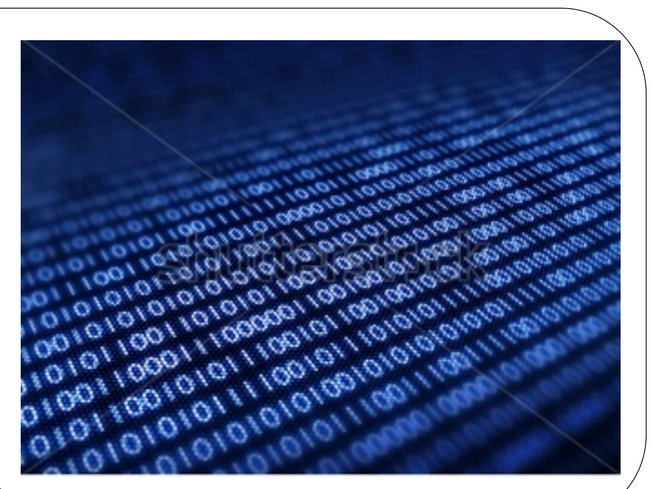
Analyzing ARM Native Code for Tracking Information Flow

Woo-Yeon Lee, Seo-Yoon Choi, Tae-Hun Kim, Byung-Gon Chun,
Cloud and Mobile Systems (CMS) Laboratory, Seoul National University (SNU)

Introduction

- Third-party "apps" may leak users' privacy-sensitive data or manifest malicious behavior.
- Why do we target ARM native code?
  - More and more apps use ARM native code.
  - Android : 49% of the apps are packaged with third-party native library.
  - Tizen : Native apps are written as ARM native code.
  - Lots of studies about information flow tracking, but not in ARM-instruction level.

