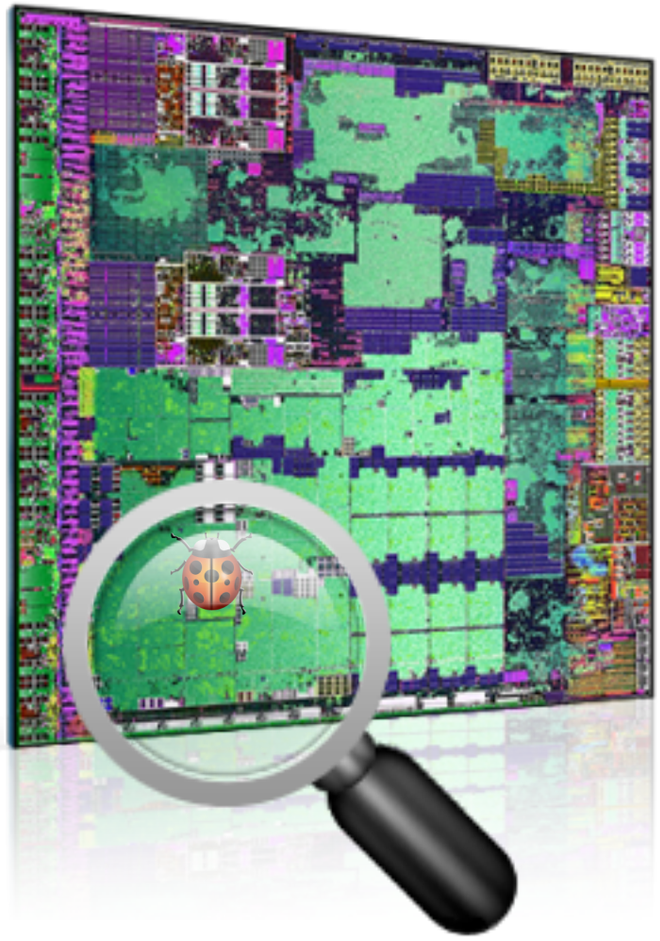
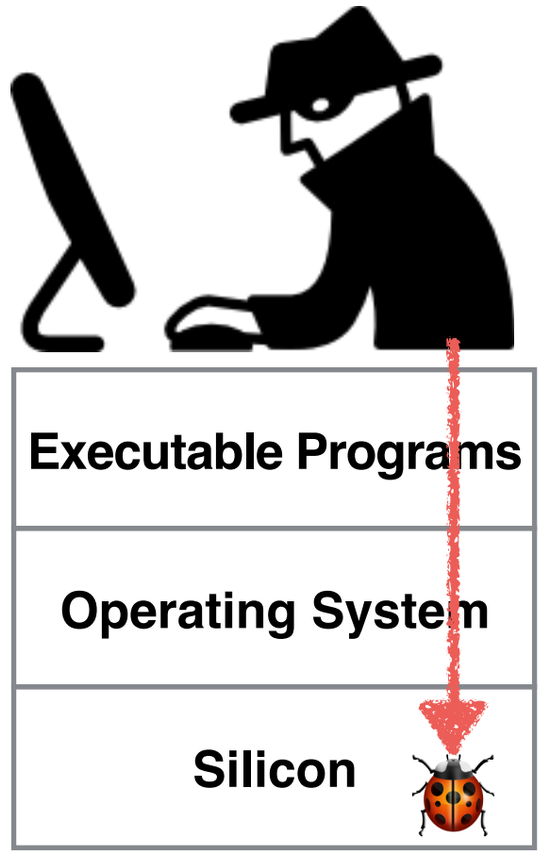
Identifying Security Critical Properties for the Dynamic Verification of a Processor

