NIST Cybersecurity White Paper
NIST CSWP 45

Metrics and Methodology for Hardware Security Constructs

Sanjay (Jay) Rekhi
Kostas Amberiadis
Computer Security Division
Information Technology Laboratory

Abir Ahsan Akib
Ankur Srivastava
Electrical and Computer Engineering
University of Maryland, College Park

This publication is available free of charge from:
https://doi.org/10.6028/NIST.CSWP.45

June 5, 2025

NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Publication History
Approved by the NIST Editorial Review Board on 2025-02-25

How to Cite this NIST Technical Series Publication:
Rekhi S, Amberiadis K, Akib AA, Srivastava A (2025) Metrics and Methodology for Hardware Security Constructs. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Cybersecurity White Paper (CSWP) NIST CSWP 45. https://doi.org/10.6028/NIST.CSWP.45

Author ORCID iDs
Sanjay (Jay) Rekhi: 0009-0008-8711-4030
Kostas Amberiadis: 0009-0000-7771-5002
Abir Ahsan Akib: 0000-0002-1455-6662
Ankur Srivastava: 0000-0002-5445-904X

Contact Information
hwsec@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Additional Information
Additional information about this publication is available at https://csrc.nist.gov/publications/cswp, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

Although hardware is commonly believed to be security-resilient, it is often susceptible to vulnerabilities that arise from design and implementation flaws. These flaws can jeopardize the hardware's security, its operations, and critical user information. This investigation presents a comprehensive methodology for assessing threats related to different hardware weaknesses and the attacks that can exploit them. The methodology results in two key metrics: a threat metric that quantifies the number of hardware weaknesses that an attack can exploit and a sensitivity metric that measures the number of distinct attacks that can target a hardware system with a specific weakness. These metrics and the accompanying analysis aim to guide security efforts and optimize the trade-offs between hardware security and associated costs.

Keywords

CAPEC attack patterns; CWE most important hardware weaknesses; security metrics.

Table of Contents

List of Tables

List of Figures

Introduction

Hardware is often regarded as more security-resilient than software because physical components are more difficult to manipulate. Hardware designers also have greater control over implementation details, which enables them to mitigate certain critical attacks.

However, weaknesses can be introduced during the design and manufacturing stages that cause hardware to leak sensitive information and compromise the accuracy of operations. For example, Spectre [1], Meltdown [2], Inception [3], Downfall [4], and Foreshadow [5] are well-known vulnerabilities that show how hardware weaknesses can be exploited in both controlled laboratory environments and real-life scenarios. Developing more secure hardware for the future requires ongoing research into different hardware weaknesses and the techniques used to exploit them.

With a goal of developing exploitation and mitigation strategies, the Hardware Common Weakness Enumeration Special Interest Group (HW CWE SIG) has analyzed 108 different hardware weaknesses that originate from hardware design issues [6]. CWE categorizes and maintains hardware and software weaknesses separately, since they and their exploitation and mitigation techniques are fundamentally different. Additionally, the Common Attack Pattern Enumeration and Classification (CAPEC) established by the U.S. Department of Homeland Security [7] categorizes possible existing attack patterns in hardware, software, communications, supply chains, social engineering, and physical security. CAPEC also analyzes the likelihood of such attacks being launched, their potential severity and flow of execution, and the skills that an adversary is likely to need.

This work utilizes a comprehensive methodology and two key metrics—threat and sensitivity—to analyze different hardware weaknesses (Sec. 3) and the specific attack patterns that can exploit them (Sec. 4). Section 5 discusses the resources needed to launch various attack patterns. Section 6 contains the overall framework for this analysis.

1. Motivation

Mell and Bojanova systematically analyzed potential hardware weaknesses identified by CWE and presented various security failure scenarios [8]. Their work explored the origins of these weaknesses and highlighted the potential consequences and implications of hardware security threats, such as unauthorized access to restricted information due to insufficient security features, cryptographic output decryption, circumventing security protections, and premature hardware failures. The real-world occurrence of these threats raises significant security concerns and underscores the need for further research to quantify risks and establish standardized metrics for consideration in integrated circuit design.

