Modeling Attacks on NoC-based SoCs

Luiz Gustavo Metzger, Cesar Albenes Zeferino
University of Vale do Itajaí
Laboratory of Embedded and Distributed Systems
Itajaí, Brazil
{metzger, zeferino}@univali.br

Martha Johanna Sepúlveda Flórez
University of São Paulo – USP
Laboratory of Microelectronics
São Paulo, Brazil
jsepulveda@lme.usp.br

ABSTRACT

Systems-on-Chip (SoCs) are used in a considerable amount of devices present in the daily lives of part of the world population. Such devices often process and store personal information, such as banking data and personal access codes. Thus, security becomes an ever-increasing concern in this domain. Networks-on-Chip (NoCs) used as the communication infrastructure in many SoCs can serve both as a target of attacks or as a barrier to defend the components of the SoC against attacks. This paper presents the application of a systematized methodology to model attacks on NoC-based SoCs. An example of Denial-of-Service attack is modeled according to the presented methodology and refined by means of a sequence diagram.

Categories and Subject Descriptors

B.8.1 [**Hardware**]: Performance and Reliability – *Reliability, Testing and Fault-Tolerance*

General Terms

Documentation, Reliability, Security, Standardization.

Keywords

Systems-on-Chip, Networks-on-Chip, Security.

1. INTRODUCTION

Systems-on-Chip (or SoCs) are integrated circuits that implement most or all of the functionality of a computational system into a single chip, including CPUs, memories and peripherals. In order to accomplish more complex tasks, more than a single CPU becomes necessary. In such cases, Multiprocessor SoCs are built, also called MPSoCs.

With more components integrated into a single chip, the drawbacks of bus-based communication architectures become more apparent. Some of these drawbacks are higher energy consumption, larger area cost, increasing parasitic capacitance and lack of scalability. Therefore, MPSoCs usually need a communication architecture that could provide, among other characteristics, higher levels of parallelism. This can reduce some of the drawbacks related to bus-based architectures [1].

Networks-on-Chip (or NoCs) are considered to be the successors of bus-based communication architectures. NoCs use routers and links that interconnect the various cores of an MPSoC, allowing for multiple communication flows to happen simultaneously.

SoCs (and MPSoCs) are integrated into various devices present in the daily lives of a significant part of the population, including smartphones, tablets and digital television sets. Some of these devices may store and process a considerable amount of personal information. Therefore, security becomes an ever-increasing concern.

Research has been done seeking to provide greater levels of security to these systems by means of implementing protection mechanisms against attacks on their security properties. However, most of the work found in the literature lacks of a standardized methodology that can be applied to document and model the attacks taken into consideration. This hampers the reuse and reproduction of the attacks by third parties, which makes it harder to validate the proposed security mechanisms.

In this context, the goal of this work is to identify and use a systematized methodology to model attacks on NoC-based SoCs.

The remainder of this paper is organized as follows. In Section 2, some of the related work is presented. Section 3 describes the methodology used to model the attacks and presents one practical example. Finally, Section 4 discusses the conclusions and future works.

2. RELATED WORK

This section presents some of the work addressing security on SoCs and NoCs. Even though this review was non-exhaustive, it could not identify previous work with a high degree of similarity. Because of that, works that address security outside the on-chip scope were considered. Also, works that propose security mechanisms were taken into consideration as well.

Although the work presented in [2] does not take into consideration SoCs or NoCs, the authors show formal definitions of attack patterns, transition states and attack scenarios in the off-chip context. The authors also propose a system that generates a database of attack scenarios. In such a system, attack patterns are extracted, classified and correlated with pre- and post-conditions in order to generate new attack scenarios.

In her thesis, [3] proposed the implementation of the Quality of Security Service (QoSS) concept on the NoC design process by means of a five-step methodology. In order to validate this concept, different scenarios were used to address extraction, modification and Denial-of-Service (DoS) attacks. Each of these scenarios presented information such as the type of attack, a brief description of the action, the target of the attack and the percentage of critical information contained in the memory. The simulations of the attacks were carried out by means of SystemC traffic generators.