Rui Zhang, Natalie Stanley, Christopher Griggs,
Andrew Chi, Cynthia Sturton

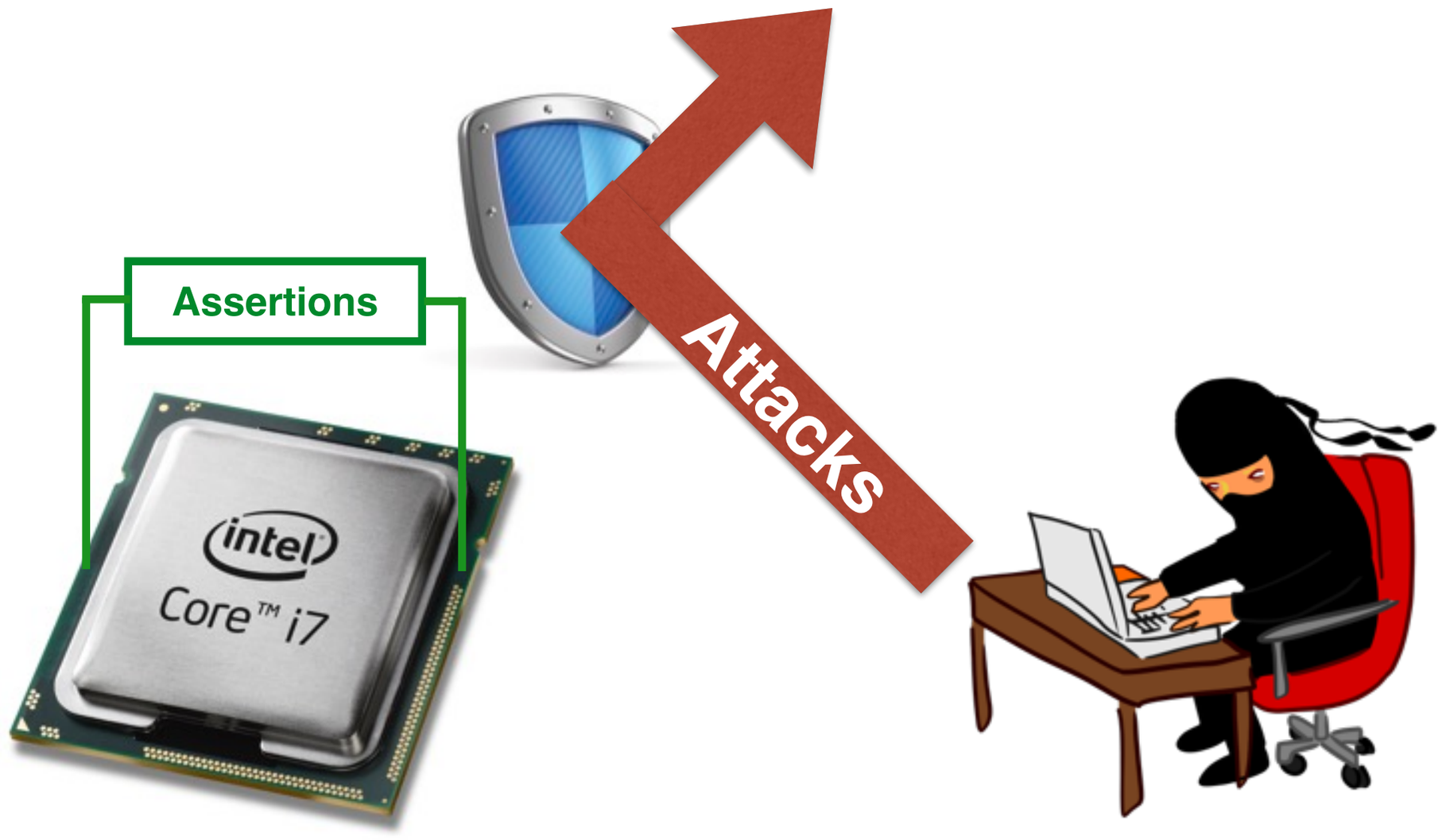
04-12-2017, ASLOS
XI'AN CHINA

Processor Bugs can Create Security Vulnerabilities





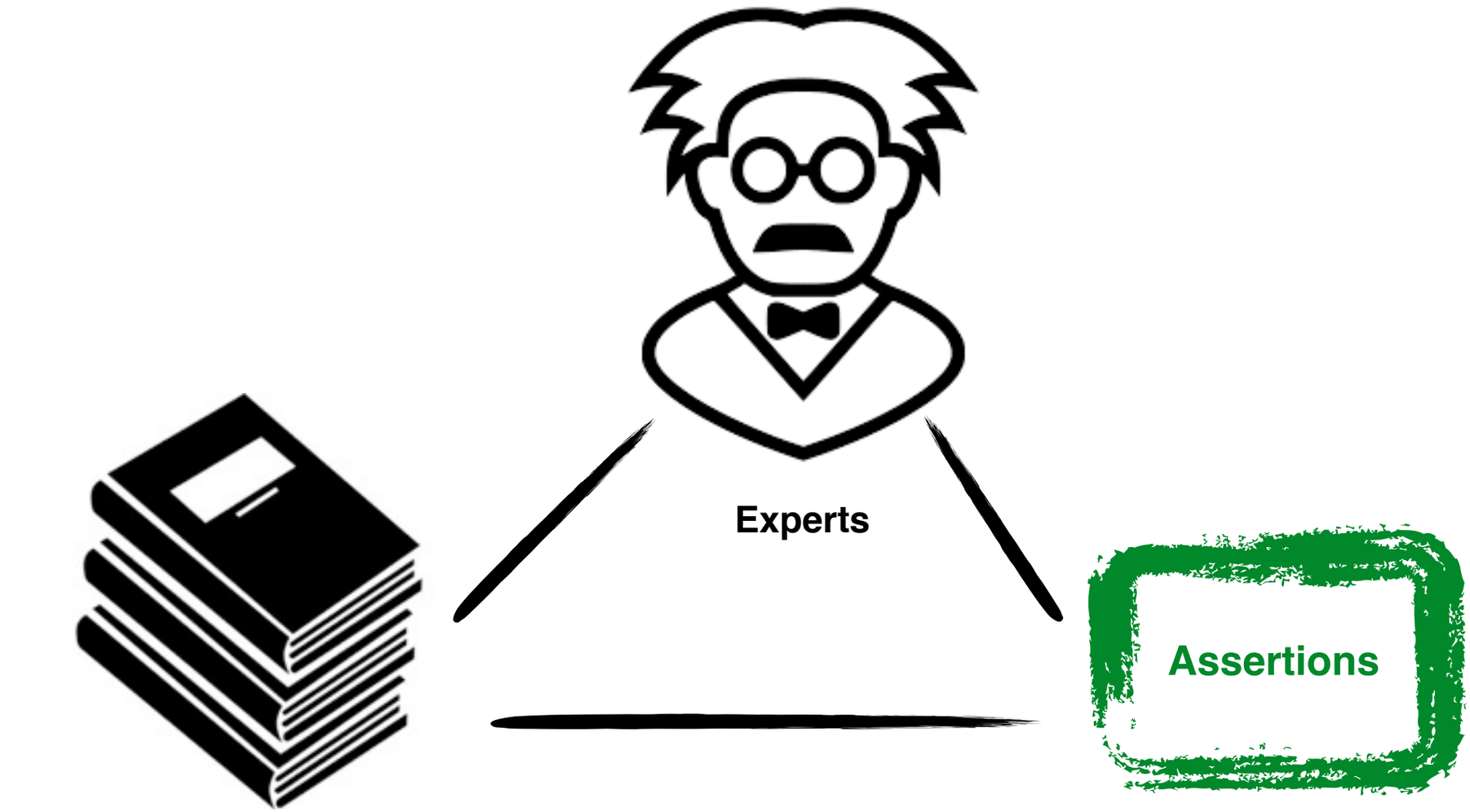
Dynamic Verification for Security



Research Question

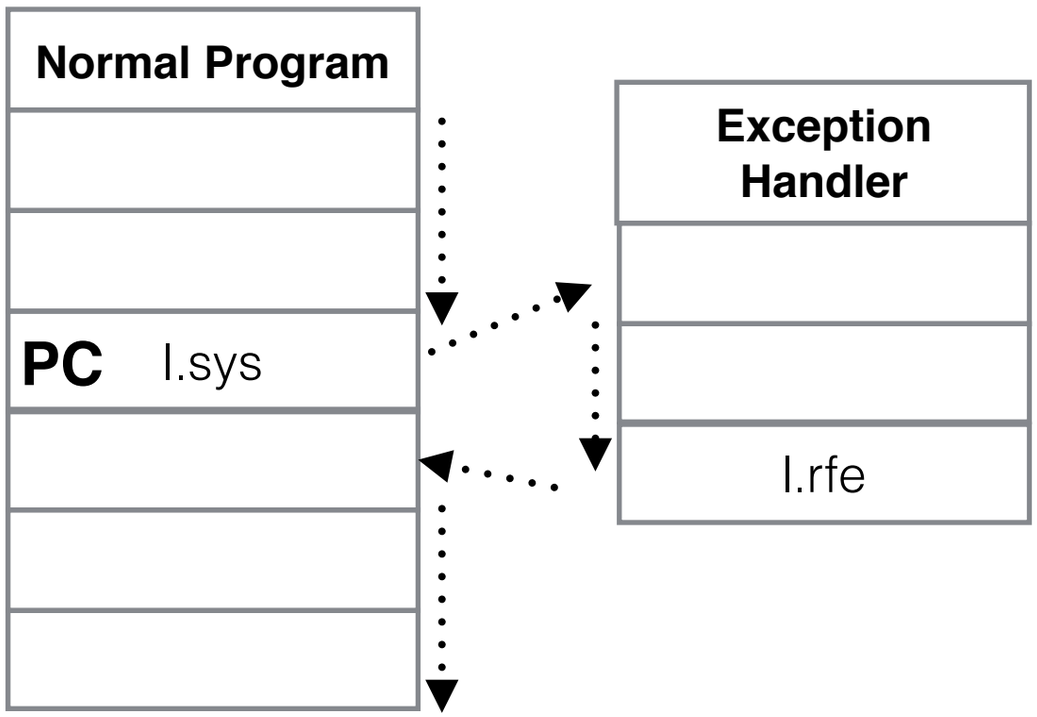
How to find the security properties we should protect for a processor?