This document builds on that analysis to introduce a comprehensive methodology for analyzing various attack strategies that exploit vulnerabilities. The approach quantifies the sensitivity of hardware weaknesses by measuring the number of attack methodologies that can exploit them. An additional metric assesses the threat level of potential attacks based on the number of weaknesses they can target. Designed to be scalable and adaptable, this methodology can evolve with emerging attack patterns and hardware vulnerabilities to support the development of more secure future hardware designs while aiding in the identification and mitigation of vulnerabilities in existing systems.

2. Hardware Weaknesses and Their Security Implications

Hardware is designed using software tools with complex encodings that could lead to the creation of bugs and weaknesses. Some of these weaknesses might cause the hardware to produce erroneous responses or stop working altogether, while others may be detected during testing and mitigated before the hardware reaches the market. However, some weaknesses do not directly affect the hardware's operation and instead allow security or sensitive data to be compromised. This work provides a methodology for analyzing such weaknesses in the hardware.

The HW CWE SIG evaluated 108 different hardware weaknesses [6] based on various factors, such as the frequency with which a weakness is detected and whether hardware modifications are necessary for mitigation. Based on that evaluation, HW CWE SIG identified the most important hardware weaknesses presented, as shown in Table 1.

Table 1. CWE most important hardware weaknesses

ID	Description
CWE-1272	Sensitive Information Uncleared Before Debug/Power State Transition
CWE-1300	Improper Protection of Physical Side Channels
CWE-1189	Improper Isolation of Shared Resources on System-on-a-Chip (SoC)
CWE-1244	Internal Asset Exposed to Unsafe Debug Access Level or State
CWE-1191	On-Chip Debug and Test Interface with Improper Access Control
CWE-1231	Improper Prevention of Lock Bit Modification
CWE-1233	Security-Sensitive Hardware Controls with Missing Lock Bit Protection
CWE-1274	Improper Access Control for Volatile Memory Containing Boot Code
CWE-1260	Improper Handling of Overlap Between Protected Memory Ranges
CWE-1240	Use of a Cryptographic Primitive with a Risky Implementation
CWE-1256	Improper Restriction of Software Interfaces to Hardware Features

These hardware weaknesses have been used to demonstrate a proposed methodology of analysis that would apply to all other hardware weaknesses.

3. CAPEC Attack Patterns

A critical stage in security analysis is identifying how different hardware weaknesses might be exploited by existing attack patterns. CAPEC is an initiative from the MITRE Corporation that provides a comprehensive list of 559 known attack patterns [7] that are classified based on the nature of attack (e.g., hardware, software, communication attack patterns). The class of interest for this work is hardware attack patterns, which have further sub-classifications.

3.1. Meta Attack Patterns

In CAPEC, a meta-level attack pattern is an abstract, high-level description of an attack methodology or technique that does not specify a particular technology or implementation. For example, a meta-level attack pattern can describe privilege escalation, where an adversary exploits a vulnerability to raise their privilege and take unauthorized actions. However, it does not describe the specific techniques that can be used to escalate that privilege. Therefore, meta-level attack patterns are most helpful during threat analysis activities at the architecture and design levels. Table 2 shows the list of meta-level attack patterns that correlate to one or more of the most important hardware weaknesses shown in Table 1.

Table 2. CAPEC meta-level attack patterns

ID	Description
26	Leveraging Race Conditions
113	Interface Manipulation
114	Authentication Abuse
122	Privilege Abuse
124	Shared Resource Manipulation
176	Configuration/Environment Manipulation
188	Reverse Engineering
192	Protocol Analysis
233	Privilege Escalation
441	Malicious Logic Insertion
624	Hardware Fault Injection

3.2. Standard Attack Patterns

In CAPEC, standard-level attack patterns are a sub-group of meta-level attack patterns that focus on a particular attack methodology or technique. For example, a standard attack pattern can describe a subversion of code-signing mechanisms, which falls under the meta-level attack pattern of privilege escalation. Code signature facilities are used by several programming languages to verify the identity of code and link it to its designated privileges in an appropriate environment. Subverting this mechanism can be instrumental to an attacker escalating privilege. Thus, a standard-level attack pattern describes a specific mechanism for escalating privilege. Table 3 shows a list of standard attack patterns that are sub-classes of one of the meta-level attack patterns in Table 2.

