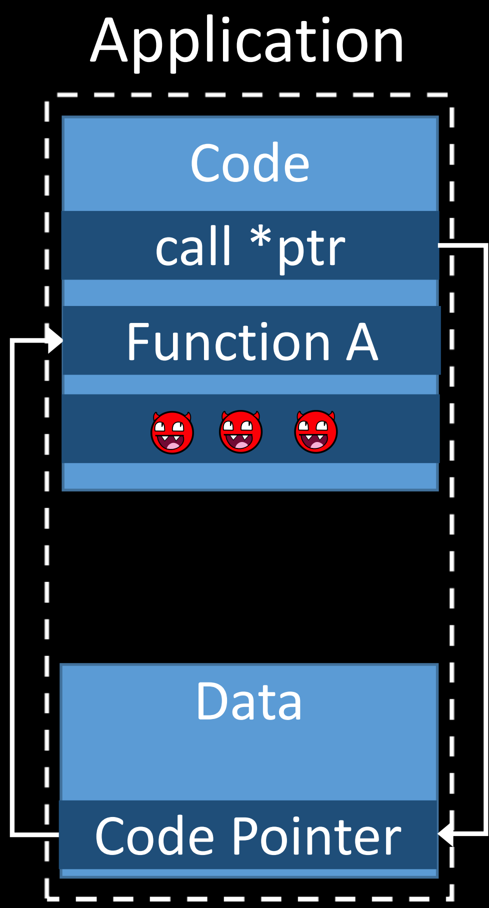
LR^2: Leakage-Resilient Layout Randomization for Mobile Devices

Kjell Braden†§, Stephen Crane‡, Lucas Davi†,
Michael Franz∗, Per Larsen∗‡, **Christopher Liebchen**†,
Ahmad-Reza Sadeghi†

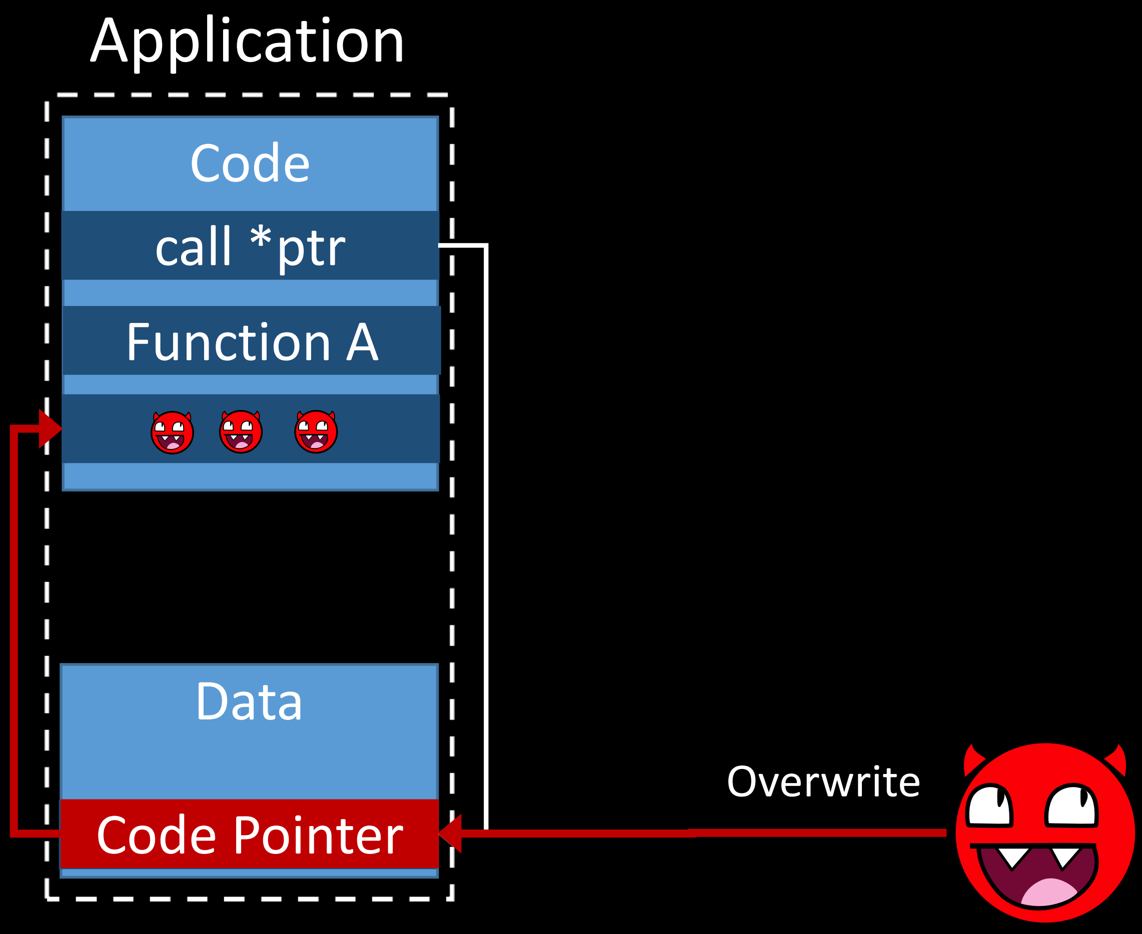
†TU Darmstadt §EURECOM
*UC Irvine ‡Immunant, Inc.