The State of the Art:
Human Expertise and Judgment



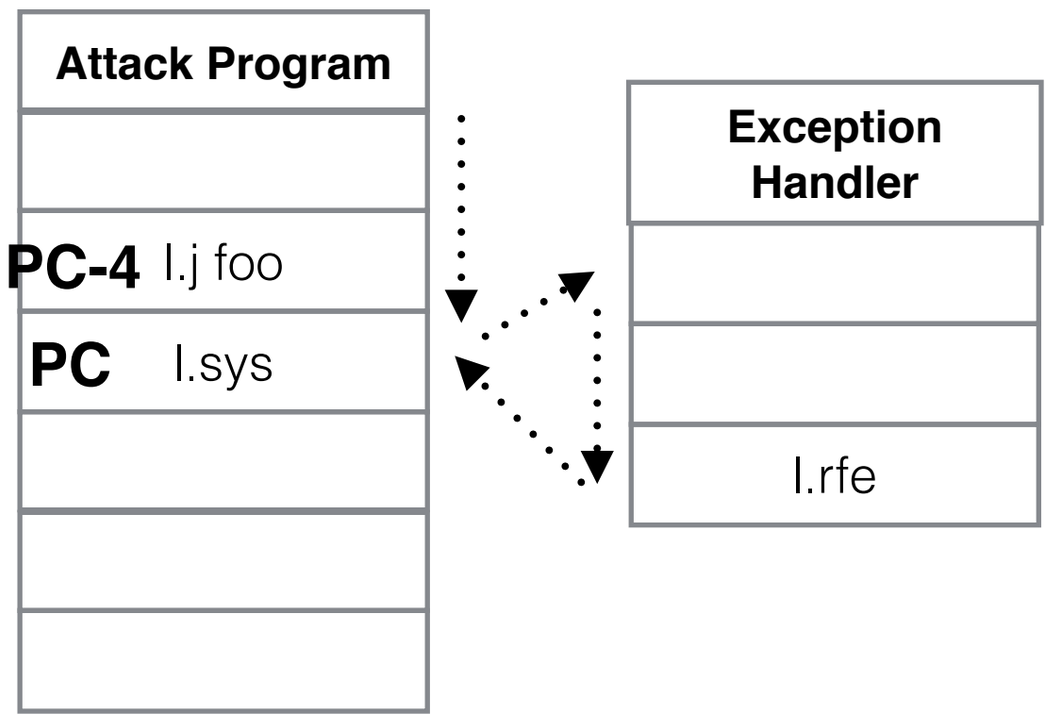
Vulnerability Example: DoS Attack

Normal



EPCR = PC + 4

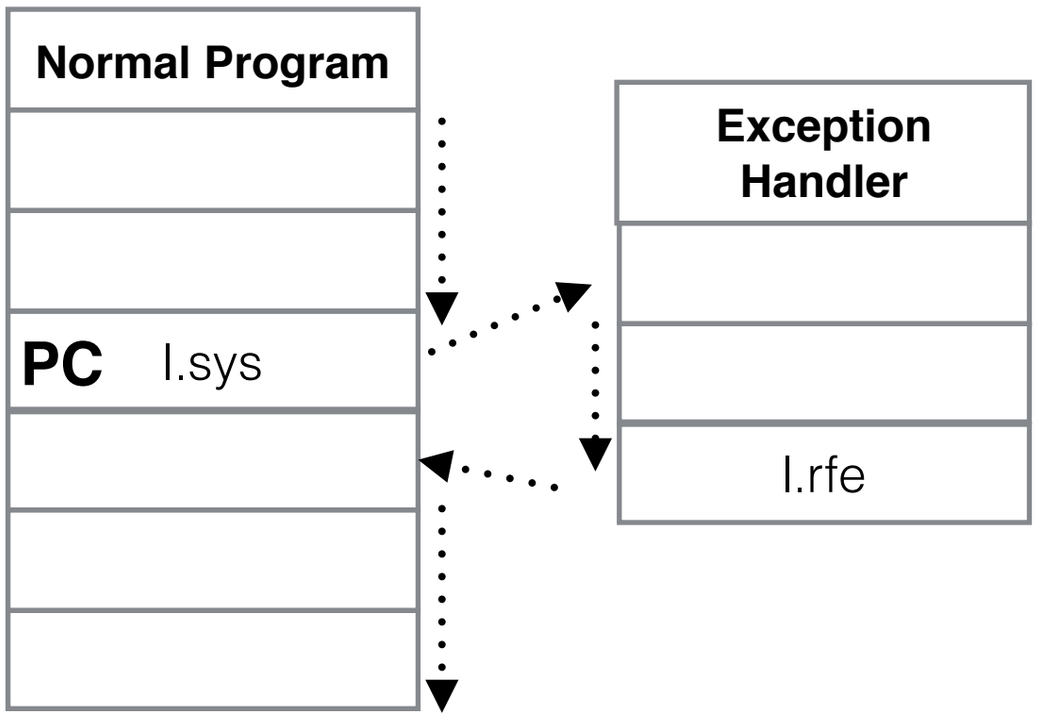
Syscall in Delay Slot



EPCR = PC

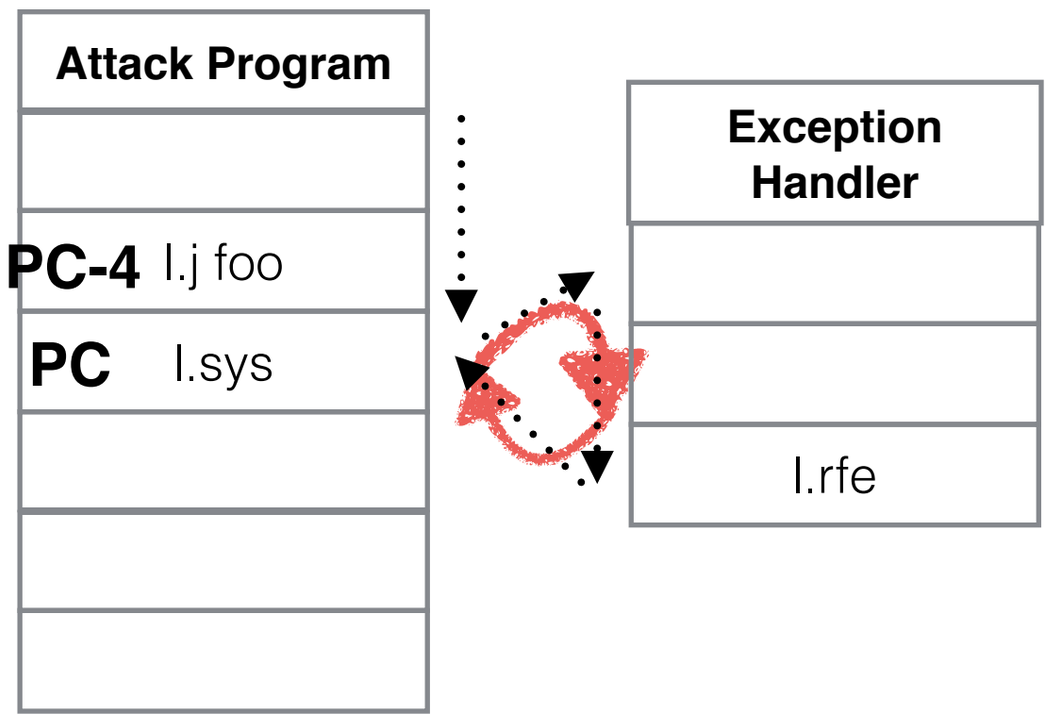
Vulnerability Example: DoS Attack

Normal



EPCR = PC + 4

Syscall in Delay Slot



EPCR = PC

Observation

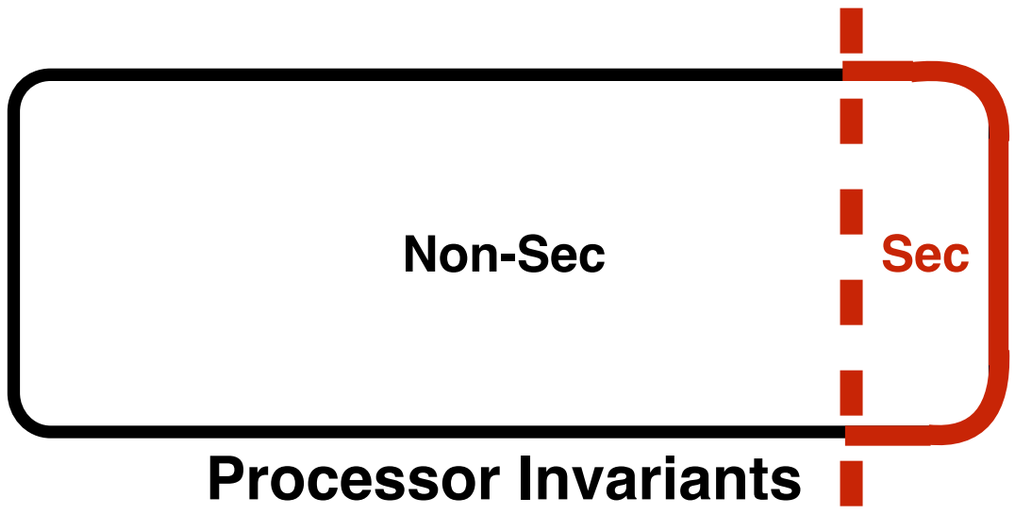
Observation:

- Security-critical bugs are vulnerabilities precisely because they violate some underlying security property

Goal:

- Identify security properties for a processor

Our Approach



Known Security Bugs → Security Properties

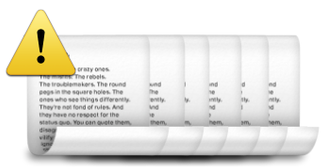
Machine Learning → Additional Security Properties