Table 3. CAPEC standard-level attack patterns

ID	Description
167	White Box Reverse Engineering
189	Black Box Reverse Engineering
121	Exploit Non-Production Interfaces
36	Using Unpublished Interfaces or Functionality
1	Accessing Functionality Not Properly Constrained by ACLs
180	Exploiting Incorrectly Configured Access Control Security Level
68	Subvert Code-signing Facilities
452	Infected Hardware
456	Infected Memory
97	Cryptanalysis
625	Mobile Device Fault Injection

4. Resources Required to Exploit CAPEC Attack Patterns

Adversaries require specific resources to exploit a hardware weakness or launch an attack: knowledge and access.

The knowledge level of an adversary is divided into two classes:

1. **High knowledge level:** Adversaries are considered to have a high knowledge level if they have prior experience working with a large number of designs that could help them identify the nature of the design under attack or if they have some knowledge about the internal workings of a chip (e.g., how some hardware is shared across programs).
2. **Low knowledge level:** Adversaries with no such experience are considered to have a low knowledge level.

An attack that can be launched by an adversary with a low knowledge level can also be launched by an adversary with a high knowledge level.

The access level of an adversary is classified into three groups:

1. **Black-box access:** The adversary has a working system with which they can provide an input and observe the output.
2. **Gray-box access:** The adversary has a working system and readily available information from some internal registers without having to launch an attack.
3. **White-box access:** The adversary knows everything about the internal workings of the design.

These access levels are hierarchical, meaning that an attack that can be launched using black-box access can also be launched with gray- or white-box access.

5. Methodology for Hardware Weakness Threat Analysis

This document proposes a systematic approach to analyzing hardware security threats, designed to adapt to newly discovered attack patterns and hardware weaknesses. The framework establishes connections between hardware weaknesses and specific attack techniques or standard attack patterns. By mapping these relationships, it identifies how many attack techniques can exploit a given hardware weakness and how many weaknesses a particular attack can target, helping prioritize mitigation efforts effectively.

A comprehensive analysis would involve an exhaustive search to evaluate all known attack patterns against each hardware weakness for potential exploitability. As new attack patterns emerge, this approach would require repeated, extensive evaluations, making it inefficient and inflexible. Instead, a more scalable and adaptable methodology is necessary to keep pace with the evolving landscape of hardware security. Mapping the relationships between hardware weaknesses and attack patterns can help prioritize mitigation efforts more efficiently.

This work presents a systematic yet versatile methodology for linking hardware weaknesses to both known and emerging attack patterns. The CAPEC database provides limited mappings between CWE-defined hardware weaknesses and corresponding attack patterns, and it does not encompass the full spectrum of vulnerabilities. For instance, CWE-1274 describes a weakness involving improper access control for volatile memory that contains boot code. This situation arises when a device lacks adequate safeguards during the secure boot process, specifically when the bootloader is transferred from non-volatile to volatile memory. One possible exploitation method aligns with CAPEC-1, which involves accessing functionality that is not adequately protected by access control mechanisms. This attack pattern includes threats that target memory protections, hardware registers, and hardware-based identifiers. However, CAPEC does not currently map CWE-1274 to CAPEC-1, illustrating a gap in the coverage. Additionally, manually performing such mappings is both time-consuming and difficult to maintain in the face of evolving attack vectors. The proposed framework provides a more comprehensive and scalable approach to mapping hardware vulnerabilities to relevant attack patterns.

