Using Metrics to Drive a Proactive Hardware Security Program

Jason Oberg (jason@cycuity.com)
Co-founder and CTO
About Me

• Co-founder Chief Technology Officer of Cycuity
  • Cycuity provides software products to increase security assurance in semiconductor designs
• PhD in Hardware Security from UC San Diego
• 15+ years in semiconductors and security
• Member of the Common Weakness Enumeration (CWE) board
Using Metrics to Drive a Security Development Lifecycle

Security Requirements
Collected before or during early stage of system development

Security Verification
Regularly applied as chip and system is developed

Security Signoff
Confirmed before manufacturing

Cycuity helps customers to...

- Align stakeholders on business needs and verifiable security requirements
- Apply automated security verification at each stage of the design process
- Document security compliance by providing verification evidence
What is CWE?

Common root-cause weaknesses that can lead to vulnerabilities

1194 - Hardware Design View
https://cwe.mitre.org/data/definitions/1194.html

- Manufacturing and Life Cycle Management Concerns - (1195)
  - Defects that arise in the semiconductor-manufacturing process.

- Security Flow Issues - (1196)
  - Full-system security flows, including secure boot, secure update, and hardware-device attestation.

- Integration Issues - (1197)
  - Integration of multiple IP cores, from SoC subsystem interactions, or from hardware platform subsystem interactions.

- Privilege Separation and Access Control Issues - (1198)
  - Hardware-based isolation and access control (e.g., identity, policy, locking control) of sensitive shared hardware resources such as registers and fuses.

- General Circuit and Logic Design Concerns - (1199)
  - Hardware-circuit design and logic as well as issues related to hardware description languages such as System Verilog and VHDL.

- Core and Compute Issues - (1201)
  - CPUs, Graphics, Vision, AI, FPGA, and microcontrollers.

- Memory and Storage Issues - (1202)
  - Memory (e.g., DRAM, SRAM) and storage technologies (e.g., NAND Flash, OTP, EEPROM, and eMMC).

- Peripherals, On-chip Fabric, and Interface/I0 Problems - (1203)
  - Peripheral devices, I/O interfaces, on-chip interconnects, NoCs, and buses.

- Security Primitives and Cryptography Issues - (1205)
  - Cryptographic protocols and other hardware-security primitives such as PUFs and RNGs.

- Power, Clock, and Reset Concerns - (1206)
  - System power, voltage, current, temperature, clocks, system state saving/restoring, and resets at the platform and SoC level.

- Debug and Test Problems - (1207)
  - Hardware debug and test interfaces such as JTAG and scan chain.

- Cross-Cutting Problems - (1208)
  - Weaknesses in this category can arise in multiple areas of hardware design or can apply to a wide cross-section of components.

- Physical Access Issues and Concerns (1388)
  - Weaknesses related to physical tamper and environmental issues
Why use CWE as a Security Metric?

• Technology products are exposed to a rising number of software and hardware vulnerabilities over their lifetimes
  • Over 160K vulnerabilities between 1999 to 2019
  • Over half were reported in the last 5 years
• Exposure increases for technologies with longer product lives
• Majority of these vulnerabilities have been reported before
  • Repeated mistakes suggest effective assurance practices are yet to be widely adopted
  • First step: Gain a deep understanding of common vulnerability patterns
• CWE provides a proactive framework to assess how secure a design is
Using CWE to Drive a Security Development Lifecycle

1. **RESEARCH** common weaknesses impacting related products to strengthen product threat models
2. **IDENTIFY** effective mitigation options per threat
3. **EDUCATE** to avoid common design & implementation pitfalls
4. **EVALUATE** relevant weakness categories to enumerate security properties & effective detection techniques
5. **TRIAGE** and classify escapes to facilitate PSIRT handling & training, process improvements

Organizations can use CWE can be used effectively across each of these 5 stages