Our Approach

Processor Design

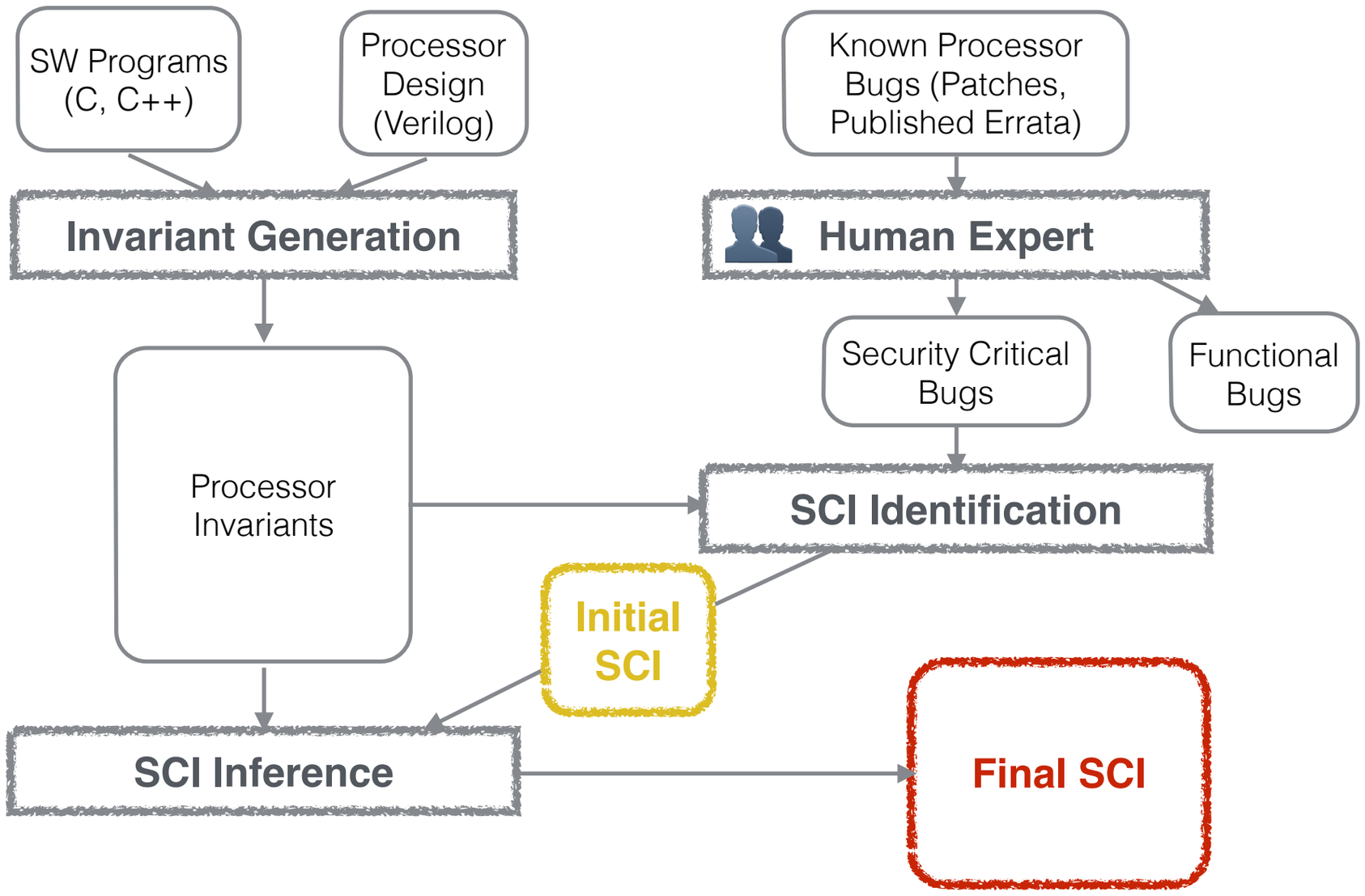
Security-Critical Bugs

1. Invariant Generation
2. Recognition of Security-Critical Invariants
3. Machine Learning



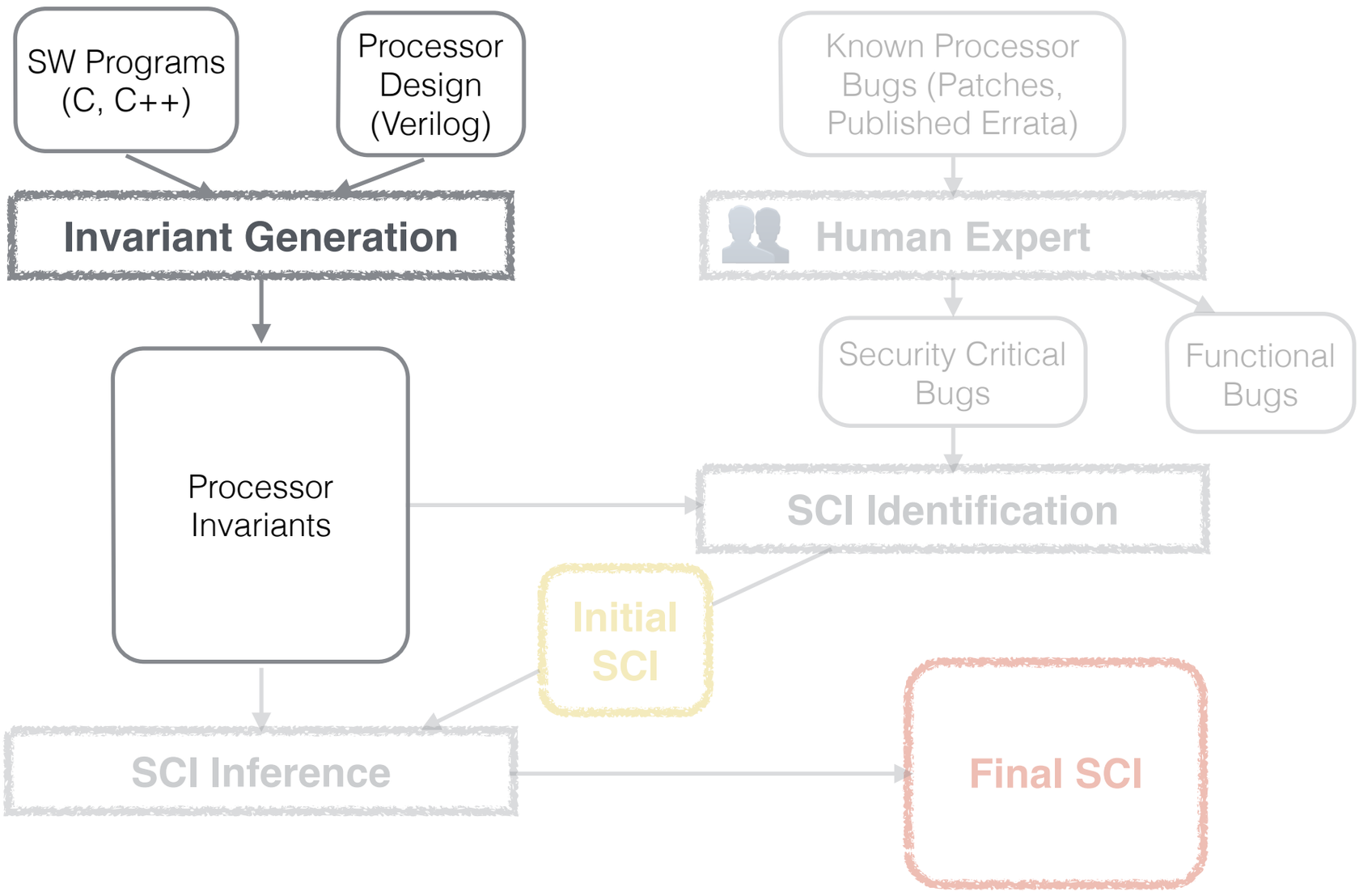
Final Security Properties

SCIFinder Tool Chain



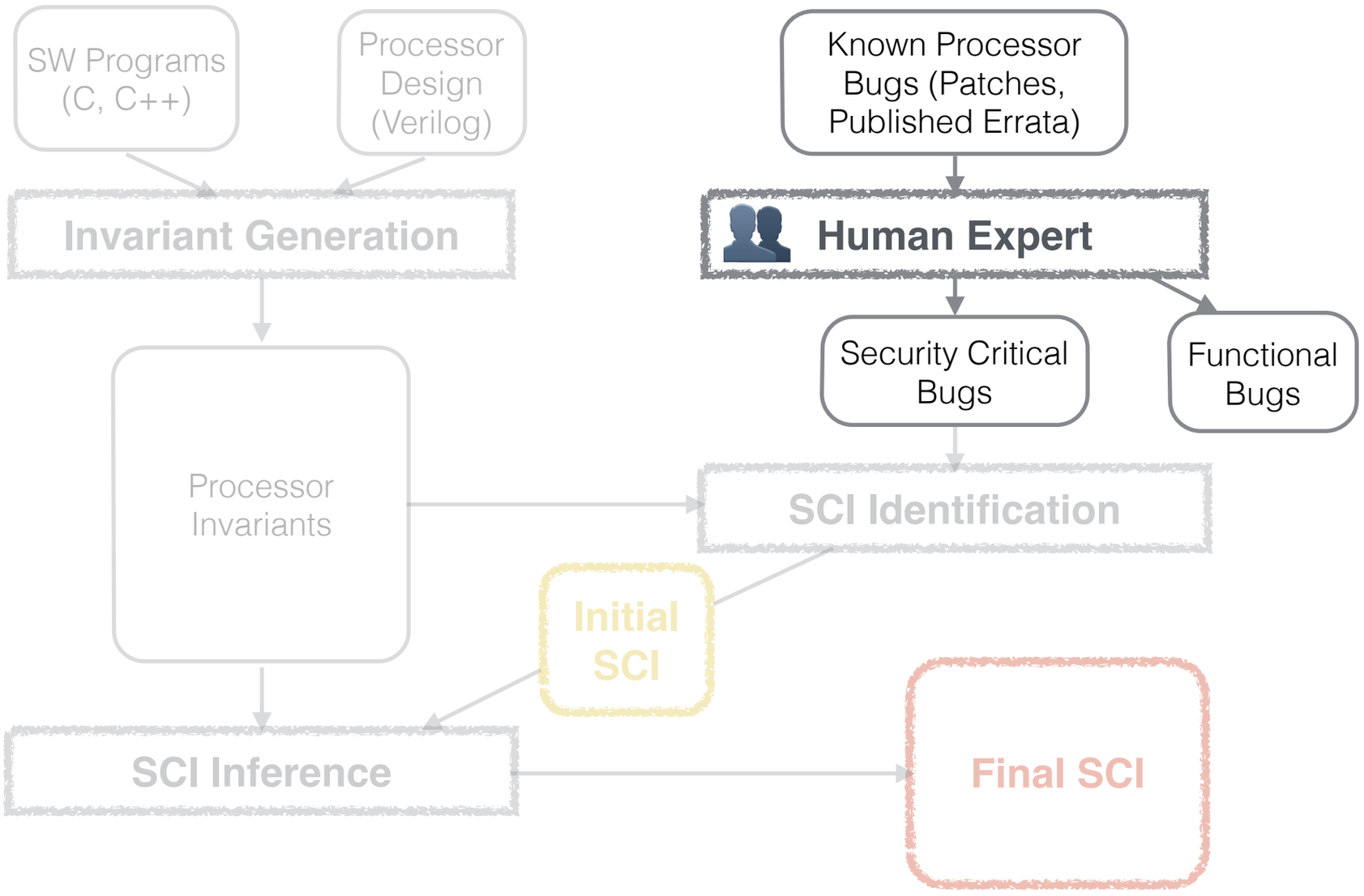
Workflow of SCIFinder

SCIFinder Tool Chain



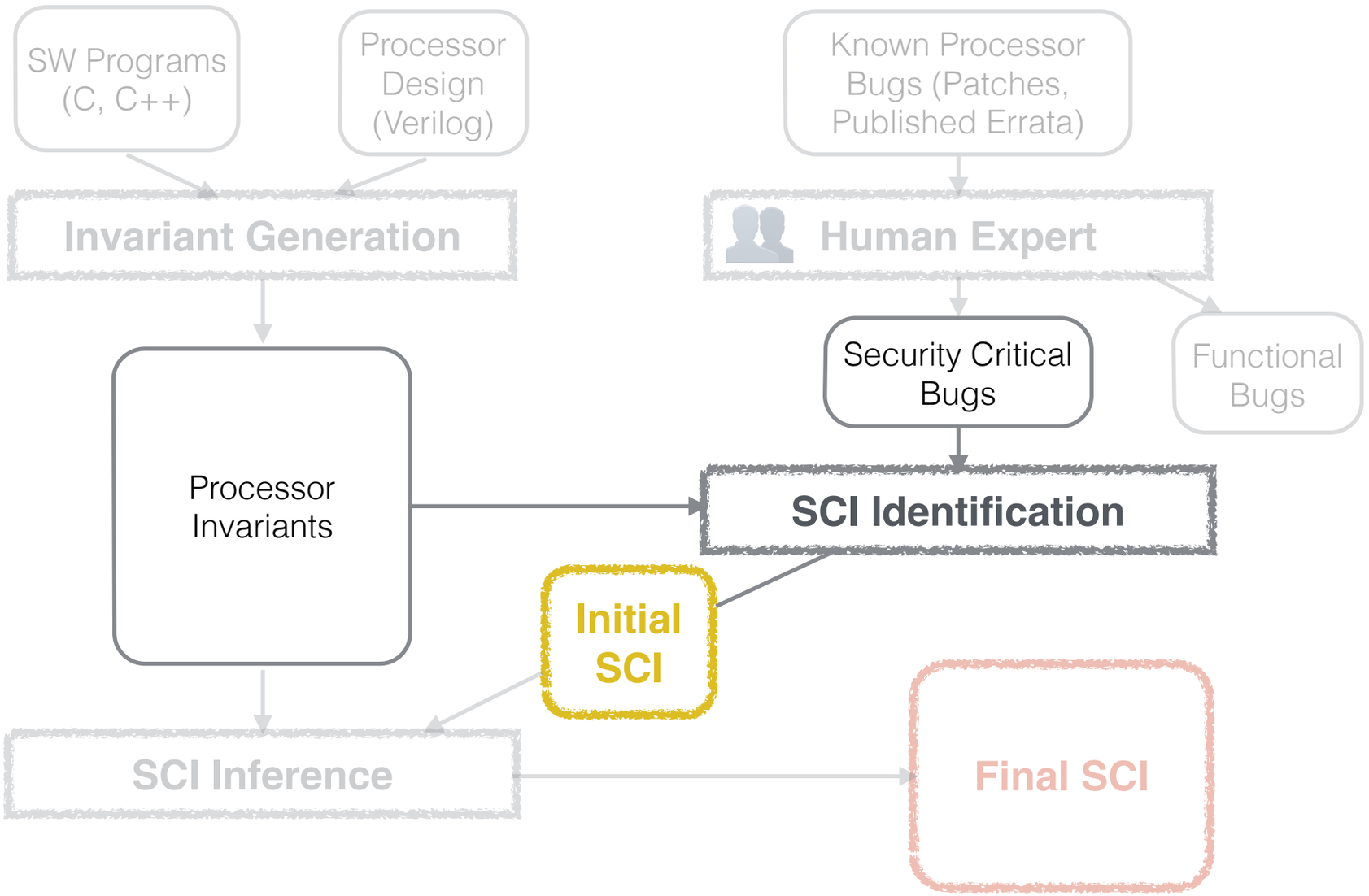
Workflow of SCIFinder

SCIFinder Tool Chain



