Metrics for Assessing Security of System-on-Chip

Sujan Kumar Saha, Joel Mandebi Mbongue and Christophe Bobda
Dept. of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32603
Email: {sujansaha, jmandebimbongue}@ufl.edu and cbobda@ece.ufl.edu

***Abstract*—Due to the increasing complexity of modern heterogeneous System-on-Chips (SoC) and the growing vulnerabilities, security risk assessment and quantification is required to measure the trustworthiness of a SoC. This paper describes a systematic approach to model the security risk of a system for malicious hardware attacks. The proposed method uses graph analysis to assess the impact of an attack and the Common Vulnerability Scoring System (CVSS) is used to quantify the security level of the system. To demonstrate the applicability of the proposed metric, we consider two open source SoC benchmarks with different architectures. The overall risk is calculated using the proposed metric by computing the exploitability and impact of attack on critical components of a SoC.**

***Index Terms*—SoC, metrics, impact, threat, vulnerability**

I. Introduction

The computation demand of modern System-on-Chips (SoC) is increasing to meet the performance requirement of the diverse software applications. Hence, more dedicated hardware functionalities are integrated into the System alongside CPU core. Nowadays, modern SoCs have multiple CPUs, dedicated crypto cores for encryption/decryption acceleration, security primitives such as Physical Unclonable Function (PUF), True Random Number Generator (TRNG), digital signal processing units, artificial intelligence engine, GPU, FPGA all in a single chip such as Xilinx Versal ACAP, Apple M1, Google TPUs etc. The decreasing size of transistors have made these large scale components integration viable. But, to meet the time-to-market deadlines, many of these components or hardware Intellectual Properties (IP) are developed by third-party vendor companies that do not often consider a complete verification and security concerns while developing those IPs. Hence, the security issues remain inherent to SoCs throughout design phases and production deployment.

Due to the increasing hardware level threats, the design of a secured System-on-Chip is an important task for safety critical applications. Various defense mechanisms are being developed to mitigate those attacks. Still, it is quite challenging to design a SoC which is resistant from all kinds of attacks. Hence, security assessment and quantification are required to measure the trustworthiness of a system. Quantifying security and the cost of related infrastructure requires careful analysis of the system assets and attack models to devise a sound metric. Existing security quantification methods have been developed for software and network security. Very few works have been done considering SoC security [1]. Hence, a systematic approach is required to measure the security level of a SoC for hardware based attacks.

This paper proposes a security assessment and quantification methodology to represent the security risk of an attack performed by a malicious component of the system. We use a graph analysis to model the impact of an attack and the probability theory is applied to measure the impact. The CVSS scoring technique is used to calculate the risk of a critical component. The major contributions of this paper are the followings.

- We represent the System-on-Chip (SoC) components using graph node and interfaces as graph edge to model the impact of an attack.
- The probability theory is used to calculate the impact propagation of a vulnerable node and a protected node when an attack is launched.
- The Common Vulnerability Scoring System (CVSS) based risk metric is proposed.

These techniques provide us a methodology to assess the impact of an attack and represent the security level of a SoC.

The rest of the paper is outlined as follows. We mention the related works in Section II. The proposed security assessment methodology is described in Section III. The Section IV represents the result analysis, and finally, we conclude in Section V.

II. Related Work

So far in literature, different security assessment methodologies have been proposed for software security analysis and network security analysis [11]. Software security analysis mostly relies on vulnerability [6] and malware detection [9], and the corresponding metrics. Goitom et. al. [15] proposed an application security metric by analysing security requirements denoted by goal question metric and the CVSS scoring for quantification. Attack graph based probability metric is used in network security analysis [14]. In [7], Gabriel proposed cyber security situation assessment method by representing the impact of an attack using impact dependency graph. Jin-Hee et. al. discusses the factors contributing to system security, trust, resiliency and agility metrics [3]. Relatively less research has been accomplished in determining hardware security analysis. [12] proposed a hardware trojan vulnerability assessment method using the testability of a circuit. In [4], Cruz et. al. proposes a trust metric for hardware IPs by considering functional coverage, structural coverage and asset coverage. Bulbul et. al. present a thorough analysis of security quantification of individual IP in SoC and discusses the challenges to determine the overall security metric at platform level [1]. In

