System Hardware Assurance

Authors: *Michael Bear, Donald Davidson, Shawn Fetterolf, Darin Leonhardt, Daniel Radack, Karen Johnson, Elizabeth A. McDaniel* ***Contributors:*** *Michael Berry, Brian Cohen, Diganta Das, Houman Homayoun, Thomas McDermott*

This article describes the discipline of hardware assurance, especially as it relates to systems engineering. It is part of the SE and Quality Attributes Knowledge Area.

□

Contents

Overview

System hardware assurance is a set of system security engineering activities (see System Security for more information) undertaken to quantify and increase the

confidence that electronics function as intended and only as intended throughout their life cycle, and to manage identified risks. The term *hardware* refers to electronic components, sometimes called integrated circuits or chips. As products of multi-stage processes involving design, manufacturing and post-manufacturing, packaging, and test, they must function properly under a wide range of circumstances. Hardware components – alone and integrated into subcomponents, subsystems, and systems – have weaknesses and vulnerabilities enabling exploitation. Weaknesses are flaws, bugs, or errors in design, architecture, code, or implementation. Vulnerabilities are weaknesses that are exploitable in the context of use (Martin 2014).

Hardware assurance is conducted to minimize risks related to hardware that can enable adversarial exploitation and subversion of functionality, counterfeit production, and loss of technological advantage. Challenges include increasing levels of sophistication and complexity of hardware architectures, integrated circuits, operating systems, and application software, combined with supply chain risks, emergence of new attack surfaces, and reliance on global sources for some components and technologies.

After identifying concerns and applicable mitigations, hardware assurance offers a range of possible activities and processes. At the component level, hardware assurance focuses on the hardware itself and the supply chain used to design and manufacture it; at the subcomponent, subsystems, and system levels, hardware assurance incorporates the software and firmware integrated with the component.

Engineering efforts to enhance trust in hardware have increased in response to complex hardware architectures, the increasing sophistication of adversarial attacks on hardware, and globalization of supply chains. These factors raise serious concerns about the security, confidentiality, integrity, and availability as well as the provenance and authenticity of hardware. The “root of trust” (NIST 2020) of a system is typically contained in the processes, steps, and layers of hardware components and across the systems engineering development cycle. System hardware assurance focuses on hardware components and their interconnections with software and firmware to reduce risks to proper function or other compromises of the hardware throughout the complete life cycle of components and systems. Advances in hardware assurance tools and techniques will strengthen designs,

and enhance assurance during manufacturing, packaging, test, and deployment and operational use.

Life Cycle Concerns of Hardware Components

Hardware assurance should be applied at various stages of a component's life cycle from hardware architecture and design, through manufacturing and testing, and finally throughout its inclusion in a larger system. The need for hardware assurance then continues throughout its operational life including sustainment and disposal.

As semiconductor technology advances the complexity of electronic components, it increases the need to "bake-in" assurance. Risks created during architecture, design, and manufacturing are challenging to address during the operational phase. Risks associated with interconnections between and among chips are also a concern. Therefore, improving a hardware assurance posture must occur as early as possible in the life cycle, thereby reducing the cost and schedule impacts associated with "fixing" components later in the life cycle of the system.

A conceptual overview of the typical hardware life cycle (Figure 1) illustrates the phases of the life cycle of components, as well as the subsystems and systems in which they operate. In each phase multiple parties and processes contribute a large set of variables and corresponding attack surfaces. As a result, the potential exists for compromise of the hardware as well as the subcomponents and systems in which they operate; therefore, matching mitigations should be applied at the time the risks are identified.

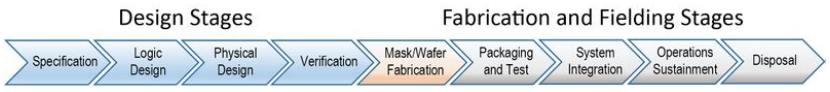

Figure 1. Component Life Cycle. (SEBoK Original)

Both the value of the hardware component and the associated cost of mitigating risks increase at each stage of the life cycle. Therefore, it is important to identify and mitigate vulnerabilities as early as possible. It takes longer to find and fix defects later, thereby increasing the complexity of replacing hardware with "corrected" designs that create system integration issues. In addition to cost savings, early correction and mitigation avoid delays in creating an operational system. It is essential to re-assess risks associated with hardware components

throughout the life cycle periodically, especially as operational conditions change.

