ConFIRM: Evaluating Compatibility and Relevance of Control-flow Integrity Protections for Modern Software

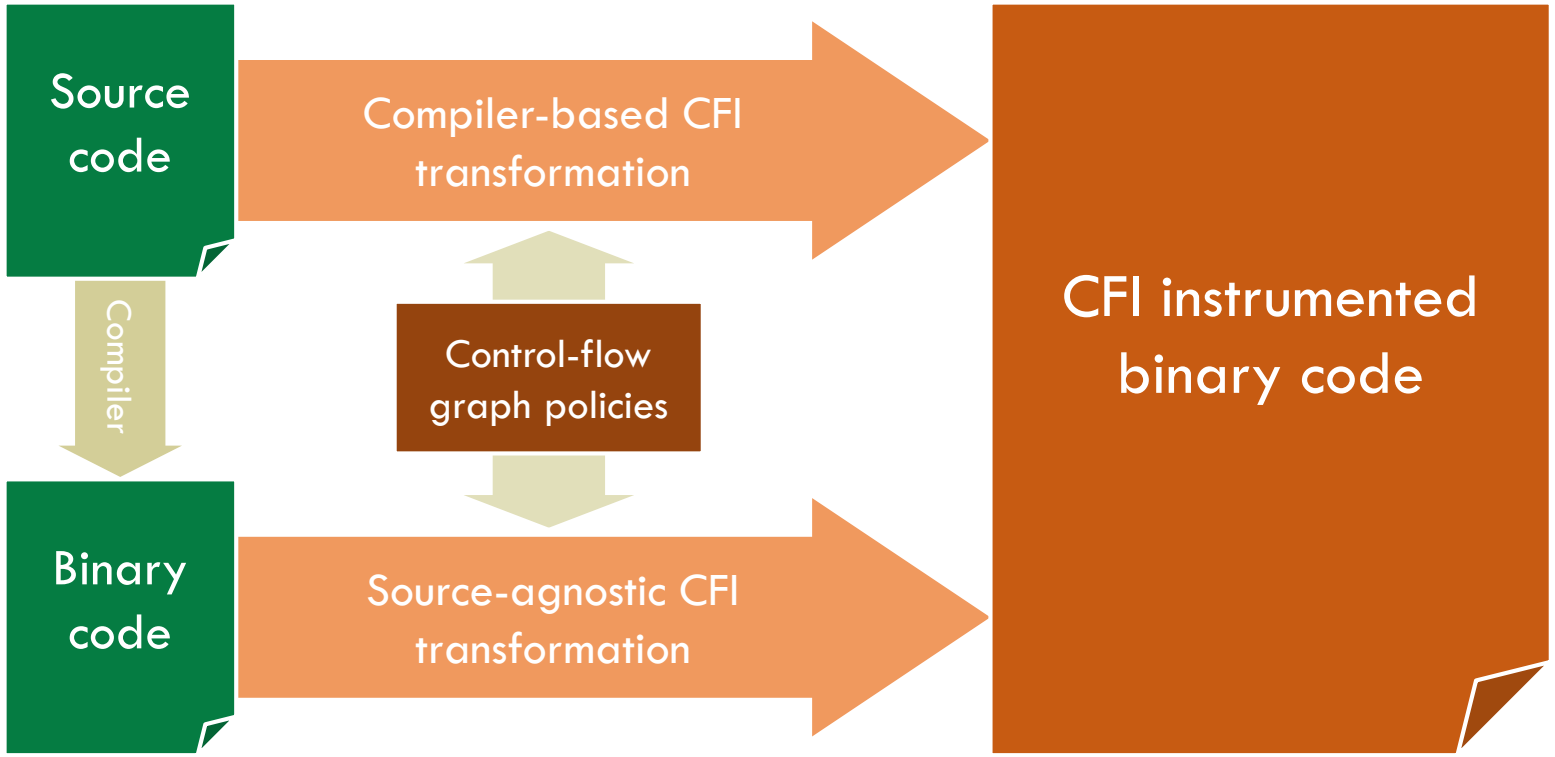
Xiaoyang Xu, Masoud Ghaffarinia,
Wenhao Wang, and Kevin W. Hamlen
The University of Texas at Dallas