PROTECT
ALL THE NATIVE
CODES

Today's Exploits & Mitigations

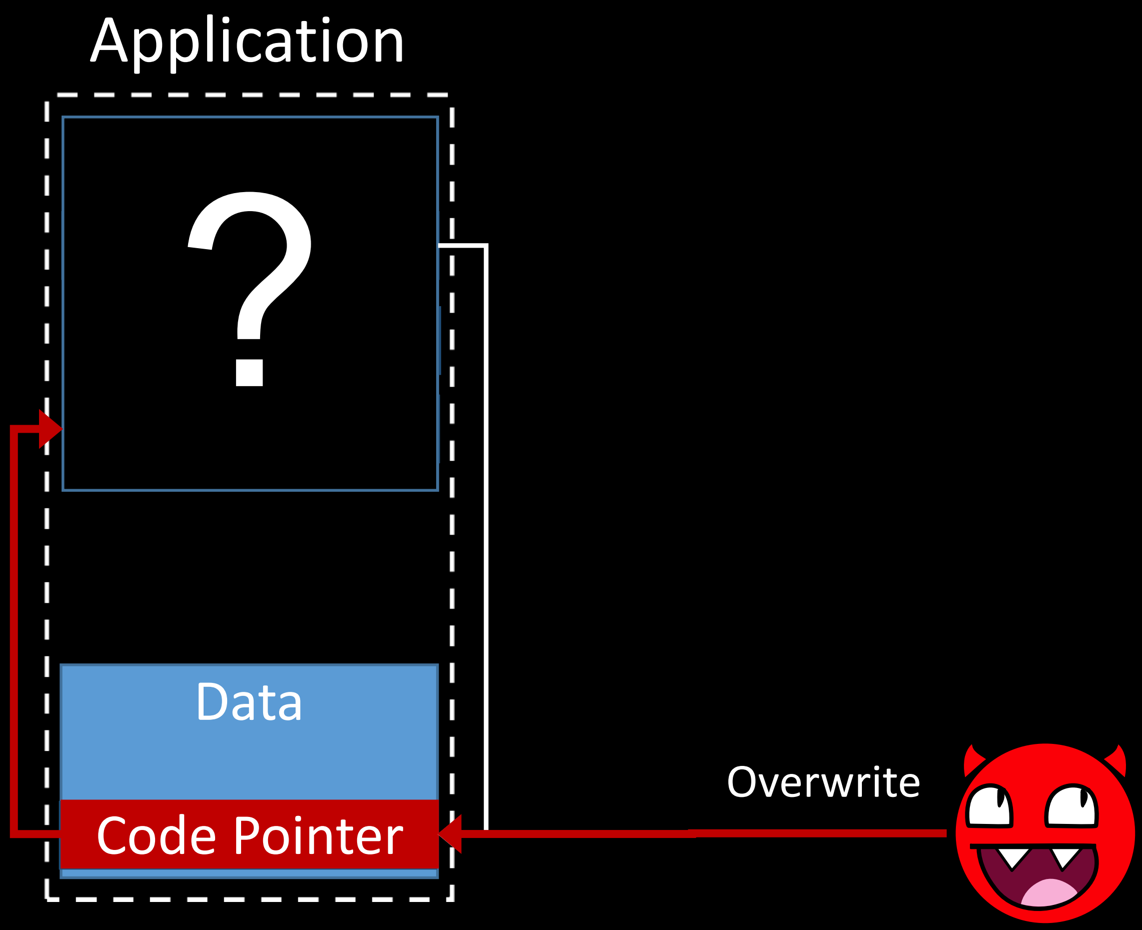


Today's Exploits & Mitigations



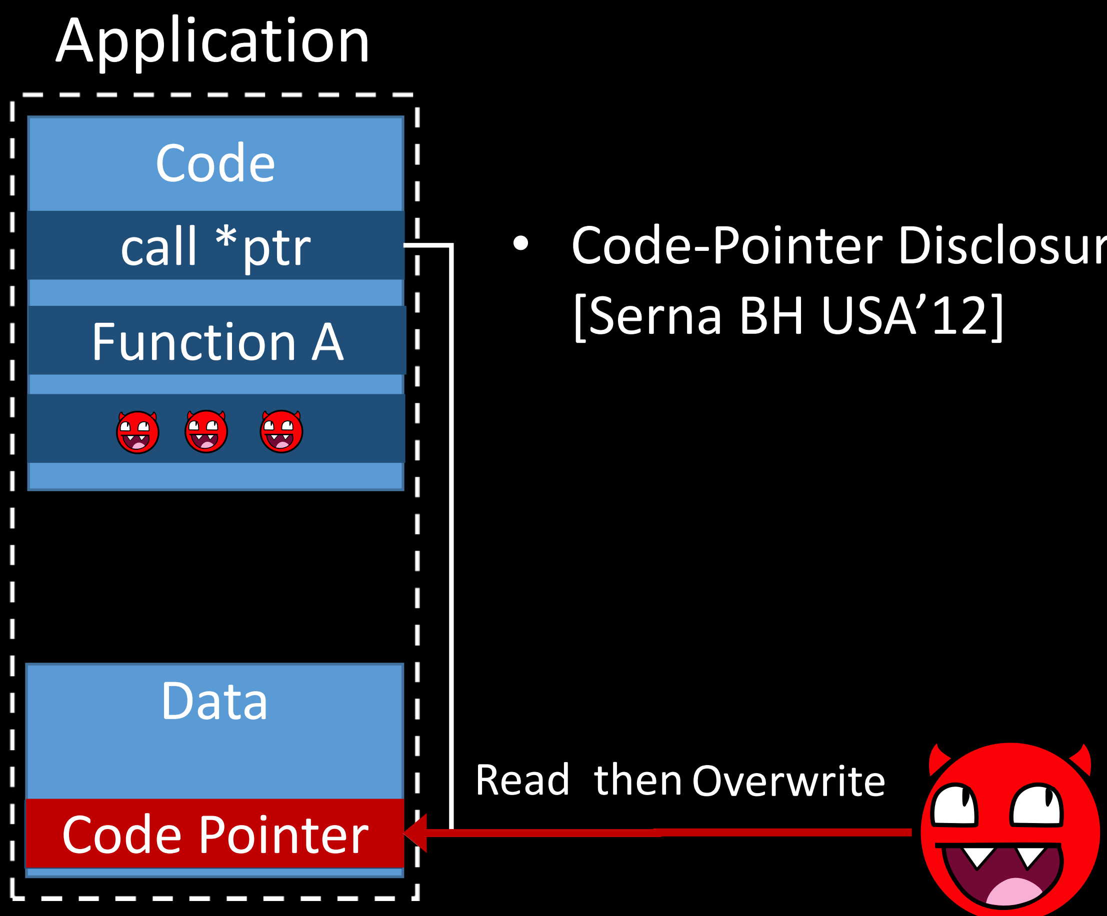
Today's Exploits & Mitigations

- Address Space Layout Randomization



Today's Exploits & Mitigations

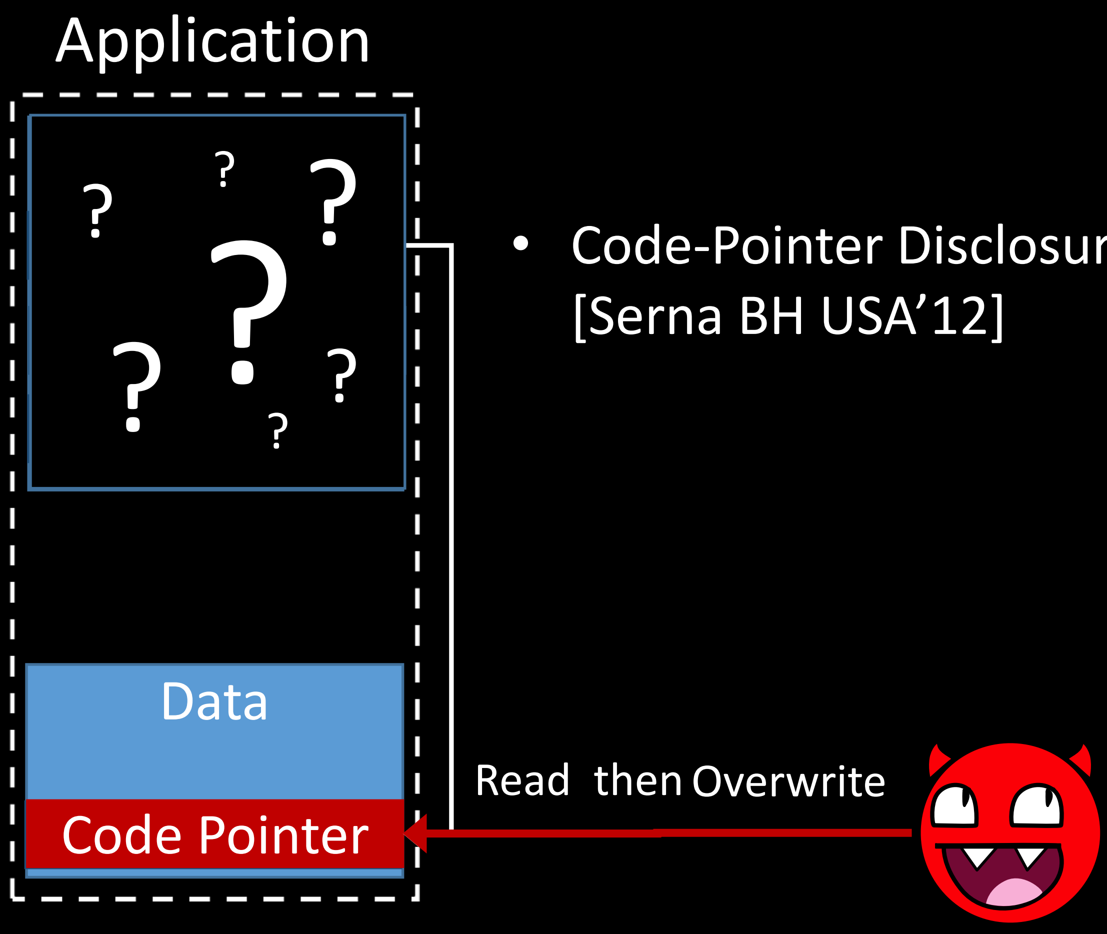
- Address Space Layout Randomization



- Code-Pointer Disclosure [Serna BH USA'12]

Today's Exploits & Mitigations

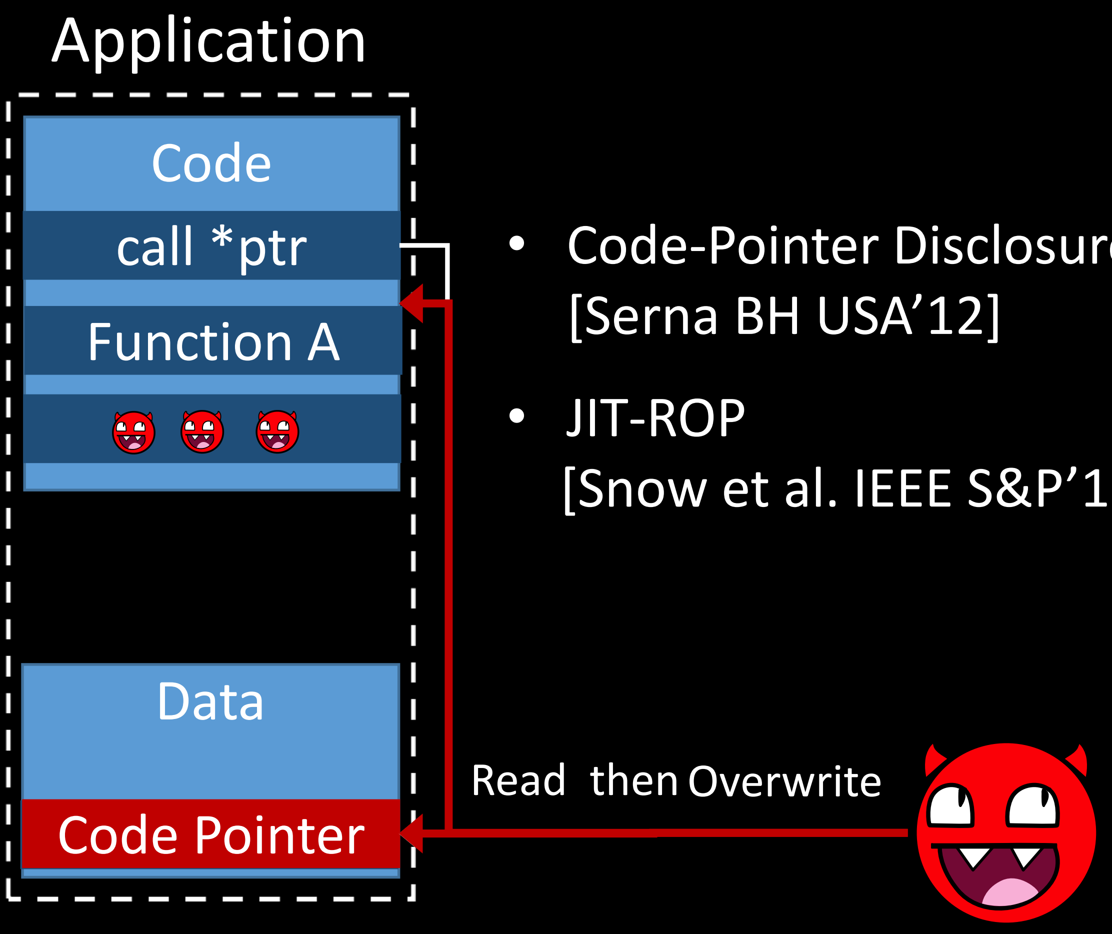
- Address Space Layout Randomization
- Fine-grained Code Randomization



