Author Speaker: Rachana Maitra, Senior Principal Engineer, Corporate SDL, Marvell Technology

Title: Testing beyond spec for product security assurance!

Abstract: This paper will bring to light the aspects of product security testing not covered by conventional hardware verification/validation methods and explain the challenges stemming from the fundamental shift in mindset in trying to break a product like a hacker. Contrary to looking for the presence of functional correctness and compliance to power and performance expectations per spec, security testing should ensure the absence of behavior or characteristics that a hacker could utilize for malicious intent such as stealing or counterfeiting IP, disrupting, or corrupting functionality, leaking secret key or confidential information, etc. For this reason, when creating a security test plan, the test parameters or what to look for while analyzing test results must not be bounded by spec. One must also accept that, the threat landscape is continually evolving, and the sophistication of bad actors is exponentially growing. Hence, security testing strategies must also be open to continuous improvement in test methods and tools. This paper will demonstrate how Marvell is building best known method (BKM) to address these challenges, as part of corporate initiatives to continuously improve its security development lifecycle process (SDL). While elaborating the concept of testing beyond spec, it will enforce the need for a layered approach to testing to achieve product security assurance before production. It will do so by categorizing security testing requirements throughout various pre- and post-silicon development phases, with strategies and tools appropriate for the phase. In every phase, the goal of the testing will be to intentionally look for potential security violations that may remain hidden within a functionally clean design!

Introduction

Security development lifecycle (SDL) is a proactive process for secure product development applicable to all products at Marvell. It is an essential part of Marvell's charter to move, store and *secure* the world's data with semiconductor solutions. Adhering to industry standards, it ensures product security requirements are analyzed and approved, mitigations for potential vulnerabilities are designed in and *tested* before the product is released to the customer. This paper will summarize Marvell's approach to ensure resilience in product security by continuously improving hardware security verification (pre-silicon testing) and validation (post-silicon testing) strategies, as an integral part of overall product development.

Challenge

The first challenge of product security testing starts with product security planning and implementation itself. Accepting the fact that the hacker mindset is unbounded, the planning of security features should start at the product definition phase. The security plan begins by analyzing probable threats of the target market and customer use cases, as guided by the application and associated security objectives of the product. This exercise is referred to as threat modeling during architecture development within the SDL process. The threat modeling of a product should be conducted considering its system with goal to implement layers of defense (with hardware, firmware, or both) from the outer to inner boundaries of the product to protect all its security sensitive assets and overall function. To ensure a robust product security implementation, the challenge lies in not being bounded by the specification of the security features themselves or the overall product. As the goal of security testing is to find failures violating the three key security properties (CIA: Confidentiality-Integrity-Availability), which may exist in a functionally clean product that adequately even meets specified performance and power expectations, we need to have test coverage beyond specified functionality and silicon process-voltage-temperature (PVT) characteristics. We must also use specialized tools that have security specific lenses.

The second challenge of security testing comes from the potential for introduction or resurfacing of security vulnerabilities at different phases of product development. For example, optimization at synthesis or silicon fabrication may introduce potential vulnerabilities. Hence as protection from vulnerabilities must be implemented in layers, the testing of it also must be conducted in layers, with phase appropriate methods and tools.

The third challenge of security testing comes from the ever-evolving threat landscape. Product teams must acknowledge the need to retrigger threat modeling, in the event of a change in design or change in threat landscape in target market that may impact security assumptions of the product. If this results in revised implementation of security features, it must be followed by revised testing.

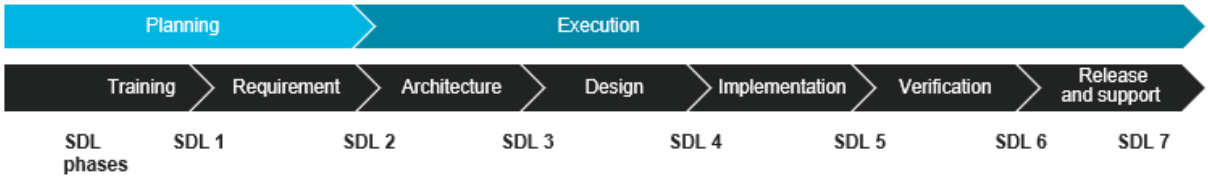
How Marvell is addressing the challenges

Marvell addresses these challenges head-on by

- Integrating SDL into its Product Lifecycle (PLC) and acknowledging, that without product security verification/validation the whole purpose of SDL is defeated
- Adopting shift-left strategy to product security testing and adding phase appropriate coverage up to product release

SDL phases aligned to PLC

While product security testing must be completed by verification/validation phase for production release and mass production approval, it is kicked off as early as in architecture phase at initial code development.



