

Eternal War in Memory

Systematization of Knowledge

László Szekeres, Mathias Payer, Tao Wei, Dawn Song

Stony Brook University

University of California, Berkeley

Peking University

Problem

- C/C++ is unsafe
- Everybody runs C/C++ code
- They surely have exploitable vulnerabilities

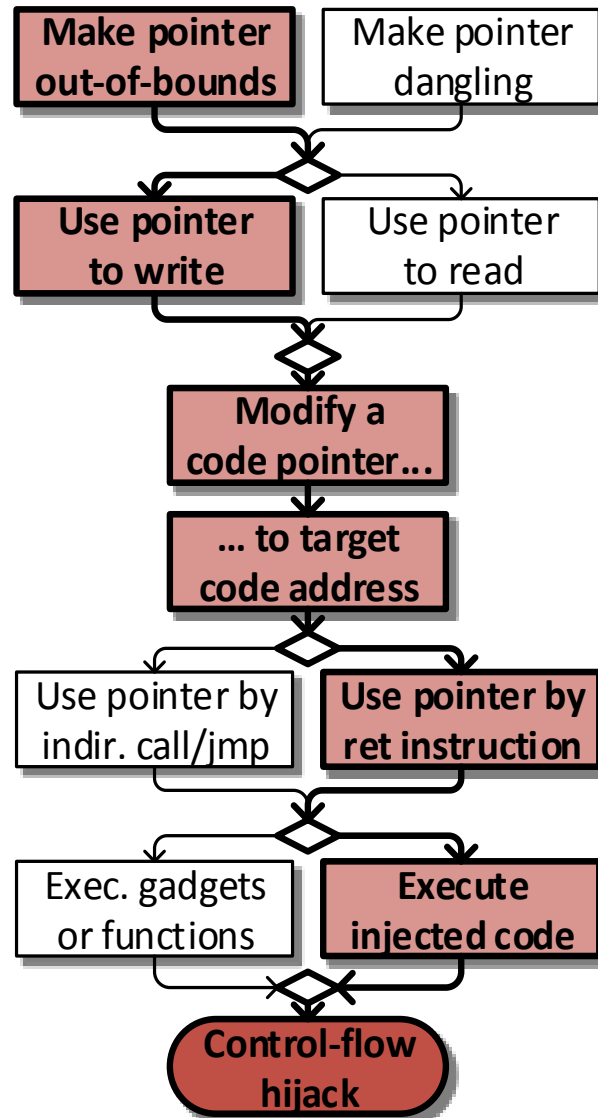


Overview

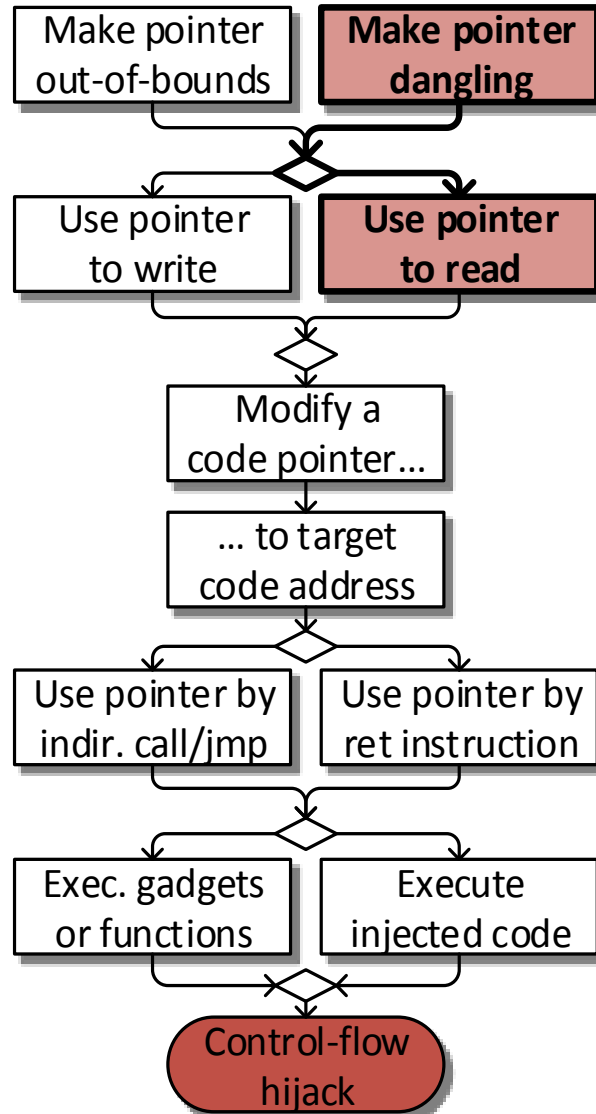
- What are the attacks?
- What are the deployed protections?
- What are the *not* deployed protections?
- Why aren't they deployed?

