Data Confidentiality & Integrity

PIs: Mathias Payer (Purdue University)
https://hexhive.github.io/projects/

hexhive

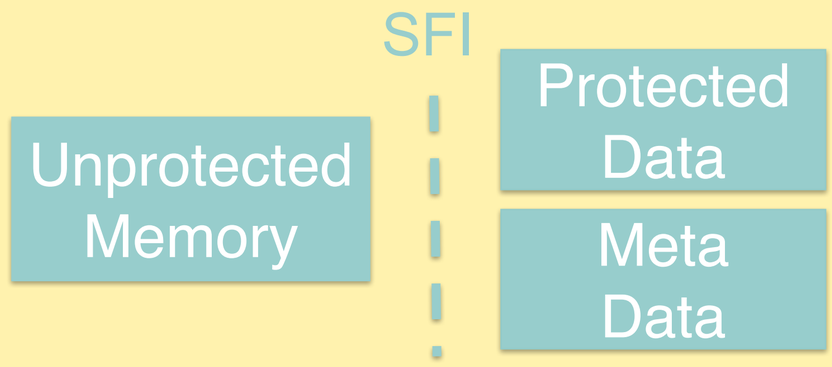
Challenge:
- Applications written in C/C++ are prone to memory corruption
- Existing solutions are incomplete or have high overhead

Solution:
- Key finding: only some data is sensitive
- Provide integrity and confidentiality for sensitive data, coarse protection for all
- Assumes CFI in place

Scientific Impact:
- Selective security policies
- Tunable security vs. overhead trade-off

Broader Impact:
- Protects critical data
- Easy to use: compiled code is protected
- Taught in new software security course

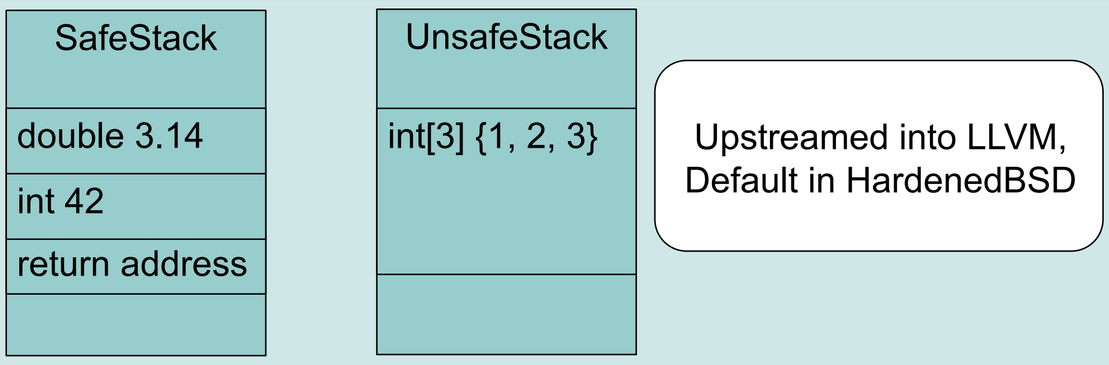
```
void vulnerable() {
  key *secret;
  int cmd[5];
  secret = load_key();
  input(cmd); // vulnerability
}
sensitive key *secret;
```



Kernel CFI [7]:
Enforce data-flow restrictions for code pointers to increase CFG precision:
- Code pointers can only be assigned or dereferenced.
- Prohibit data flow from data pointers to code pointers.

VTrust [5]:
- Verify virtual function based on type
- Sanitize vtable pointers to ensure pointee is valid

SafeStack:
- Unsafe data on separate stack
- Prevents corruption of return addresses
- Compiler-based transformation
- Relies on detailed type information



TypeSanitizer [6]:
- Verify C++ casts dynamically
- Downcasts (from base class to subclass) are unsafe
- Low overhead (SPEC: 4.6%, Firefox: 14.3%)

```
void draw(Shape* s) {
  Square* sq =
    static_cast<Square*>(s);
  s->foo();
}
```

PUBLICATIONS

1. Control-Flow Integrity 3P: Protection, Precision, and Performance. Nathan Burow, Scott A. Carr, Joseph Nash, Per Larsen, Michael Franz, Stefan Brunthaler, and Mathias Payer. In CSUR'17: ACM Computing Surveys, 2017 (to appear).
2. Automatic Contract Insertion with CCBot Scott A. Carr, Francesco Logozzo, and Mathias Payer. In TSE'16: IEEE Transactions on Software Engineering, 2016
3. Enforcing Least Privilege Memory Views for Multithreaded Applications. Terry Ching-Hsiang Hsu, Kevin Hoffman, Patrick Eugster, and Mathias Payer. In CCS'16: ACM Conf on Computer and Communication Security, 2016
4. TypeSanitizer: Practical Type Confusion Detection Istvan Haller, Yuseok Jeon, Hui Peng, Mathias Payer, Herbert Bos, Cristiano Giuffrida, and Erik van der Kouwe. In CCS'16: ACM Conf on Computer and Communication Security, 2016
5. VTrust: Regaining Trust on Your Virtual Calls Chao Zhang, Scott A. Carr, Tongxin Li, Yu Ding, Chengyu Song, Mathias Payer, and Dawn Song. In NDSS'16: Network and Distributed System Security Symposium, 2016
6. TypeSanitizer: Practical Type Confusion Detection. Istvan Haller, Yuseok Jeon, Hui Peng, Mathias Payer, Herbert Bos, Cristiano Giuffrida, and Erik van der Kouwe. In CCS'16: ACM Conf on Computer and Communication Security, 2016
7. Fine-Grained Control-Flow Integrity for Kernel Software Xinyang Ge, Nirupama Talele, Mathias Payer, and Trent Jaeger. In EuroS&P'16: IEEE European Symposium on Security and Privacy, 2016

Interested in meeting the PIs? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

NSF Secure and Trustworthy Cyberspace Inaugural Principal Investigator Meeting
Nov. 27 -29th 2012
National Harbor, MD

PURDUE UNIVERSITY