- Code-Pointer Disclosure [Serna BH USA'12]

Today's Exploits & Mitigations

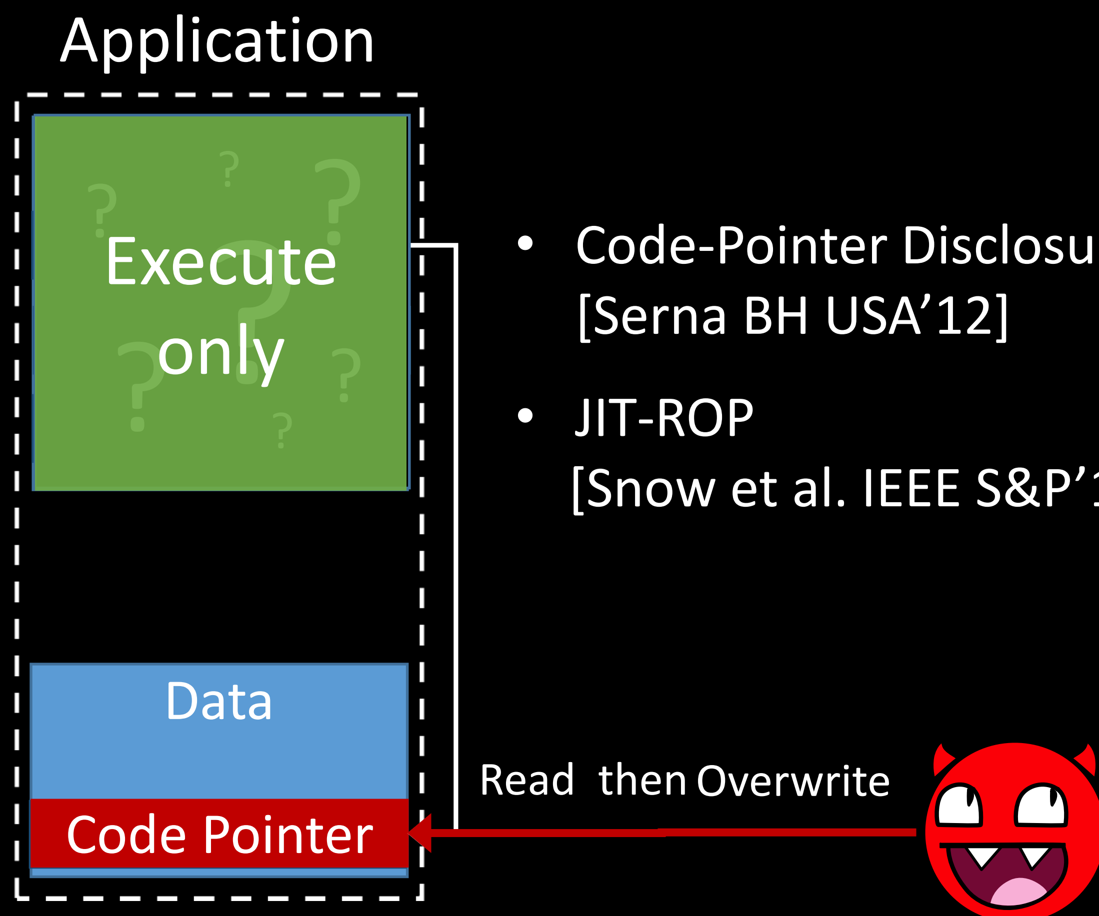
- Address Space Layout Randomization
- Fine-grained Code Randomization



- Code-Pointer Disclosure [Serna BH USA'12]
- JIT-ROP [Snow et al. IEEE S&P'13]

Today's Exploits & Mitigations

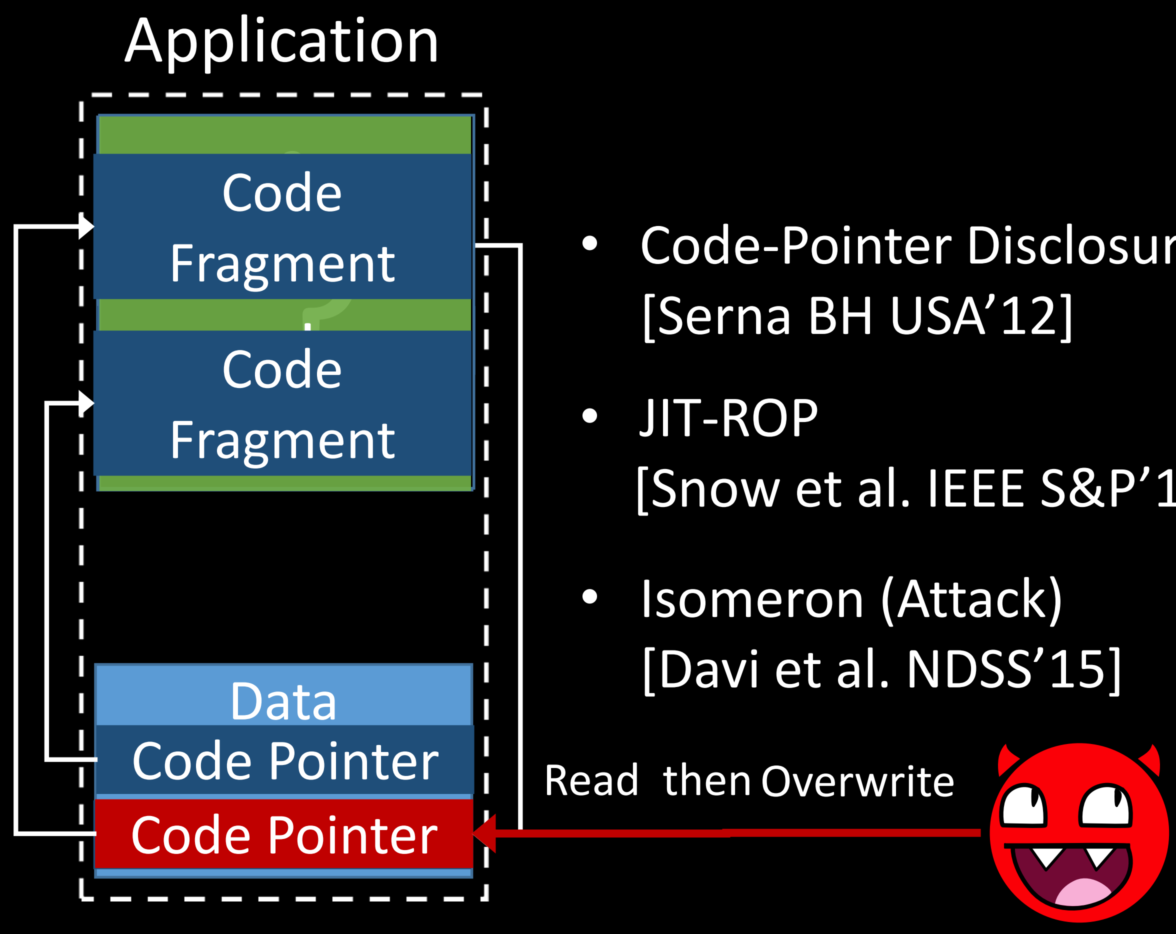
- Address Space Layout Randomization
- Fine-grained Code Randomization
- Execute-only Memory



- Code-Pointer Disclosure [Serna BH USA'12]
- JIT-ROP [Snow et al. IEEE S&P'13]

Today's Exploits & Mitigations

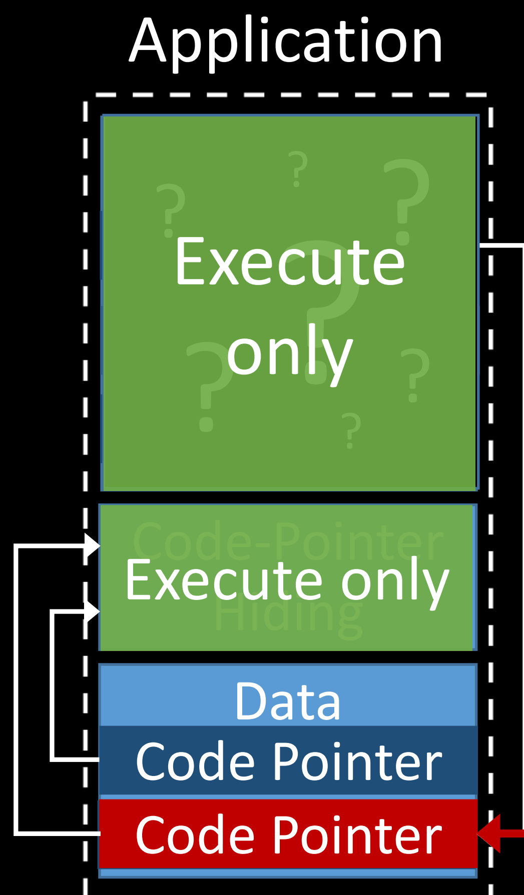
- Address Space Layout Randomization
- Fine-grained Code Randomization
- Execute-only Memory