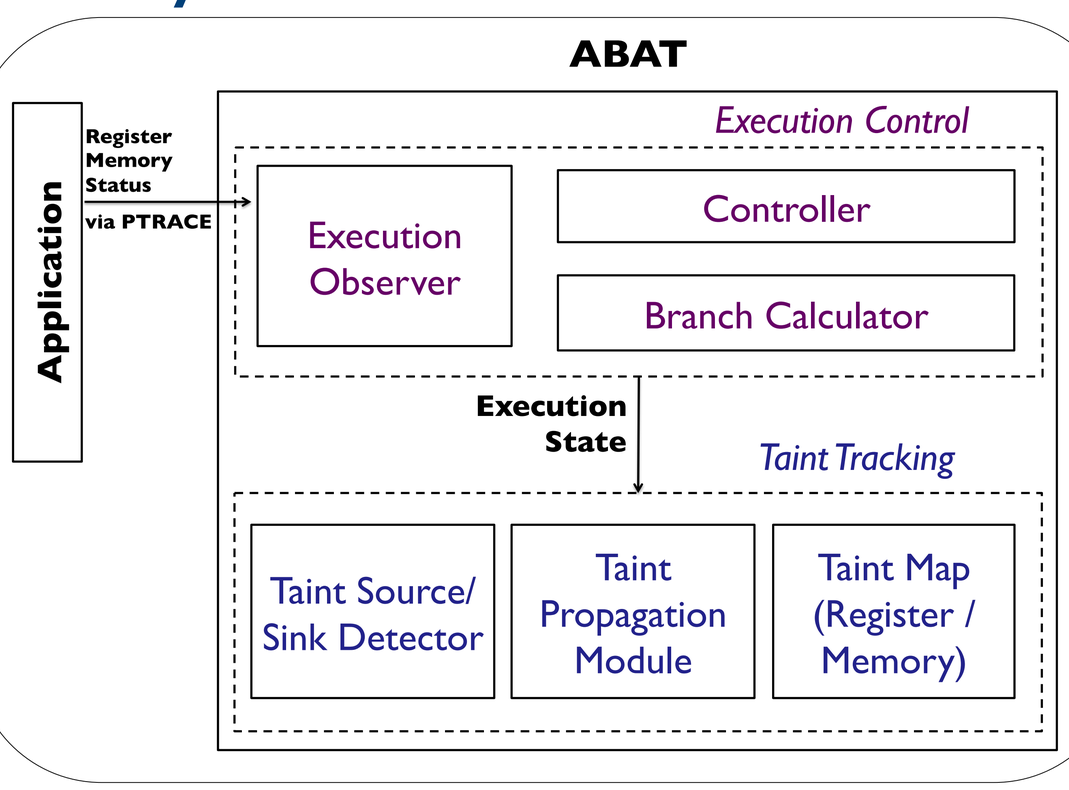


ARM Architecture

- Advanced RISC architecture
  - 32bit-fixed instruction length
  - PC as a general register
  - Single execution cycle
  - Conditional execution
- Extension
  - Thumb / Thumb-2 mode (16bit)

Some of these features are challenging to handle.

System Architecture



Concurrency Handling

- Multi-thread application support
  - Different Taint Map(register) for each thread
  - Coherent Taint Map(memory) for whole threads

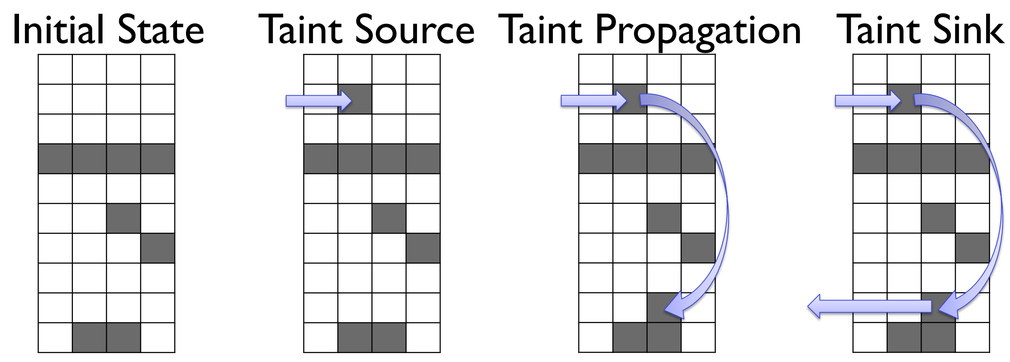
Dynamic Taint Tracking

- *Taint Tracking* is a technique used to track information dependencies from an origin.

- Three Factors
  - Taint Source
  - Taint Propagation
  - Taint Sink

```
v1 = taint_source()
…
v3 = v2 + v1
…
taint_sink(v3)
```

- Update Taint map during execution



- Taint Map API

Type	API	Description
Register	Tag source (threadId, regIndex, Set<Tag>);	Set an union of taint Tags to regIndex of threadId's register taint map
	Tag propagate (threadId, sourceIndex, destIndex);	Copy taint tag from sourceIndex to destIndex of threadId's register taint map
	Tag sink (threadId, regIndex);	Return Tag located in regIndex of threadId's register taint map
Memory	Tag source (memAddr, Set<Tag>);	Set an union of taint Tags to memAddr in memory taint map
	Tag propagate (sourceAddr, destAddr);	Copy taint tag from sourceAddr to destAddr of memory taint map
	Tag sink (memAddr);	Return Tag located in memAddr of memory taint map

- Taint Propagation Logic
  - We handle over 800 instructions
  - Taint Map Function τ ( )
  : τ (A) *retrieves the taint tag for 'A' from Taint Map.*

Operation Type	Assembly Representation	Action	Taint Propagation	Description
ADD <immediate>	ADD Rd, Rn, <immediate>	Rd := Rn + <immediate>	τ (Rd)$\leftarrow \tau$ (Rn)	Set Rd taint to Rn taint
ADD <register>	ADD Rd, Rn, Rm	Rd := Rn + Rm	τ (Rd)$\leftarrow \tau$ (Rn)$\cup \tau$ (Rm)	Set Rd taint to Rn taint OR Rm taint
MOV <immediate>	MOV Rd, <immediate>	Rd := <immediate>	τ (Rd)$\leftarrow \emptyset$	Clear Rd taint
MOV <register>	MOV Rd, Rn	Rd := Rn	τ (Rd)$\leftarrow \tau$ (Rn)	Set Rd taint to Rn taint