- Prioritizing security testing and repurposing conventional methods to intentionally look for common vulnerabilities based on STRIDE

STRIDE

Mnemonic developed by Microsoft widely used for modeling threats to system

	Threat	Property Violated	Threat Definition
S	Spoofing identify	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying something on disk, network, memory, or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information disclosure	Confidentiality	Providing information to someone not authorized to access it
D	Denial of service	Availability	Exhausting resources needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to do

- Adding beyond spec parameters to uncover hidden loopholes that can be taken advantage of by hackers for malicious intent

- Partnering with industry experts and consultants to evaluate products with specialized tools, including enhanced automation and AI capability focused on ensuring resilience in security

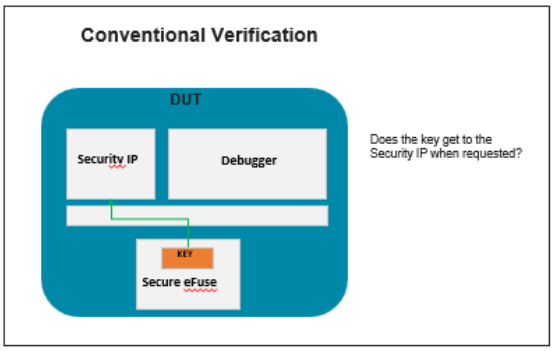
Security sensitive coverage prioritization in conventional verification/validation

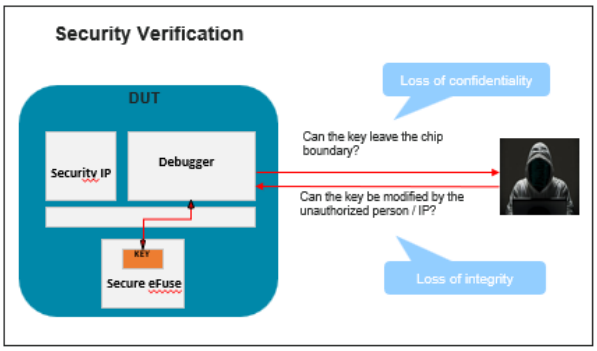
- Functional robustness of all security sensitive critical assets (Ex: Root of Trust IP, Secure BOOT hardware or firmware, Hardware security module, Secure processor and memory, Anti tamper module, Debugger), chip interconnects (Ex: Fabric, DMA), system interfacing peripherals (Ex: I/O ports, Debug ports) etc.
- Negative conditions (error detection features) at all security boundaries
- Silicon robustness of all components enlisted in threat model across spec PVT range
- System level stress of threat model with random input (hardware and firmware fuzz testing)

Inspection beyond expected behavior to ensure absence of unexpected

Inspection of what should not happen, typically not covered by conventional test methods

- Who should not be able to grant access - Check for wrongful authority of agents permitting access to specific block
- Who should not be able to access - Check for unauthorized agents accessing specific block
- Where data should not go - Check for secure data propagating to non-secure region
- What should not be accessible - Check for secure blocks being accessible in non-secure mode

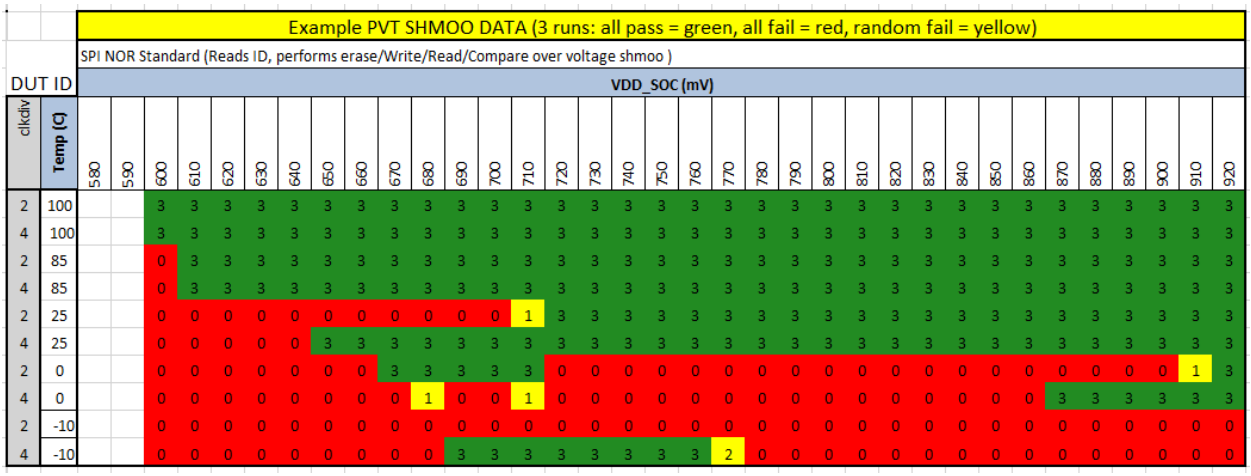




Coverage addition beyond spec

Coverage beyond functional correctness, typically not included in conventional test plans

- Invalid input (don't care logic) in directed testing (focus test) and random testing (fuzz testing) of Security features and security sensitive interfaces
- Long duration and iterative stress of negative conditions (error detection features) beyond use case limits
- Threat model component characterization beyond specified silicon target. Shmoo voltage and temperature on corner silicon down to failure (example of a PVT shmoo shown in chart below). Inspect all random or rare failures within passing range, followed by absolute failures at margin.