Zhiqiang Lin
The Ohio State University

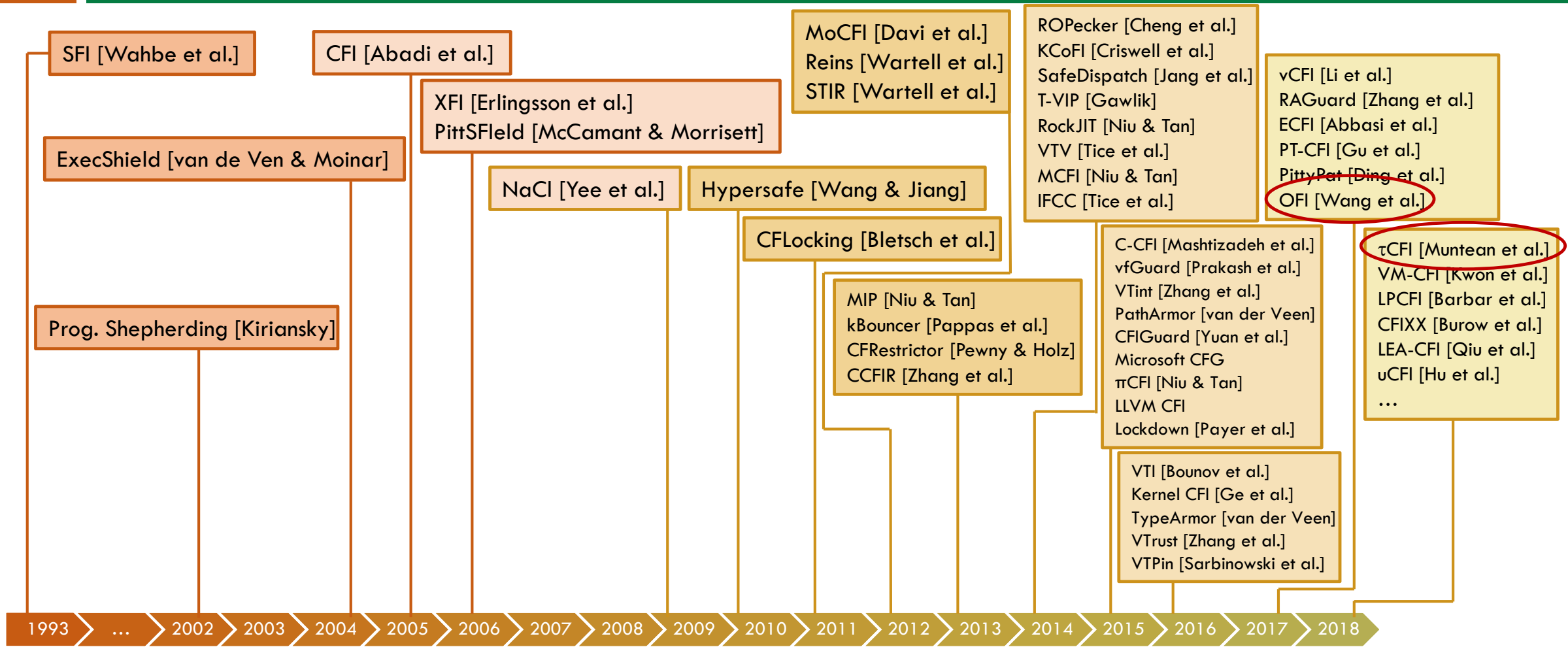
Supported in part by:
ONR award N00014-17-2995,
DARPA award FA8750-19- C-0006,
NSF awards #1513704 and #1834215,
and an NSF I/UCRC Award from Lockheed Martin

Control-Flow Integrity

[M. Abadi, M. Budiu, Ú. Erlingsson, and J. Ligatti; CCS'05.]