The first step of the methodology involves identifying meta-level attack patterns and the hardware weaknesses they could exploit. For example, meta-level attacks 188, 124, and 26 can exploit hardware weakness CWE-1189, as shown in Table 4. Since meta-level attack patterns represent a generalized class of attack strategies rather than specific implementation techniques, linking hardware weaknesses to meta-level attack patterns is more efficient and less time-consuming.

Table 4. Mapping of most common hardware weaknesses and CAPEC meta-level attack patterns

		CAPEC Meta Attack Patterns										
		188	**124**	**26**	**113**	**114**	**122**	**233**	**176**	**441**	**192**	**624**
CWE Most Important Hardware Weakness	**CWE-1272**	1	0	0	0	0	0	0	0	0	0	0
	CWE-1300	1	0	0	0	0	0	0	0	0	0	0
	CWE-1189	1	1	1	0	0	0	0	0	0	0	0
	CWE-1244	0	0	0	1	1	0	0	0	0	0	0
	CWE-1191	0	0	0	1	0	1	1	0	0	0	0
	CWE-1231	0	0	0	0	0	1	0	1	0	0	0
	CWE-1233	0	0	0	0	0	1	0	1	0	0	0
	CWE-1274	0	0	0	0	0	1	1	0	1	0	0
	CWE-1260	0	0	1	0	0	1	0	0	1	0	0
	CWE-1240	0	0	0	0	0	0	1	0	0	1	0
	CWE-1256	0	0	0	0	0	1	0	0	0	0	1

The second step of this methodology analyzes the relationships between meta-level attack patterns and their corresponding standard attack patterns, as shown in Table 5.

Table 5. Mapping of CAPEC meta and standard attack patterns

		CAPEC Standard Attack Patterns										
		167	**189**	**121**	**36**	**1**	**180**	**68**	**452**	**456**	**97**	**625**
CAPEC Meta Attack Patterns	**188**	1	1	0	0	0	0	0	0	0	0	0
	124	0	0	0	0	0	0	0	0	0	0	0
	26	0	0	0	0	0	0	0	0	0	0	0
	113	0	0	1	1	0	0	0	0	0	0	0
	114	0	0	0	0	0	0	0	0	0	0	0
	122	0	0	0	0	1	1	0	0	0	0	0
	233	0	0	0	0	0	0	1	0	0	0	0
	176	0	0	0	0	0	0	0	0	0	0	0
	441	0	0	0	0	0	0	0	1	1	0	0
	192	0	0	0	0	0	0	0	0	0	1	0
	624	0	0	0	0	0	0	0	0	0	0	1

For example, standard attack patterns 167 and 189 fall under meta-level attack pattern 188. Mapping a newly discovered attack technique to an existing generalized class of attacks is a relatively straightforward process.

The third step of the methodology involves performing a simple matrix multiplication to identify correlations between the most common hardware weaknesses and standard attack patterns. Table 6T shows the result of multiplying the matrices in Table 4 and Table 5.

Table 6. Correlation matrix of most important hardware weaknesses and CAPEC standard attack patterns

		CAPEC Standard Attack Patterns											
		167	**189**	**121**	**36**	**1**	**180**	**68**	**452**	**456**	**97**	**625**	**Sensitivity**
CWE Most Common Hardware Weakness	**CWE-1272**	1	1	0	0	0	0	0	0	0	0	0	**2**
	CWE-1300	1	1	0	0	0	0	0	0	0	0	0	**2**
	CWE-1189	1	1	0	0	0	0	0	0	0	0	0	**2**
	CWE-1244	0	0	1	1	0	0	0	0	0	0	0	**2**
	CWE-1191	0	0	1	1	1	1	1	0	0	0	0	**5**
	CWE-1231	0	0	0	0	1	1	0	0	0	0	0	**2**
	CWE-1233	0	0	0	0	1	1	0	0	0	0	0	**2**
	CWE-1274	0	0	0	0	1	1	1	1	1	0	0	**5**
	CWE-1260	0	0	0	0	1	1	0	1	1	0	0	**4**
	CWE-1240	0	0	0	0	0	0	1	0	0	1	0	**2**
	CWE-1256	0	0	0	0	1	1	0	0	0	0	1	**3**
	Threat	**3**	**3**	**2**	**2**	**6**	**6**	**3**	**2**	**2**	**1**	**1**	

When a new attack technique is discovered, a column is added to Table 5. The matrix multiplication then updates Table 6 to automatically link hardware weaknesses to standard attacks. This approach simplifies the otherwise complex and time-consuming process of directly mapping hardware weaknesses to specific exploit techniques by breaking it into two manageable tasks. As a result, the methodology becomes more robust, scalable, and efficient.

