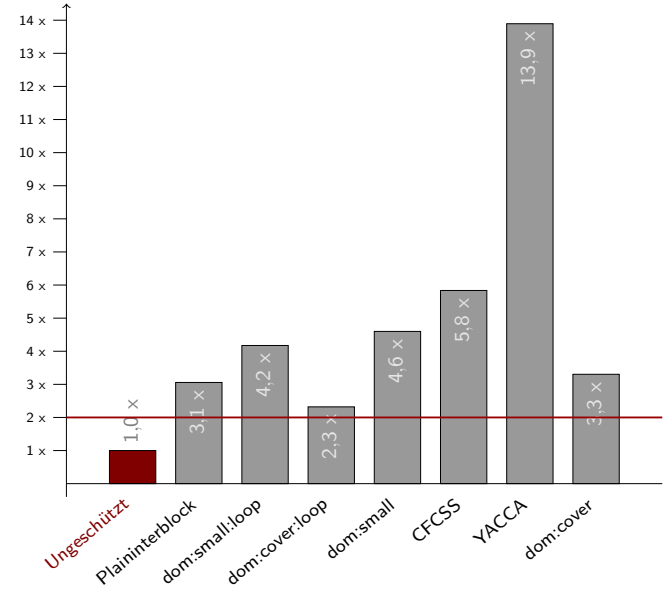




Laufzeit



Laufzeit

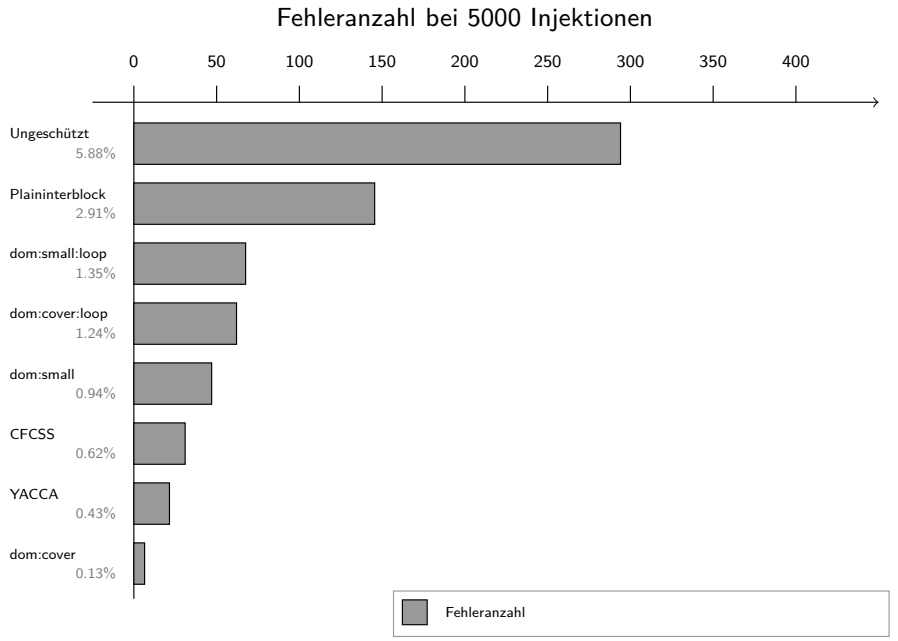


Literaturvergleich

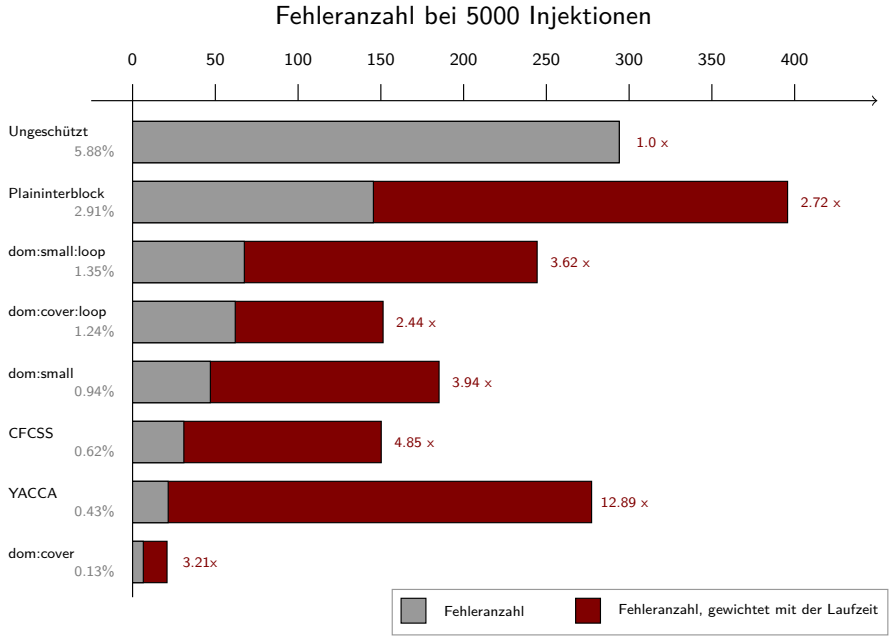
- IA32 / TC1796
- g++, v4.9.1, v4.6.3
- Optionen: `-O2`
- Signatur als `volatile` Variable
- Speicherzugriff TC1796: 2 Zyklen
- ...

- R4400 (MIPS), Intel 8051, TIMA T225, ...
- ?
- ?
- ?
- ?
- ...