Hardware assurance during system sustainment is a novel challenge given legacy hardware and designs with their associated supply chains. In long-lived high-reliability systems, hardware assurance issues are compounded by obsolescence and diminished sourcing of components, thereby increasing concerns related to counterfeits and acquisitions from the gray market.

Function as Intended and Only as Intended

Exhaustive testing can check system functions against specifications and expectations; however, checking for unintended functions is problematic. Consumers have a reasonable expectation that a purchased product will perform as advertised and function properly (safely and securely, under specified conditions) – but consumers rarely consider if additional functions are built into the product. For example, a laptop with a web-conferencing capability comes with a webcam that will function properly when enabled, but what if the webcam also functions when turned off, thereby violating expectations of privacy? Given that a state-of-the-art semiconductor component might have billions of transistors, "hidden" functions might be exploitable by adversaries. The statement "function as intended and only intended" communicates the need to check for unintended functions.

Hardware specifications and information in the design phase are needed to validate that components function properly to support systems or missions. If an engineer creates specifications that support assurance that flow down the system development process, the concept of "function as intended" can be validated for the system and mission through accepted verification and validation processes. "Function only as intended" is also a consequence of capturing the requirements and specifications to assure the product is designed and developed without extra functionality. For example, a Field Programmable Gate Array (FPGA) contains programmable logic that is highly configurable; however, the programmable circuitry might be susceptible to exploitation.

Given the specifications of a hardware component, specialized tools and processes can be used to determine with a high degree of confidence whether the

component's performance meets specifications. Research efforts are underway to develop robust methods to validate that a component does not have capabilities that threaten assurance or that are not specified in the original design. Although tools and processes can test for known weaknesses, operational vulnerabilities, and deviations from expected performance, all states of possible anomalous behavior cannot currently be determined or predicted.

Data and information can be used to validate the component's function and should be collected from multiple sources including designers, developers, and members of the user community. Designers and developers can provide deep understanding of the component's intended function and provide tests used to verify its functional performance before fielding. The merging of component design and development information with extensive field data, including third-party evaluation, contributes to assurance that the component is performing specified functions and that no unintended functionality is observed.

Risks to Hardware

Modern systems depend on complex microelectronics, but advances in hardware without attention to associated risks can expose critical systems, their information, and the people who rely on them. "Hardware is evolving rapidly, thus creating fundamentally new attack surfaces, many of which will never be entirely secured". (Oberg 2020) Therefore, it is imperative that risk be modeled through a dynamic risk profile and be mitigated in depth across the entire profile. Hardware assurance requires extensible mitigations and strategies that can and do evolve as threats do. Hardware assurance methods seek to quantify and improve confidence that weaknesses that can become vulnerabilities that create risks are mitigated.

Most hardware components are commercially designed, manufactured, and inserted into larger assemblies by multi-national companies with global supply chains. Understanding the provenance and participants in complex global supply chains is fundamental to assessing risks associated with the components.

Operational risks that derive from unintentional or intentional features are differentiated based on the source of the feature. Three basic operational risk areas related to goods, products, or items are: failure to meet

quality standards, maliciously tainted goods, and counterfeit hardware. Counterfeits are usually offered as legitimate products, but they are not. They may be refurbished or mock items made to appear as originals, re-marked products, the result of overproduction, or substandard production parts rejected by the legitimate producer. Counterfeit risks and substandard quality offer avenues for malware insertion and potential impacts to overall system performance and availability.

Failure to follow quality standards including safety and security standards, especially in design, can result in unintentional features or flaws being inadvertently introduced. These can occur through mistakes, omissions, or lack of understanding how features might be manipulated by future users for nefarious purposes. Features introduced intentionally for specific purposes can also make the hardware susceptible to espionage or control of the hardware at some point in its life cycle.

Quantify and Improve Confidence

The quantification of hardware assurance is a key technical challenge because of the complex interplay among designer, manufacturer and supply chains, and adversarial intent, as well as the challenge of defining "security" with respect to hardware function. Quantification is necessary to identify and manage hardware risks within program budgets and timeframes. It enables a determination of the required level of hardware assurance and whether quantification is achievable throughout the hardware's life cycle.

Current methods for quantifying hardware assurance are adapted from the fields of quality and reliability engineering, which use methods like Failure Mode and Effects Analysis (FMEA). (SAE 2021) FMEA is semi-quantitative and combines probabilistic hardware failure data and input from experts. Adapting FMEA to quantify hardware assurance is hampered when it relies on assigning probabilities to human behavior that may be motivated by money, malicious intent, etc. Expert opinion often varies when quantifying and weighting factors used in generating risk matrices and scores. In response, recent efforts are attempting to develop quantitative methods that reduce subjectivity.