TABLE I
TRUST-HUB TROJAN BENCHMARK SUMMARY

Trojan Location	Trojan Name	Malicious Functionality
Processor	B15-T100∼T400	DoS
	B19-T300∼T500	Change Functionality
	MC8051	Change Functionality, DoS
	PIC16F84	DoS, Information Leakage
	MULTPYRAMID	Performance Degradation, DoS
	S15850, S35932, S38417	DoS, Change Functionality, Information Leakage, Performance Degradation
	VGA_LCD-T100	Change Functionality, DoS
Memory	MEMCTRL-T100	Performance Degradation, DoS
Micro-UART	RS232	DoS, Functional Change
Crypto IP	AES	Information Leakage, DoS
	BasicRSA	Information Leakage, DoS
I/O	WB_CONMAX	DoS, Change Functionality
	MC8051	Change Functionality
Power Supply	EthernetMAC10GE	DoS
Clock Grid	B15-T100∼T400	DoS
	EthernetMAC10GE	DoS
Ethernet	EthernetMAC10GE	Performance Degradation, DoS, Change Functionality

[8], Sandhya proposed a CVSS scoring based security metric for server hardware architecture. Although many of these techniques propose software and network security metrics, and some of these address the hardware IP level security metric, SoC or platform level security metric has not been adequately addressed in literature yet.

III. SECURITY ASSESSMENT AND METRIC

In this section, we start our discussion by presenting the behaviour of malicious hardware and their impacts on SoC. Then, we model the impact propagation by representing the SoC using a graph. The CVSS scoring metric is used to assess the risk value of critical nodes.

A. Malicious Hardware and vulnerability assessment

Over the last two decades, studies demonstrated that hardware vulnerabilities offer an entry point to launch attacks in a system. To understand the impact of these vulnerabilities and their physical location, we summarize the Trust-hub trojan benchmark [13] suit in Table I. We observe that trojans are inserted in all critical components of a SoC such as processor, memory, Crypto IP, I/O port, power supply unit etc. The impact of these trojan attacks are information leakage, change of functionality, Denial-of-Service (DoS), and performance degradation. We represent information leakage as Confidentiality (C) impact, change functionality as Integrity (I) impact and DoS and performance degradation as Availability (A) impact. Hence, it is concluded that malicious hardware can affect all three components of CIA triad.

Identifying vulnerabilities of an IP is important to predict the impact of an attack due to malicious functionality. Several metrics have been proposed in the literature to evaluate the hardware vulnerabilities in IPs. Some of these are controllability and observability, statement hardness, hard-to-detect, code coverage, observation hardness [1] and Hardware Trojan Horse (HTH) vulnerability metric [12]. In this research, it is assumed that either the IP provider or system integrator will perform the security assessment of the IP and provide the vulnerability score for that IP.

B. Impact Analysis using graph

A SoC can be mapped to a graph where each component of the SoC is represented as vertex and the interfaces are represented as edges. Figure 1 shows a SoC architecture and its graph representation. The graph-tool [5] is used to identify all the possible paths to reach a vertex. When an attack is launched due to a malicious component, the impact is propagated from the source node to the other nodes of the graph. To analyze the impact, directed sub-graphs are created from the original graph. The critical node is identified as the target node of the sub-graph and all other nodes of the sub-graph are considered as source node. The impact of attack on the source node is propagated to the target node. Let, I_s denote the impact probability of the source node due to an attack on that node and the I_t denote the impact probability of the target node due to an attack on the source node. The impact of attack on the target node can be calculated using the following way.

$$I = I_s \times I_{i1} \times I_{i2} ... \times I_t \tag{1}$$

Where, I_i is the impact probability of the nodes along the path from source to target node. Here, the assumption is that the node with higher distance will have less impact on the target node. Hence, the total impact of a critical node is the summation of the impact of other nodes and it's own impact.

$$I_{c,total} = I_c + \sum_{i=1}^{N} I_i \tag{2}$$

Where, N is the number of nodes that have an impact on the critical node. Each individual Impact value is calculated using the CVSS scoring metric mentioned in the following section.