The correlation matrix in Table 6 shows how sensitive hardware becomes to attacks in the presence of a hardware weakness. It also shows how an attack can exploit multiple weaknesses. From the interpretation of the correlation matrix, two metrics are proposed. The threat metric counts the number of different hardware weaknesses that a specific attack pattern can exploit. The higher the number, the greater the threat. The sensitivity metric counts the number of different standard attack patterns than can exploit a particular weakness. The hardware's vulnerability increases with the number of attack patterns that can exploit it. Figure 1 and Fig. 2 show the metrics with respect to CAPEC standard attack patterns and CWE most important hardware weakness, respectively.

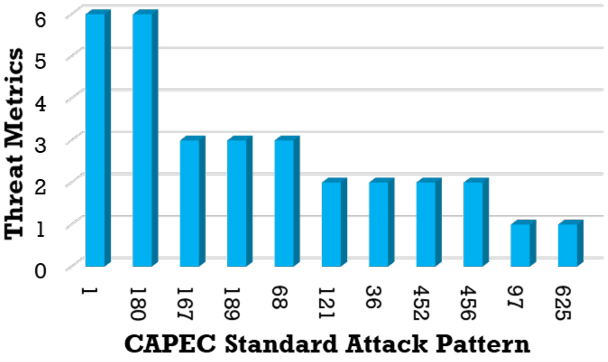

Fig. 1. Threat metrics of CAPEC standard attack patterns

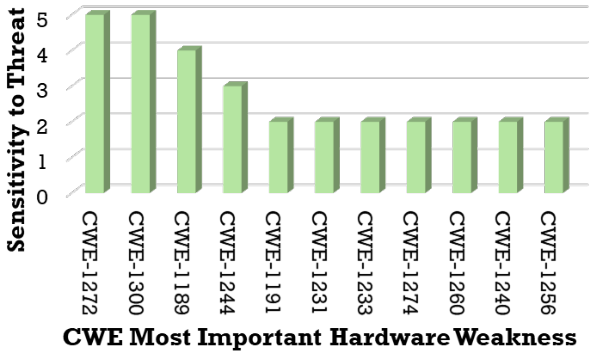

Fig. 2. Threat sensitivity of CWE most important hardware weaknesses

While the matrices are binary for simplicity (i.e., a weakness can either be linked to an attack pattern or not), weights can be selected based on a variety of factors, such as an attack's probability of success, potential impact, and the resources required to launch it. These weights can be subjective and dependent on user requirements.

The fourth and final step of the methodology is identifying the resources required to launch a standard attack, as shown in Fig. 3.