Attack model

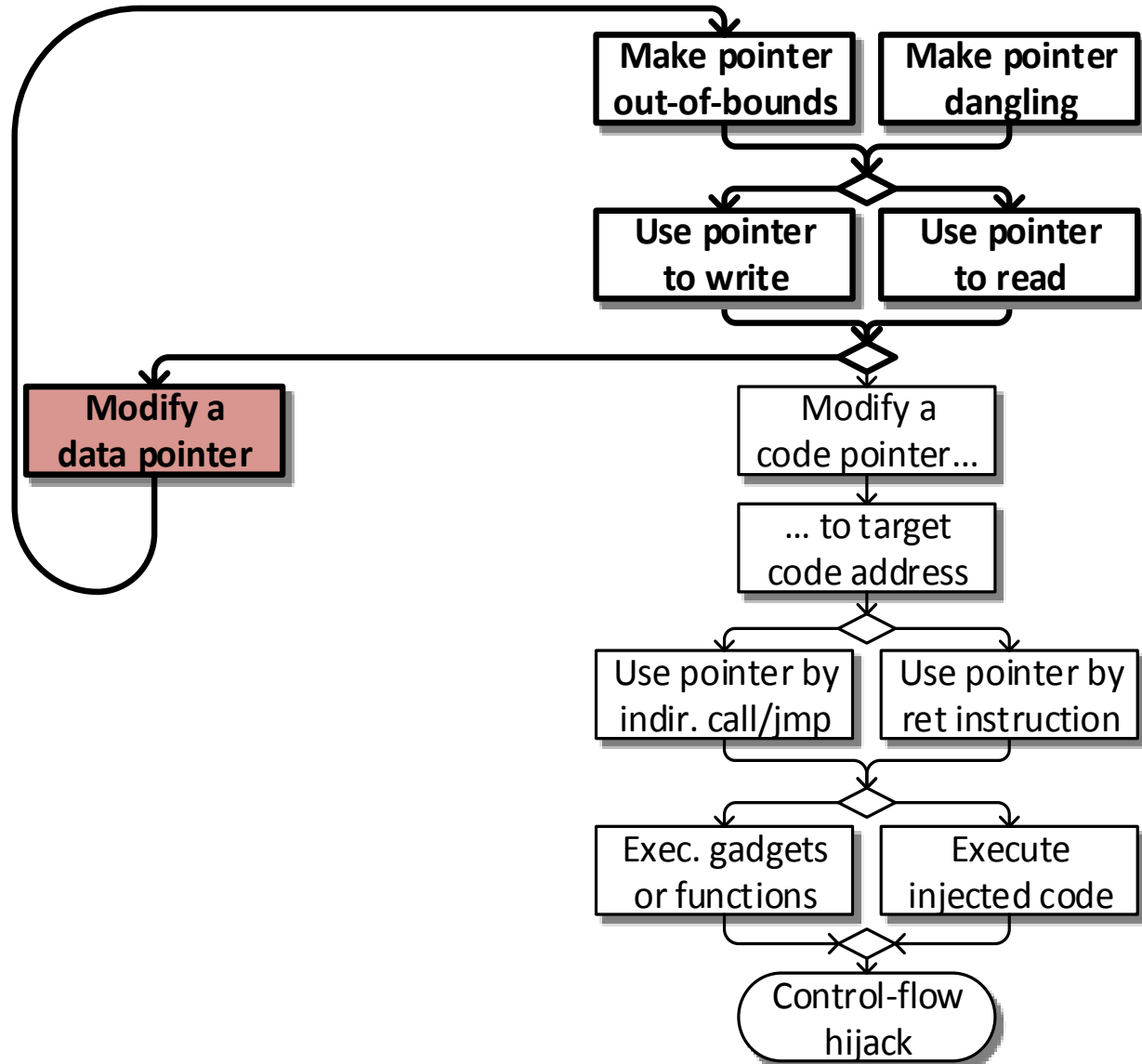
Classic stack smashing attack



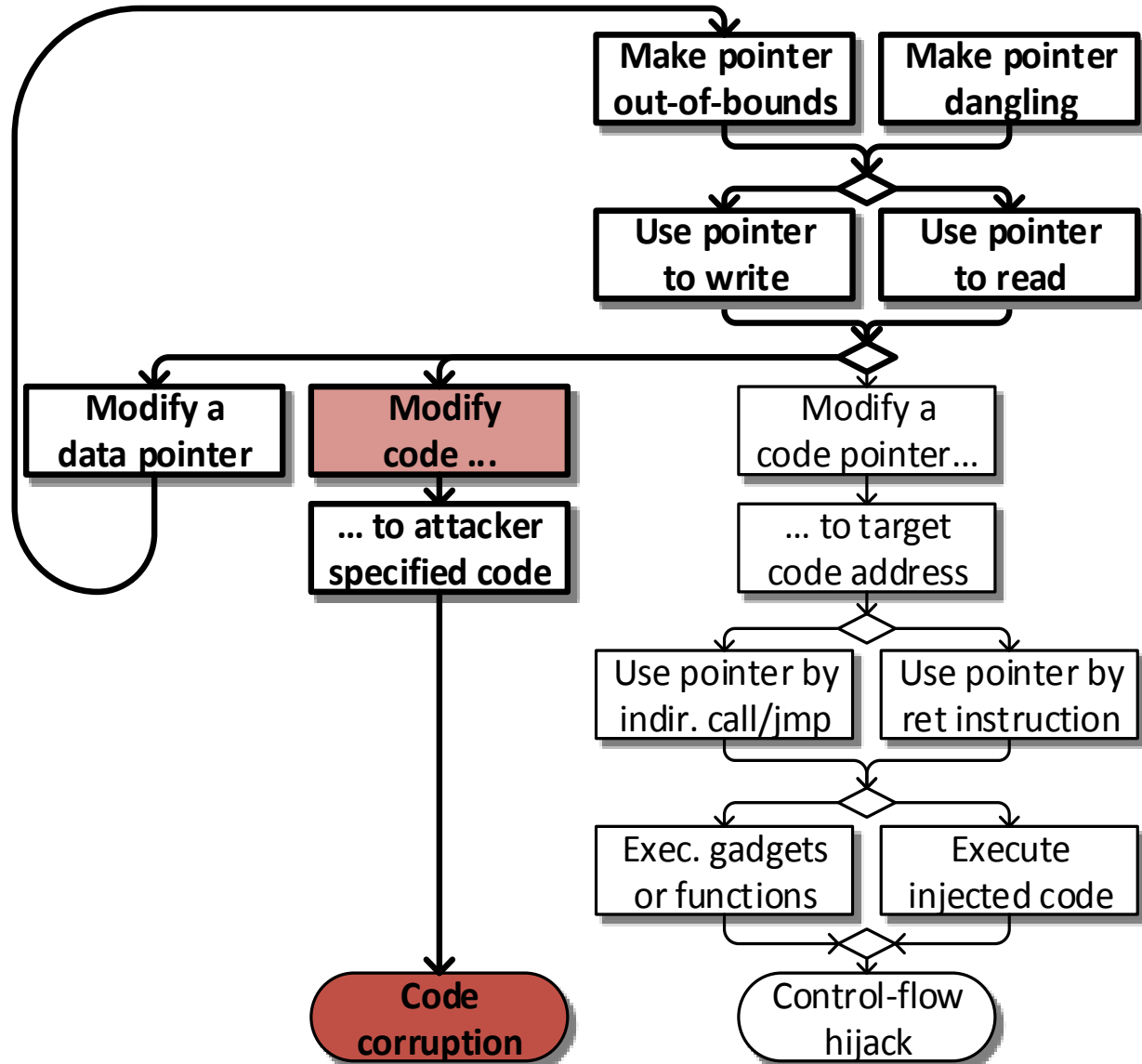
Use-after-free exploits



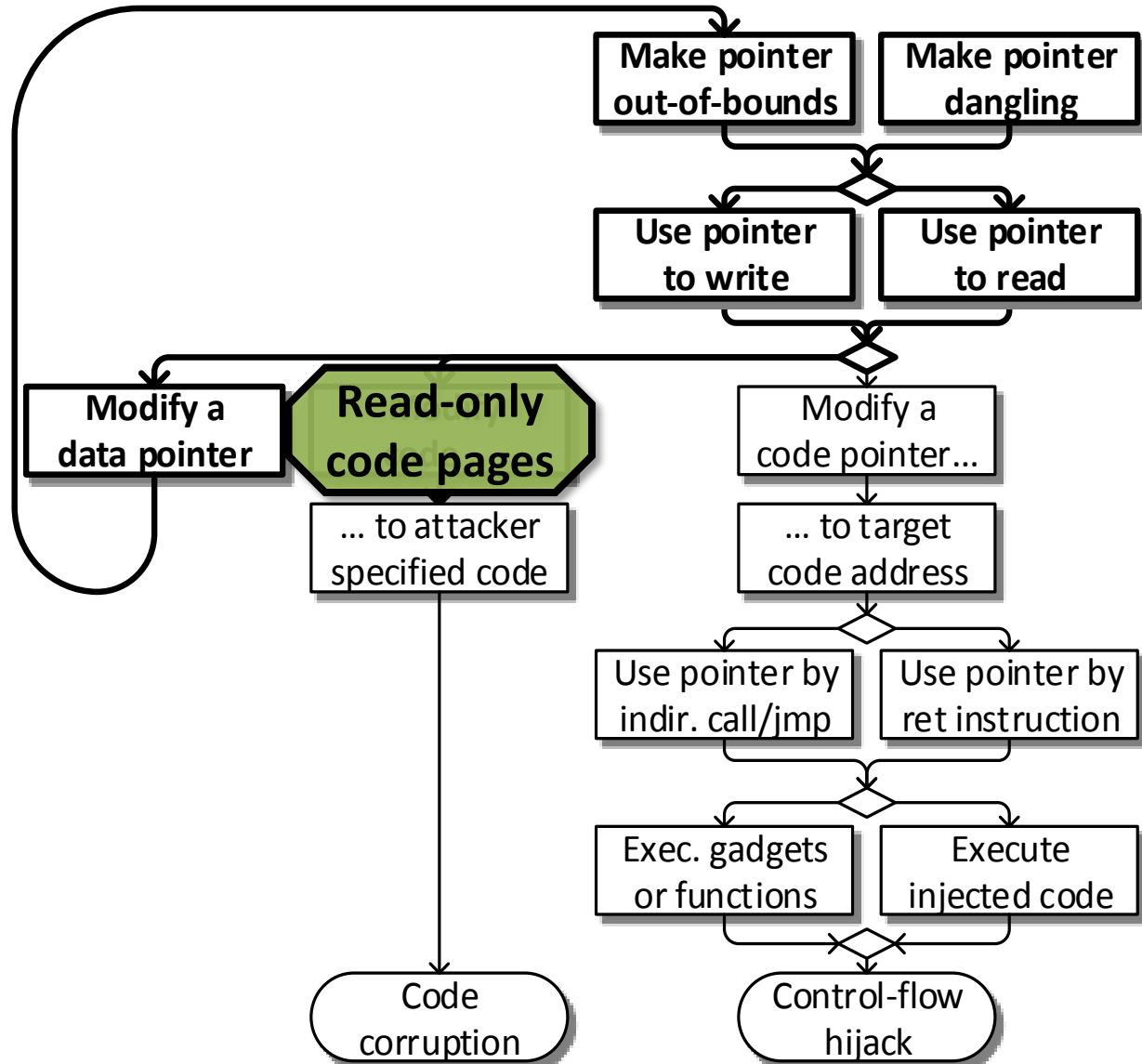
Corrupting newer and newer pointers



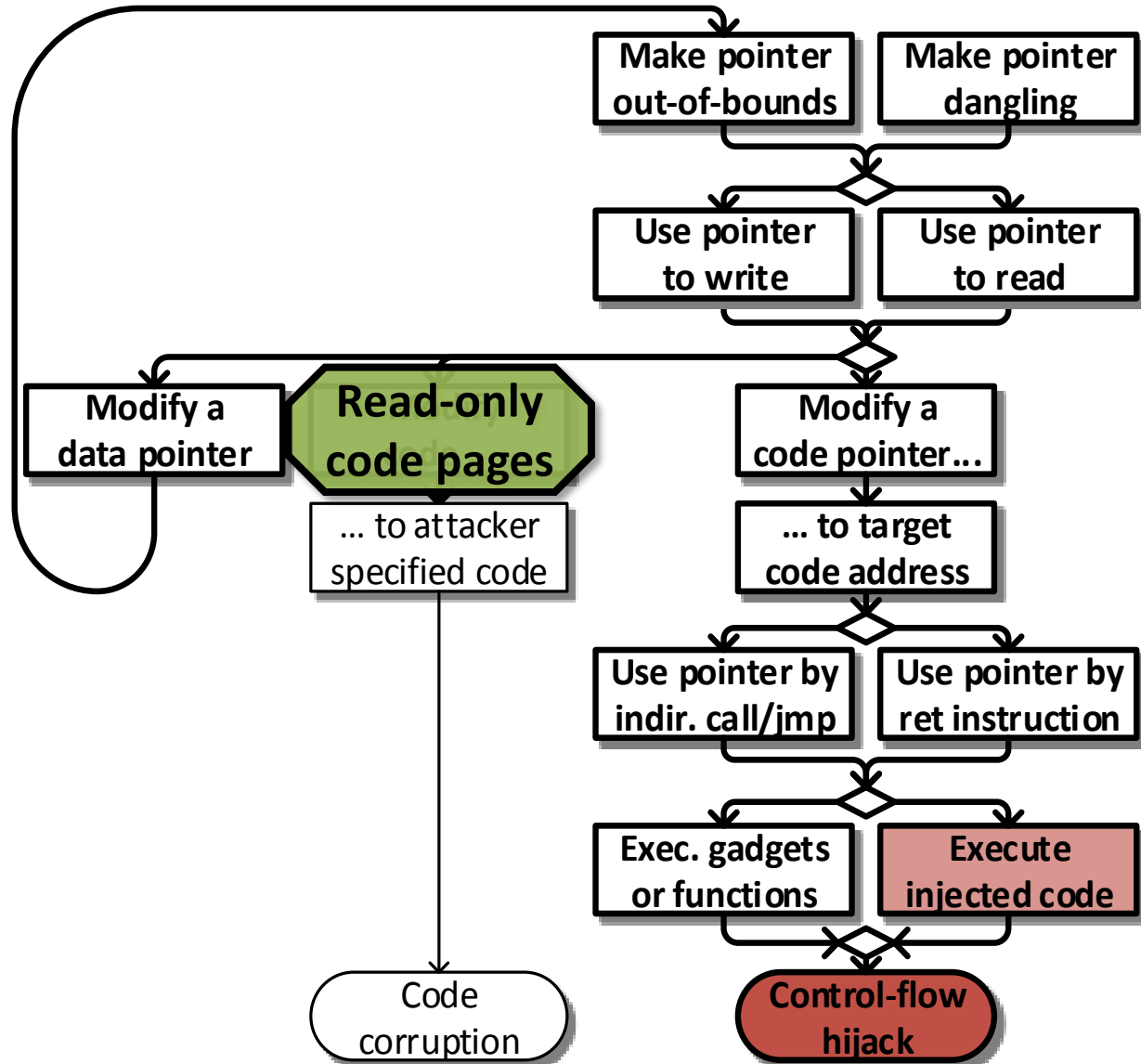
Modifying the code itself



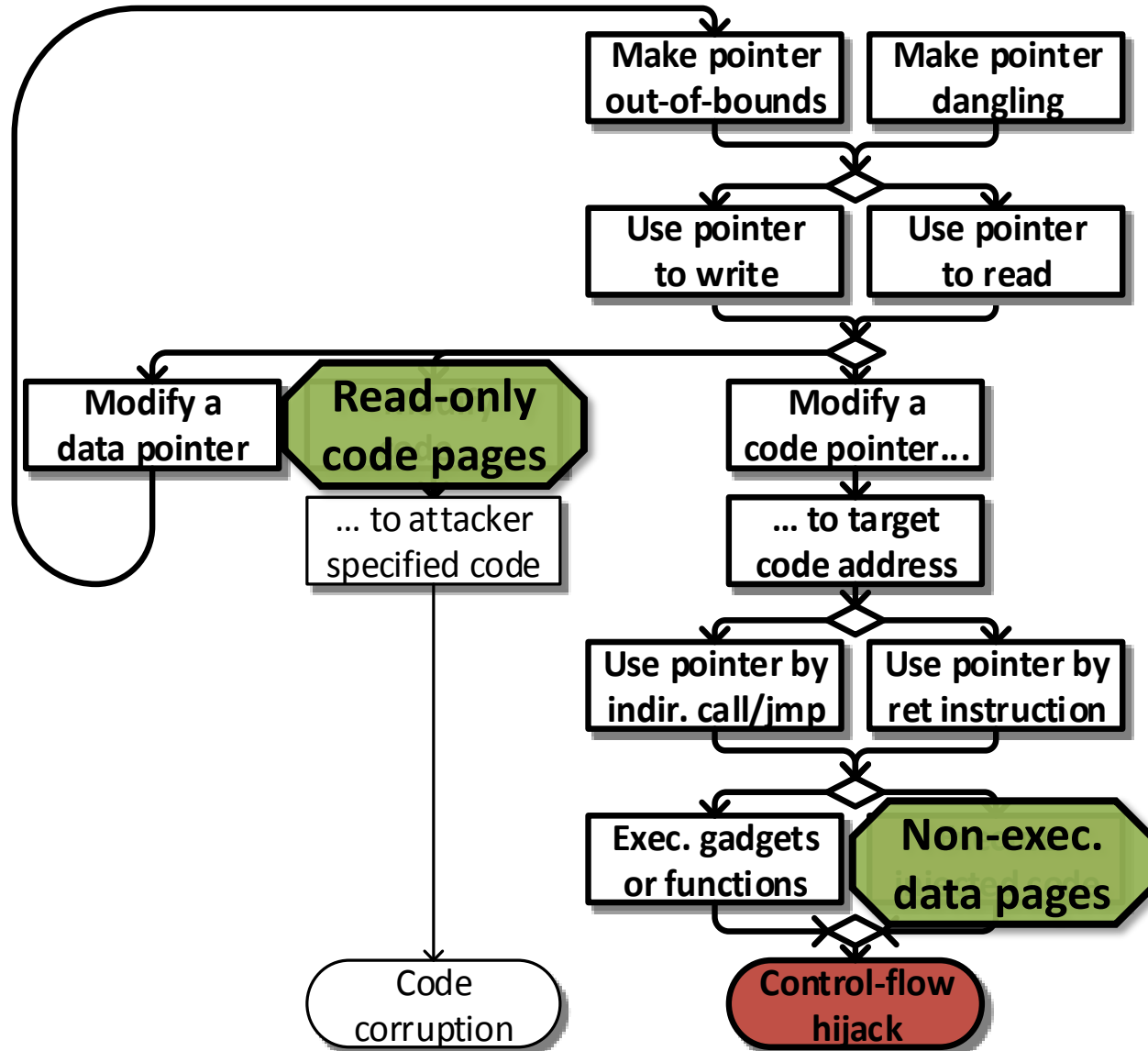
Code integrity



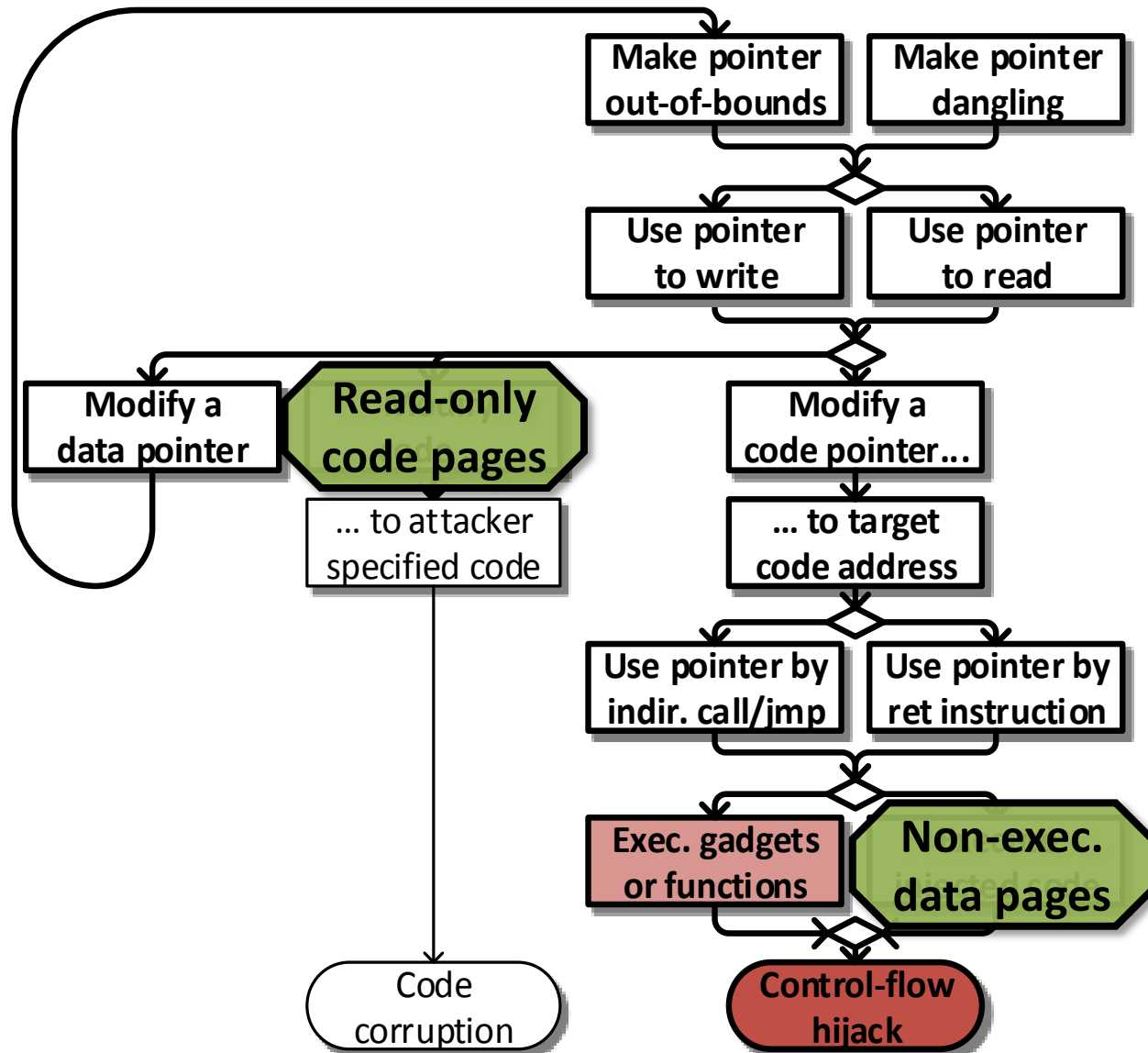
Non-executable data