C. CVSS metric in the context of SoC

The Common Vulnerability Scoring System is a standard used in software vulnerability assessment. The vulnerabilities reported in the NIST CVE database are assessed using CVSS metric [10]. The CVSS specification v3.1 describes about three metrics: Base metric, Temporal metric and Environmental metric. First, the Base score is calculated for a software vulnerability and then considering other factors temporal score and environmental score are calculated. In this research, we will use only the Base metric for evaluating SoC level trust. The Base score is defined as the following.

$$BaseScore = Min[(Impact + Exploitability), 10] \tag{3}$$

The impact is calculated using the following equation.

$$Impact = K_1 \times [1 - (1 - Conf) \times (1 - Int) \times (1 - Avail)] \tag{4}$$

The K_1 is set to 6.42 in the CVSS specification. The $Confidentiality$, $Integrity$, and $Availability$ values can

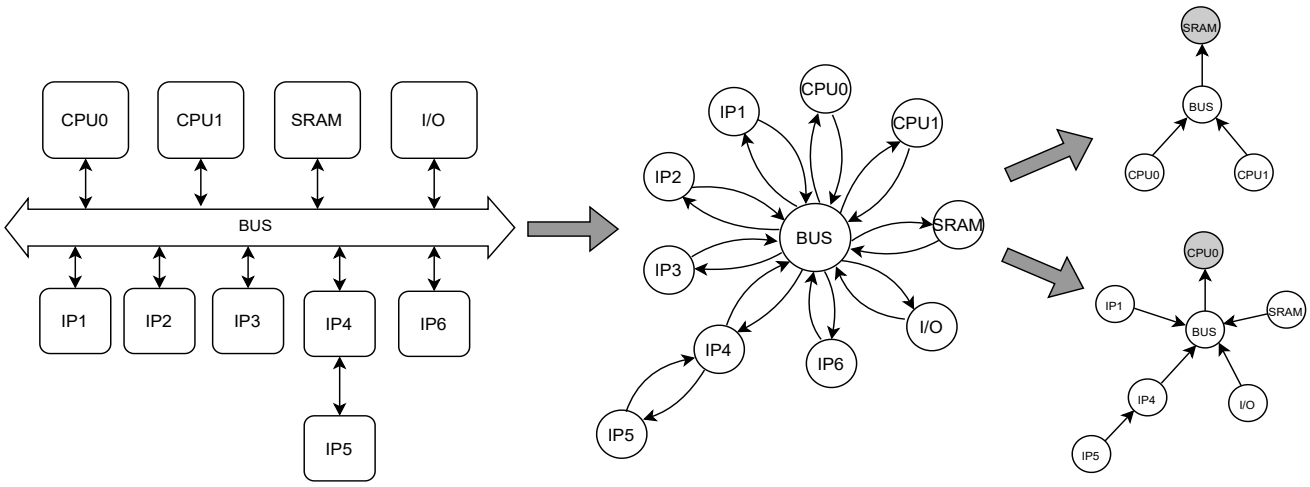

Fig. 1. System-on-Chip Architecture, it's derived graph, and associated sub-graph

be None(0), Low(0.22), and High(0.56) according to CVSS specification. We calculate the impact of an attack of each individual node using equation 4, then multiplied with the corresponding impact probability value of each node and aggregate those to find the overall impact using equation 2 mentioned in the previous subsection.

In CVSS standard, the $Exploitability$ is calculated by multiplying the attack vector, attack complexity, privileges required, and user interaction. But, we modify the $Exploitability$ equation in the context of malicious hardware attack on SoC. As, the exploitability is proportional to the vulnerability in an IP, we use the following equation.

$$Exploitability = K_2 \times V \tag{5}$$

Where, the V is the vulnerability of the considering IP and K_2 is set to 3.89 as mentioned in [8]. Here, the assumption is the vulnerability score coming from the IP provider is normalized to the range of 0 to 1.

