Guarded Control-Flow and Data Privacy for Sensitive Data

MATHIAS PAYER, PURDUE UNIVERSITY

Cybersecurity Research Acceleration Workshop and Showcase, Indianapolis, IN
October 11, 2017

hexhive

Lockdown: Guarded Control-Flow and Data-Privacy for Sensitive Data

Mathias Payer, Purdue University
http://hexhive.github.io

Cybersecurity Research Acceleration Workshop and Showcase

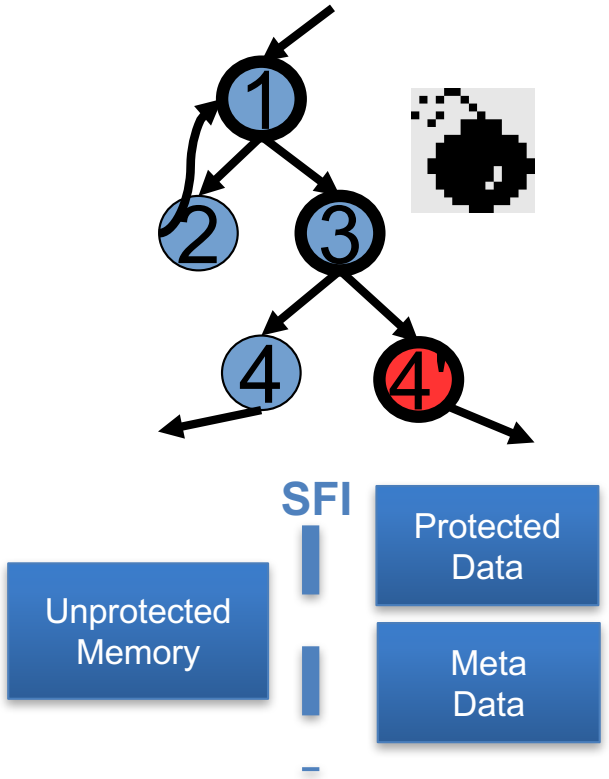
October 11, 2017 | Indianapolis, IN

Quad Chart for: Guarded Control-Flow and Data Privacy for Sensitive Data

Challenge:
Applications are written in low level languages such as C/C++ and prone to vulnerabilities. Complete mitigations result in prohibitive performance overhead.

Solution:

- Develop fine-grained policies to guard control flow at all times
- Develop selective policy to protect sensitive data only
- Compiler-based analysis allows reasoning about types
- Compartmentalize, apply different data policies depending on sensitivity



NSF CNS-1464155
Purdue University

PI: Mathias Payer

Value proposition:

- Increase public outreach and interaction with community
- Educate developers about security policies, develop defaults
- Build full products, not just research prototypes

What we need to TTP

- Transitioning from research prototype to usable mitigation
- Code review and upstream into framework (e.g., LLVM)

Contact us

- mpayer@purdue.edu

Software is unsafe and insecure*

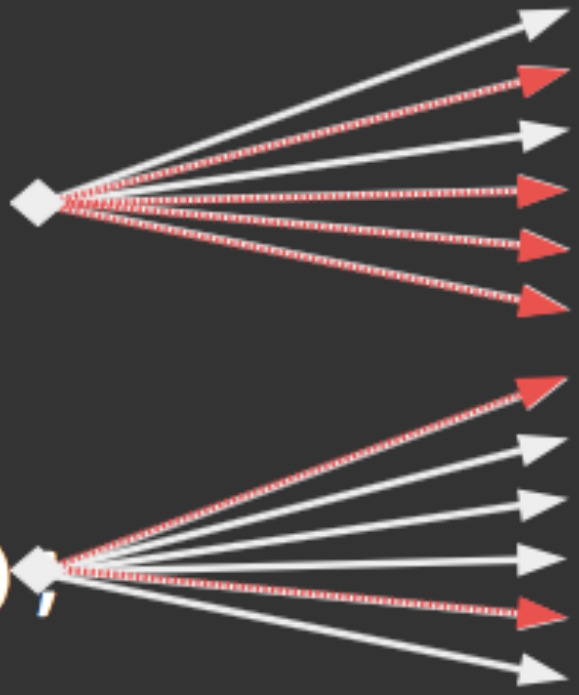
- Low-level languages (C/C++) trade type safety and memory safety for performance
  - Our systems are implemented in C/C++
  - Too many bugs to find and fix manually

Google Chrome:	**76 MLoC**
Gnome:	**8.6 MLoC**
Xorg:	**1 MLoC**
glibc:	**1.5 MLoC**
Linux kernel:	**14 MLoC**

* SoK: Eternal War in Memory. Laszlo Szekeres, Mathias Payer, Tao Wei, and Dawn Song. In IEEE S&P'13

Control-Flow Integrity (CFI)*

```
CHECK(fn);
(*fn)(x);
```



```
CHECK_RET();
return 7
```

* ***Control-Flow Integrity.*** Martin Abadi, Mihai Budiu, Ulfar Erlingsson, Jay Ligatti. CCS '05
Control-Flow Integrity: Protection, Security, and Performance. Nathan Burow, Scott A. Carr, Joseph Nash, Per Larsen, Michael Franz, Stefan Brunthaler, Mathias Payer. ACM CSUR '18, preprint: https://nebelwelt.net/publications/files/18CSUR.pdf

Data Confidentiality

- Only some data is sensitive
  - Strong protection for sensitive data
  - Loose protection for other data
- Compartmentalization is crucial
  - Annotate sensitive data (types)
  - Compiler and runtime system enforce separation

Conclusion

- Protect systems despite vulnerabilities
- Selective mitigations
  - Stack integrity, precise CFI, and locality
  - Context awareness is key for effectiveness
- Low overhead, open-source
- Transitioning to practice
  - Enable as default defense in compiler
  - Outreach and awareness for developers