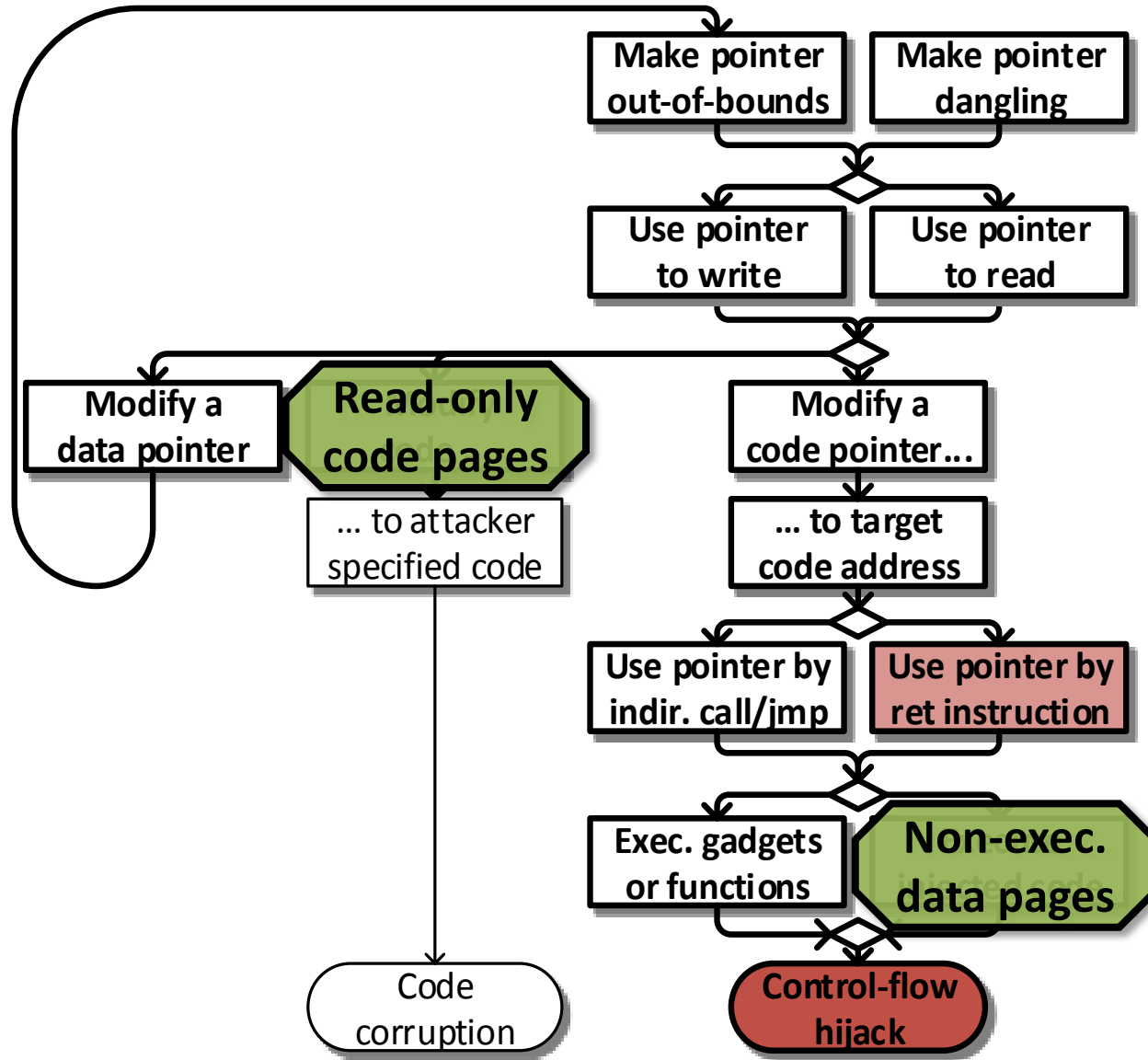
Non-executable data



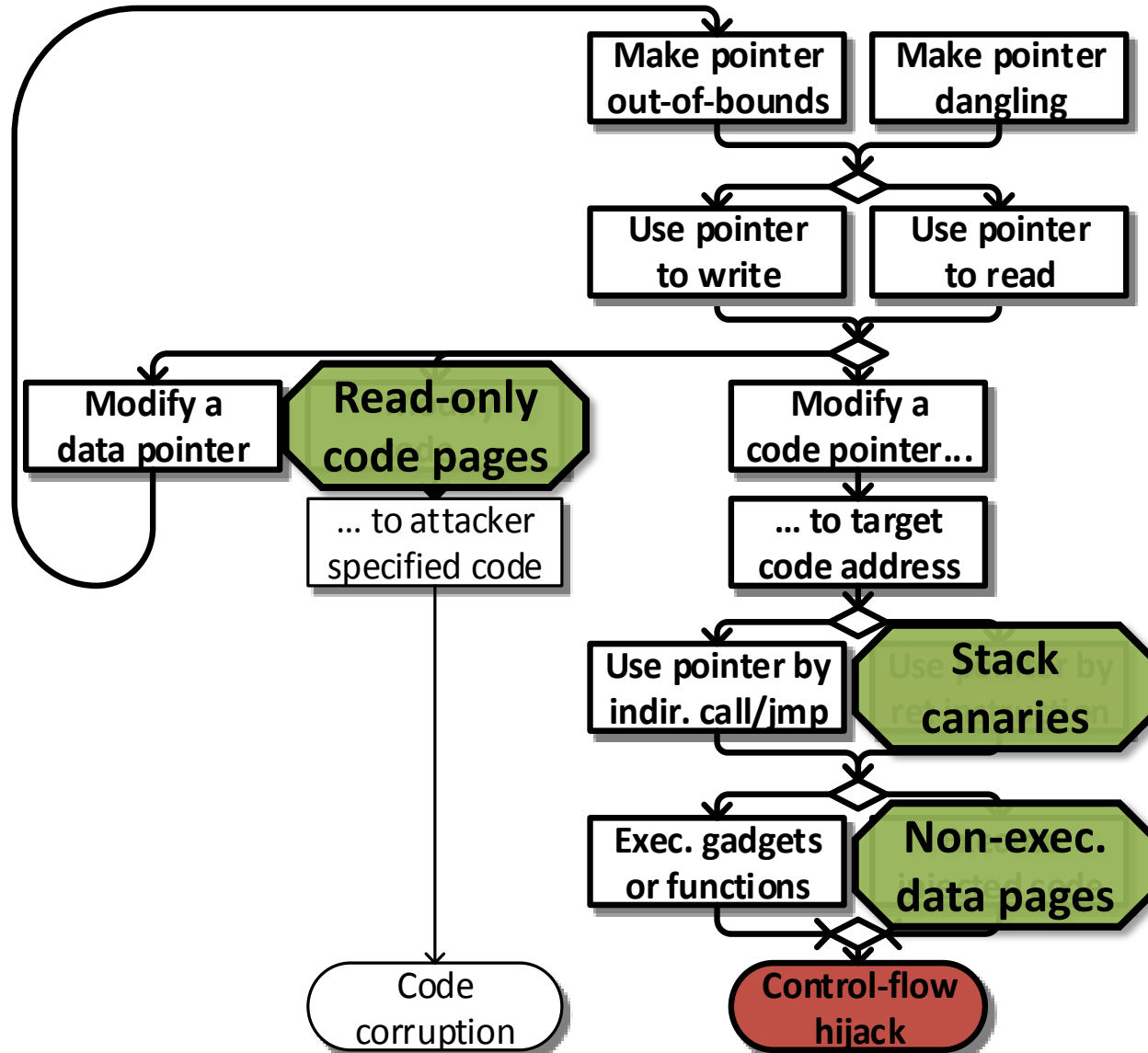
Return-oriented programming



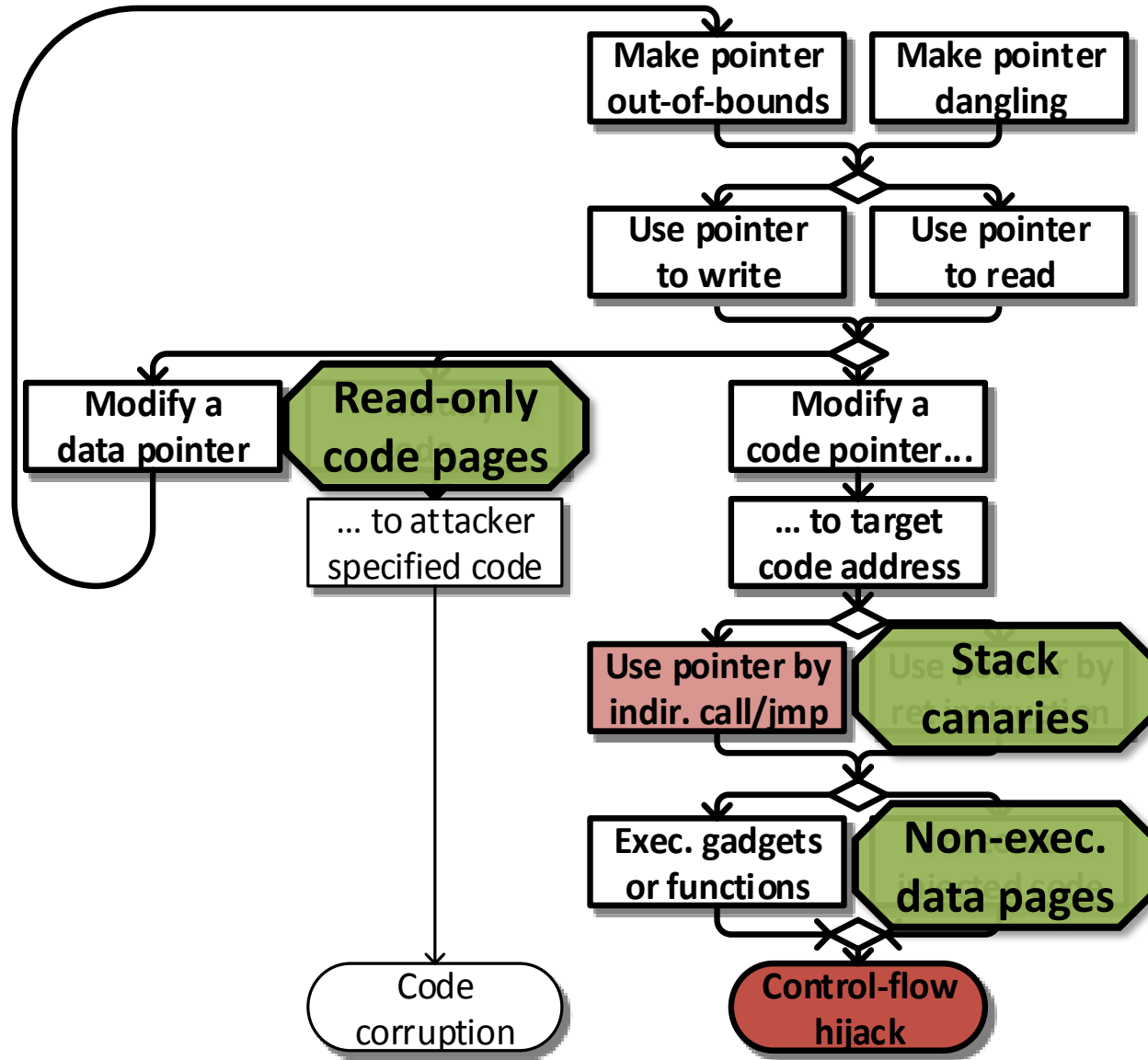
Return integrity



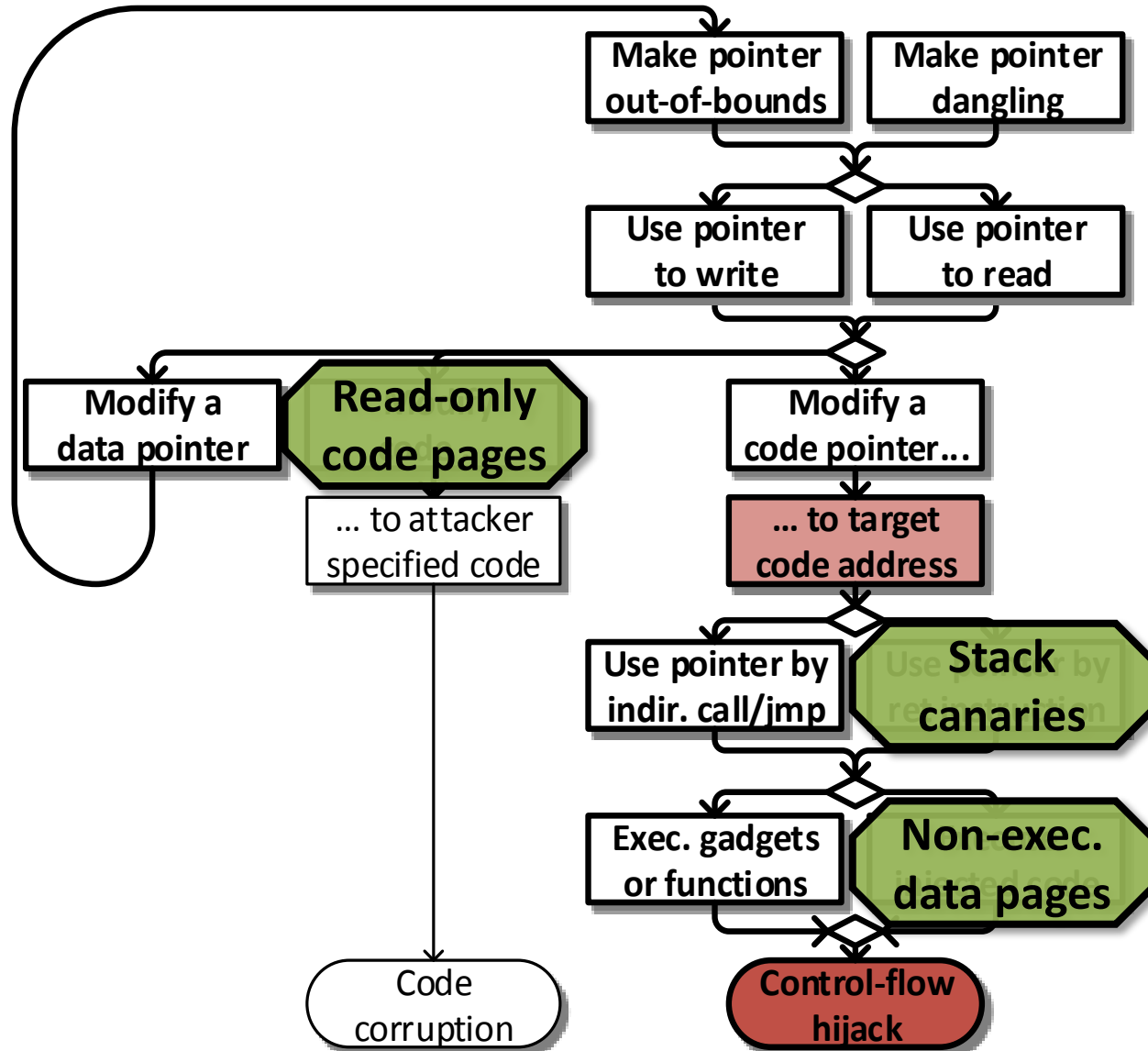
Return integrity



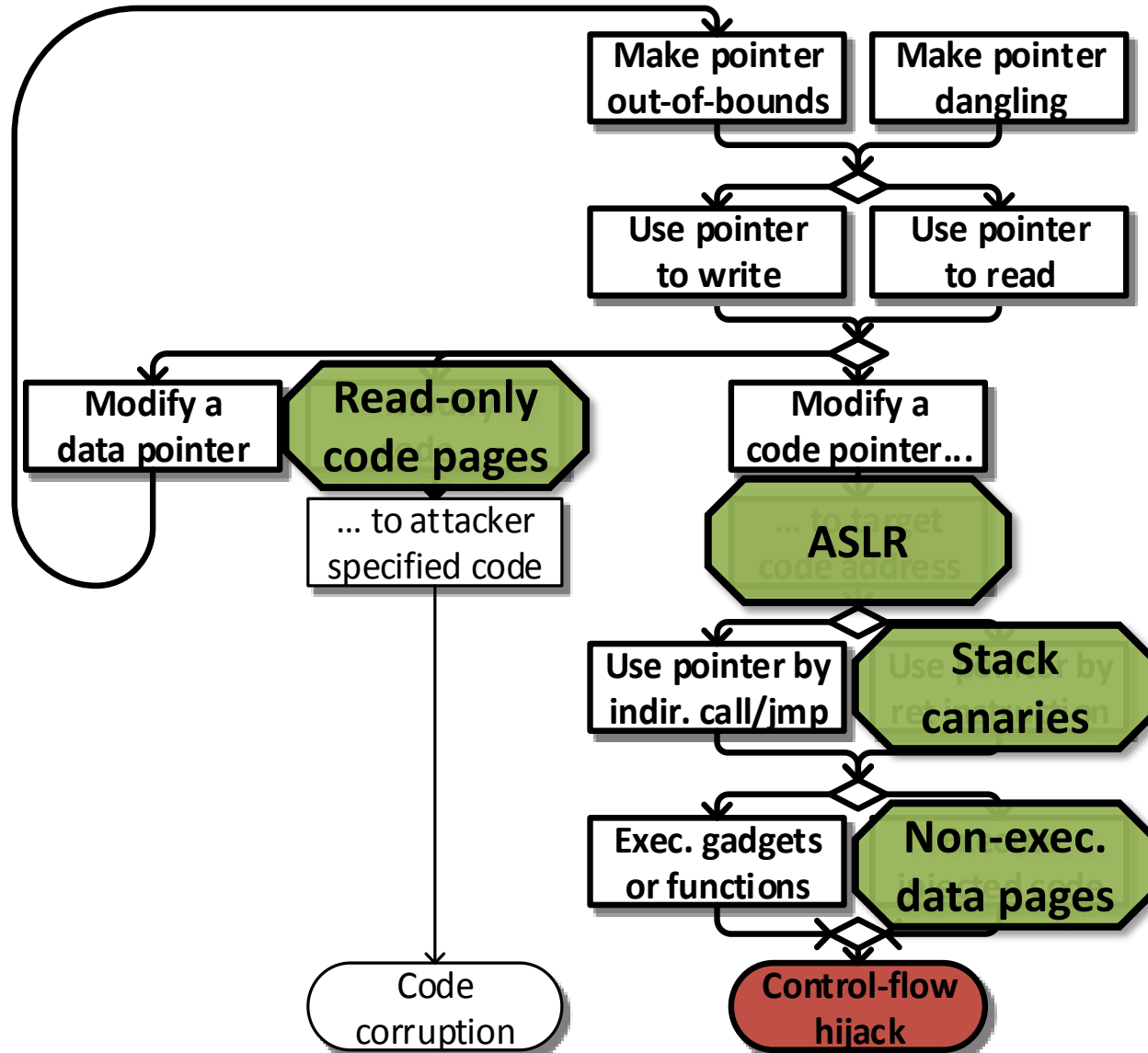
Hijacking indirect calls and jumps



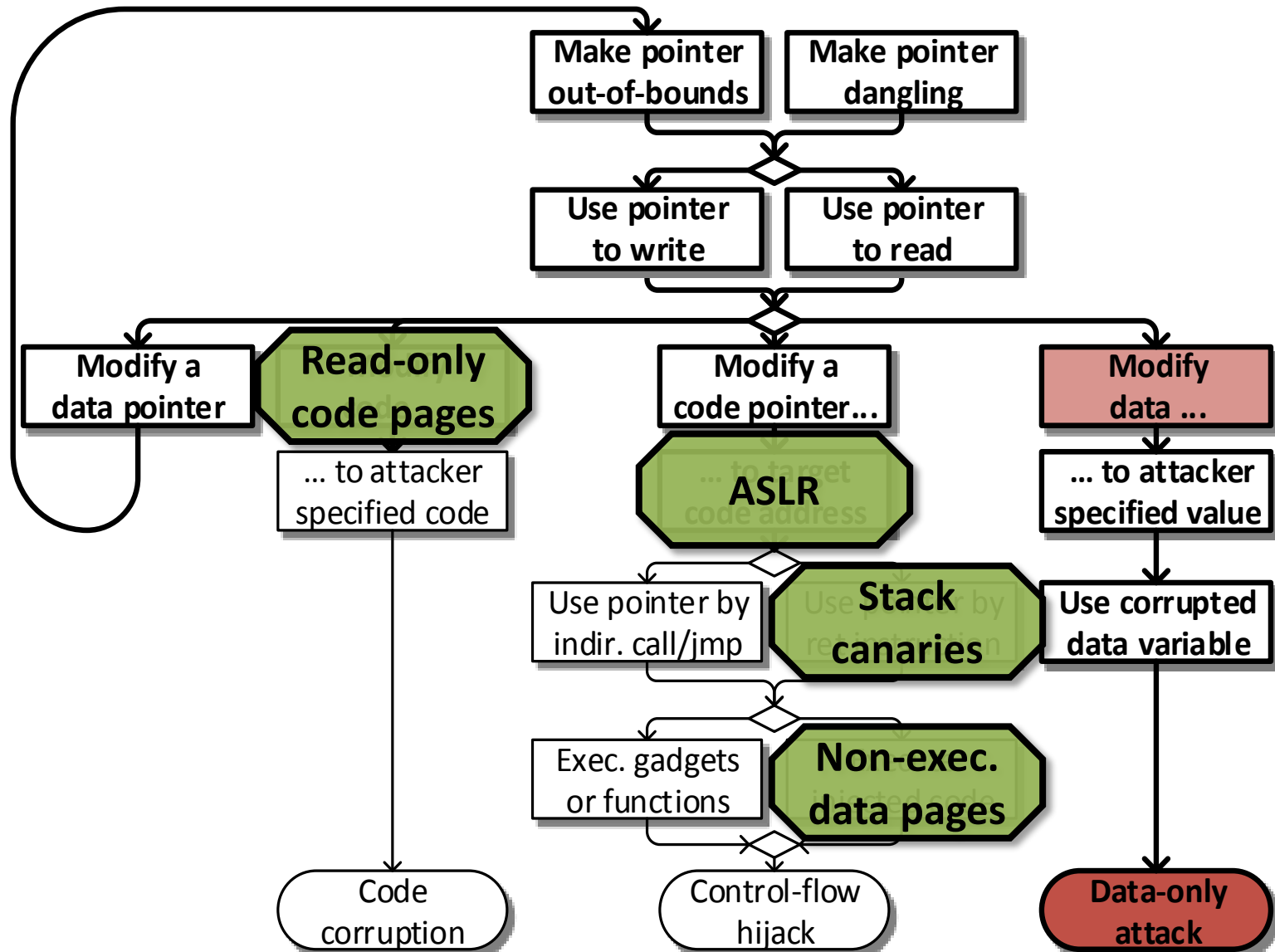
Address space randomization



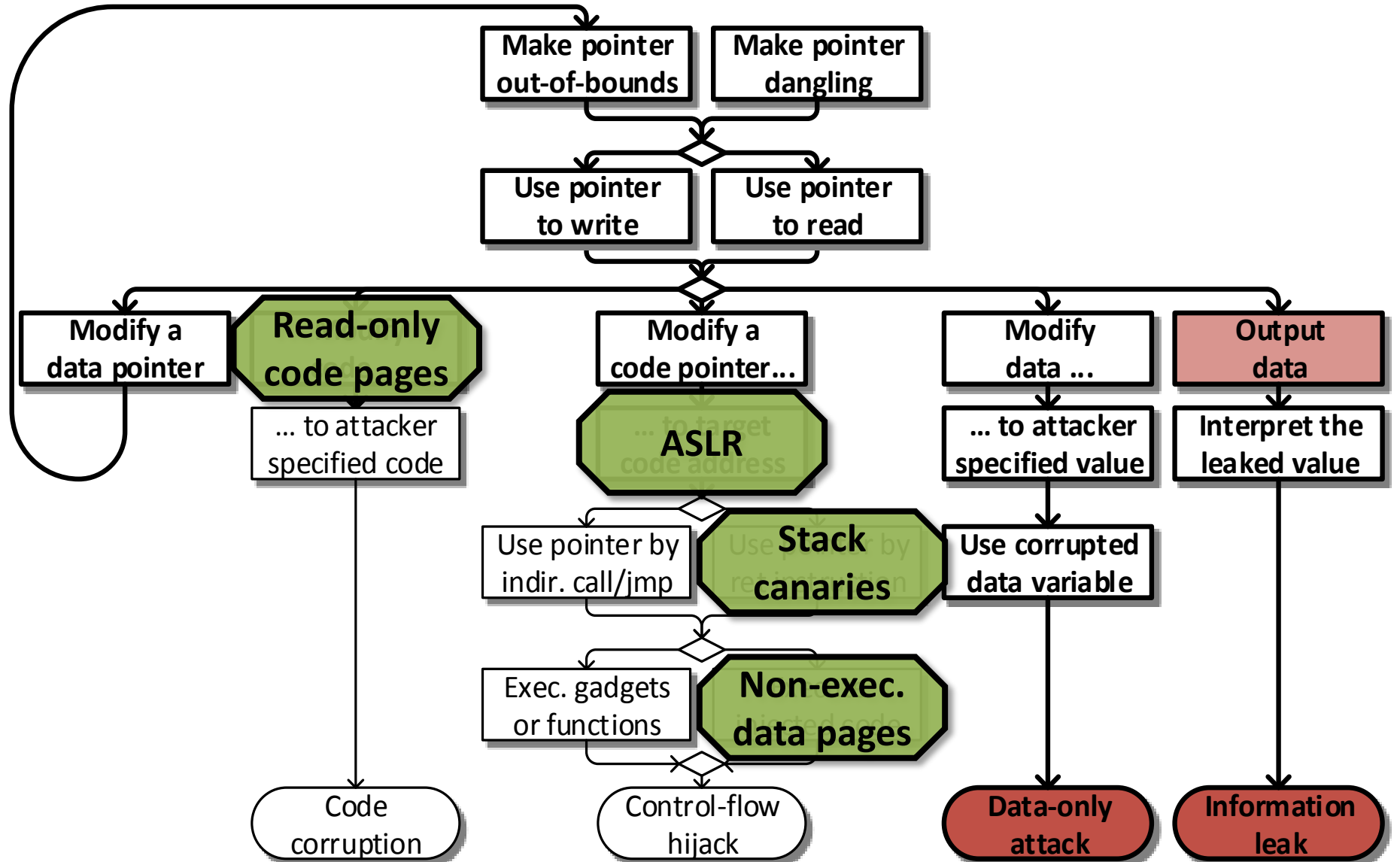