The CVSS Base Score metric ranges from 0 to 10. The severity of the risk values are 0→None, 0.1-3.9→Low, 4.0-6.9→Medium, 7.0-8.9→High and 9.0-10.0→Critical.

D. Practical Consideration

As several factors contribute to the final base score, these factors need to be carefully considered in the practical scenario. There are many scaling factors such as K_1, K_2, vulnerability score normalization etc. As a result, this metric should be used as a relative score rather than an absolute value. Also, the assignment of CIA values for impact calculation are dependent on the security assessment. As the sub-graph creation is a process of security assessment, it also should be carefully assessed that which nodes have an impact on the considering critical node.

IV. Evaluation

To evaluate the proposed security assessment, we consider two SoC benchmarks from CAD for Assurance SoC benchmark suits [2]: AXISoC and ClusterSoC. We use python based Graph-tool to map the SoC in to a graph and python script to calculate the risk values.

A. Case Study 1: AXISoC

The AXISoC is AXI bus based System-on-Chip benchmark which has 1 PicoRV32 processor, 1 SRAM, 3 crypto IP core, 1 UART, 1 MD5, 4 hardware accelerators for FIR, IIR, DFT and IDFT. We consider AES crypto core as a critical node. As CPU access the AES core and save data to memory, we consider CPU and SRAM as two source nodes that has impact on AES node. We change the impact probability from 0.0 to 0.8. We consider two vulnerability scores of AES as 0.3 (low) and 0.8 (high). Also, we set the confidentiality, integrity and availability values as 0.56 (high). The base score or risk values are calculated using the proposed equation for the two cases, one is there is no protection for CPU and SRAM, and another is CPU is protected while SRAM is not protected. There are several methods available for trojan mitigation or IP protection in literature. Our assumption is that when the IP is protected, its impact does not propagate throughout the path. The result has been shown in Figure 2. We can see that at low probabilities, the risk value is medium for low and high vulnerability score of AES. But for impact probability higher than 0.5 the risk score becomes higher. Also, it is observed that after the CPU is protected, risk score becomes lower.

B. Case Study 2: ClusterSoC

The ClusterSoC is a NoC based SoC benchmark which has 1 CPU, 2 SRAM, 3 crypto IP, power management unit, 3 accelerators for FFT, FIR, IDFT, 5 Routers, and 11 network adapters. In this case study, again we consider AES as the critical node and CPU and SRAM are the source nodes. But in this benchmark, there are two SRAMs. So, in total there are three paths for impact propagation. We set the confidentiality, integrity and availability values same as case study 1 and AES vulnerability scores are also the same. For the same Impact probability range, we calculate the risk shown in Figure 3. Here, the risk score has a less increment for lower