- Code-Pointer Disclosure [Serna BH USA'12]
- JIT-ROP [Snow et al. IEEE S&P'13]
- Isomeron (Attack) [Davi et al. NDSS'15]

Today's Exploits & Mitigations

- Address Space Layout Randomization
- Fine-grained Code Randomization
- Execute-only Memory
- Code-Pointer Hiding

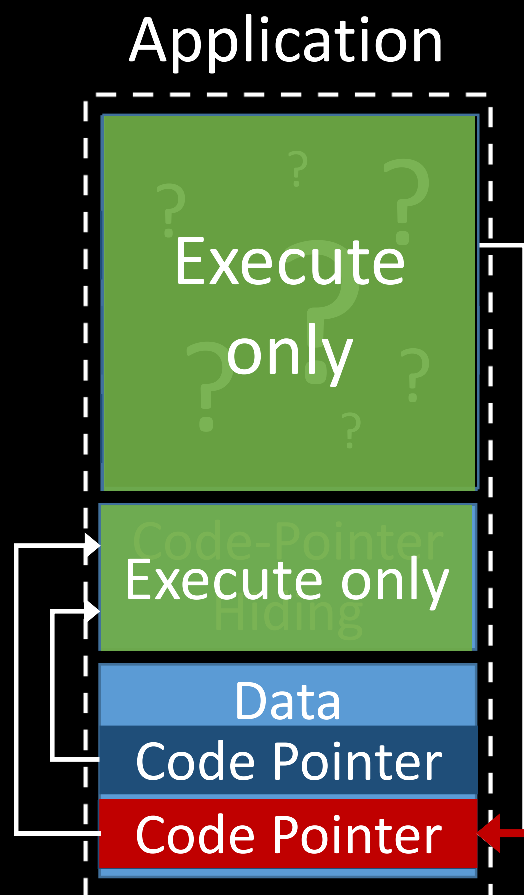


- Code-Pointer Disclosure [Serna BH USA'12]
- JIT-ROP [Snow et al. IEEE S&P'13]
- Isomeron (Attack) [Davi et al. NDSS'15]

Today's Exploits & Mitigations

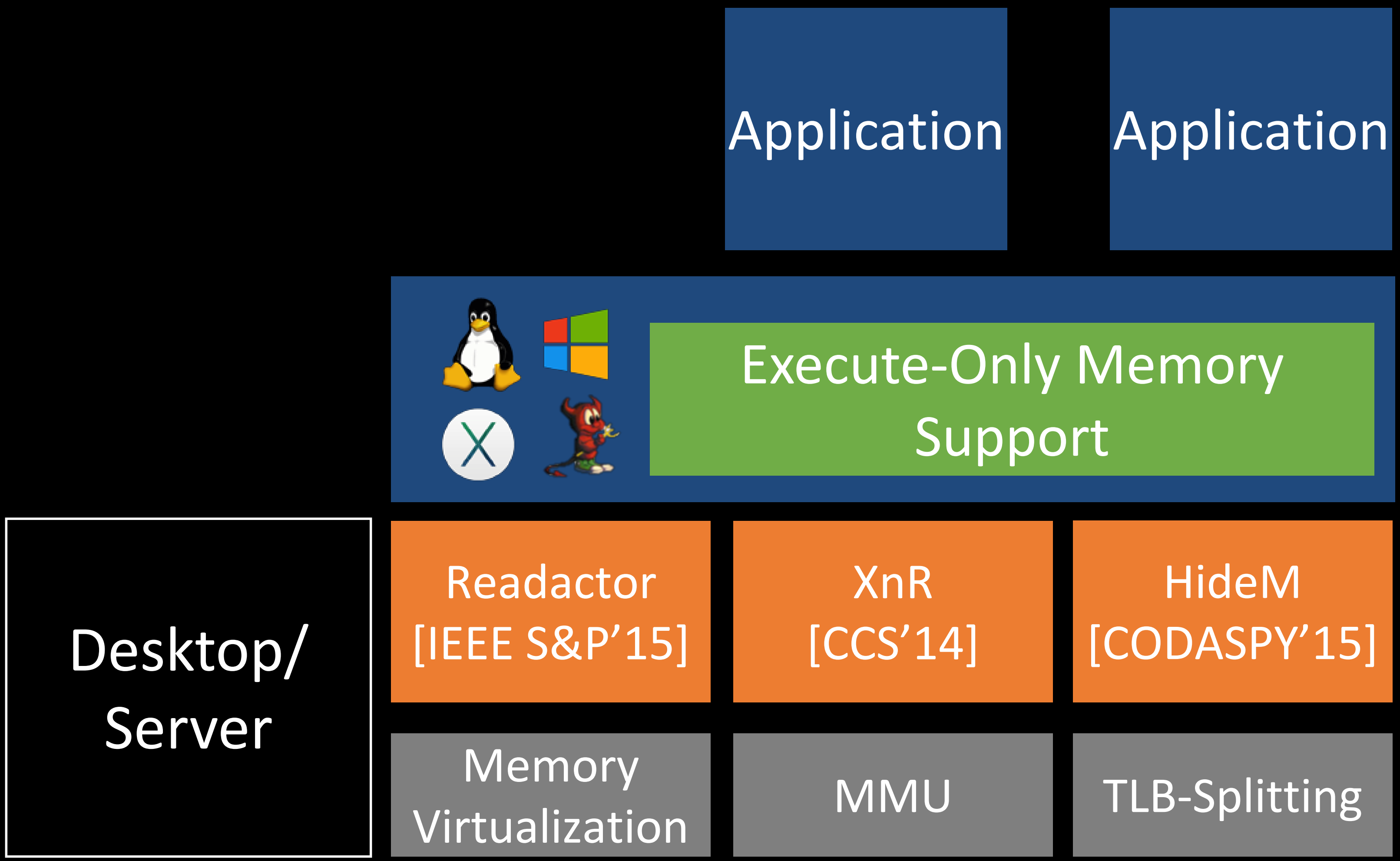
Readactor [IEEE S&P'15]
Readactor++ [CCS'15]

- Address Space Layout Randomization
- Fine-grained Code Randomization
- Execute-only Memory
- Code-Pointer Hiding

