30. CFI & secure data flow

CFI: CONTROL FLOW INTEGRITY

- overview(P7)

  Main idea: pre-determine control flow graph (CFG) of an application(
- **Branch Types(P10)**
- **Binary Instrumentation(P11)**

  Use *binary rewriting* to instrument code with runtime checks
  - CFG example(P12)
- **Control Flow Enforcement(P13-14)**
  - 有imprecise问题
- **CFI: Example of Instrumentation（P15-16）**
- **Improving CFI Precision（P17-18）**
  - **Shadow Stack（P20）**
  - **Shadow Stack Mode（P21）**
  - **ENDBRANCH（P22-23）**
  - **WAIT_FOR_ENDBRANCH State（P24）**
- **CFI: Security Guarantees（P25）**
  - 保证了控制流没有保证数据流

DATA FLOW PROTECTION

- **Two Usages of Data Flow Tracking（P28）**
- **Ways an Attacker Can Steal Your Secrets（P35）**
- Data Protection：**TAINT TRACKING（P36-**
  - **Data Lifetime（P37）**

    - **Tainting: Data Flow Tracking（P38）**
    - **Dynamic Taint Analysis example（P44，P49）**
    - application：**TaintDroid（P50-51）**
- **Defending Malicious Input（P53**
    - **How does a Hacker Search a Bug：example & steps（P55-63）**
        - **Step-0: Chose a library (open-source, of course)**
        - **Step-1: List the demuxers of ffmpeg**
        - **Step-2: Identify the input data**
        - **Step-3: Trace the input data**
    - solution：**TaintCheck: Basic Ideas（P64-65）**
    - **TaintCheck Detection Modules（P66）**
    - **Performance（P67-68）**