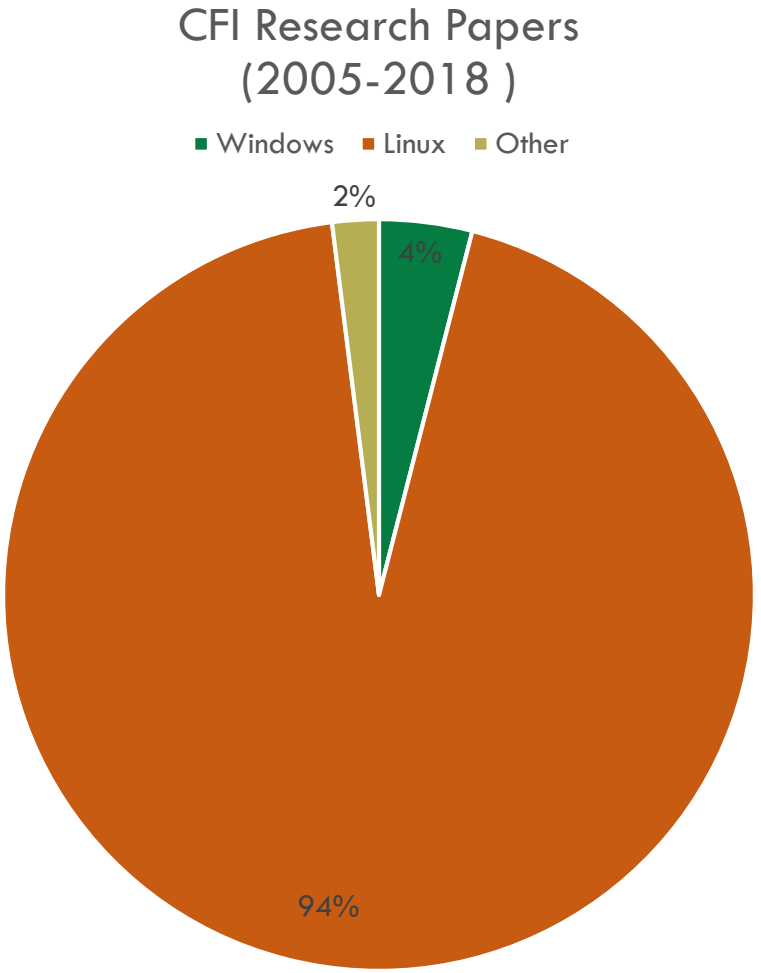


CFI Research Timeline



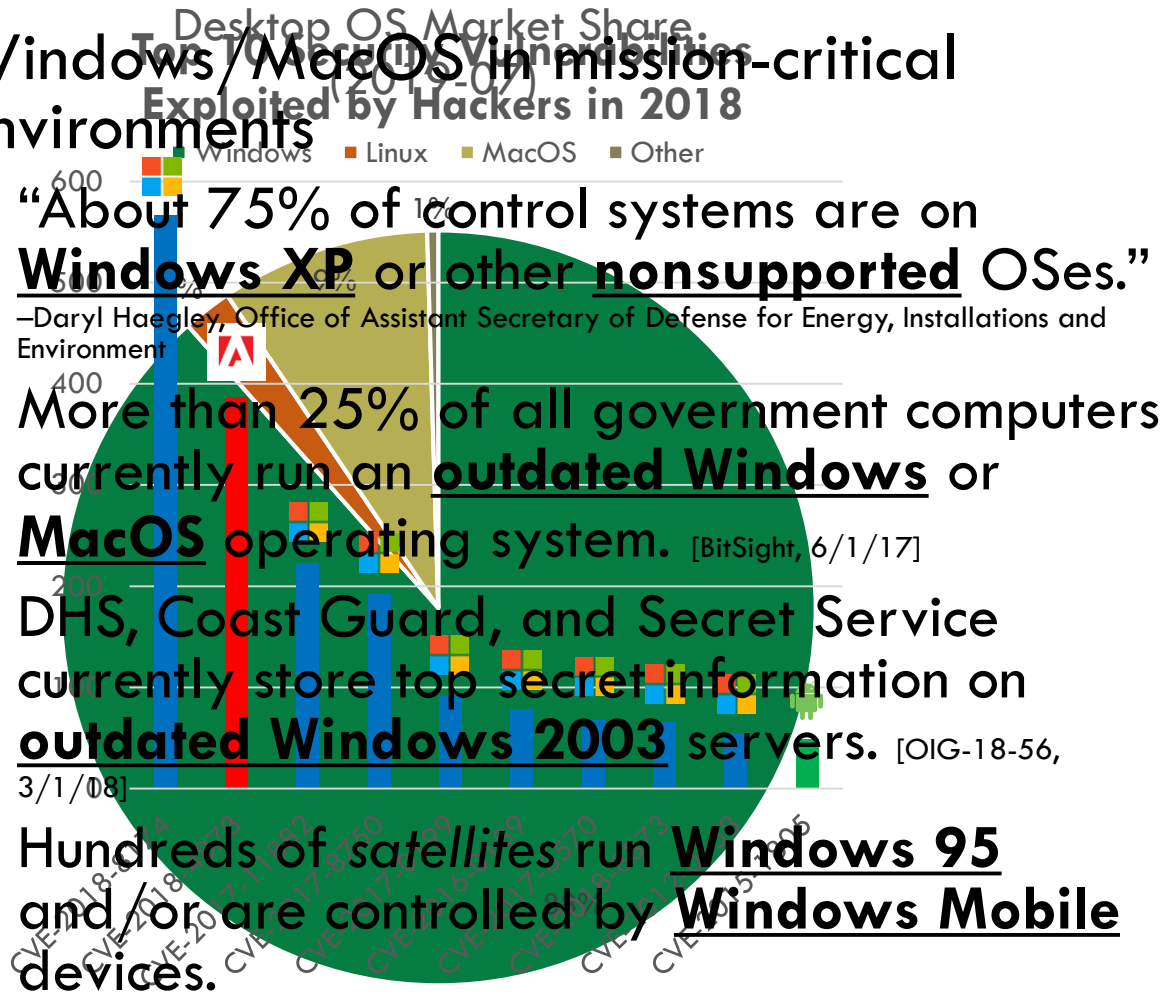
CFI: Precision, Security, and Performance *[Burow et al., CSUR'17]*

Scalability Gap



*Papers containing at least one experiment where at least one **COMPLETE** non-benchmark application for the indicated OS was rewritten & secured

- Windows/MacOS in mission-critical environments
  - "About 75% of control systems are on **Windows XP** or other **nonsupported** OSes."
    –Daryl Haegley, Office of Assistant Secretary of Defense for Energy, Installations and Environment
  - More than 25% of all government computers currently run an **outdated Windows** or **MacOS** operating system. [BitSight, 6/1/17]
  - DHS, Coast Guard, and Secret Service currently store top secret information on **outdated Windows 2003** servers. [OIG-18-56, 3/1/18]
  - Hundreds of *satellites* run **Windows 95** and/or are controlled by **Windows Mobile** devices.



ConFIRM (CONtrol-Flow Integrity Relevance Metrics)

Problems

- Compatibility of CFI solutions are under-studied
- CFI implementations are commonly evaluated in terms of performance and security
- CPU benchmarks are widely adopted for CFI evaluation

Our solution: ConFIRM

- A set of 23 synthetic tests for CFI compatibility problems identified
- A testing suite designed specifically for CFI evaluation
- Reevaluation of 12 CFI implementations
  - These CFI implementations pass 53% of ConFIRM's compatibility and security tests
- Correlation with CPU benchmarks