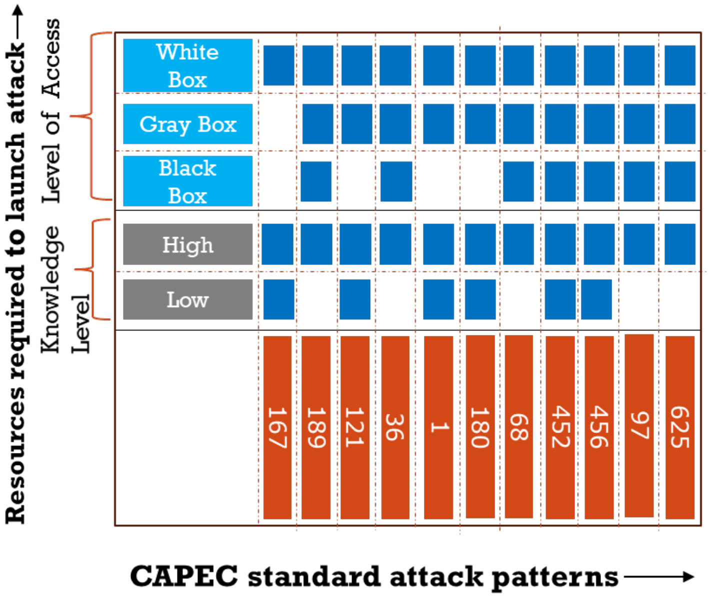

Fig. 3. Resource mapping of CAPEC standard attack patterns

For example, white-box access is required to launch white-box reverse engineering attack 167 against an exploitable weakness. However, black-box access is needed to launch black-box reverse engineering attack 189. If there is a higher probability for an adversary to have black-box access than white-box access, mitigation of 189 should be a higher priority than mitigation of 167.

6. Conclusion

This work addresses the critical issue of threats that arise from hardware weaknesses by presenting a comprehensive methodology to quantify the vulnerabilities that these weaknesses introduce and the potential threats from various attacks that can exploit them. The methodology helps designers understand different hardware weaknesses and attack patterns in order to prioritize and plan effective mitigation efforts.

Given the evolving nature of hardware security, new weaknesses and attack strategies will inevitably emerge. This work emphasizes the importance of continuously updating and expanding the analysis, and the methodology's robust design allows for seamless modifications. While this work focuses on hardware weaknesses, the methodology is applicable to all identified vulnerabilities, including those that may arise in the future. As such, the findings contribute to a growing body of knowledge to help mitigate existing vulnerabilities and guide the development of more secure hardware architectures. Ongoing research will be crucial in supporting decision-making and design considerations for security-resilient hardware.

References

[1] Kocher P, Horn J, Fogh A, Genkin D, Gruss D, Haas W, Hamburg M, Lipp M, Mangard S, Prescher T, Schwarz M, Yarom Y (2019) Spectre attacks: Exploiting speculative execution. *2019 IEEE Symposium on Security and Privacy (SP 2019)* (IEEE, San Francisco, California), pp 1-19. https://doi.org/10.1109/SP.2019.00002

[2] Lipp M, Schwarz M, Gruss D, Prescher T, Haas W, Fogh A, Horn J, Mangard S, Kocher P, Genkin D, Yarom Y, Hamburg M (2018) Meltdown: Reading kernel memory from user space. *Proceedings of the 27th USENIX Security Symposium* (USENIX, Baltimore, Maryland), pp 973-990. Available at https://www.usenix.org/conference/usenixsecurity18/presentation/lipp

[3] Advanced Micro Devices (2024) Return Address Security Bulletin, AMD-SB-7005. Available at https://www.amd.com/en/resources/product-security/bulletin/amd-sb-7005.html

[4] Moghimi D (2023) Downfall: Exploiting Speculative Data Gathering. *Proceedings of the 32nd USENIX Security Symposium* (USENIX, Anaheim, California), pp 7179–7193. Available at https://www.usenix.org/conference/usenixsecurity23/presentation/moghimi

[5] Van Bulck J, Minkin M, Weisse O, Genkin D, Kasikci B, Piessens F, Silberstein M, Wenisch TF, Yarom Y, Strackx R (2018) Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. *Proceedings of the 27th USENIX Security Symposium* (USENIX, Baltimore, Maryland), pp 991-1008. Available at https://www.usenix.org/conference/usenixsecurity18/presentation/bulck

[6] MITRE, HSSEDI (2025) CWE - Common Weakness Enumeration. Available at https://cwe.mitre.org/index.html

[7] MITRE, HSSEDI (2025) CAPEC - Common Attack Pattern Enumeration and Classification. Available at https://capec.mitre.org/

[8] Mell P, Bojanova I (2024) Hardware Security Failure Scenarios. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Internal or Interagency Report (IR) NIST IR 8517. https://doi.org/10.6028/NIST.IR.8517