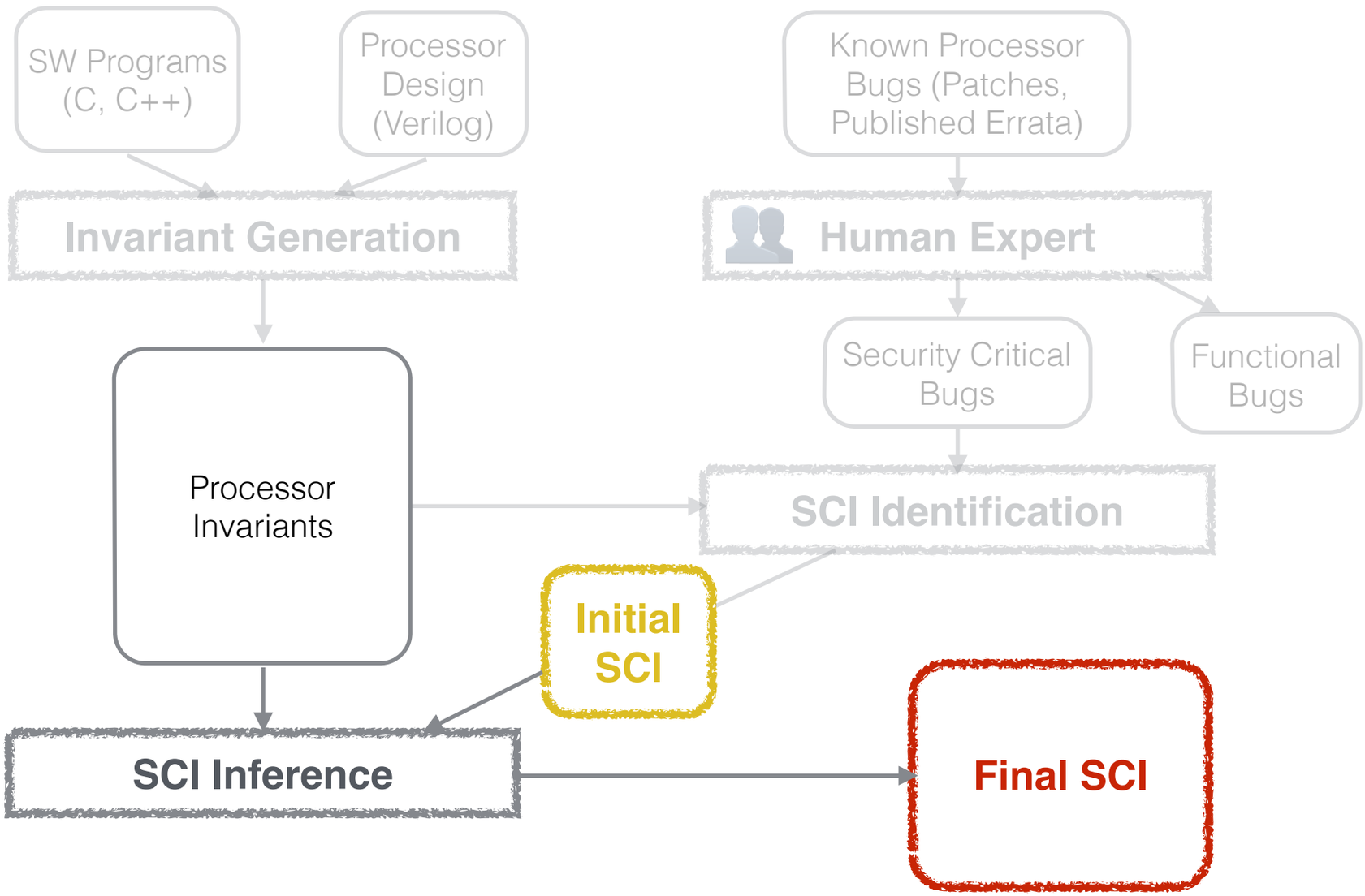
Workflow of SCIFinder

SCIFinder Tool Chain



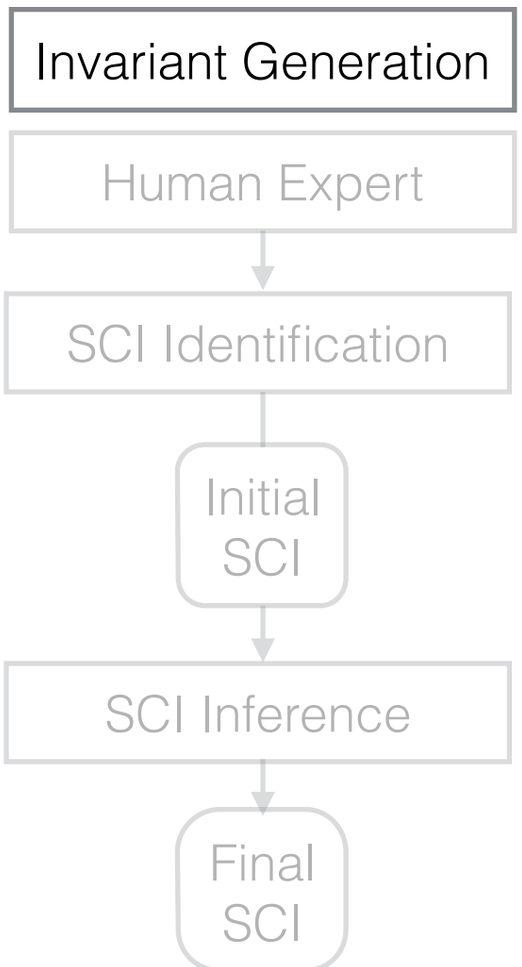
Workflow of SCIFinder

SCIFinder Tool Chain



Workflow of SCIFinder

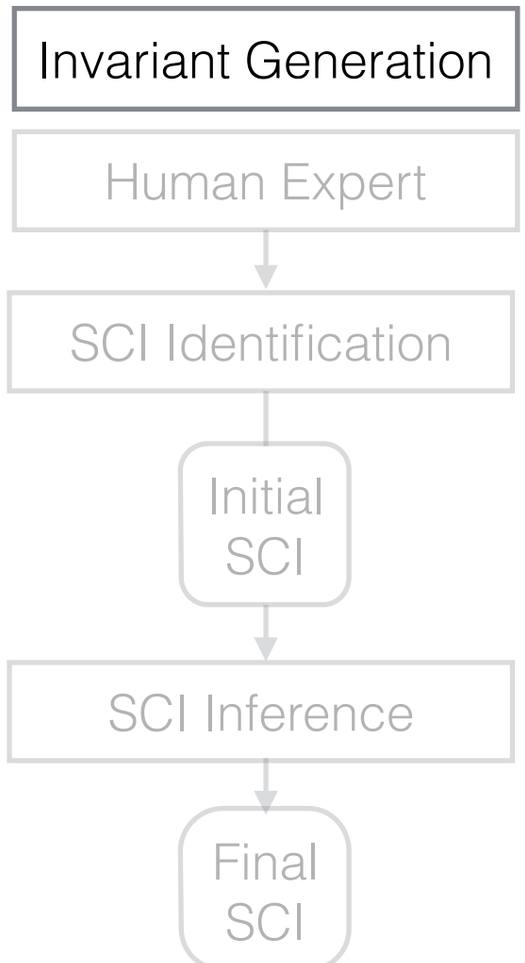
Invariant Generation



Adaptation:

- New Daikon Instrumenter: adapt Daikon to processor execution traces
- ISA-level variables: registers and signals visible to software
- Configurable: patterns unknown to Daikon