https://github.com/SoftwareLanguagesSecurityLab/Confirm

20 Widespread Classes of CFI Compatibility Problems

Compatibility Problem	Real-world Software Examples
Function Pointers	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Callbacks	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Dynamic Linking	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Delay-Loading	Adobe Reader, Calculator, Chrome, Firefox, JVM, MS Paint, MS Powerpoint, …
Exporting/Importing Data Symbols	7-Zip, Apache, Calculator, Chrome, Dropbox, Firefox, MS Paint, MS Powerpoint, …
Virtual Functions	7-Zip, Adobe Reader, Calculator, Chrome, Dropbox, Firefox, JVM, Notepad, …
Writable Vtables	programs with UI's based on GTK+ (Linux) or COM (Windows)
Tail Calls	programs compiled with tail-call optimization (e.g., -O2 or /O2)
Switch-Case Statements	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Returns	almost every benign program
Unmatched Call/Return Pairs	Adobe Reader, Apache, Chrome, Firefox, JVM, MS PowerPoint, Visual Studio, …
Exceptions	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Calling Conventions	almost every program has functions
Multithreading	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
TLS Callbacks	Adobe Reader, Chrome, Firefox, MS Paint, TeXstudio, UPX
Position-Independent Code	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Memory Management	7-Zip, Adobe Reader, Apache, Chrome, Dropbox, Firefox, MS PowerPoint, …
JIT Code	Adobe Flash, Chrome, Dropbox, Firefox, JVM, MS PowerPoint, PotPlayer, …
Self-Unpacking	programs decompressed by self-extractors (e.g., UPX, NSIS)
Runtime API Hooking	Microsoft Office, including MS Excel, MS PowerPoint, etc.

ConFIRM: Control-Flow Integrity Relevance Metrics

	LLVM (Windows)					LLVM (Linux)							
Test	**CFI**	**ShadowStack**	**MCFG**	**OFI**	**Reins**	**GCC-VTV**	**CFI**	**ShadowStack**	**MCFI**	**πCFI**	**πCFI (nto)**	**PathArmor**	**Lockdown**
fptr	6.35%	⚠	20.13%	4.35%	4.08%	⚠	6.97%	⚠	✗	-14.00%	-13.79%	⚠	174.92%
callback	⚠	⚠	⚠	128.39%	114.84%	⚠	⚠	⚠	✗	✗	✗	⚠	✗
load_time_dynlnk	2.74%	⚠	8.83%	3.36%	2.66%	⚠	1.33%	⚠	30.83%	31.52%	34.05%	74.54%	1.45%
run_time_dynlnk	⚠	⚠	17.63%	12.57%	11.48%	⚠	4.44%	⚠	✗	✗	✗	1,221.48%	✗
delay_load	N/A	N/A	8.16%	3.61%	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
data_symbl	✓	⚠	✓	✓	✗	✓	✓	⚠	✓	✓	✓	✓	✓
vtbl_call	5.62%	⚠	27.71%	35.94%	31.17%	33.56%	5.94%	⚠	✗	-8.19%	-9.31%	⚠	227.82%
code_coop	⚠	⚠	⚠	✓	✗	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
tail_call	6.17%	⚠	9.51%	0.05%	0.05%	⚠	6.82%	⚠	✗	-17.69%	-17.37%	⚠	178.06%
switch	-5.80%	⚠	3.51%	22.82%	17.69%	⚠	-6.93%	⚠	-29.01%	-27.19%	-28.46%	⚠	85.85%
ret	⚠	18.04%	⚠	49.34%	48.49%	⚠	⚠	20.88%	70.72%	72.40%	71.52%	⚠	106.71%
unmatched_pair	⚠	⚠	⚠	✓	✓	⚠	⚠	⚠	✓	✓	✓	⚠	⚠
signal	✓	⚠	✓	✗	✗	✓	✓	⚠	✓	✓	✓	✗	✓
cppeh	✓	⚠	✓	✓	✗	✓	✓	⚠	✓	✓	✓	✗	✓
seh	✓	⚠	✓	✓	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
veh	⚠	⚠	⚠	✓	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
convention	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
multithreading	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
tls_callback	N/A	N/A	N/A	✓	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
pic	✓	✓	✓	⚠	⚠	✓	✓	✓	✓	✓	✓	✓	✓
mem	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	✗	✗	✗	✓	✗
jit	⚠	⚠	⚠	✗	✗	⚠	⚠	⚠	✗	✗	✗	⚠	✗
unpacking	N/A	N/A	N/A	✗	✗	N/A	N/A	N/A	N/A	N/A	N/A	✗	✗
api_hook	⚠	⚠	⚠	✗	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