In his dissertation, [4] conducted an analysis of the vulnerabilities present in the SoCIN NoC. After identifying them, the author proposed and implemented solutions that raise the network availability, which were developed as wrappers that can be integrated into the network interface of the NoC. The simulations

of the attacks used by the author to validate his solution were carried out by means of both VHDL test benches and SystemC traffic generators. Also, the attacks were modeled using only a non-systematized text-based approach.

In [5], a study was done on the main research papers that seek to provide security mechanisms to NoCs. The authors concluded that the most discussed attacks are message modification, release of message contents and DoS. Also, the most used security mechanisms are the ones that deal with access control and data integrity.

From this study, we concluded that the authors do not use a standardized method to model the attacks used to validate their proposals. Therefore, we propose the use of an approach to model the attacks in order to ease their reuse and reproduction by third parties. Also, another limitation is that the main method used to simulate the attacks is by means of traffic generators. This limitation will be covered in future works.

3. METHODOLOGY

3.1 Attack Trees

A structured and reusable approach to document information about attacks on enterprise systems was presented in [6]. Firstly, the authors describe the concept of attack trees. These structures refine the information about the attacks by classifying the compromise of security or survivability of the enterprise as the root node of the tree. The ways by which an attacker can cause this compromise are represented as nodes in the lower levels of the structure. Thus, the attack tree elaborates and enumerates the actions an attacker could take for a particular event to occur.

Each node of an attack tree can be decomposed into a set of sub-goals. In an "OR" decomposition, only one of the sub-goals has to be fulfilled for its parent goal to be reached. Similarly, in an "AND" decomposition, all of the sub-goals must be fulfilled. An "OR" decomposition is illustrated in Figure 1. In such case, if any of the sub-goals *G1* through *Gn* is fulfilled, goal *G0* is reached.

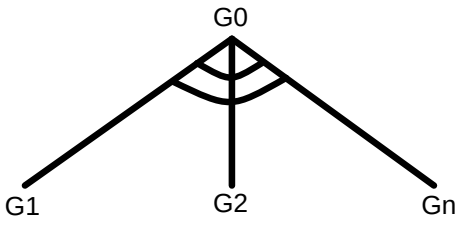

Figure 1. Attack tree with "OR" decomposition

3.2 Attack Patterns

In order to increase the feasibility of the attack trees, the authors in [6] also define the attack patterns. These patterns are generic representations of attacks that take place in specific contexts. They include the main goal of the attack, a set of pre- and post-conditions, and the attack tree that defines the steps to carry out the attack. The pre-conditions include assumptions about the attacker or the state of the target system that are necessary for the attack to be successful. The post-conditions include the knowledge obtained by the attacker and the changes in the system caused by the successful execution of the attack.

3.3 Attack pattern applied to a SoC attack

Using the approach proposed by [6], we modeled some possible attacks to a NoC-based SoC.

The attack pattern shown in Figure 2 is a type of DoS attack that aims to block some of the resources of the NoC. To do that, an attacker (*i.e.* a CPU infected with malicious code) may send out packets with invalid destination address or send packets without trailer.

Goal: Block resources of the NoC

Pre-condition:
1. The infected core can alter the routing table
2. The NoC does not discard packets

Attack:
OR 1. Send packets with invalid destination address
OR 1.1. Send packets with destination address outside the boundaries of the NoC
1.2. Send packets with destination address equal to the origin address
2. Send packet without trailer

Post-condition: The NoC will have blocked resources

Figure 2. NoC DoS-type attack pattern

In the first case (represented as the node 1 in the textual description of the attack tree contained in the attack pattern), the packets can be sent to a destination address outside the boundaries of the NoC (node 1.1) or even to the same core (node 1.2). If the NoC does not discard packets (listed as one of the pre-conditions), all of these approaches will result in the blocking of some of the network resources (*e.g.* links and buffers), which is shown in the post-condition section of the attack pattern.