Address space randomization



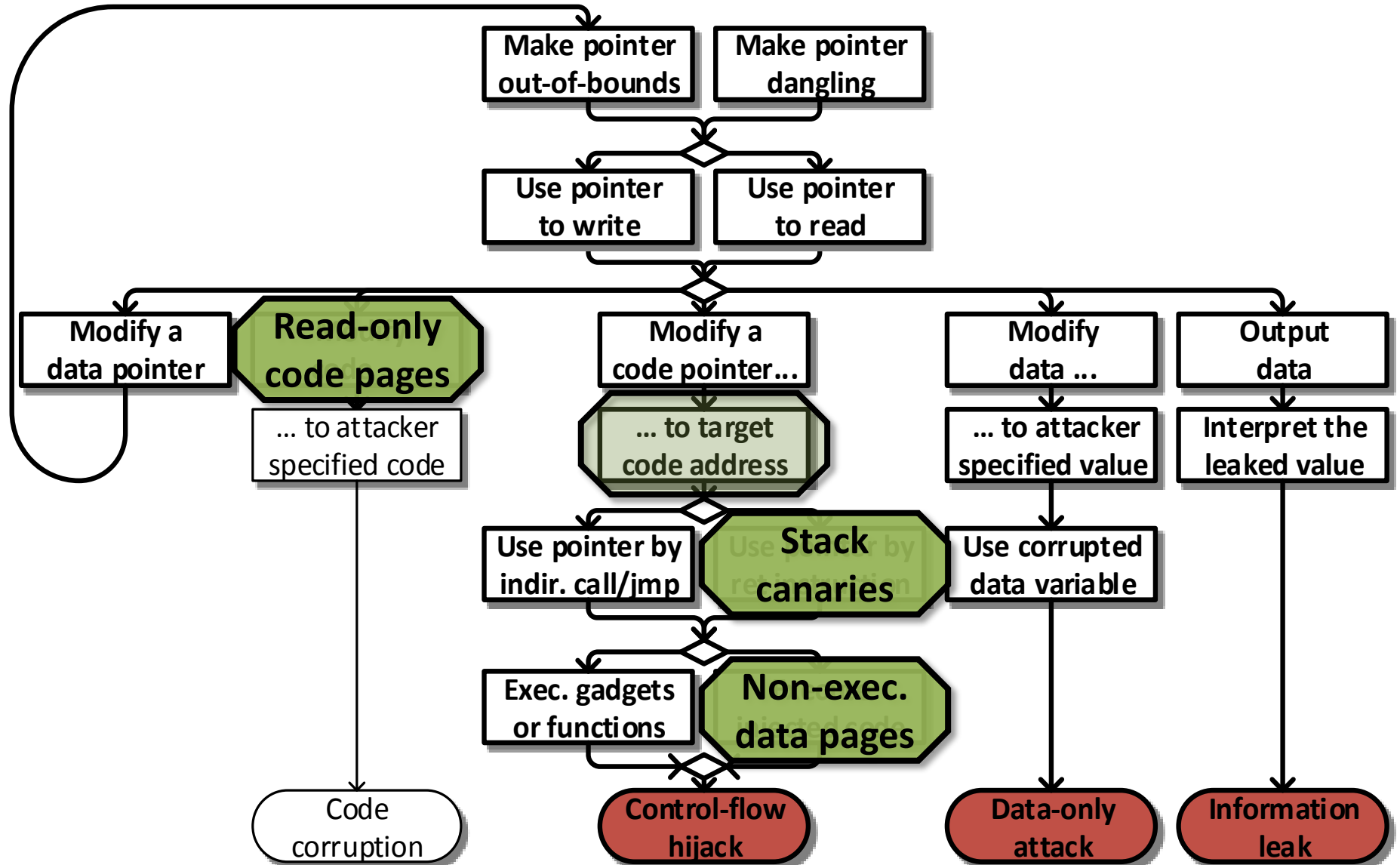
Data-only attack



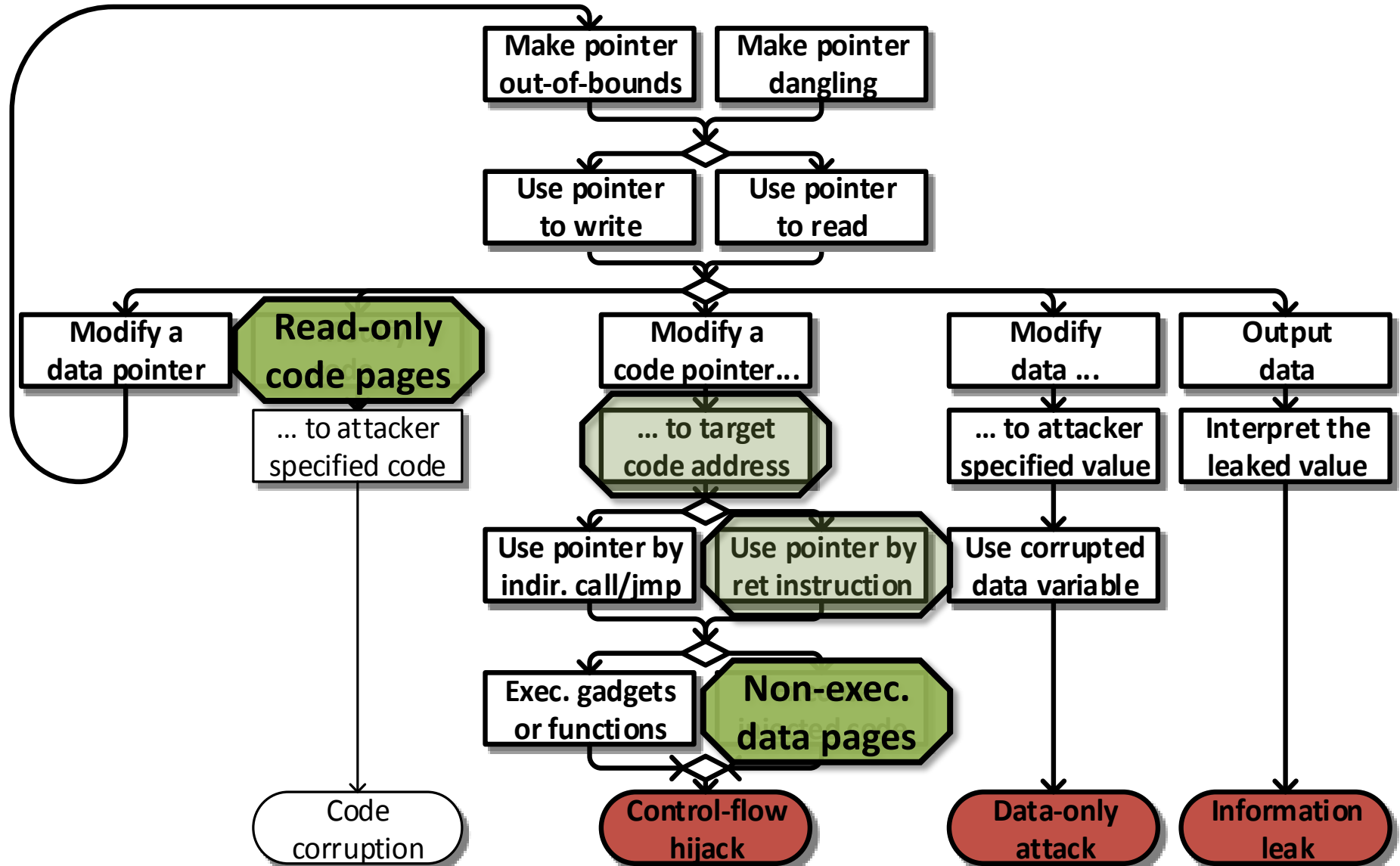
Information leakage



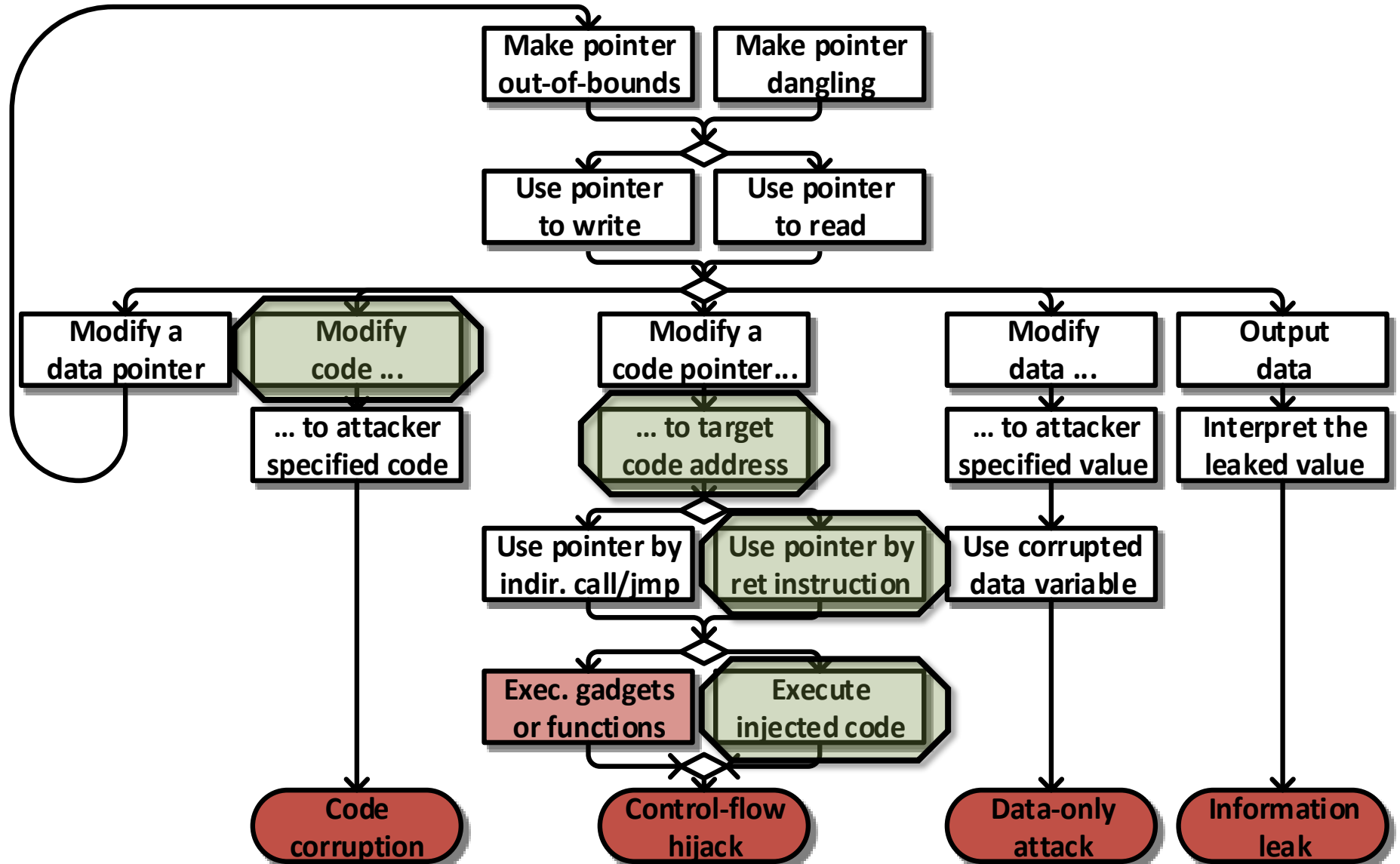
Bypassing ASLR with user scripting



Bypassing stack cookies



Problems due to JIT compilation

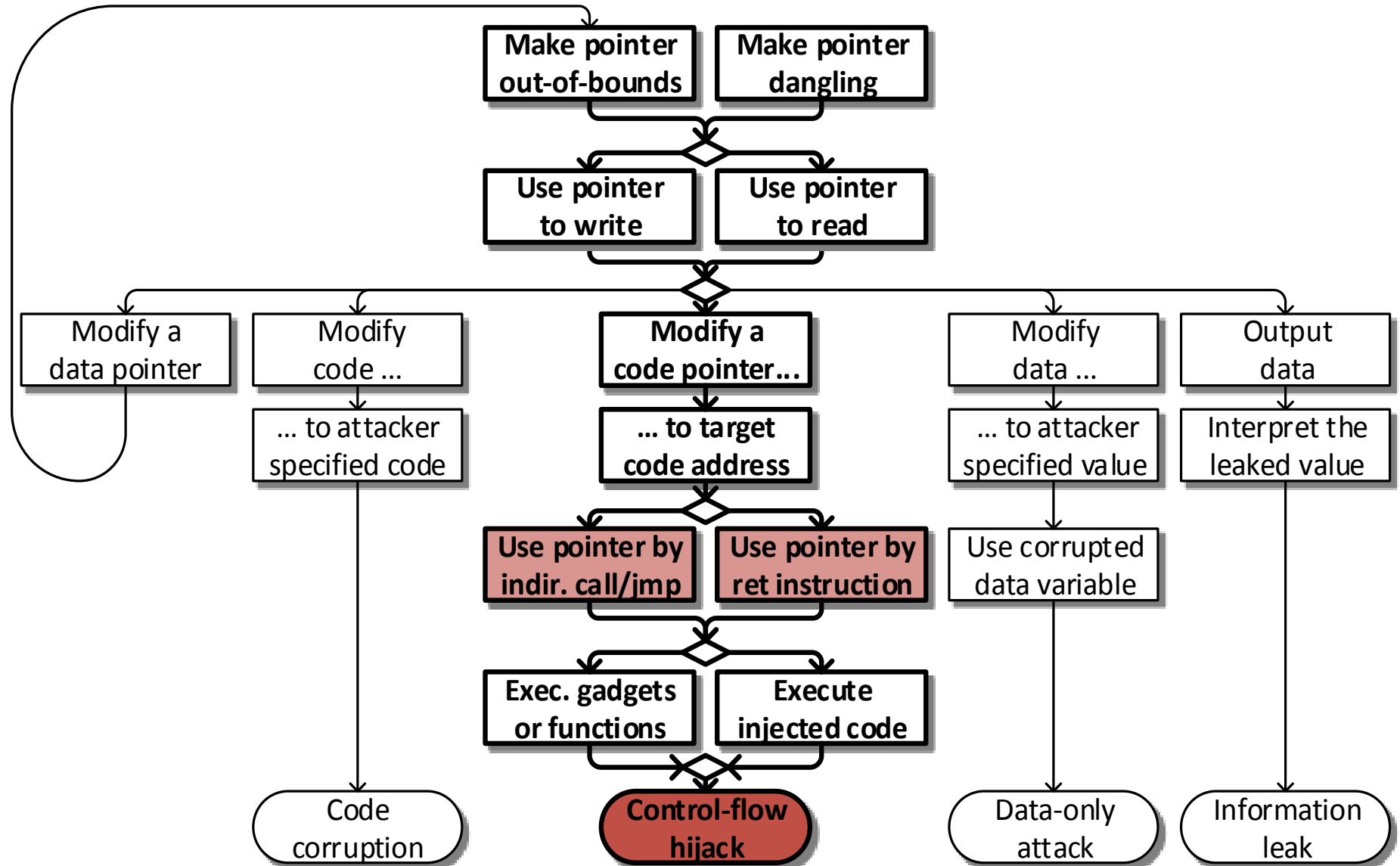


Deployed protections

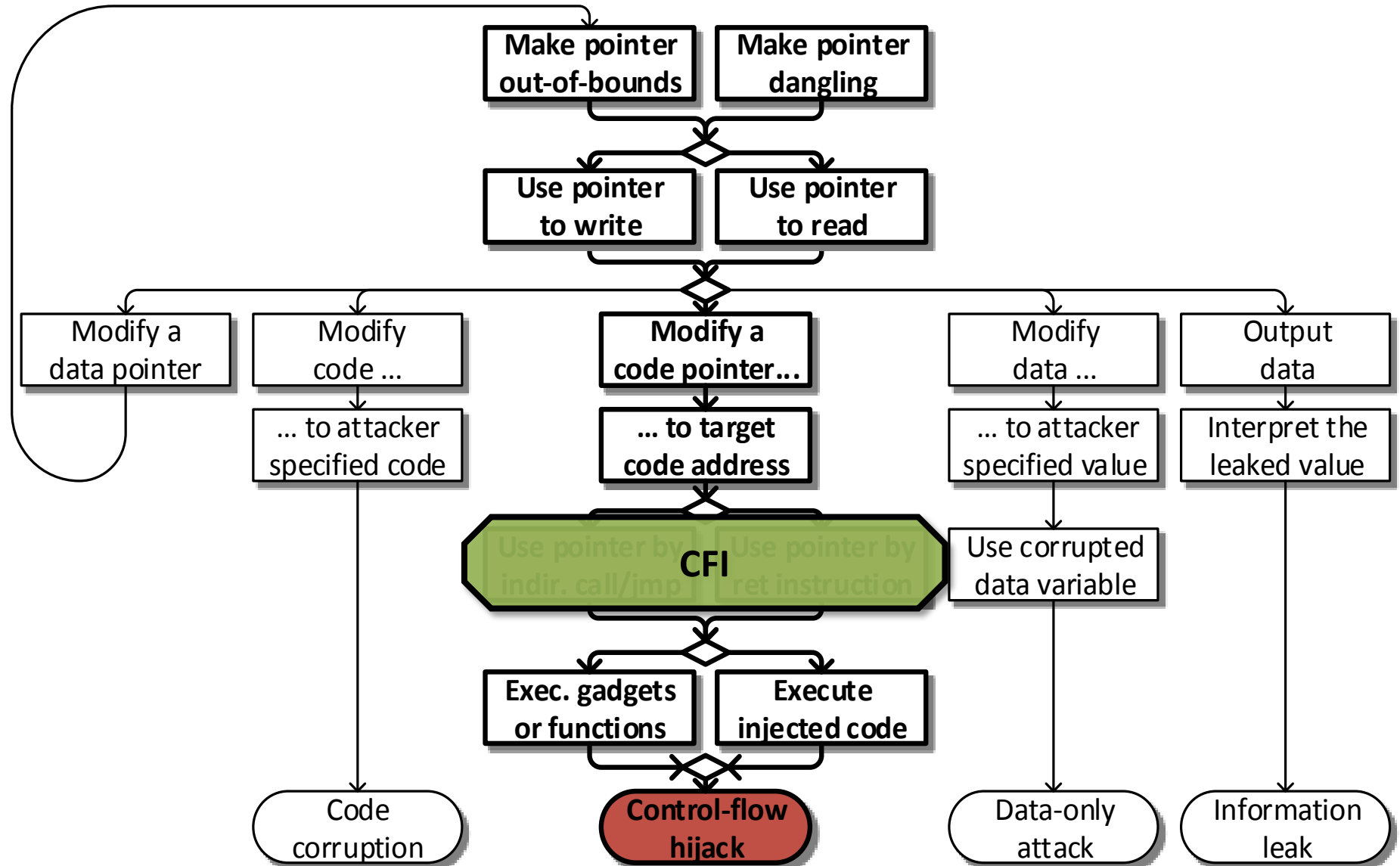