A Compatibility Problem Example — Returns

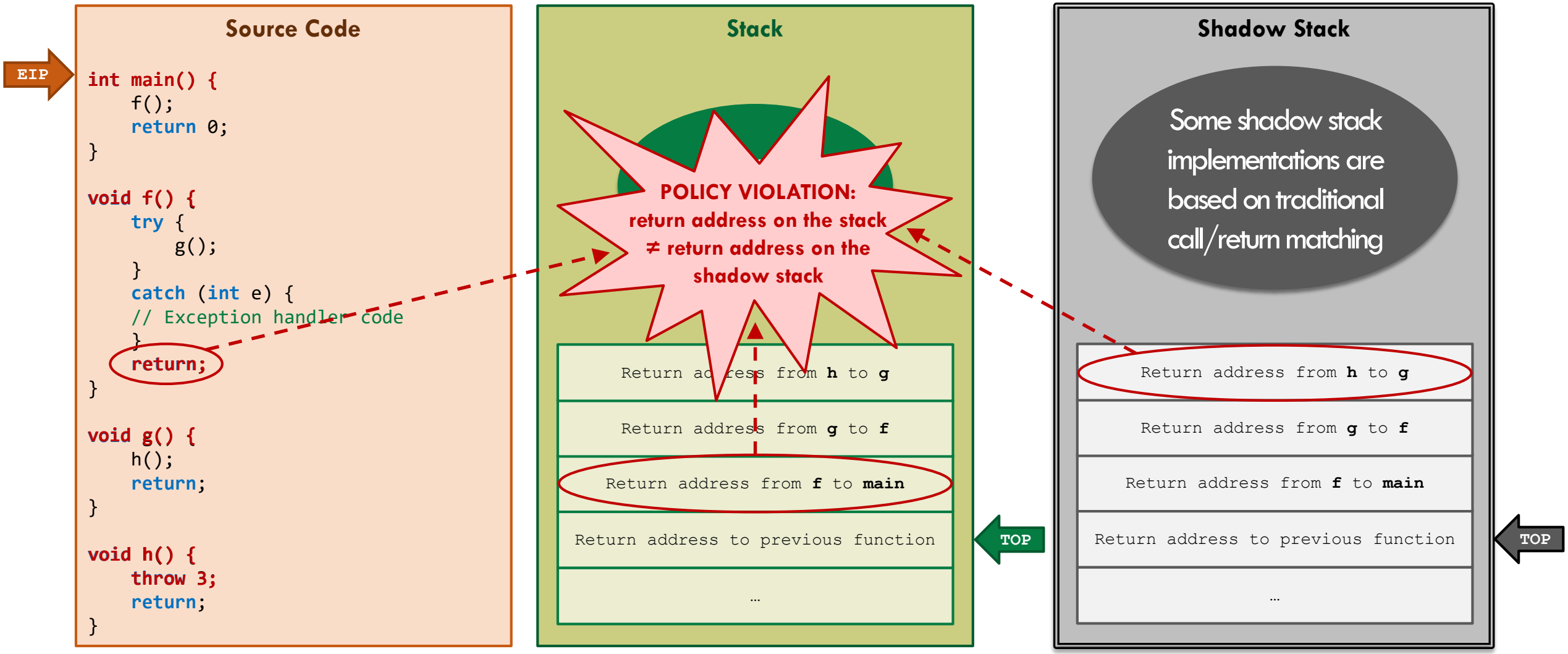
Source Code

```
1   void authenticate() {
2       …
3       f();
4       authenticated = 1;
5       …
6   }
7   void print_prompt() {
8       …
9       f();
10      …
11  }
12  void f() {
13      …
14      return;
15  }
```

CFI Hardened Assembly Code

```
1   _authenticate:
2   …
3   call _f
4   mov [authenticated], 1
5   …
6   _print_prompt:
7   …
8   call _f
9   …
10  _f:
11  …
12  ret
13      jmp security_abort
14
```