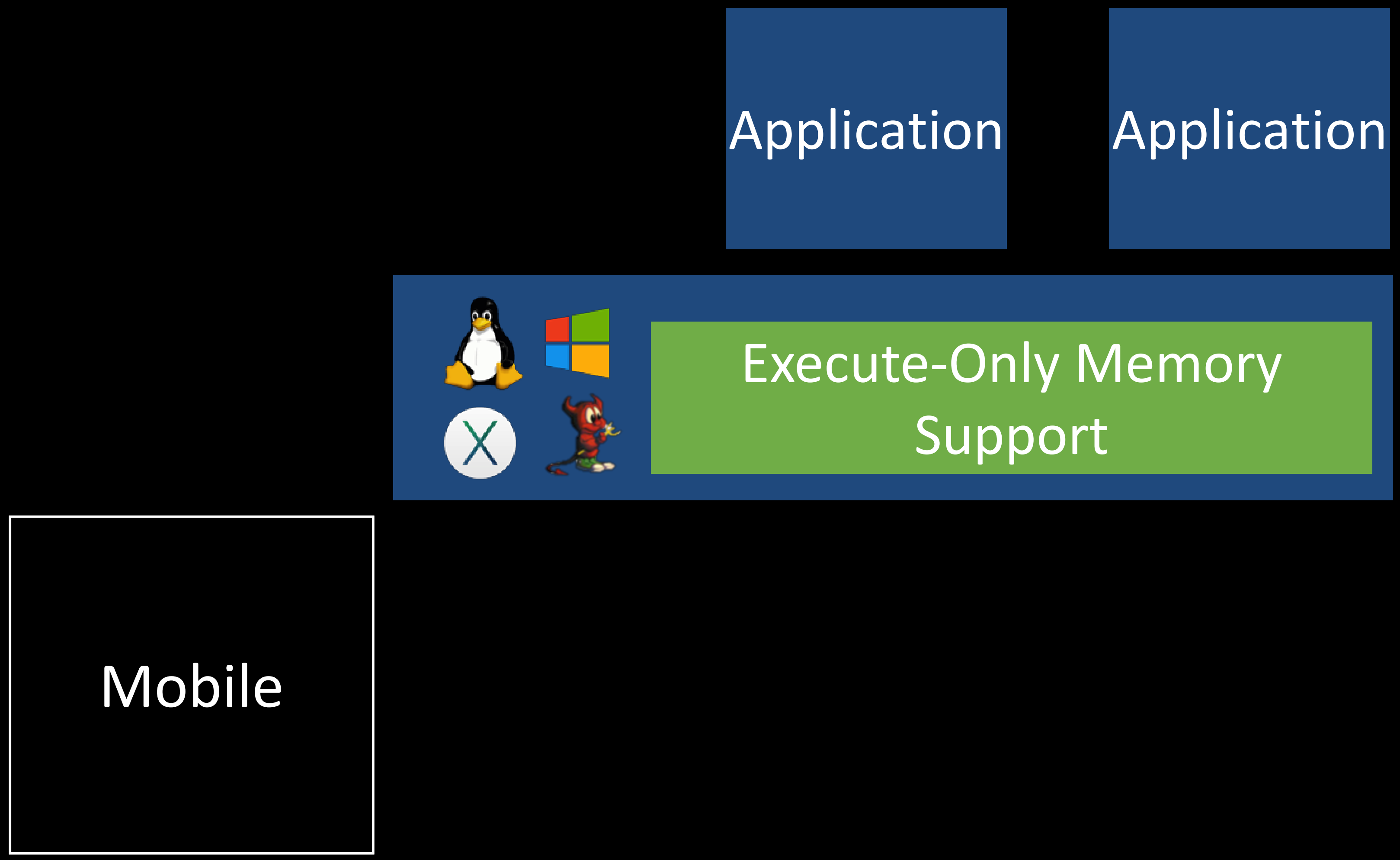


- Code-Pointer Disclosure [Serna BH USA'12]
- JIT-ROP [Snow et al. IEEE S&P'13]
- Isomeron (Attack) [Davi et al. NDSS'15]

Execute-only Memory



Execute-only Memory



Execute-only Memory

Application

Application

Execute-Only Memory Support

Mobile

This Talk:

Execute-only Memory without Hardware Support

Threat Model

- Read Memory (Information Disclosure)
- Write Memory (Memory Corruption Vulnerability)
- Perform Computations (Scripting Engine or Locally)
- Cannot Inject New Code (DEP, W^X)

LR^2: Leakage-Resilient Layout Randomization

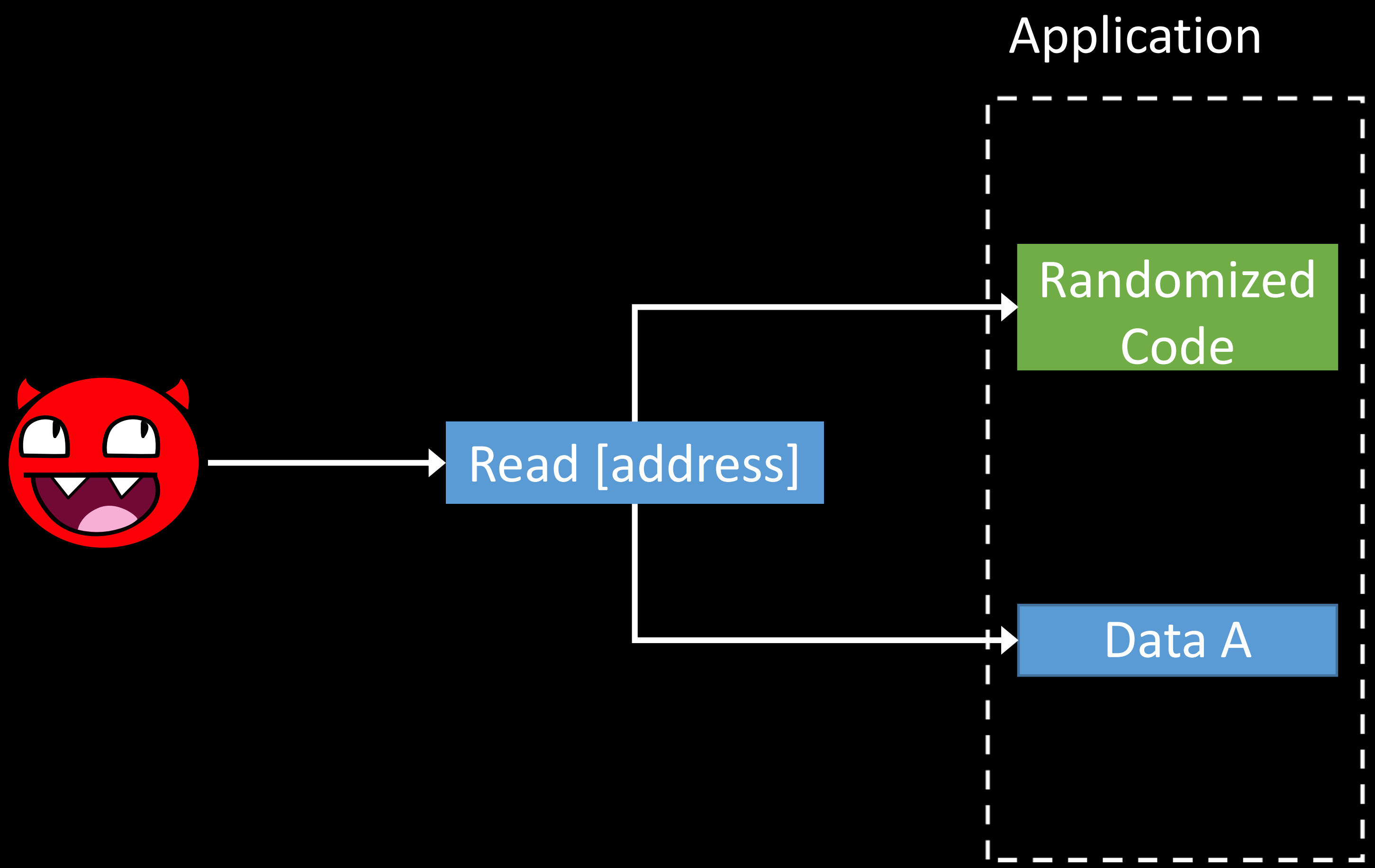
LR^2 Overview

- Fine-grained Code Randomization
- Software eXecute-only Memory (XoM)
- Code-Pointer Hiding
  - Return Addresses
  - Forward Pointers

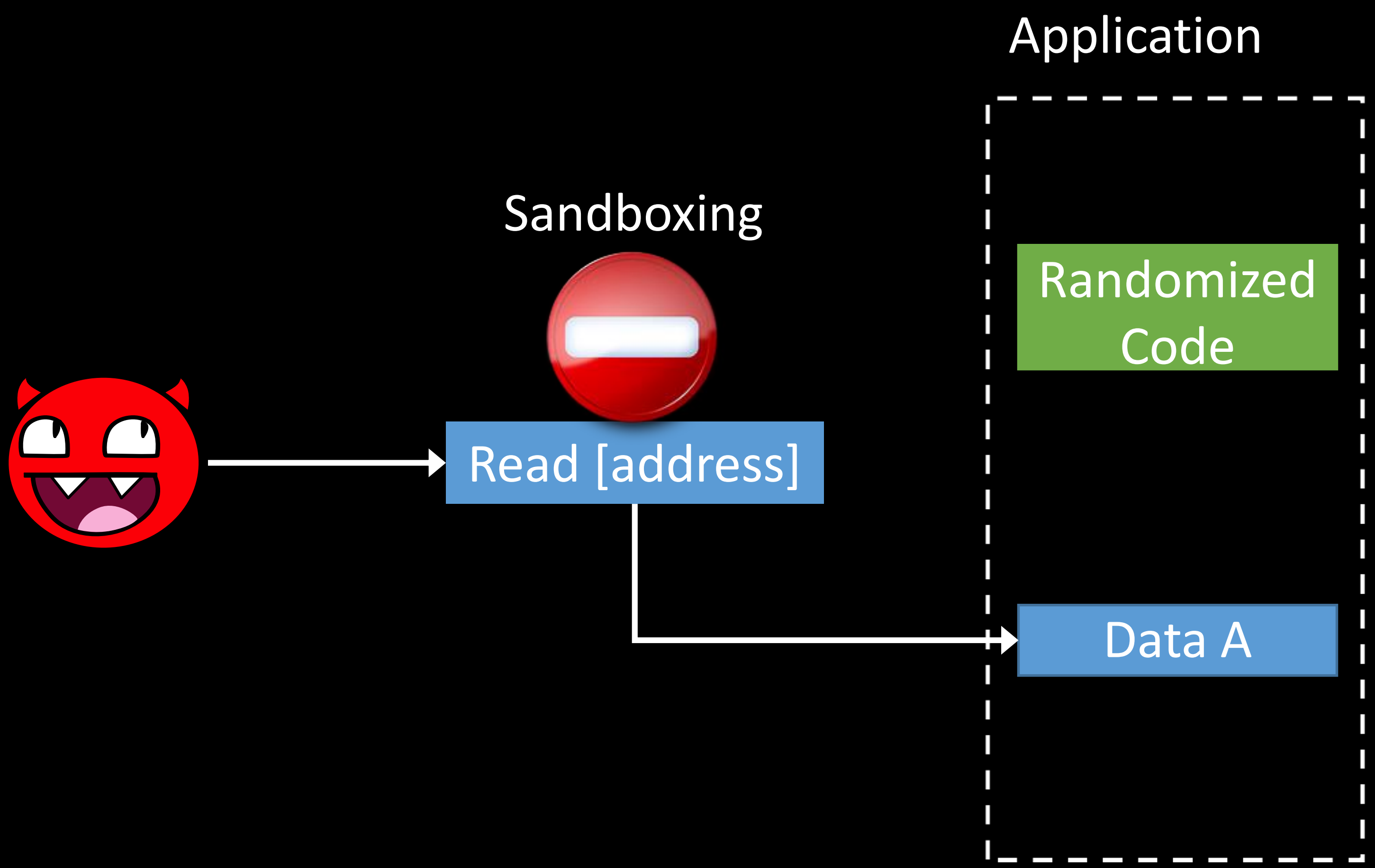
LR^2 Overview

- Fine-grained Code Randomization
- Software eXecute-only Memory (XoM)
- Code-Pointer Hiding
  - Return Addresses
  - Forward Pointers