	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	W \oplus R	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good

Proposed solutions

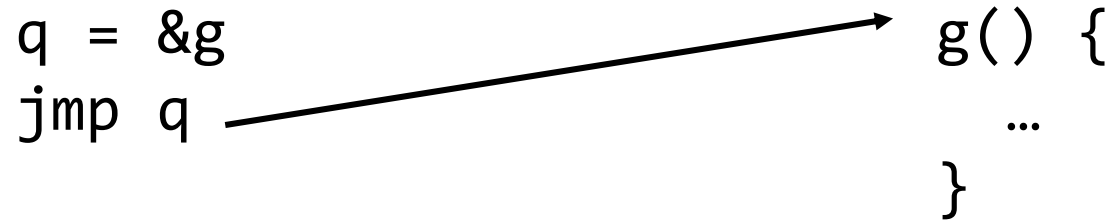
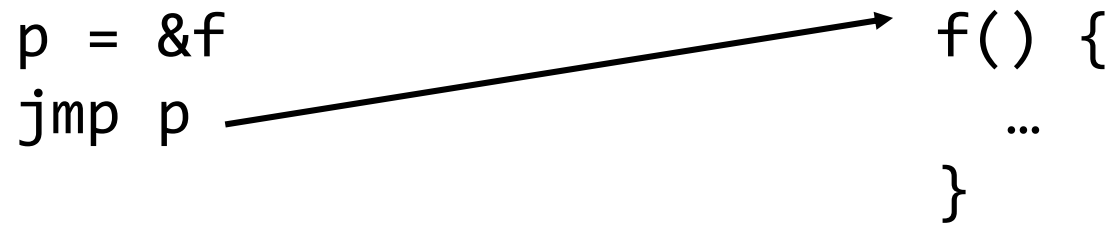
Control-flow integrity



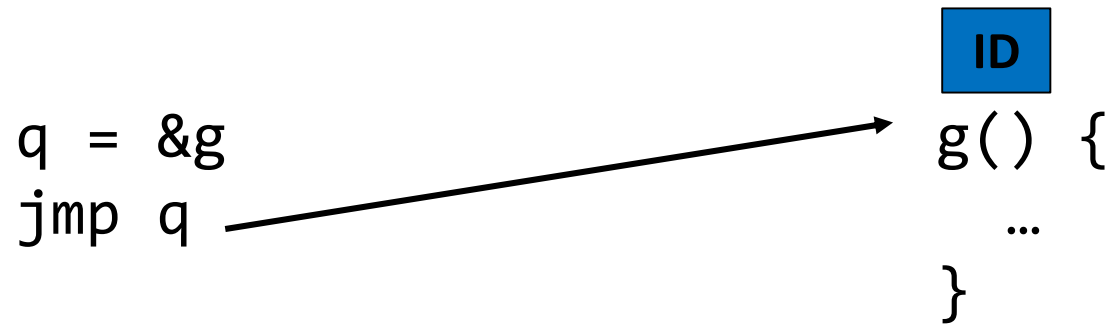
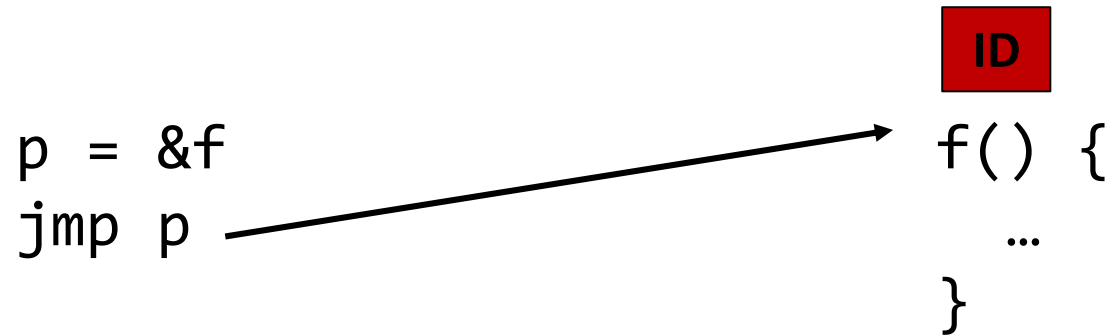
Control-flow integrity



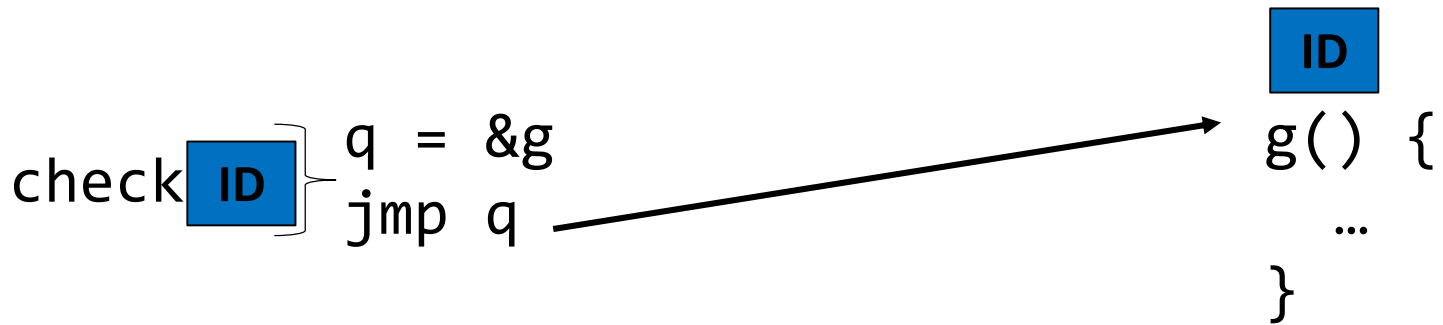
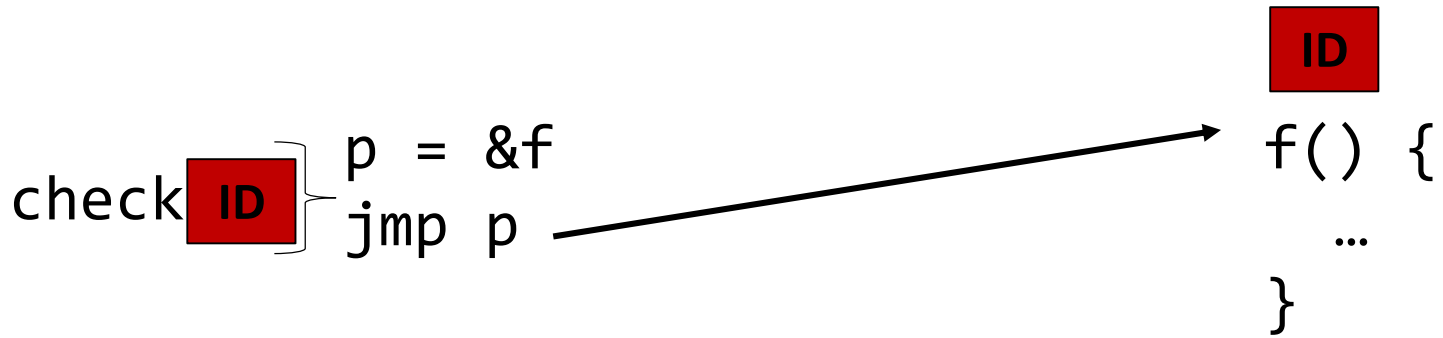
CFI