- Carefully handle processor optimizations

Invariant Generation



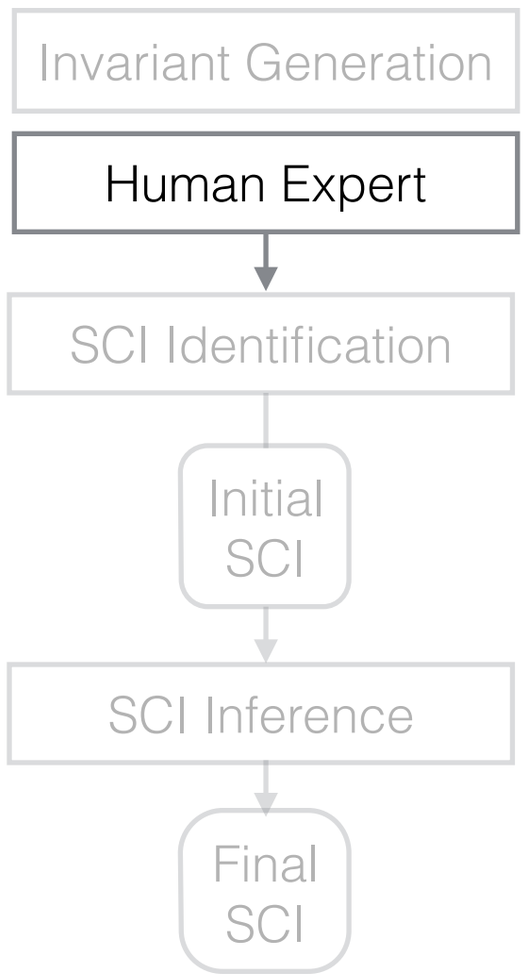
Invariant Format:

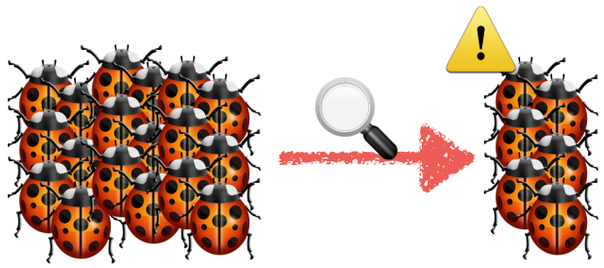
- $I \doteq risingEdge(INSN) \rightarrow EXPR$

Invariant Example:

- $I \doteq risingEdge(\text{I.rfe}) \rightarrow \text{SR} = orig(\text{ESR0})$

Manually Classifying Bugs





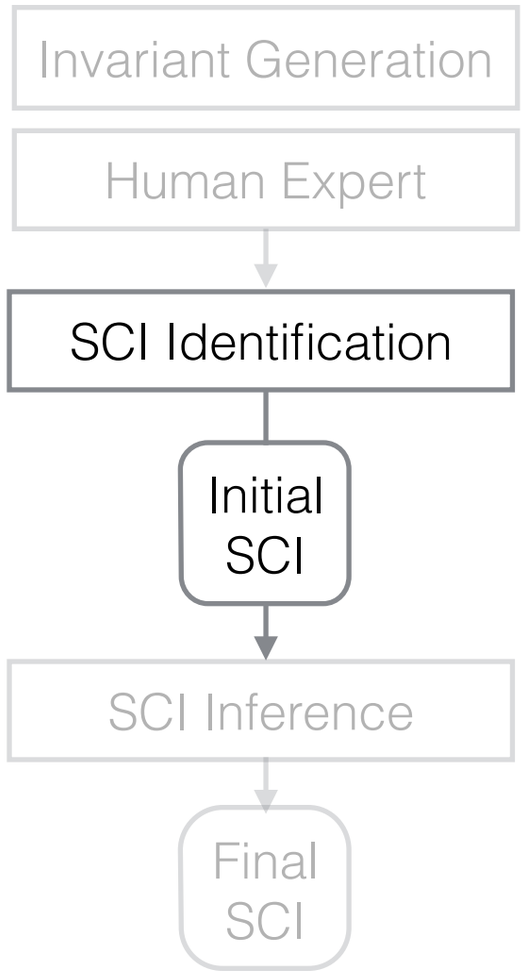
Collected Bugs (185)

Security-Critical Bugs (25, reproduced 17)

Sources:

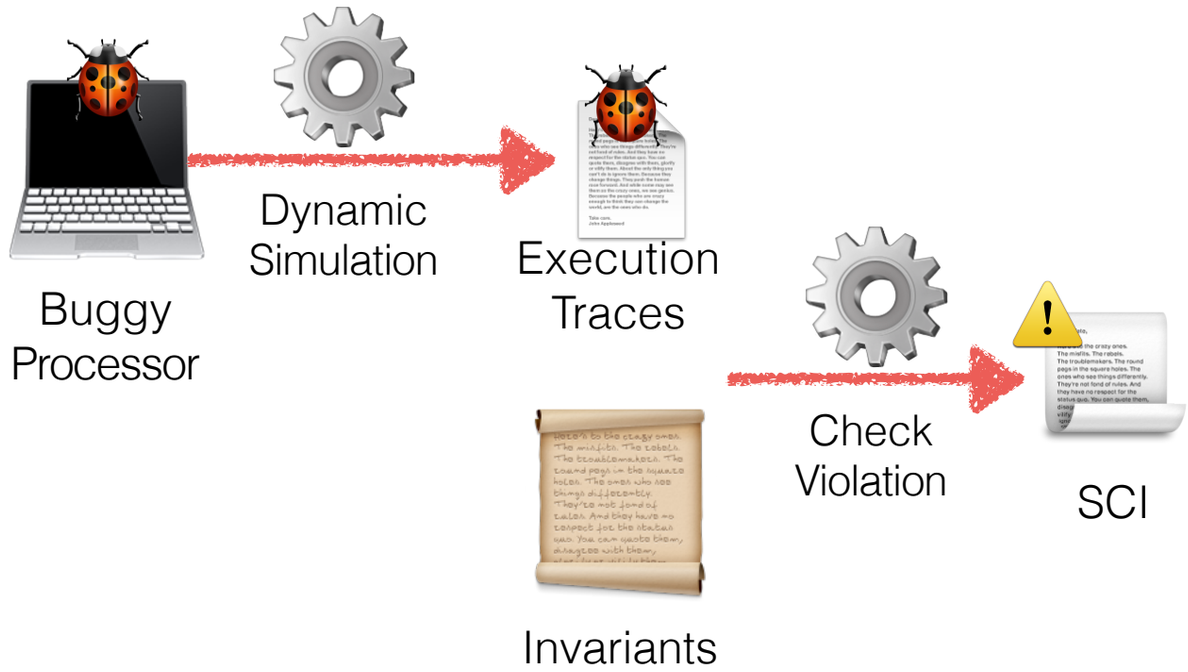
- Processors' bug tracker or Bugzilla sites
- Developers' mail archives
- Commits to the source repository
- Comments in the source code
- Published list of errata

Security-Critical Invariant Identification

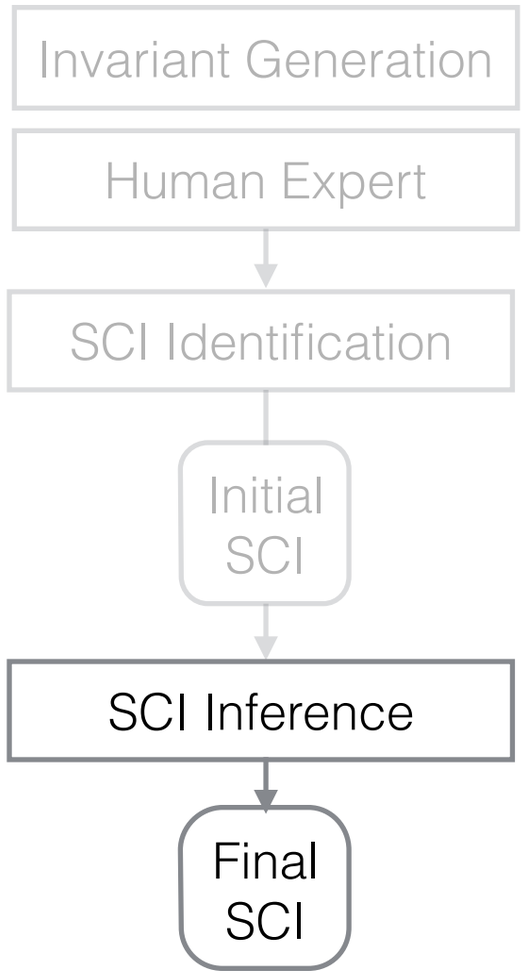


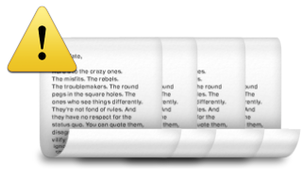
Key Observation:

- Security-critical bugs are vulnerabilities precisely because they violate some underlying security property

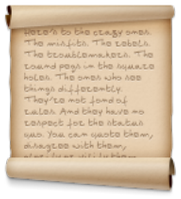


Security-Critical Invariant Inference





Initial SCI



Invariants

Logistic Regression with Elastic Net Penalty

Final SCI

The constructed model can:

- Predict whether a given invariant is likely an SCI
- Help hardware designers understand which features are critical to security

Evaluation Methodology

Gather Real-world Security Vulnerabilities:

- Reproduce 17 security-critical bugs from open source processors
- Write attack programs that exploit the vulnerabilities

Generate Security-Critical Properties:

- Run normal programs and attack programs on affected processors
- Record execution traces
- Use SCIFinder to generate SCI

Compare with Prior Work:

- Collect 22 manually written security-critical properties from prior work
- Compare SCI generated by SCIFinder with manually written ones
- Add assertions to detect unknown bugs

Main Results

Security Properties

22

Manually Intensive

SCI Finder

19 + 3 ← **New**

Semi-automatic

Manual Efforts: classifying bugs, validating the reported SCI

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	SPR equals GPR in register move instruction	b12
p3	Updates to exception registers make sense	b4 b9 b15
p4	Destination matches the target	✓
p5	Memory value in equals register value out	b14
p6	Register value in equals memory value out	b16 b17
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	Jumps update the LR correctly	b13
p12	Instruction is in a valid format	b11
p13	Continuous Control Flow	b5
p14	Exception return updates state correctly	b1 b5
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	Interrupt implies handled	b8
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	Exception handling implies exception mechanism activated	b8
p22	Unspecified custom instructions are not allowed	✗
p23	Exception handler accessed only during exception, in supervisor mode, or on reset	b8
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

Properties from SPECS

[H.S.K. ASPLOS 2015]

Properties from Security-Checker

[B.H.I. HOST 2011]

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	SPR equals GPR in register move instruction	b12
p3	Updates to exception registers make sense	b4 b9 b15
p4	Destination matches the target	✓
p5	Memory value in equals register value out	b14
p6	Register value in equals memory value out	b16 b17
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	Jumps update the LR correctly	b13
p12	Instruction is in a valid format	b11
p13	Continuous Control Flow	b5
p14	Exception return updates state correctly	b1 b5
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	Interrupt implies handled	b8
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	Exception handling implies exception mechanism activated	b8
p22	Unspecified custom instructions are not allowed	✗
p23	Exception handler accessed only during exception, in supervisor mode, or on reset	b8
p24	Page fault generated if MMU detects an access control violation	--
p25	**UART output changes on a write command from CPU**	--
p26	**Only transmit command or initialization change Ethernet data output**	--
p27	**Debug Unit's value and control registers only accessible from supervisor mode**	--

Properties Outside of Processor Core

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	SPR equals GPR in register move instruction	b12
p3	Updates to exception registers make sense	b4 b9 b15
p4	Destination matches the target	✓
p5	Memory value in equals register value out	b14
p6	Register value in equals memory value out	b16 b17
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	Jumps update the LR correctly	b13
p12	Instruction is in a valid format	b11
p13	Continuous Control Flow	b5
p14	Exception return updates state correctly	b1 b5
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	Interrupt implies handled	b8
p18	**Instruction unchanged in pipeline**	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	Exception handling implies exception mechanism activated	b8
p22	Unspecified custom instructions are not allowed	✗
p23	Exception handler accessed only during exception, in supervisor mode, or on reset	b8
p24	**Page fault generated if MMU detects an access control violation**	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

Properties Needing Micro-architectural States

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	**SPR equals GPR in register move instruction**	**b12**
p3	**Updates to exception registers make sense**	**b4 b9 b15**
p4	Destination matches the target	✓
p5	**Memory value in equals register value out**	**b14**
p6	**Register value in equals memory value out**	**b16 b17**
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	**Jumps update the LR correctly**	**b13**
p12	**Instruction is in a valid format**	**b11**
p13	**Continuous Control Flow**	**b5**
p14	**Exception return updates state correctly**	**b1 b5**
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	**Interrupt implies handled**	**b8**
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	**Exception handling implies exception mechanism activated**	**b8**
p22	Unspecified custom instructions are not allowed	✗
p23	**Exception handler accessed only during exception, in supervisor mode, or on**	**b8**
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

Properties Found in the Identification Step

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	**SPR equals GPR in register move instruction**	**b12**
p3	**Updates to exception registers make sense**	**b4 b9 b15**
p4	Destination matches the target	✓
p5	**Memory value in equals register value out**	**b14**
p6	**Register value in equals memory value out**	**b16 b17**
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	**Jumps update the LR correctly**	**b13**
p12	**Instruction is in a valid format**	**b11**
p13	**Continuous Control Flow**	**b5**
p14	**Exception return updates state correctly**	**b1 b5**
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	**Interrupt implies handled**	**b8**
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	**Exception handling implies exception mechanism activated**	**b8**
p22	Unspecified custom instructions are not allowed	✗
p23	**Exception handler accessed only during exception, in supervisor mode, or on**	**b8**
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