Game theoretic analysis (game theory) is the creation of mathematical models of conflict and cooperation between intelligent and rational decision-makers.

(Myerson 1991) Models include dynamic, as opposed to static, interactions between attackers and defenders that can quantify the risks associated with potential interactions among adversaries, hardware developers, and manufacturing processes. (Eames and Johnson 2017) Creation of the models forces one to define attack scenarios explicitly and to input detailed knowledge of hardware development and manufacturing processes. Outputs of the model may include a ranking of the most likely attacks to occur based on cost-benefit constraints on the attackers and defenders. (Graf 2017) The results can empower decision-makers to make quantitative trade-off decisions about hardware assurance.

Another quantification method that results in a confidence interval for detecting counterfeit/suspect microelectronics is presented in the SAE AS6171 standard. (SAE 2016) Confidence is based on knowing the types of defects associated with counterfeits, and the effectiveness of different tests to detect those defects. Along the same lines, a standard for hardware assurance might be developed to quantify the confidence interval by testing against a catalogue of known vulnerabilities, such as those documented in the MITRE Common Vulnerabilities and Exposures (CVE) list. (MITRE 2020)

Distributed ledger technology (DLT) is an example of an emerging technology that could enable a standardized approach for quantifying hardware assurance attributes such as data integrity, immutability, and traceability. DLT can be used in conjunction with manufacturing data (such as dimensional measurement, parametric testing, process monitoring, and defect mapping) to improve tamper resistance using component provenance and traceability data. DLT also enables new scenarios of cross-organizational data fusion, opening the door to new classes of hardware integrity checks.

Manage Risks

The selection of specific components for use in subsystems and systems should be the outcome of performance-risk and cost-benefit trade-off assessments in their intended context of use. The goal of risk management and mitigation planning is to select mitigations with the best overall operational risk reduction and the lowest cost impact. The required level of hardware assurance varies with the criticality of a component's use and the system in which it is used.

During a typical development life cycle of a system – architecture, design, code, and implementation – various

types of problems can pose risks to the operational functionality of the hardware components provided. These risks include weaknesses or defects that are inadvertent (unintentional), as well as counterfeits that may be either inadvertent or intentionally injected into the supply chain for financial motivations or malicious components designed to change functionality.

Managing risk in the context of hardware assurance seeks to decrease the risk of weaknesses that create attack surfaces that can be exploited, while improving confidence that an implementation resists exploitation. Ideally, risk management reduces risk and maximizes assurance to an acceptable level. Often, risks are considered in the context of likelihood of consequences and the costs and effectiveness of mitigations. However, new operationally impactful risks are recognized continuously over the hardware life cycle and supply chains of components. At the same time hardware weaknesses are often exploited through software or firmware. Therefore, to maximize assurance and minimize operationally impactful risks mitigation-in-depth across all constituent components must be considered. This highlights the need for a dynamic risk profile.

An example of a post-manufacturing mitigation involves a new hardware risk identified during field operation. A dynamic risk profile can be used to characterize the issue and identify possible resources to address the suspect component function. This profile can also be used to track and address risks throughout its life, including obsolescence-related risk. One means of mitigating this kind of hardware life cycle risk is the use of existing programmable logic.

Just as with software patches and updates, new attack surfaces on hardware may become exposed through the mitigation being applied, and they will likely take a long time to discover. In the example above, the programmable logic is updated to provide a new configuration to protect the hardware. In this context, access to hardware reconfiguration must be limited to authorized parties to prevent an unauthorized update that introduces weaknesses on purpose or by accident. While programmable logic may have mitigated a specific attack surface or type of weakness, additional mitigations are needed to minimize risk more completely. This is mitigation-in-depth – multiple mitigations building upon one another.

Throughout the entire supply chain, critical pieces of

information can be inadvertently exposed. The exposure of such information directly enables the creation and exploitation of new attack surfaces. Therefore, the supply chain infrastructure must also be assessed for weaknesses, and the development, use, and maintenance of hardware components assured. The dynamic risk profile offers a framework to balance mitigations in the context of risk and cost throughout the complete hardware and system life cycles.