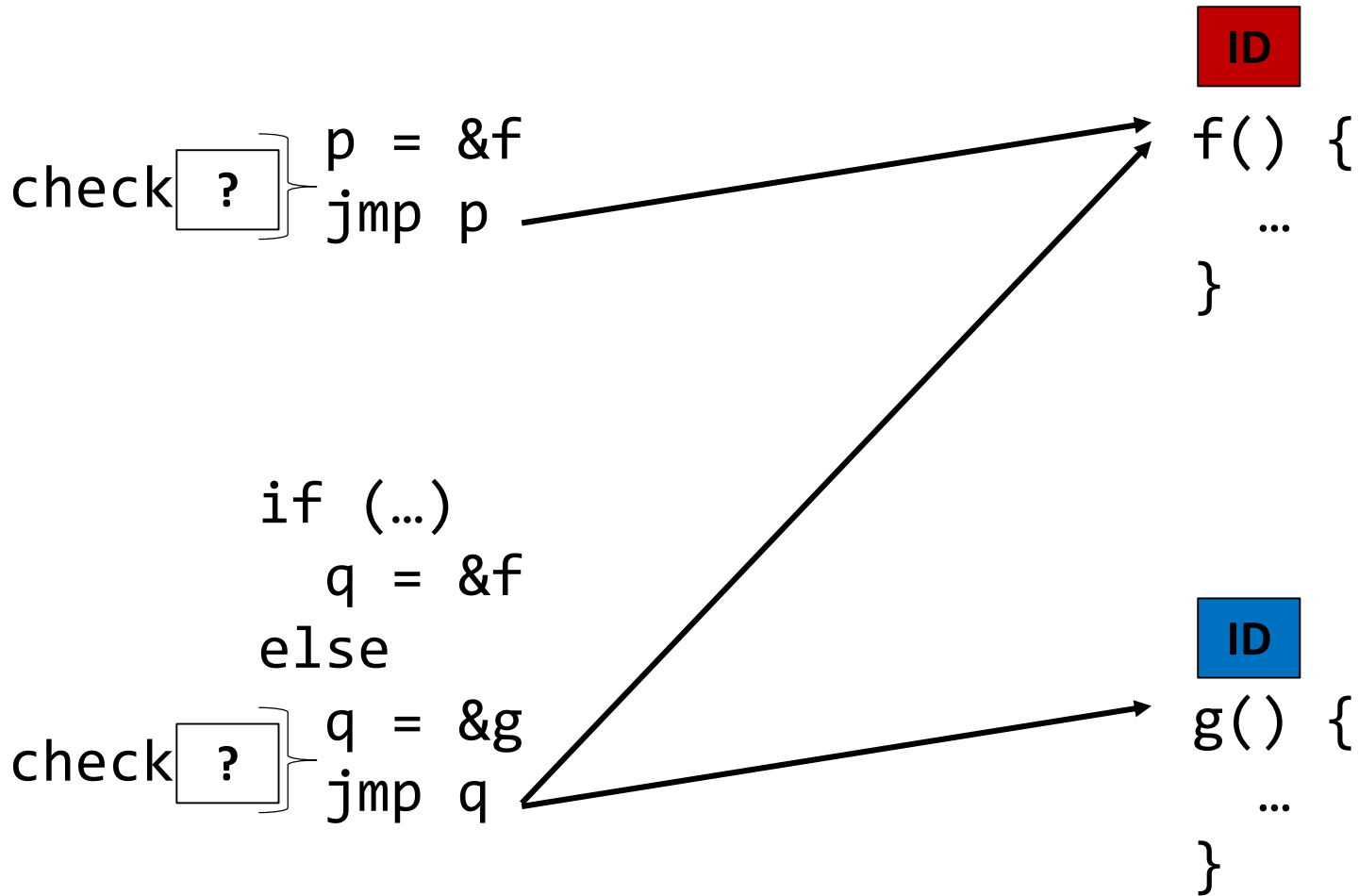
CFI



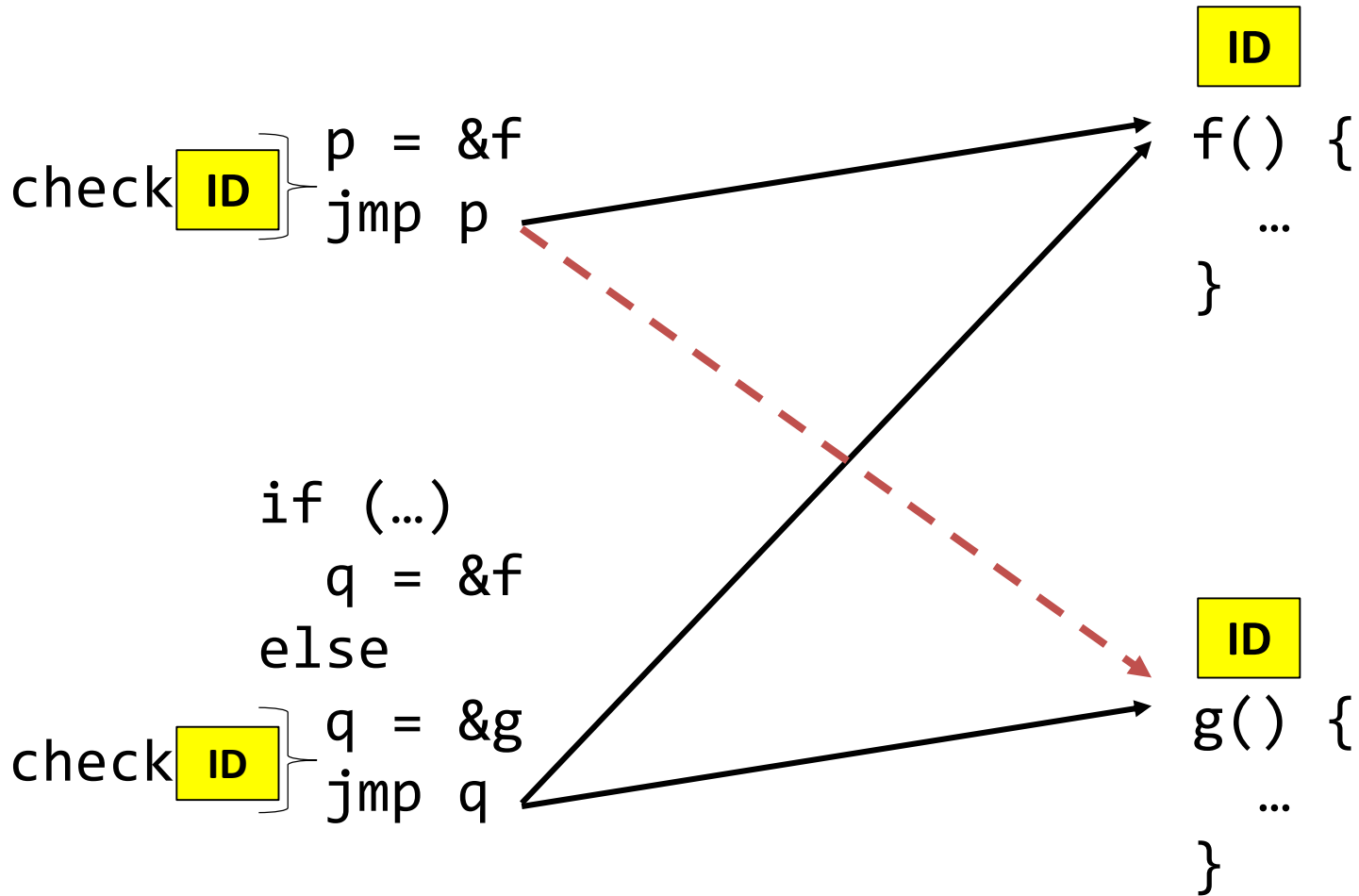
CFI



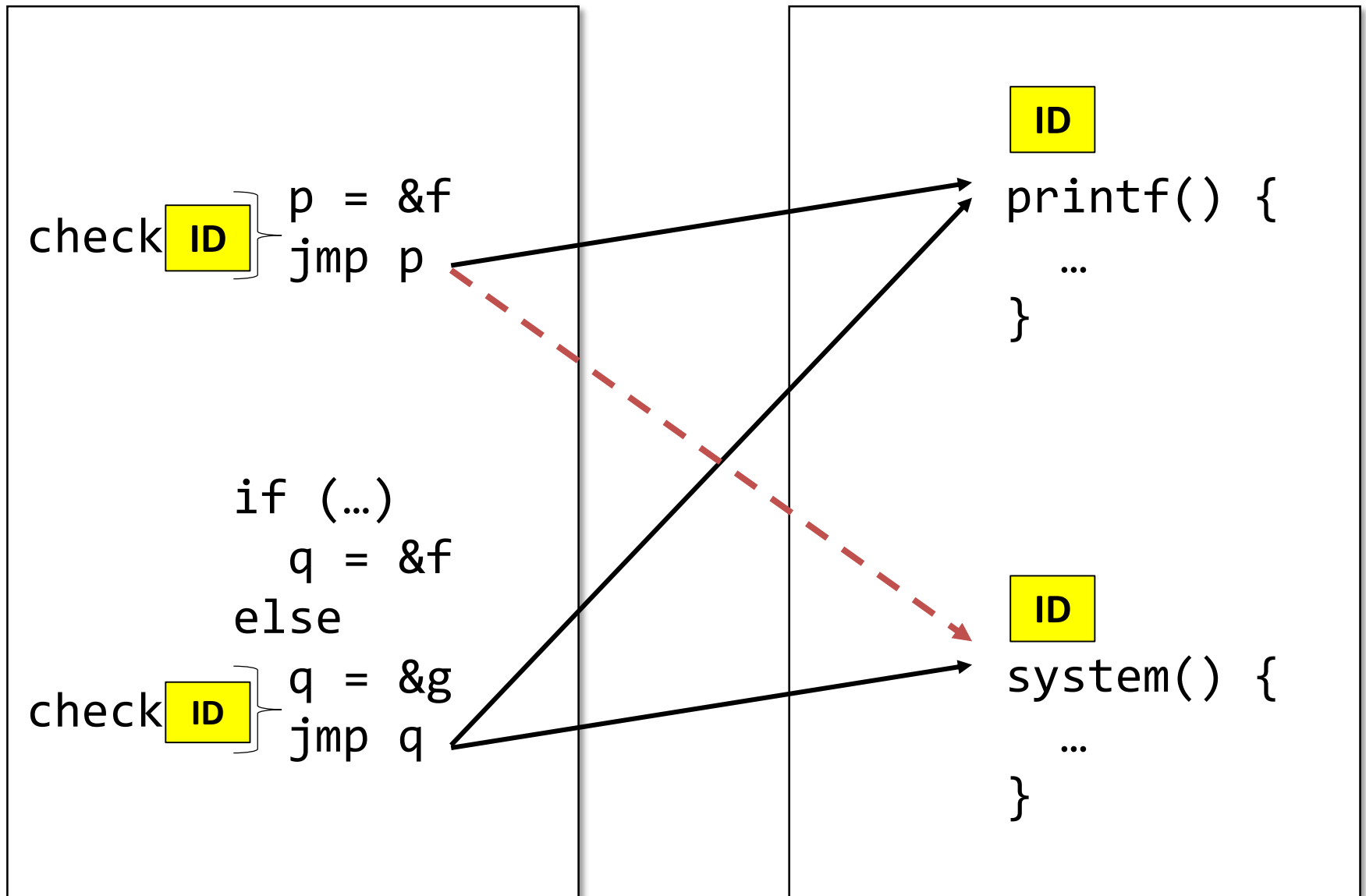
CFI



Over-approximation problem



Over-approximation problem



Modularity problem

ID

```
printf() {  
    ...  
}
```

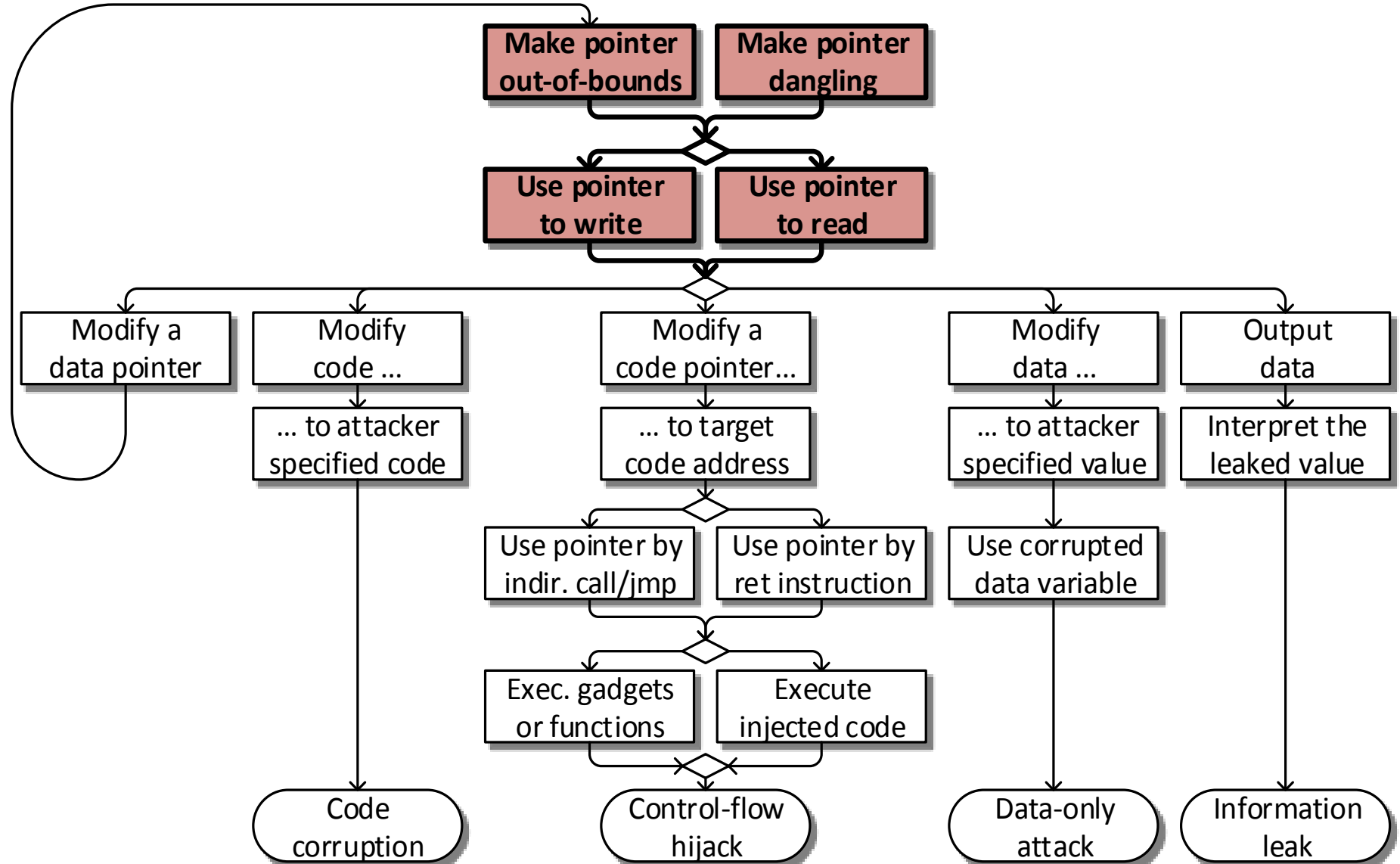
ID

```
system() {  
    ...  
}
```

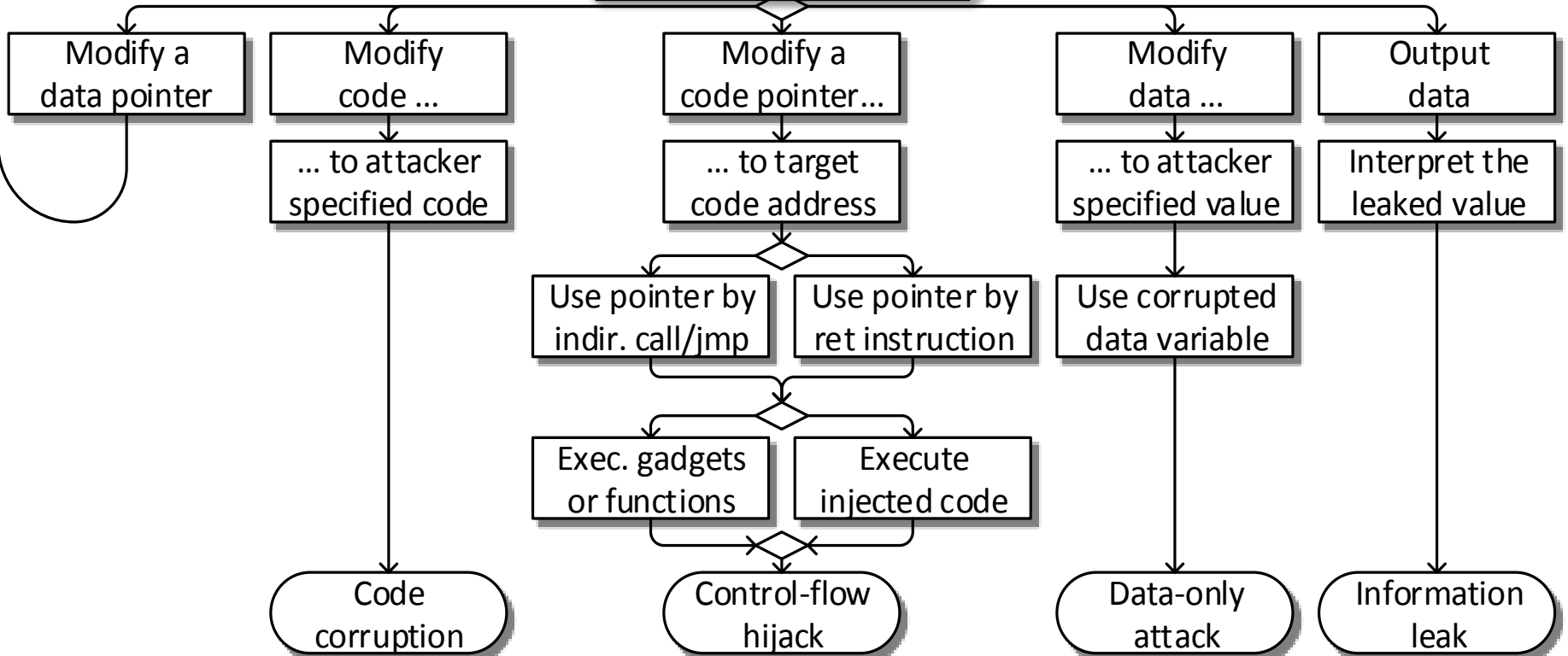
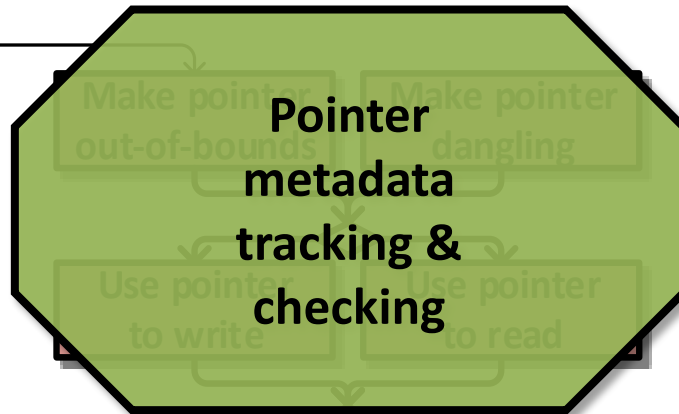
CFI

	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	W \oplus R	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good
	Control-flow integ.	CFI	Over-approx.	1.4x	Libraries