Fehlerrate und -abdeckung



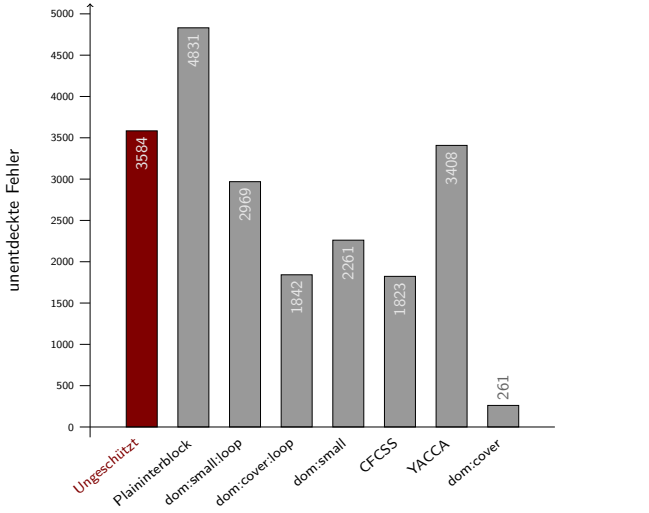
Fehlerrate und -abdeckung



Fazit

Fazit

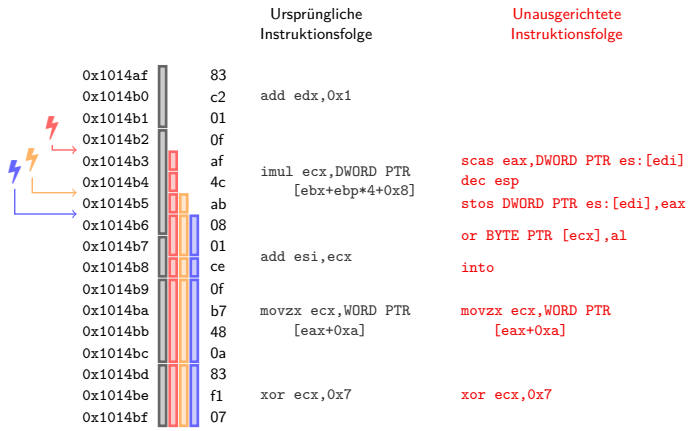
Kontrollflussfehler



Fazit

Kontrollflussfehler

Unausgerichtete Sprünge

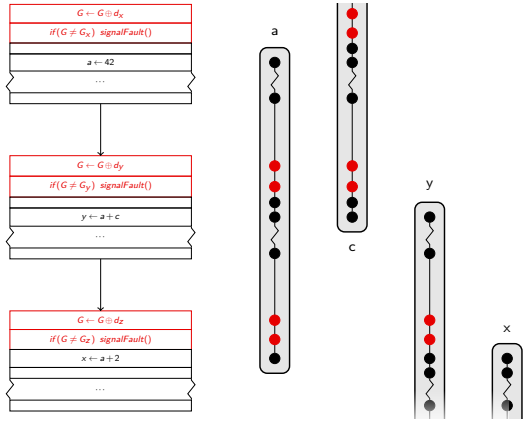


Fazit

Kontrollflussfehler

Unausgerichtete Sprünge

Datenfehler



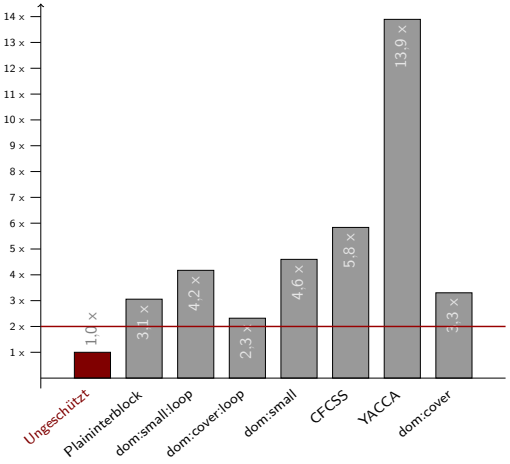
Fazit

Kontrollflussfehler

Unausgerichtete Sprünge

Datenfehler

Laufzeit



Referenzen

[1] M. Rebaudengo, M. S. Reorda, M. Torchiano und M. Violante, „Soft-error detection through software fault-tolerance techniques," in **Defect and Fault Tolerance in VLSI Systems, 1999. DFT'99. International Symposium on**. IEEE, 1999, S. 210–218.

[2] P. Cheynet, B. Nicolescu, R. Velazco, M. Rebaudengo, M. Sonza Reorda und M. Violante, „Experimentally evaluating an automatic approach for generating safety-critical software with respect to transient errors," **IEEE Transactions on Nuclear Science**, Vol. 47, Nr. 6, S. 2231–2236, 2000.

[3] Z. Alkhalifa, V. S. Nair, N. Krishnamurthy und J. A. Abraham, „Design and evaluation of system-level checks for on-line control flow error detection," **Parallel and Distributed Systems, IEEE Transactions on**, Vol. 10, Nr. 6, S. 627–641, 1999.

[4] N. Oh, P. Shirvani und E. McCluskey, „Control-flow checking by software signatures," **Reliability, IEEE Transactions on**, Vol. 51, Nr. 1, S. 111–122, Mar. 2002.

[5] O. Goloubeva, M. Rebaudengo, M. S. Reorda und M. Violante, „Soft-error detection using control flow assertions," in **Defect and Fault Tolerance in VLSI Systems, 2003. Proceedings. 18th IEEE International Symposium on**. IEEE, 2003, S. 581–588.

[6] O. Goloubeva, M. Rebaudengo, M. Reorda und M. Violante, **Software-implemented hardware fault tolerance**. Springer, 2006.

[7] C. Dietrich, „Global Optimization of Non Funcitonal Properties in OSEK Real-Time Systems by Static Cross-Kernel Flow Analyses," Masterarbeit, Universität Erlangen, Deutschland, Sep. 2014.