Example PVT SHMOO DATA (3 runs: all pass = green, all fail = red, random fail = yellow)

SPI NOR Standard (Reads ID, performs erase/Write/Read/Compare over voltage shmoo)

DUT ID		VDD_SOC (mV)																																						
clkdiv	Temp (C)	580	590	600	610	620	630	640	650	660	670	680	690	700	710	720	730	740	750	760	770	780	790	800	810	820	830	840	850	860	870	880	890	900	910	920				
2	100			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3				
4	100			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3				
2	85			0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3				
4	85			0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3				
2	25			0	0	0	0	0	0	0	0	0	0	0	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3				
4	25			0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3				
2	0			0	0	0	0	0	0	0	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3				
4	0			0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	3				
2	-10			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
4	-10			0	0	0	0	0	0	0	0	0	3	3	3	3	3	3	3	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Security testing throughout product development

In sequence

- Linting at code development
- Formal testing at IP level
- Simulation at block and full chip level
- Post-synthesis netlist inspection at gate level
- Emulation at full chip and system level
- System stress on silicon
- Silicon characterization across PVT

Inclusion of toolset with security specific lens

In partnership with specialized suppliers

- Static lint tool for detecting security rule violations in RTL code
- Formal tool for detecting potential vulnerability in IP connectivity
- Dynamic tools for data leak or SCA or FIA vulnerability analysis in pre-silicon, at RTL and gate level
- Dynamic tools for data leak or SCA or FIA vulnerability analysis in post-silicon, at block and system level

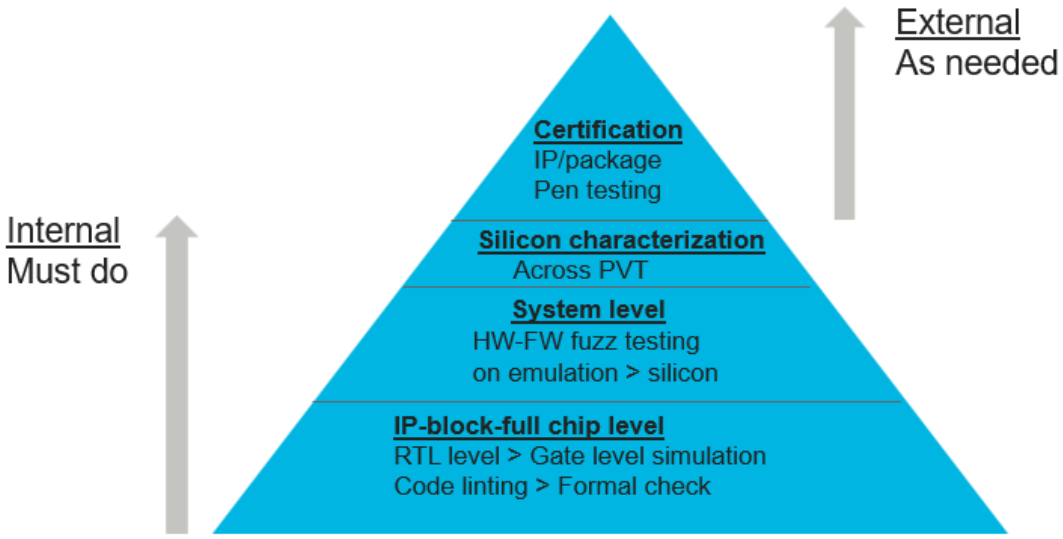
Certification from external security consultants

As needed

- Closed or gray box testing by external ethical hackers or pen testing service providers
- IP or product or package certification by security consultants or certified labs (Ex: NIST)

Security testing pyramid

From coding to production release



Conclusion

Marvell is committed to security assurance of its products to protect itself and its customer from the ever-evolving threat landscape. Through a comprehensive approach to product security verification and validation, Marvell is committed to not only differentiating its products, but also to partner with industry for continuous improvement in strategy and toolset to combat against the growing sophistication of bad actors affecting the entire industry.

References

1. IEEE-HOST 2024 paper by Professor Prabhat Mishra and Ankur Srivastava: Hardware Security and Trust verification
2. Article by Anders Nordstrom (Principal Engineer – Cycuity) on Data leak and timing side channel attack: Timing is of the Essence in Hardware Security
3. Course on Fault injection attack: https://course.ece.cmu.edu/~ece749/docs/faultInjectionSurvey.pdf
4. Articles on threat modeling:
    a. https://en.wikipedia.org/wiki/STRIDE_model
    b. https://www.synopsys.com/content/dam/synopsys/sig-assets/whitepapers/wp-threat-modeling-decoded.pdf

Appendix

Acronyms

SCA	Side Channel Attack
FIA	Fault Injection Attack
IP	Intellectual Property
RTL	Register Transfer Level
DUT	Device Under Test
HW	Hardware
FW	Firmware
AI	Artificial Intelligence
NIST	National Institute of Standards and Technology

Disclaimers

Product security can never be 100% guaranteed.

Marvell representative should be contacted for product specific implementation of security verification/validation strategy.