One property can be identified from different bugs

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	**SPR equals GPR in register move instruction**	**b12**
p3	**Updates to exception registers make sense**	**b4 b9 b15**
p4	Destination matches the target	✓
p5	**Memory value in equals register value out**	**b14**
p6	**Register value in equals memory value out**	**b16 b17**
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	**Jumps update the LR correctly**	**b13**
p12	**Instruction is in a valid format**	**b11**
p13	**Continuous Control Flow**	**b5**
p14	**Exception return updates state correctly**	**b1 b5**
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	**Interrupt implies handled**	**b8**
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	**Exception handling implies exception mechanism activated**	**b8**
p22	Unspecified custom instructions are not allowed	✗
p23	**Exception handler accessed only during exception, in supervisor mode, or on**	**b8**
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

One property can be identified from different bugs

Different properties can be identified from the same bug

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	**SPR equals GPR in register move instruction**	**b12**
p3	**Updates to exception registers make sense**	**b4 b9 b15**
p4	Destination matches the target	✓
p5	**Memory value in equals register value out**	**b14**
p6	**Register value in equals memory value out**	**b16 b17**
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	Jumps update the PC correctly	✗
p11	**Jumps update the LR correctly**	**b13**
p12	**Instruction is in a valid format**	**b11**
p13	**Continuous Control Flow**	**b5**
p14	**Exception return updates state correctly**	**b1 b5**
p15	Register change implies that it is the instruction target	✓
p16	SR is not written to a GPR in user mode	✗
p17	**Interrupt implies handled**	**b8**
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	**Exception handling implies exception mechanism activated**	**b8**
p22	Unspecified custom instructions are not allowed	✗
p23	**Exception handler accessed only during exception, in supervisor mode, or on**	**b8**
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

One property can be identified from different bugs

Different properties can be identified from the same bug

A single SCI can concisely represent multiple manually written properties

Results: Comparison to State-of-the-Art

No.	Security Property Description	Found?
p1	**Execution privilege matches page privilege**	✓
p2	SPR equals GPR in register move instruction	b12
p3	Updates to exception registers make sense	b4 b9 b15
p4	**Destination matches the target**	✓
p5	Memory value in equals register value out	b14
p6	Register value in equals memory value out	b16 b17
p7	**Memory address equals effective address**	✓
p8	**Privilege escalates correctly**	✓
p9	**Privilege deescalates correctly**	✓
p10	Jumps update the PC correctly	✗
p11	Jumps update the LR correctly	b13
p12	Instruction is in a valid format	b11
p13	Continuous Control Flow	b5
p14	Exception return updates state correctly	b1 b5
p15	**Register change implies that it is the instruction target**	✓
p16	SR is not written to a GPR in user mode	✗
p17	Interrupt implies handled	b8
p18	Instruction unchanged in pipeline	--
p19	**SPR modified only in supervisor mode**	✓
p20	**Enter supervisor mode is on reset or exception**	✓
p21	Exception handling implies exception mechanism activated	b8
p22	Unspecified custom instructions are not allowed	✗
p23	Exception handler accessed only during exception, in supervisor mode, or on reset	b8
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

Properties Found in the Inference Step

Results: Comparison to State-of-the-Art

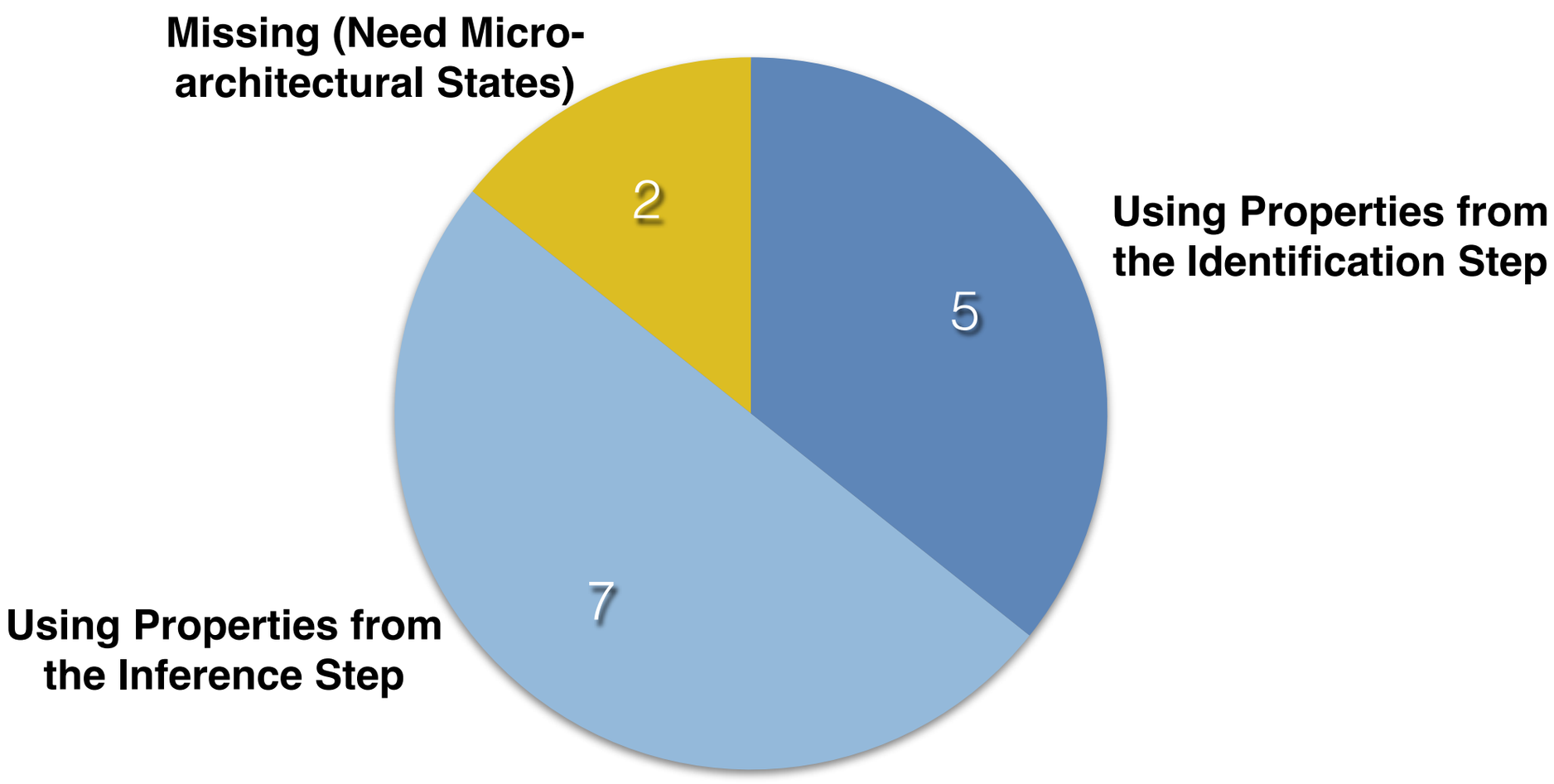
No.	Security Property Description	Found?
p1	Execution privilege matches page privilege	✓
p2	SPR equals GPR in register move instruction	b12
p3	Updates to exception registers make sense	b4 b9 b15
p4	Destination matches the target	✓
p5	Memory value in equals register value out	b14
p6	Register value in equals memory value out	b16 b17
p7	Memory address equals effective address	✓
p8	Privilege escalates correctly	✓
p9	Privilege deescalates correctly	✓
p10	**Jumps update the PC correctly**	✗
p11	Jumps update the LR correctly	b13
p12	Instruction is in a valid format	b11
p13	Continuous Control Flow	b5
p14	Exception return updates state correctly	b1 b5
p15	Register change implies that it is the instruction target	✓
p16	**SR is not written to a GPR in user mode**	✗
p17	Interrupt implies handled	b8
p18	Instruction unchanged in pipeline	--
p19	SPR modified only in supervisor mode	✓
p20	Enter supervisor mode is on reset or exception	✓
p21	Exception handling implies exception mechanism activated	b8
p22	**Unspecified custom instructions are not allowed**	✗
p23	Exception handler accessed only during exception, in supervisor mode, or on reset	b8
p24	Page fault generated if MMU detects an access control violation	--
p25	UART output changes on a write command from CPU	--
p26	Only transmit command or initialization change Ethernet data output	--
p27	Debug Unit's value and control registers only accessible from supervisor mode	--

Properties Not Found

Results: New Security Properties Found

No.	Security Property Description	Found?
p28	Flags that influence control flow should be set correctly	b6 b7
p29	Calculation of memory address or memory data is correct	b3 b10
p30	Link address is not modified during function call execution	✓

Results: Stopping New Bugs



Result of detecting 14 AMD errata from SPECS project
(bugs not used in the development of the assertions).

Summary

SCIFinder:

- Generates security-critical invariants semi-automatically
- Requires a list of known security-critical bugs and a processor design

Main Results:

- The final SCI set covers 86.4% of the manually crafted security properties
- We identify 3 new properties not seen in prior work