Memory safety



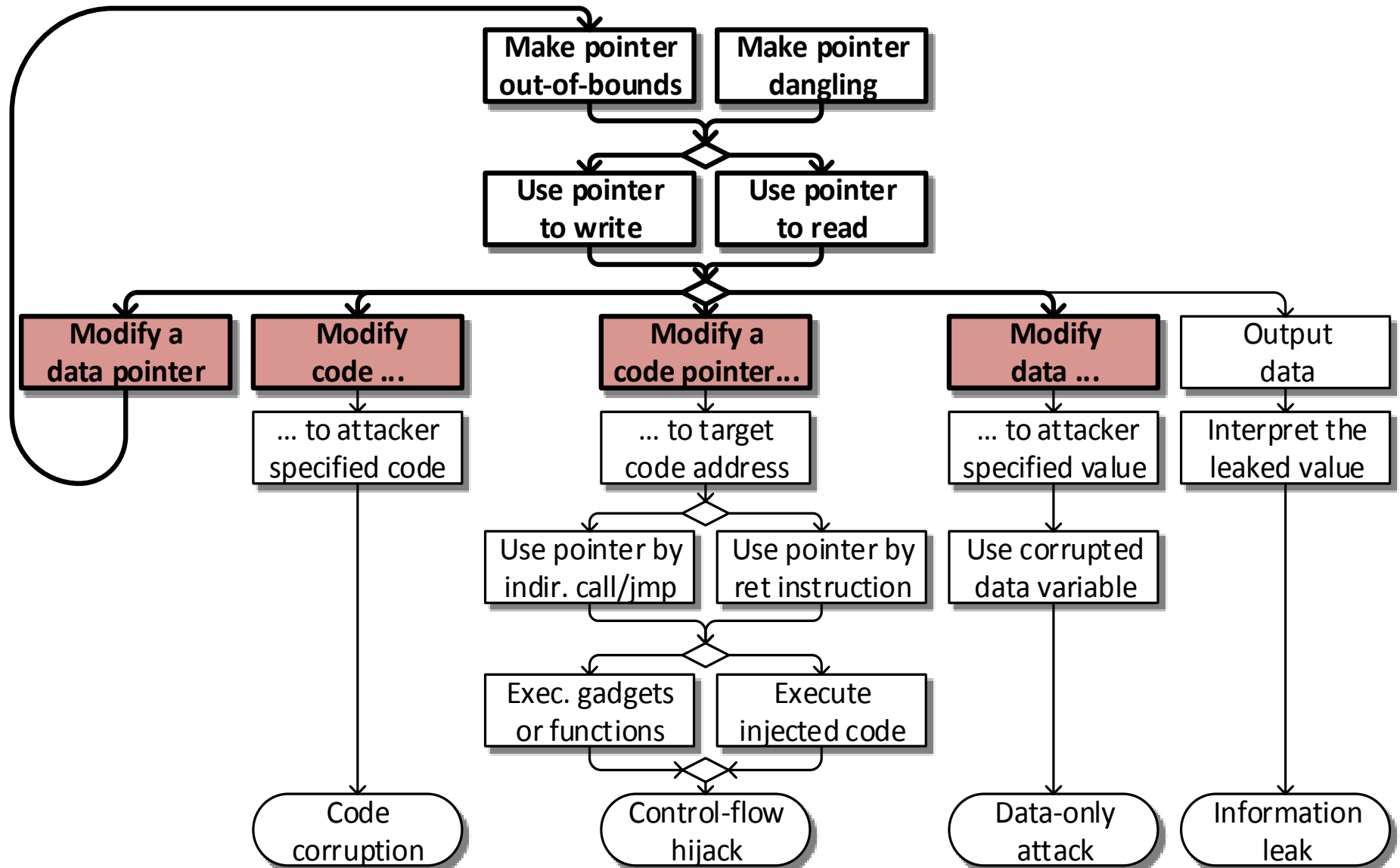
Memory safety



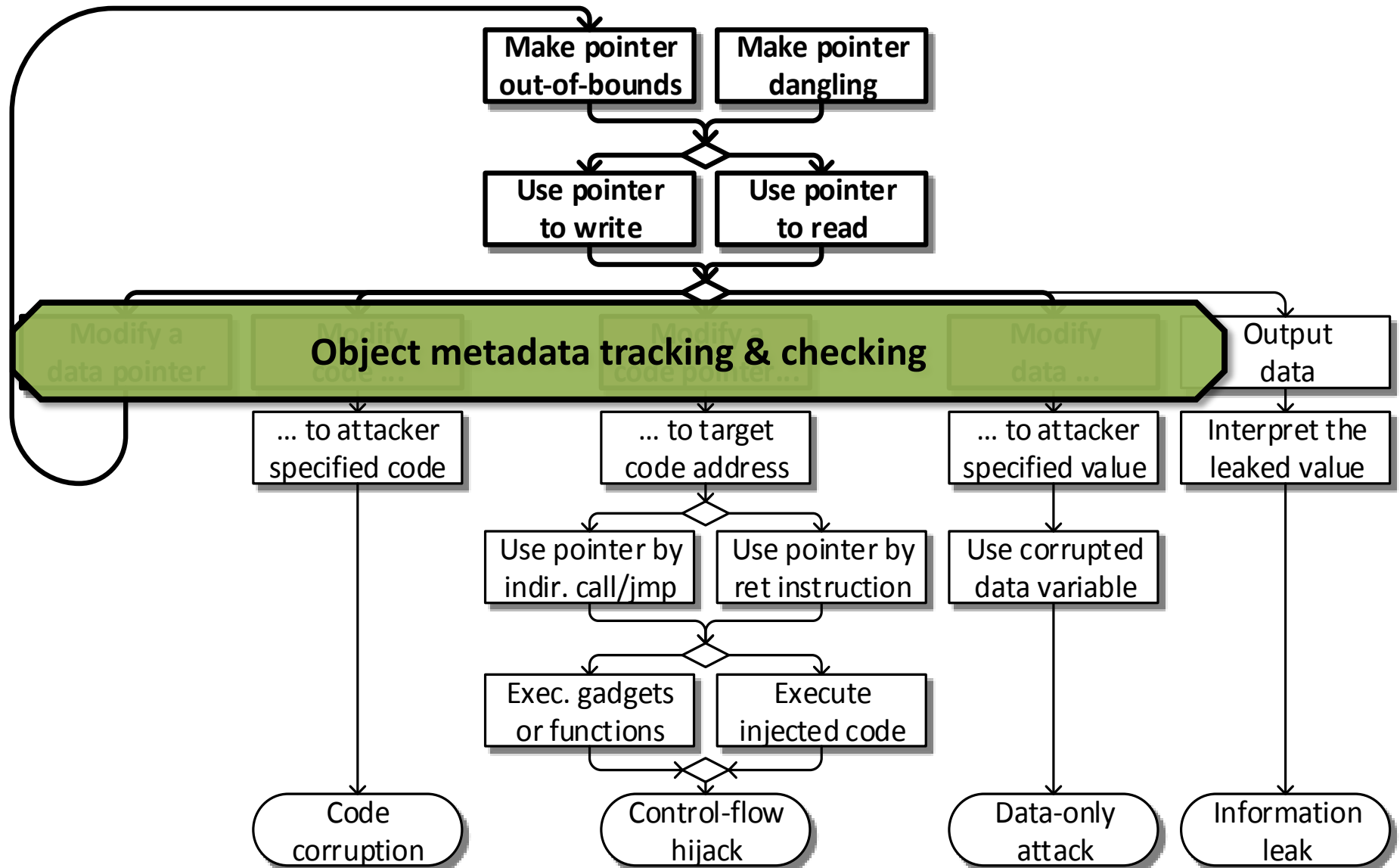
SoftBounds+CETS

	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	$W \oplus R$	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good
	Control-flow integrity	CFI	Over-approx.	1.4x	Libraries
Generic protection	Memory safety	SB+CETS	None	2-4x	Good


