Inline Reference Monitors

- All jumps in untrusted section is made to 32 byte aligned targets.
- There are specific entry points in trusted code for untrusted code to jump.
- These 32 byte chunks are called Trampoline by Google. They are filled in by run time linker.
- Untrusted code can't modify these trampolines.
- Example Compiler will allocate 32 byte chunk for a function call such as safe open. For each jump to memory location that is out of range, there is a 32 byte chunk. This is similar to procedure linkage table (PLT)
• On Return, trusted code can jump back in the middle of 32 byte chunk. Better check return address

• Stack in untrusted code function has Return Address(RA) which can be overwritten

• **Function Call from untrusted module to trusted module**

  Push stack pointer on stack of stack pointers
  
  Switch stack
  
  Move Return Address of Springboard and arguments into trusted stack
  
  Jump foo ( Compiler knows type signature of foo)

• **Springboard handles return and is generated by linker. It is similar to trampoline.**

  Switch stack
  
  Copy return values
  
  Check Return Address (Do AND, OR Operation)
  
  Jump to Real Address (Can't jump to arbitrary return location)

Both the modules are in same address space

• Superfast context switch .
• It is only a few times slower than standard function call.
• It is faster than standard IPC method which involve process kernel switch because this technique is using shared memory based on software fault isolation.