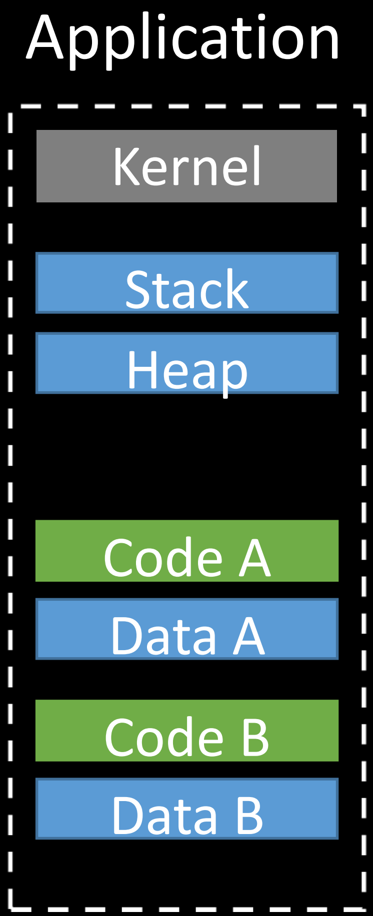
Software XoM: Idea



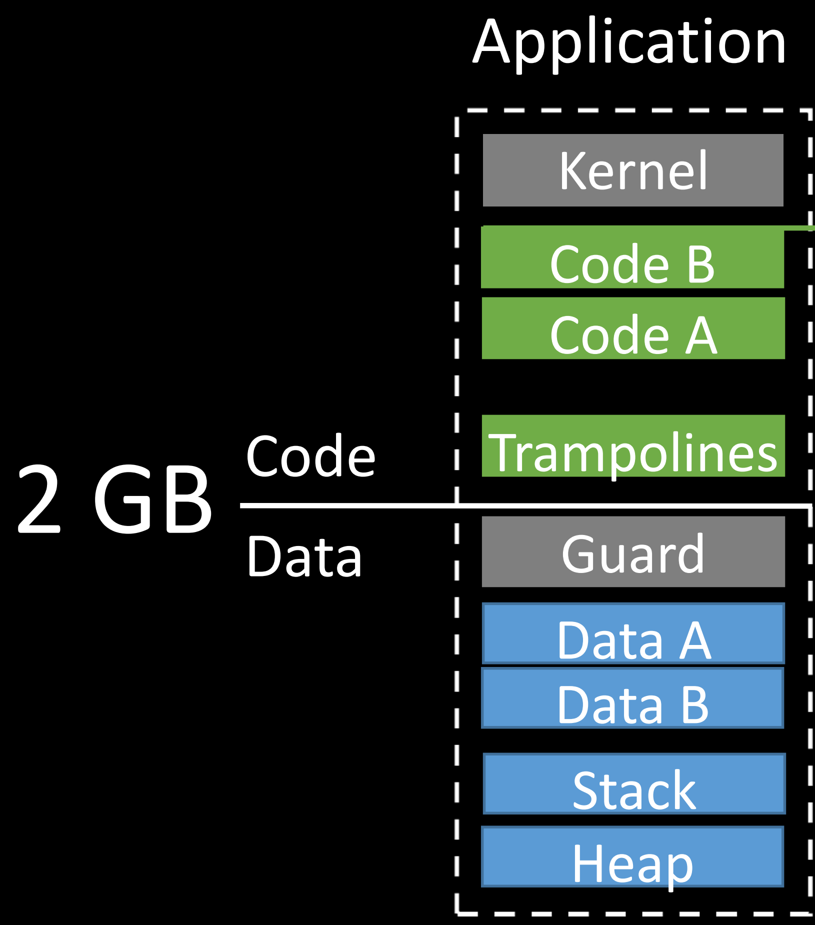
Software XoM: Idea



Software XoM: Design



Software XoM: Design

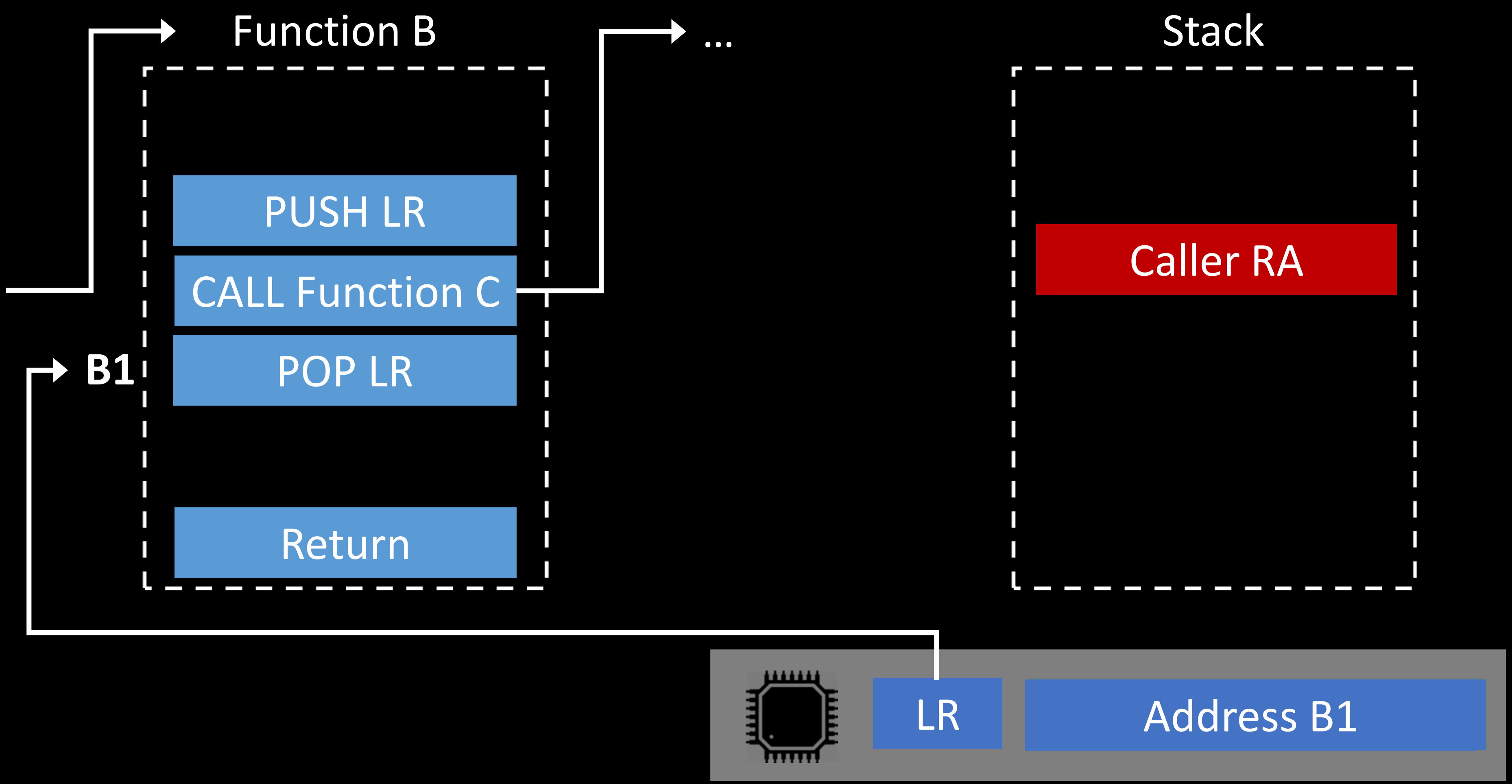


Sandboxing Read Instructions

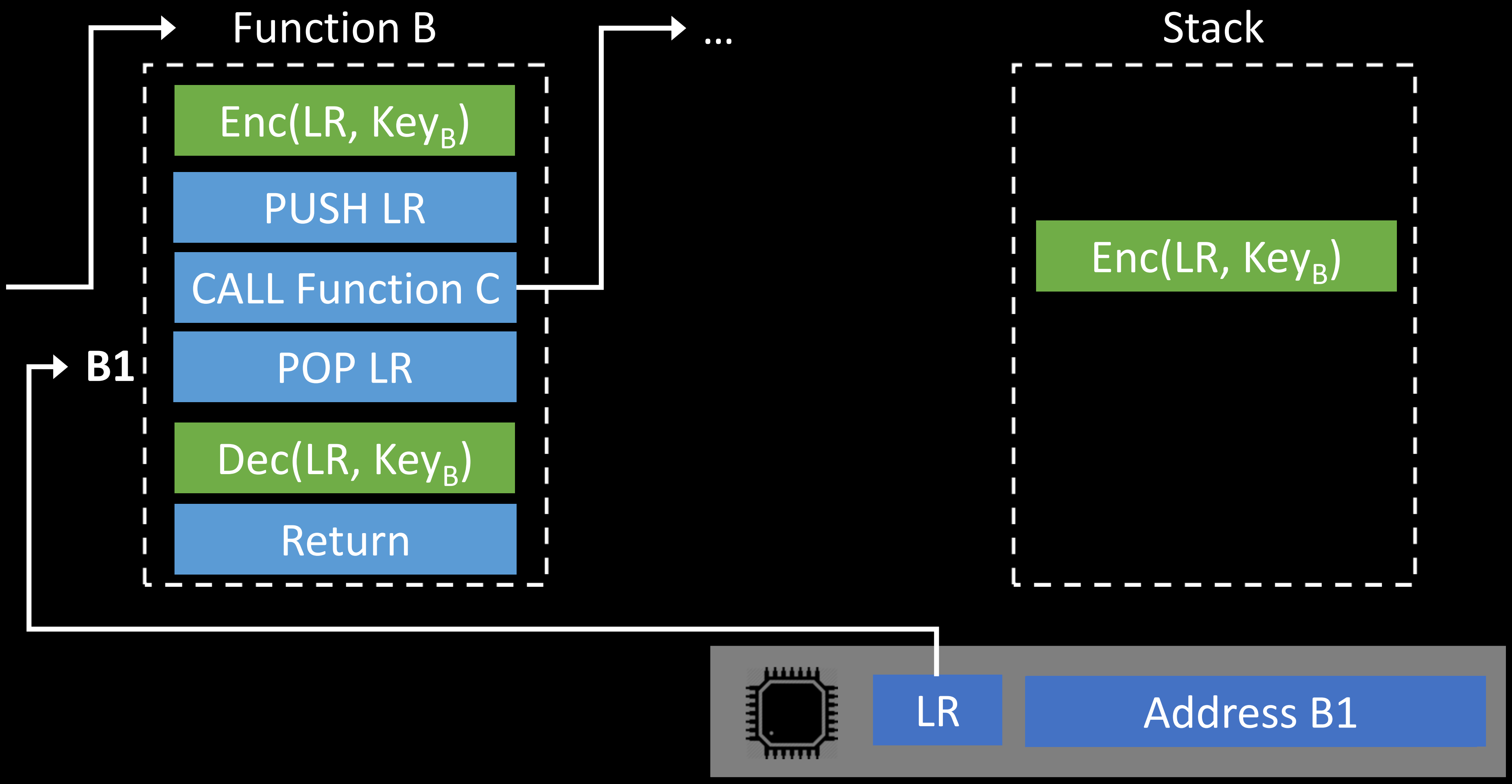
```
r1 <- addr
r1 <- r1 & 0x7FFFFFFF
r0 <- [r1]
```

Code-Pointer Hiding: Return Addresses

Code-Pointer Hiding: Return Addresses



Code-Pointer Hiding: Return Addresses



Sandboxing Reads: Optimizations