Another Compatibility Problem Example — Unmatched Call/Return Pairs



20 Widespread Classes of CFI Compatibility Problems

Compatibility Metric	Real-world Software Examples
Function Pointers	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Callbacks	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Dynamic Linking	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Delay-Loading	Adobe Reader, Calculator, Chrome, Firefox, JVM, MS Paint, MS Powerpoint, …
Exporting/Importing Data Symbols	7-Zip, Apache, Calculator, Chrome, Dropbox, Firefox, MS Paint, MS Powerpoint, …
Virtual Functions	7-Zip, Adobe Reader, Calculator, Chrome, Dropbox, Firefox, JVM, Notepad, …
Writable Vtables	programs with UI's based on GTK+ (Linux) or COM (Windows)
Tail Calls	programs compiled with tail-call optimization (e.g., -O2 or /O2)
Switch-Case Statements	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Returns	almost every benign program
Unmatched Call/Return Pairs	Adobe Reader, Apache, Chrome, Firefox, JVM, MS PowerPoint, Visual Studio, …
Exceptions	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Calling Conventions	almost every program has functions
Multithreading	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
TLS Callbacks	Adobe Reader, Chrome, Firefox, MS Paint, TeXstudio, UPX
Position-Independent Code	7-Zip, Adobe Reader, Apache, Calculator, Chrome, Dropbox, Firefox, JVM, …
Memory Management	7-Zip, Adobe Reader, Apache, Chrome, Dropbox, Firefox, MS PowerPoint, …
JIT Code	Adobe Flash, Chrome, Dropbox, Firefox, JVM, MS PowerPoint, PotPlayer, …
Self-Unpacking	programs decompressed by self-extractors (e.g., UPX, NSIS)
Runtime API Hooking	Microsoft Office, including MS Excel, MS PowerPoint, etc.

Cross-Thread Stack-Smashing Attack

Thread 1 (malicious)

```
1  while (1) {
2        // smash thread 2's
3        // return address
4        *p = 0xDEADBEEF
5  }
```

Thread 2 (CFI instrumented)

```
1  _f:
2  …
3  if (!is_valid_target([esp]))
4     jmp security_abort
5  ret
```

TOCTOU window

ConFIRM: Control-Flow Integrity Relevance Metrics

	LLVM (Windows)					LLVM (Linux)							
Test	**CFI**	**ShadowStack**	**MCFG**	**OFI**	**Reins**	**GCC-VTV**	**CFI**	**ShadowStack**	**MCFI**	**πCFI**	**πCFI (nto)**	**PathArmor**	**Lockdown**
fptr	6.35%	⚠	20.13%	4.35%	4.08%	⚠	6.97%	⚠	✗	-14.00%	-13.79%	⚠	174.92%
callback	⚠	⚠	⚠	128.39%	114.84%	⚠	⚠	⚠	✗	✗	✗	⚠	✗
load_time_dynlnk	2.74%	⚠	8.83%	3.36%	2.66%	⚠	1.33%	⚠	30.83%	31.52%	34.05%	74.54%	1.45%
run_time_dynlnk	⚠	⚠	17.63%	12.57%	11.48%	⚠	4.44%	⚠	✗	✗	✗	1,221.48%	✗
delay_load	N/A	N/A	8.16%	3.61%	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
data_symbl	✓	⚠	✓	✓	✗	✓	✓	⚠	✓	✓	✓	✓	✓
vtbl_call	5.62%	⚠	27.71%	35.94%	31.17%	33.56%	5.94%	⚠	✗	-8.19%	-9.31%	⚠	227.82%
code_coop	⚠	⚠	⚠	✓	✗	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
tail_call	6.17%	⚠	9.51%	0.05%	0.05%	⚠	6.82%	⚠	✗	-17.69%	-17.37%	⚠	178.06%
switch	-5.80%	⚠	3.51%	22.82%	17.69%	⚠	-6.93%	⚠	-29.01%	-27.19%	-28.46%	⚠	85.85%
ret	⚠	18.04%	⚠	49.34%	48.49%	⚠	⚠	20.88%	70.72%	72.40%	71.52%	⚠	106.71%
unmatched_pair	⚠	⚠	⚠	✓	✓	⚠	⚠	⚠	✓	✓	✓	⚠	⚠
signal	✓	⚠	✓	✗	✗	✓	✓	⚠	✓	✓	✓	✗	✓
cppeh	✓	⚠	✓	✓	✗	✓	✓	⚠	✓	✓	✓	✗	✓
seh	✓	⚠	✓	✓	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
veh	⚠	⚠	⚠	✓	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
convention	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
multithreading	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
tls_callback	N/A	N/A	N/A	✓	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
pic	✓	✓	✓	⚠	⚠	✓	✓	✓	✓	✓	✓	✓	✓
mem	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠	✗	✗	✗	✓	✗
jit	⚠	⚠	⚠	✗	✗	⚠	⚠	⚠	✗	✗	✗	⚠	✗
unpacking	N/A	N/A	N/A	✗	✗	N/A	N/A	N/A	N/A	N/A	N/A	✗	✗
api_hook	⚠	⚠	⚠	✗	✗	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