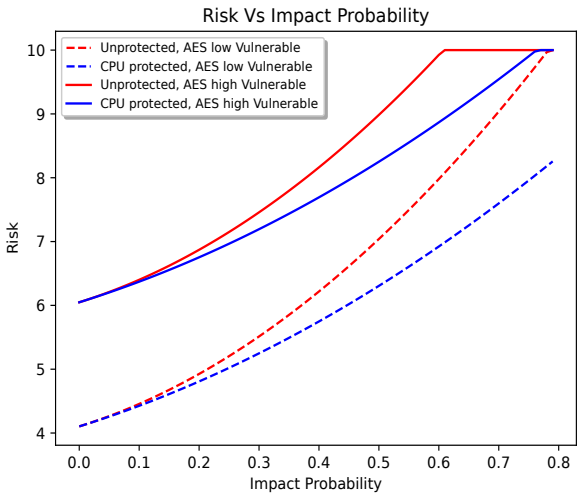

Fig. 2. Risk Vs Impact probability for AXISoC

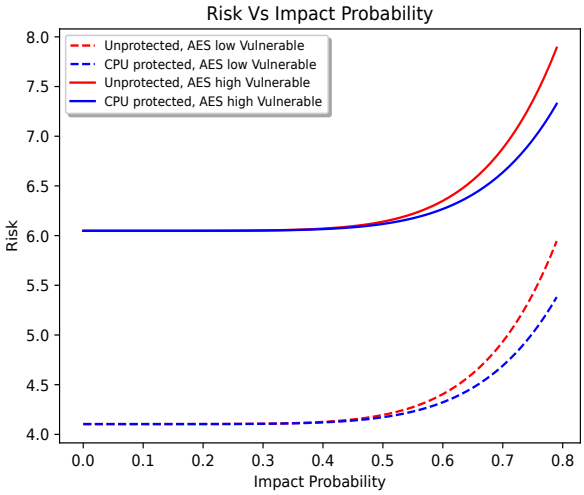

Fig. 3. Risk Vs Impact probability for ClusterSoC

impact probabilities. For higher impact probabilities, risk score increases higher. It occurs because the there are 6 nodes (routers and network adapters) in between CPU and AES, and 5 nodes between SRAMs and AES. Hence, the impact propagation is reduced.

V. Conclusion

The aim of this paper is to present a mathematical model of quantifying security risk of a System-on-Chip. We propose graph based impact analysis methodology and CVSS scoring metric for SoC to calculate the risk of a component in SOC. We apply our method on two practical SoC benchmarks with malicious component and with/without protection. The result analysis shows the practicality of the proposed security assessment metric.

In future, we plan to consider other hardware based security threats such as fault injection, side channel attack, illegal access, supply chain attack etc. in security assessment and find how to devise a security metric at SoC level.

References

[1] Bulbul Ahmed et al. *Quantifiable Assurance: From IPs to Platforms*. Cryptology ePrint Archive, Report 2021/1654. https://ia.cr/2021/1654. 2021.

[2] *CAD for Assurance*. https://cadforassurance.org/soc-platform/soc-benign-benchmark/system-on-chip-benchmarks/. 2022.

[3] Jin-Hee Cho et al. "Stram: Measuring the trustworthiness of computer-based systems". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), pp. 1–47.

[4] Jonathan Cruz, Prabhat Mishra, and Swarup Bhunia. "The Metric Matters: The Art of Measuring Trust in EIectronics". In: *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE. 2019, pp. 1–4.

[5] *Graph Tool*. https://graph-tool.skewed.de/. 2022.

[6] Forum of Incident Response and Security Teams (FIRST. org). "Common Vulnerability Scoring System v. 3.1: Specification Document". In: (2021). URL: https://www.first.org/cvss/v3.1/specification-document.

[7] Gabriel Jakobson. "Mission cyber security situation assessment using impact dependency graphs". In: *14th international conference on information fusion*. IEEE. 2011, pp. 1–8.

[8] Sandhya Koteshwara. "Security Risk Assessment of Server Hardware Architectures using Graph Analysis". In: *2021 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. IEEE.

[9] Fanny Lalonde Levesque et al. "A clinical study of risk factors related to malware infections". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pp. 97–108.

[10] *NIST Database*. https://nvd.nist.gov/vuln. 2022.

[11] Marcus Pendleton et al. "A survey on systems security metrics". In: *ACM Computing Surveys (CSUR)* 49.4 (2016), pp. 1–35.

[12] Sayandeep Saha, Rajat Subhra Chakraborty, and Debdeep Mukhopadhyay. "Testability based metric for hardware trojan vulnerability assessment". In: *2016 Euromicro Conference on Digital System Design (DSD)*. IEEE. 2016, pp. 503–510.

[13] *Trust-Hub Benchmark*. https://trust-hub.org/#/benchmarks/chip-level-trojan. 2022.

[14] Lingyu Wang et al. "An attack graph-based probabilistic security metric". In: *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer. 2008, pp. 283–296.

[15] Goitom Kahsay Weldehawaryat and Basel Katt. "Towards a quantitative approach for security assurance metrics". In: *The 12th International Conference on Emerging Security Information*. 2018.