Optimizations: Loops

r0 <- address

For i <- 0 ; i < X ; ++i

Mask r0

r1 <- [r0]

Optimizations: Loops

r0 <- address

Mask r0

For i <- 0 ; i < X ; ++i

r1 <- [r0]

Optimizations: Loops

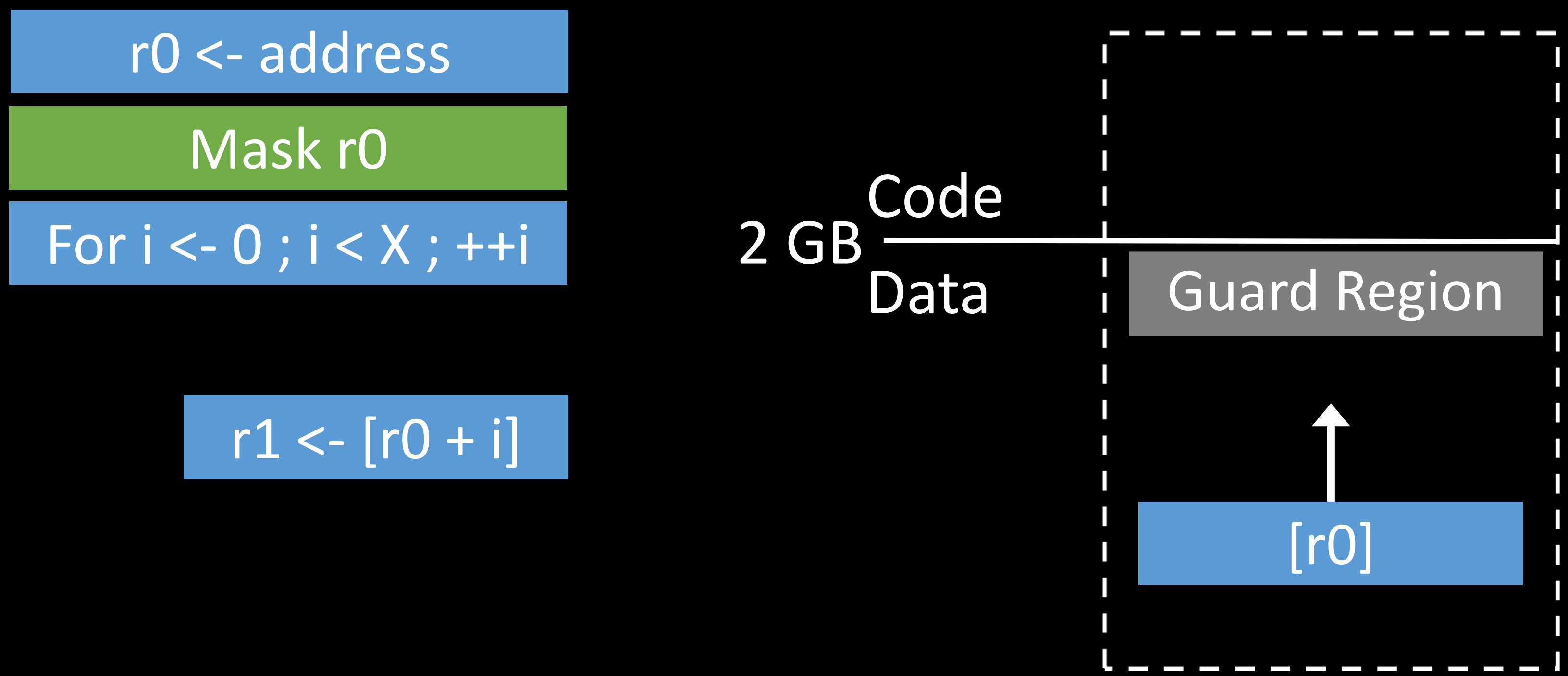
r0 <- address

Mask r0

For i <- 0 ; i < X ; ++i

r1 <- [r0 + i]

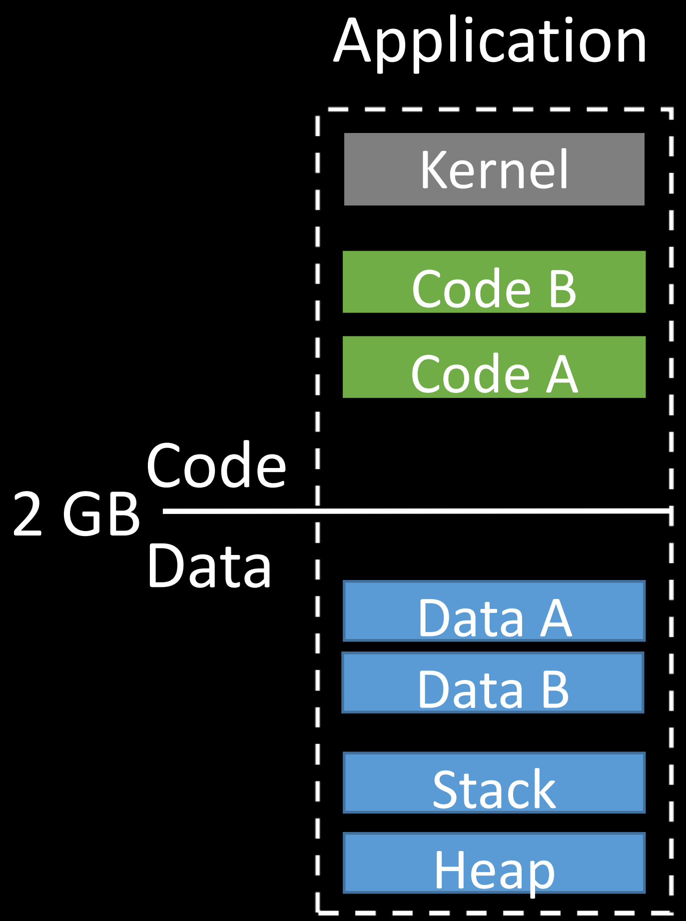
Optimizations: Loops



Implementation

Implementation

- Kernel
  - Stack and Heap Allocations
- Loader
  - Code and Data Sections
- Compiler
  - Sandbox Read Instructions