CFI Performance Measurement Problems

SPEC CPU Benchmark	CFI Solution									Benchmark Correlation
	MCFG	Reins	GCC-VTV	LLVM-CFI	MCFI	πCFI	πCFI (nto)	PathArmor	Lockdown	
perlbench				2.4	5.0	5.0	5.3	15.0	150.0	**0.09**
bzip2	−0.3	9.2		−0.7	1.0	1.0	0.8	0.0	8.0	**−0.12**
gcc					4.5	4.5	10.5	9.0	50.0	**0.02**
mcf	0.5	9.1		3.6	4.5	4.5	1.8	1.0	2.0	**−0.39**
gobmk	−0.2			0.2	7.0	7.5	11.8	0.0	43.0	**−0.09**
hmmer	0.7			0.1	0.0	0.0	−0.1	1.0	3.0	**0.33**
sjeng	3.4			1.6	5.0	5.0	11.9	0.0	80.0	**−0.03**
h264ref	5.4			5.3	6.0	6.0	8.3	1.0	43.0	**−0.09**
libquantum				−6.9	0.0	−0.3	−1.0	3.0	5.0	**0.51**
omnetpp	3.8		5.8		5.0	5.0	18.8			**−0.52**
astar	0.1		3.6	0.9	3.5	4.0	2.9		17.0	**0.92**
xalancbmk	5.5		24.0	7.2	7.0	7.0	17.6		118.0	**0.94**
milc	2.0			0.2	2.0	2.0	1.4	4.0	8.0	**0.40**
namd	0.1		−0.1	0.1	−0.5	−0.5	−0.5	3.0		**0.98**
dealII	−0.1		0.7	7.9	4.5	4.5	4.4			**−0.36**
soplex	2.3		0.5	−0.3	−4.0	−4.0	0.9	12.0		**0.89**
povray	10.8		−0.6	8.9	10.0	10.5	17.4		90.0	**0.88**
lbm	4.2			−0.2	1.0	1.0	−0.5	0.0	2.0	**−0.22**
sphinx3	−0.1			−0.8	1.5	1.5	2.4	3.0	8.0	**0.31**
ConFIRM median	9.51	4.59	33.56	5.19	30.83	−11.10	−11.60	648.01	140.82	**0.36**

Conclusions

- Compatibility of CFI solutions are under-studied
  - Complicated compatibility problems lurking in large COTS software products
- CFI implementations are commonly evaluated in terms of performance and security using CPU benchmarks.
- Proposed solution: CONFIRM
  - A set of 20 CFI-relevant compatibility problems
  - The first testing suite designed specifically for CFI solution evaluation
  - Reevaluation of 12 CFI implementations
  - Correlation with SPEC CPU benchmarks
  - https://github.com/SoftwareLanguagesSecurityLab/Confirm

THANK YOU

https://github.com/SoftwareLanguagesSecurityLab/Confirm

Supported in part by:
ONR award N00014-17-2995,
DARPA award FA8750-19- C-0006,
NSF awards #1513704 and #1834215,
and an NSF I/UCRC Award from Lockheed Martin