Data integrity




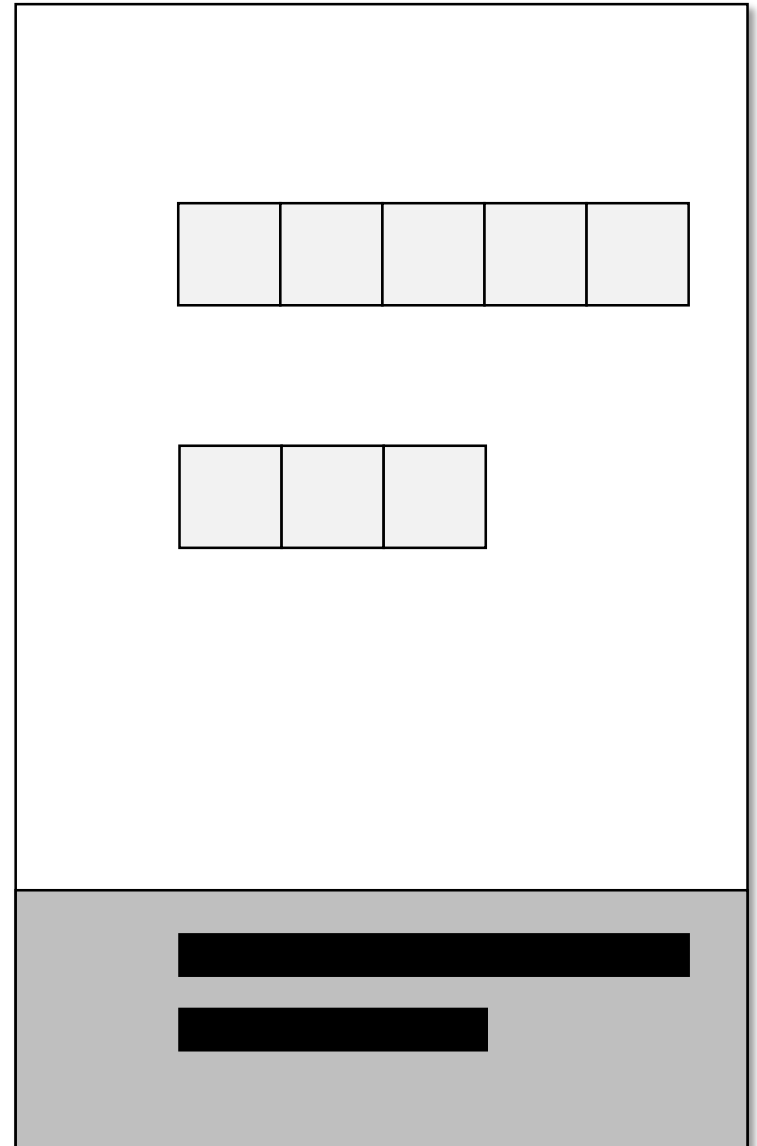
Data integrity



Valgrind / ASAN

check  } `p = alloc(5)`
`p[0] = v`

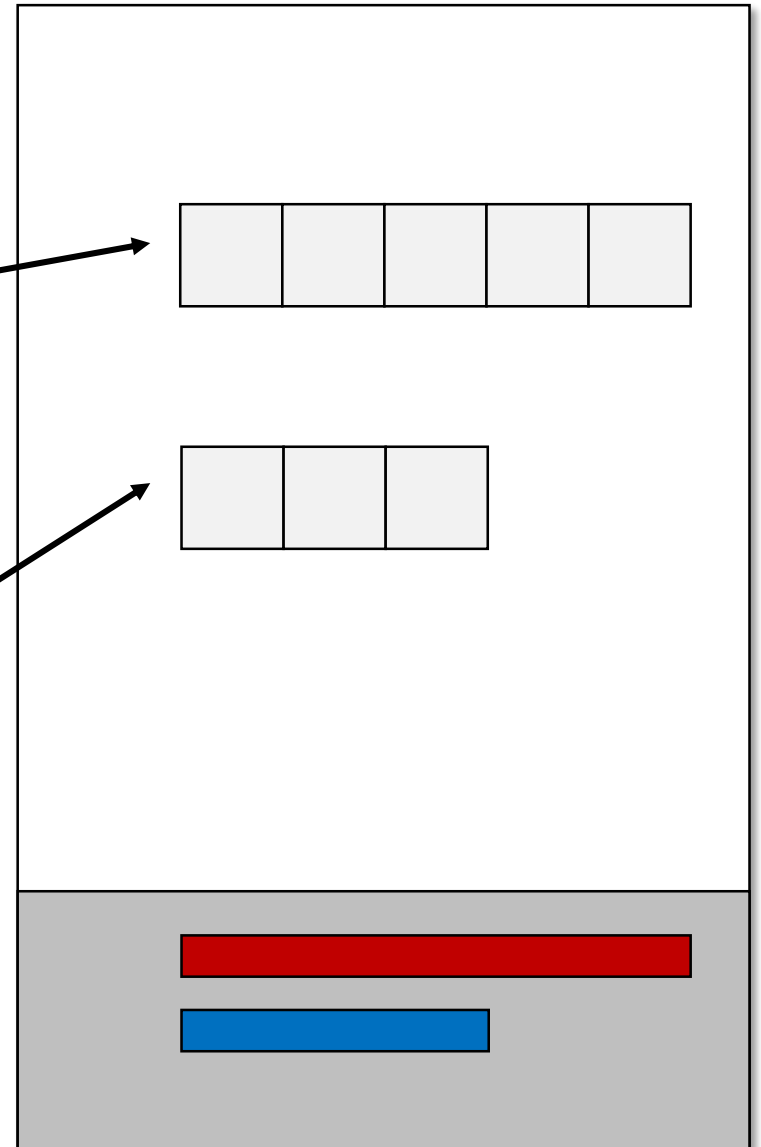
check  } `int q[3]`
`v = q[1]`



WIT

check ID } `p = alloc(5)`
`p[0] = v`

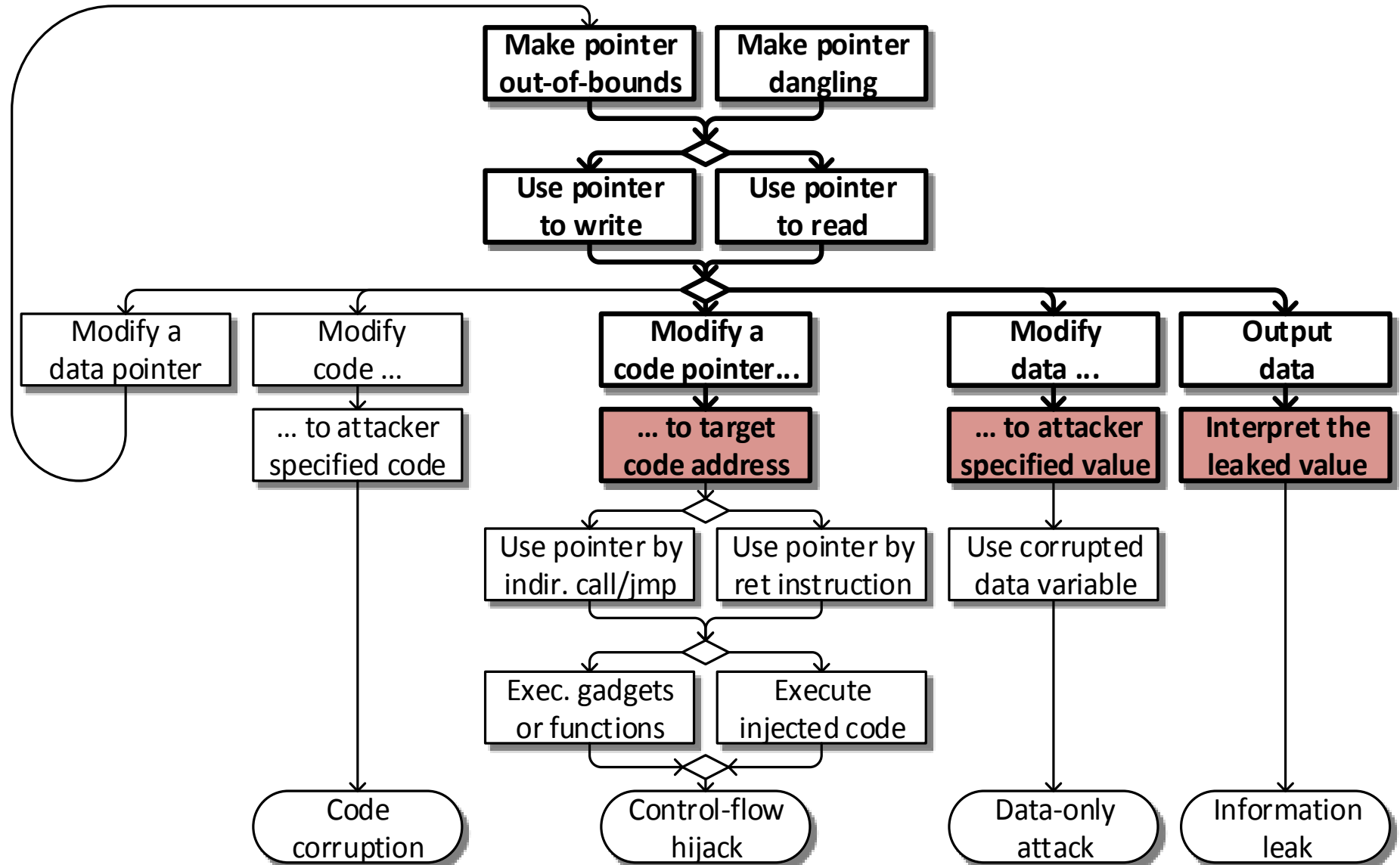
check ID } `int q[3]`
`v = q[1]`



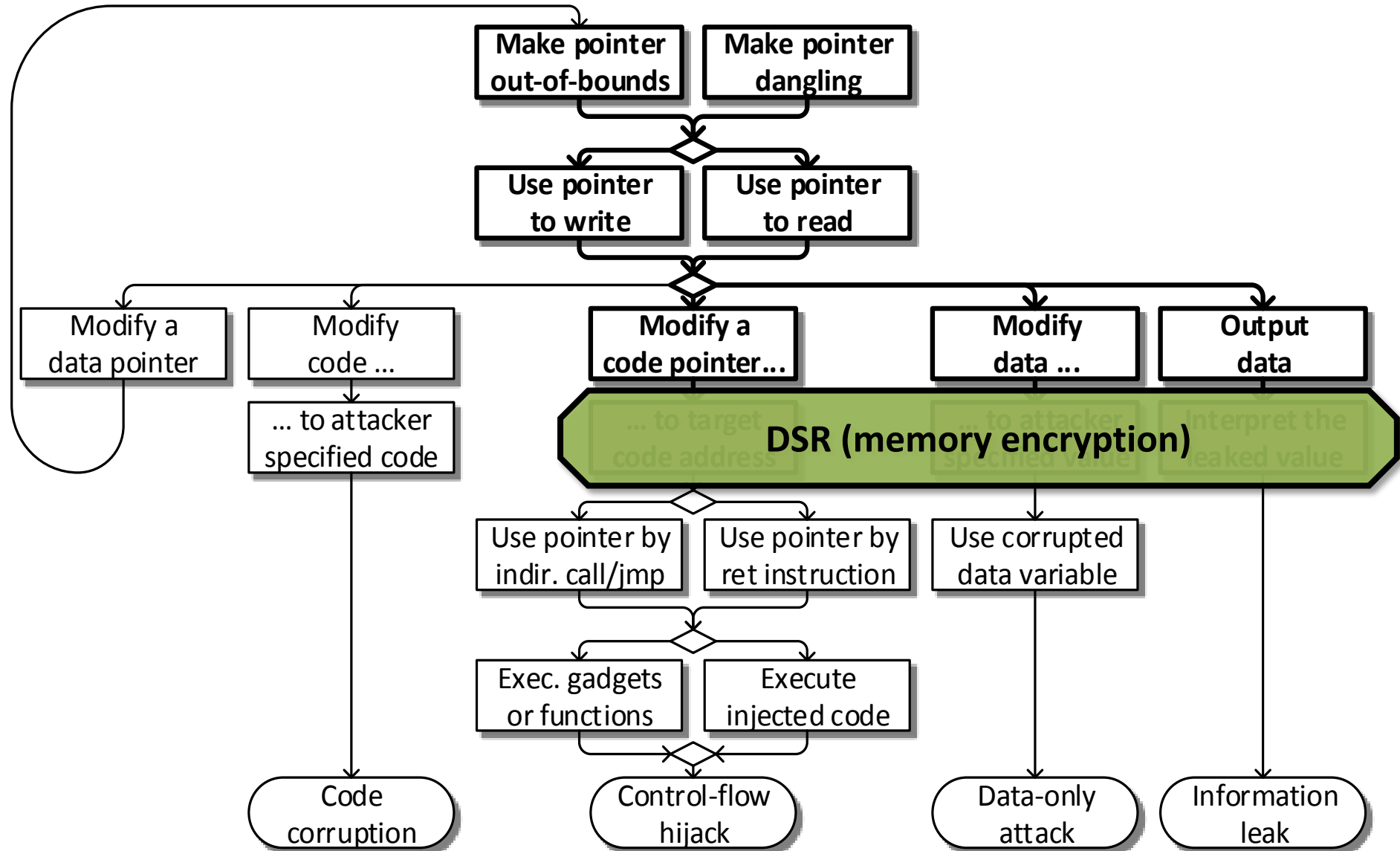
WIT

	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	W \oplus R	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good
	Control-flow integrity	CFI	Over-approx.	1.4x	Libraries
Generic protection	Memory safety	SB+CETS	None	2-4x	Good
	Data integrity	WIT	Over-approx., Use-after-frees, Invalid reads, Sub-objects	1.2x	Libraries

Data space randomization



Data space randomization

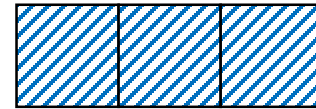


DSR

$v = v \oplus \boxed{K}$ } `p = alloc(5)`
`p[0] = v`



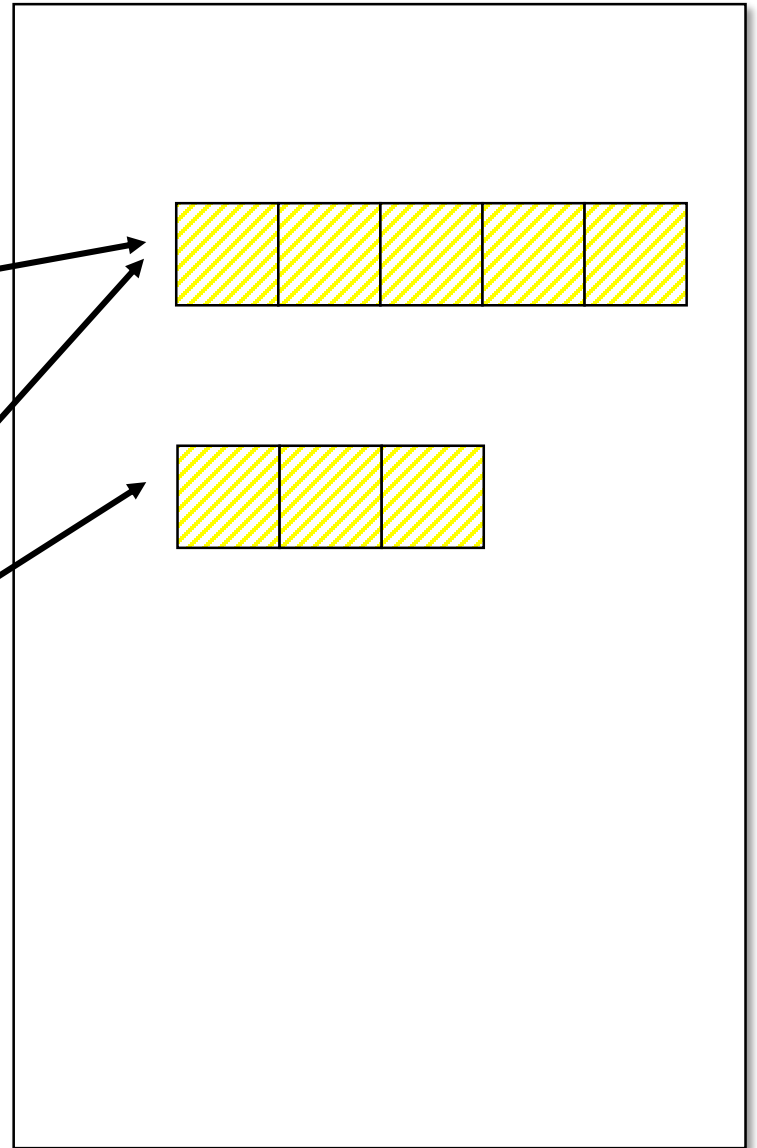
$v = v \oplus \boxed{K}$ } `int q[3]`
`v = q[1]`



DSR

$v = v \oplus \boxed{\text{K}}$ } `p = alloc(5)`
`p[0] = v`

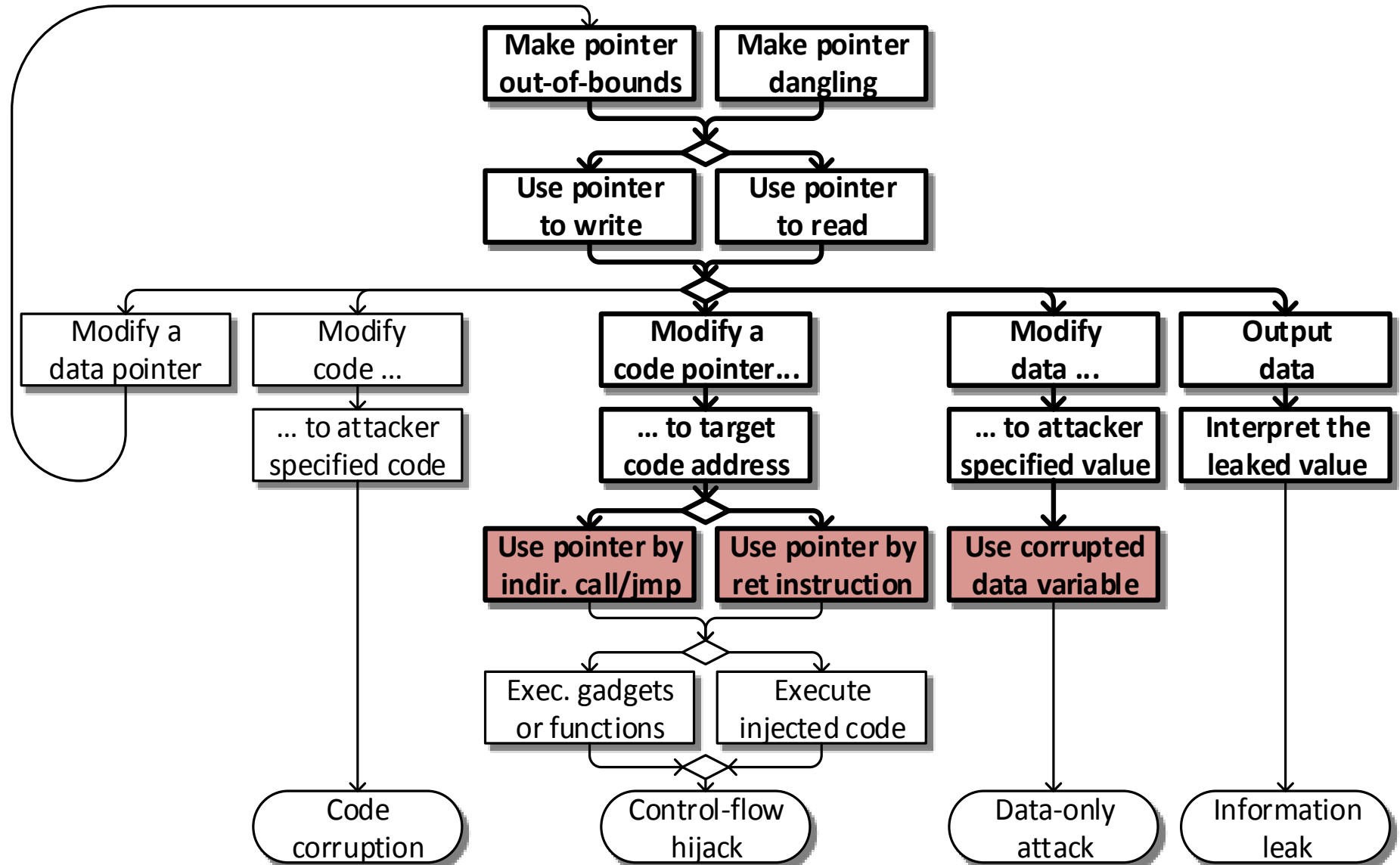
$v = v \oplus \boxed{\text{K}}$ } `int q[3]`
`v = q[1]`



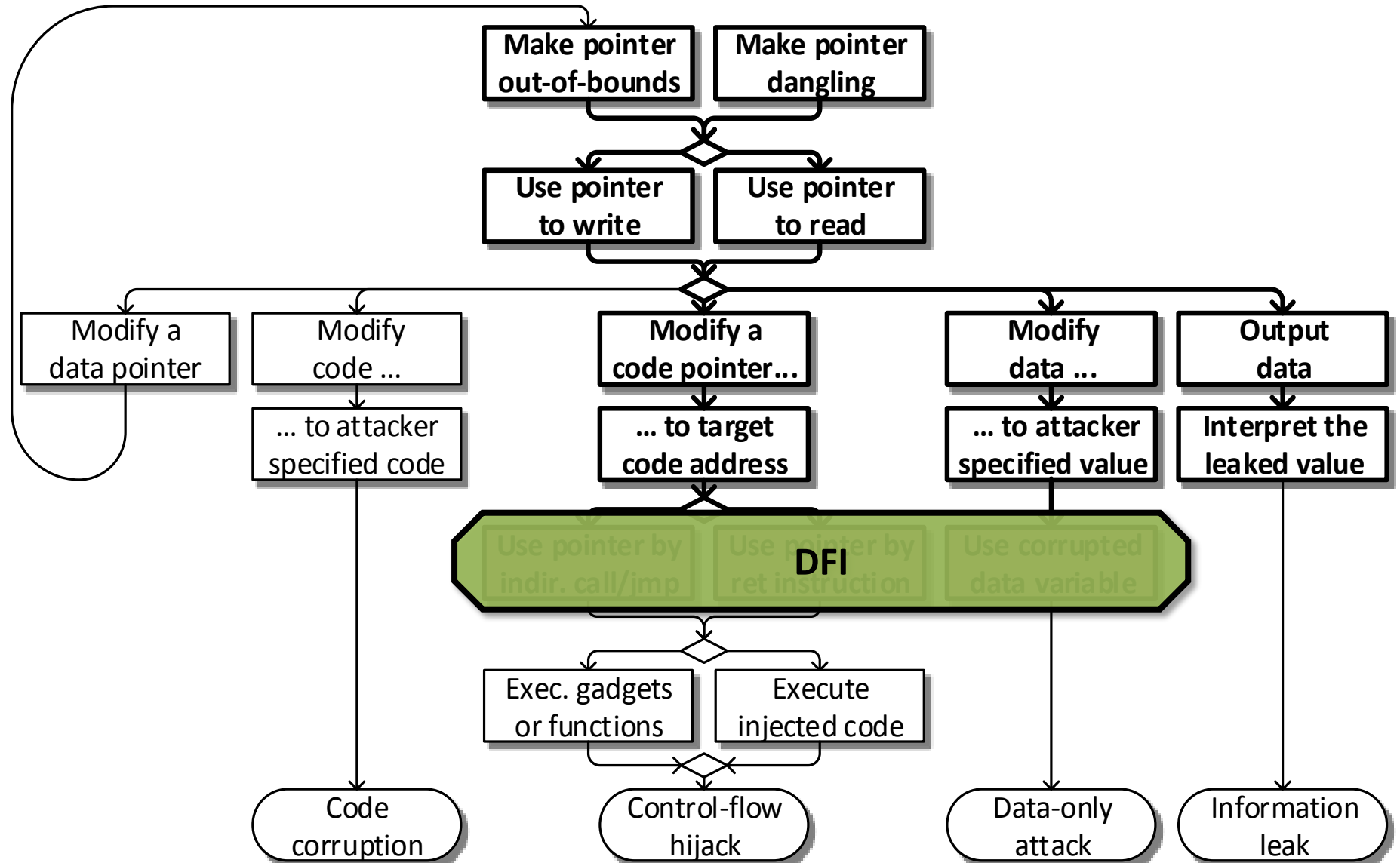
DSR

	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	W \oplus R	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good
	Control-flow integrity	CFI	Over-approx.	1.4x	Libraries
Generic protection	Memory safety	SB+CETS	None	2-4x	Good
	Data integrity	WIT	Over-approx.,...	1.2x	Libraries
	Data space rand.	DSR	Over-approx., Info-leak	1.3x	Libraries

Data-flow integrity



Data-flow integrity



DFI

	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	$W \oplus R$	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good
	Control-flow integrity	CFI	Over-approx.	1.4x	Libraries
Generic protection	Memory safety	SB+CETS	None	2-4x	Good
	Data integrity	WIT	Over-approx.,...	1.2x	Libraries
	Data space rand.	DSR	Over-approx.,...	1.3x	Libraries
	Data-flow integrity	DFI	Over-approx.	2-3x	Libraries

Summary

	Policy	Technique	Weakness	Perf.	Comp.
Hijack protection	W \oplus R	Page flags	JIT	1x	Good
	Return integrity	Stack cookies	Direct overwrite	1x	Good
	Address space rand.	ASLR	Info-leak.	1.1x	Good
	Control-flow integ.	CFI	Over-approx.	1.4x	Libraries
Generic protection	Memory safety	SB+CETS	None	2-4x	Good
	Data integrity	WIT	Over-approx.,...	1.2x	Libraries
	Data space rand.	DSR	Over-approx.,...	1.3x	Libraries
	Data-flow integrity	DFI	Over-approx.	2-3x	Libraries

Questions