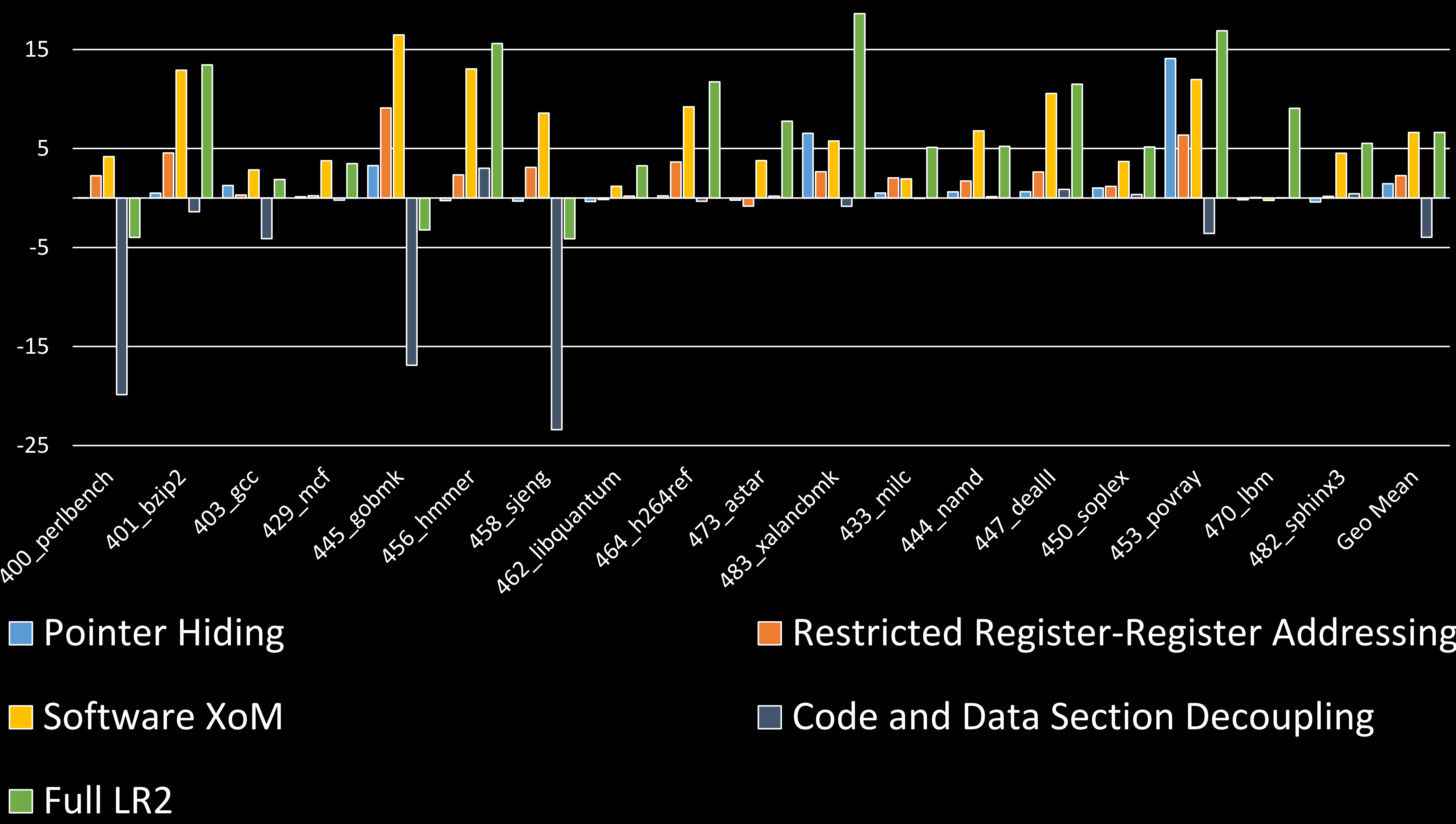


Evaluation

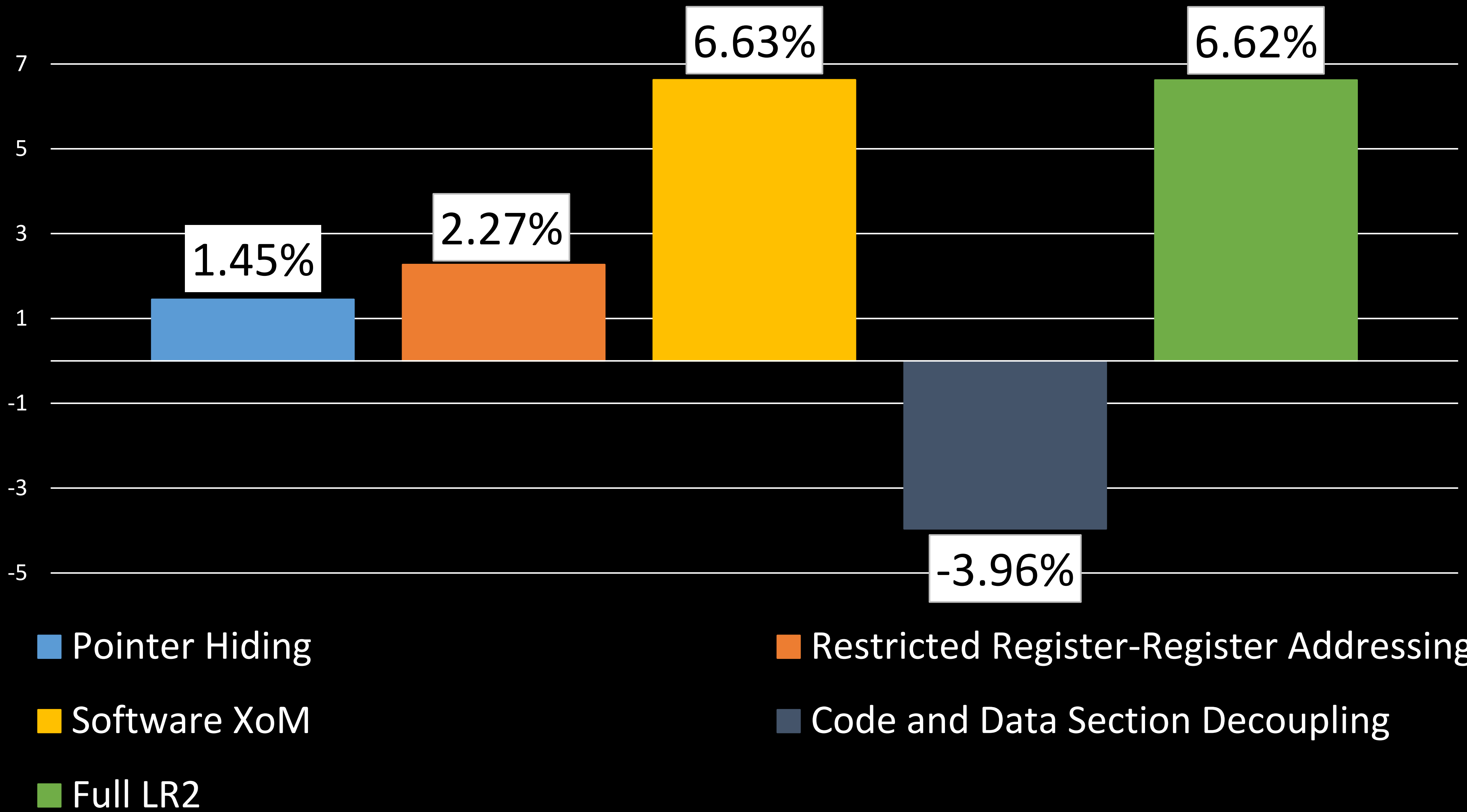
- Security:
  - Code-Reuse Attacks: Function Permutation
  - Direct disclosure: Execute-only Memory
  - Indirect disclosure:
    - Code-pointer Hiding
    - Code/Data section decoupling

- CPU: Nvidia Tegra Logan K1
- Performance:
  - 6.6% run-time overhead
  - 5.6% space overhead

SPEC CPU 2006



SPEC CPU 2006 Geometric Mean



LR^2 and Software Fault Isolation (SFI)

- Different Threat Models
  - SFI isolates untrusted code
  - LR^2 protects trusted code
- LR^2 can protect multiple load instructions by masking one address
- SFI sandboxes write and branch instructions

Conclusion

- First pure software execute-only memory technique
- Optimized return address protection scheme
- Performance and security matches state-of-the-art solutions requiring special, high-end hardware