References

Works Cited

Eames, B.K. and M.H. Johnson. 2017. “Trust Analysis in FPGA-based Systems.” Proceeding of GOMACTech 2017, March 20-23, 2017, Reno, NV.

Graf, J. 2017. “OpTrust: Software for Determining Optimal Test Coverage and Strategies for Trust.” Proceedings of GOMACTech 2017, March 20-23, 2017, Reno, NV.

Martin, R.A. 2014. “Non-Malicious Taint: Bad Hygiene is as Dangerous to the Mission as Malicious Intent.” CrossTalk Magazine. 27(2).

MITRE. 2020. “Common Vulnerabilities and Exposures.” Accessed March 31, 2021. Last Updated December 11, 2020. Available: https://cve.mitre.org/cve/

Myerson, R.R. 1991. *Game Theory: Analysis of Conflict*. Cambridge, MA: Harvard University Press.

NIST. 2020. Roots of Trust. Accessed March 31, 2021. Last Updated June 22, 2020. Available: https://csrc.nist.gov/projects/hardware-roots-of-trust

Oberg, J. 2020. Reducing Hardware Security Risk. Accessed March 31, 2021. Last Updated July 1, 2020. Available: https://semiengineering.com/reducing-hardware-security-risk/

SAE. 2016. SAE AS6171, *Test Methods Standard: General Requirements, Suspect/Counterfeit, Electrical, Electronic, and Electromechanical Parts.* SAE International. Accessed March 31, 2021. Available: https://www.sae.org/standards/content/as6171/

SAE. 2021. SAE J1739_202101, *Potential Failure Mode and Effects Analysis (FMEA) Including Design FMEA*,

Supplemental FMEA-MSR, and Process FMEA. SAE International. Accessed March 31, 2021. Last Updated January 13, 2021. Available: https://www.sae.org/standards/content/j1739_202101/

Primary References

Bhunia, S. and M. Tehranipoor. 2018. *Hardware Security: A Hands-on Learning Approach*. Amsterdam, Netherlands: Elsevier Science.

ENISA. 2017. *Hardware Threat Landscape and Good Practice Guide. Final Version 1.0*. European Union Agency for Cybersecurity. Accessed March 31, 2021. Available: https://www.enisa.europa.eu/publications/hardware-threat-landscape

TAME Steering Committee. 2019. *Trusted and Assured Microelectronics Forum Working Group Reports*. Accessed March 31, 2021. Last Updated December 2019. Available: https://dforte.ece.ufl.edu/wp-content/uploads/sites/65/2020/08/TAME-Report-FINAL.pdf

Additional References

DARPA. A DARPA Approach to Trusted Microelectronics. Accessed March 31, 2021. Available: https://www.darpa.mil/attachments/Background_FINAL3.pdf

Fazzari, S. and R. Narumi. 2019. *New & Old Challenges for Trusted and Assured Microelectronics*. Accessed March 31, 2021. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/1076110.pdf

IEEE. 2008-2020. IEEE International Symposium on Hardware Oriented Security and Trust (HOST). Annual symposium held since 2008 providing wealth of articles on hardware assurance.

Martin, R. 2019. "Hardware Assurance and Weakness Collaboration and Sharing (HAWCS)." Proceedings of the 2019 Software and Supply Chain Assurance Forum, September 17-18, 2019 in McLean, VA. Accessed March 31, 2021. Available: https://csrc.nist.gov/CSRC/media/Projects/cyber-supply-chain-risk-management/documents/SSCA/Fall_2019/WedPM2.2_Robert_Martin.pdf

NDIA. 2017. Trusted Microelectronics Joint Working Group: Team 3 White Paper: Trustable Microelectronics Standard Products. Accessed March 31, 2021. Available: https://www.ndia.org/-/media/sites/ndia/divisions/working-groups/tmjwg-documents/ndia-tm-jwg-team-3-white-paper-finalv3.ashx

Regenscheid, A. 2019. NIST SP 800-193, *Platform Firmware Resiliency Guidelines*. Accessed March 31, 2021. Available: https://csrc.nist.gov/publications/detail/sp/800-193/final

Ross, R., V. Pillitteri, R. Graubart, D. Bodeau, R. McQuaid. 2019. NIST SP 800-160 Vol. 2, *Developing Cyber Resilient Systems – A Systems Security Engineering Approach*. Accessed March 31, 2021. Available: https://csrc.nist.gov/News/2019/sp-800-160-vol2-developing-cyber-resilient-systems

---

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.5, released 15 October 2021
