HOST 2022 Microelectronics Security Challenge: SoC Security Track

Kanad Basu, Rajat Subhra Chakraborty and Ujjwal Guin
Emails: kanad.basu@utdallas.edu rajat.subhra.chakraborty@intel.com ujjwal.guin@auburn.edu

Overview: Modern computing systems extensively use System-on-Chips (SoCs), usually integrated with multiple Intellectual Property (IP) cores. The integration of different IPs has created distinctive security challenges while interacting among themselves. The individual IP cores may contain security-threatening bugs. However, when they are connected within an SoC, additional security bugs may arise even if the individual IP cores are verified as secure. A vulnerable design results from a security bug that escapes the verification phase. Hardware vulnerabilities can be introduced within a design due to incorrect/ambiguous specifications, which results in designers misinterpreting the design functionality, design mistakes such as setting a global variable to true when it should be false, and/or flawed implementations such as including knowingly insecure cryptographic cores that can be easily attacked. An adversary can exploit these vulnerabilities to launch an attack on a secure system. MITRE maintains a Common Weakness Enumeration (CWE) database [1] for vulnerabilities exploited by adversaries. These include issues in debug and test, peripherals and on-chip fabric, core and computing, etc. It is thus imperative to perform a thorough security assurance before a product is shipped. Trust-Hub also includes 50+ security properties to be checked [2]. With this perspective, we propose launching a competition so that students can analyze SoC and perform security analysis.

IP overview: A widely used open-source RISC-V-based SoC will be released to all the participants. Each participating group will be required to analyze the SoC and identify different vulnerabilities. A general overview of the SoC can be found from the "REARME.md" file inside the zipped folder. The SoC can be analyzed using any technique chosen by the participants. Common security verification techniques include simulation, emulation, formal verification, and manual code review. Though, the limitations of this competition will deem certain techniques infeasible/impractical. There will need to be supporting material included in the report that illustrates how the team deduced that a property was valid or violated, or that a vulnerability was inserted if the property was valid. This can be screenshots, highlighted portions of code, etc.

Submission Criteria:

- The participants are required to submit a zip file containing codes, sample submission, demo, and presentation. The file name should be "team_name_HOST_2022_SoCS.zip using OneDrive).
- The report should be clear. It may be prepared as a presentation or document including the following:
    - **Vulnerabilities and weaknesses:** List security issues found in the SoC and its implementation. Brief explanation as necessary.
    - **Mitigations:** List of mitigations. Brief explanation as necessary.
    - **Code:** Code for mitigations (as appendices) and anything else you think necessary to understand your work.
    - **Demo:** A demo video link (upload in OneDrive). The video should demonstrate finding an issue and/or proving a mitigation.
        - **Demo time limit**: minimum - 5 mins, maximum - 30 mins, recommended – 15 mins
- **In-person demonstrations:** Qualified groups will be invited to attend HOST'22 and demonstrate the designed crypto core in-person. Teams may also qualify for travel support. There will be final evaluation as well during the in-person demonstration time.

Evaluation Criteria:

Total points depends on the inserted vulnerabilities

- Identification of each vulnerability – 10 pts
- Mitigations for each identified vulnerability – 20 pts

Additional bonus points:

- New vulnerability Insertion: (only one: 20 points)

Poorly written, unclear reports may be rejected.

References:

[1] MITRE, "2021 CWE Most Important Hardware Weaknesses," Link: https://cwe.mitre.org/scoring/lists/2021_CWE_MIHW.html

[2] Link: https://trust-hub.org/#/data/Security-Properties-Rules