3.4 Attack refinement

In order to further refine the attacks, we propose the use of UML sequence diagrams. These diagrams allow showing the actions and reactions of different components of the SoC chronologically. The sequence diagram that refers to node 1.1 of the attack tree used in the attack pattern shown in Figure 2 is depicted in Figure 3.

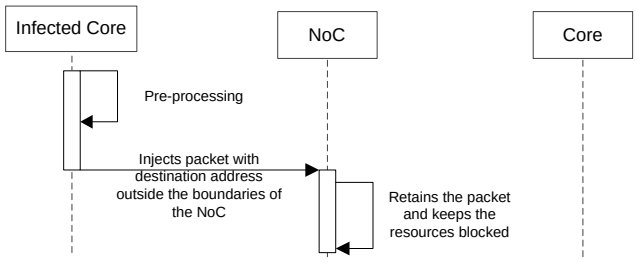

Figure 3. Sequence diagram of node 1.1 from the DoS-type attack pattern

In this situation, a pre-processing stage is illustrated. This stage may be related to the execution of another attack that will give permission to the infected core to send packets that would be invalid in a normal situation. One example of such situation is a buffer overflow attack. This attack might be executed first in order to give the infected core access to the routing table used by the NoC. After that, the packet with invalid destination address is injected by the infected core. The NoC then retains the packet

and, as a consequence, keeps some of its resources blocked for other communication flows.

Figure 4 illustrates the situation presented by node 1.2 of the attack tree shown in Figure 2.

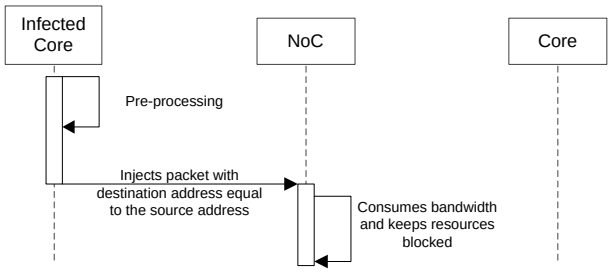

Figure 4. Sequence diagram of node 1.2 from the DoS-type attack pattern

Similar to the diagram presented in Figure 3, this diagram also depicts a pre-processing stage. This may be necessary because usually the NoC will not accept a packet with the same source and destination address. Thus, this stage can be responsible for the modification of the routing table used by the NoC, allowing such packet to be sent. When this packet is received by the NoC (consuming a small amount of bandwidth), it is blocked on the network interface. Consequently, this causes the blocking of the resources of the NoC.

Figure 5 depicts the operation of node 2 of the attack tree shown in Figure 2.

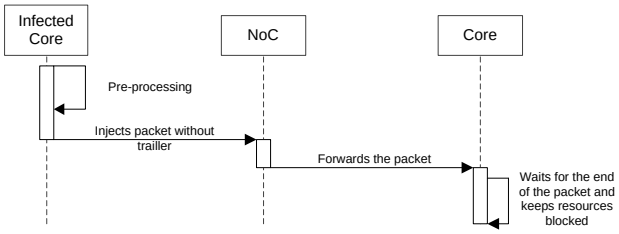

Figure 5. Sequence diagram of node 2 from the DoS-type attack pattern

A pre-processing stage is also illustrated, although it may not be necessary. The infected core then proceeds to inject a packet with no trailer. After that, the NoC simply forwards this packet to the destination core. As the NoC usually only checks the header flit of the packet, the lack of trailer will not have any direct impact on the routing or flow control mechanisms. Therefore, when the destination core receives the packet, it will keep its resources blocked waiting for the trailer, causing the availability of the system to be degraded.

4. CONCLUSIONS AND FUTURE WORK

This work presented a methodology used to model attacks in the off-chip context and applied it to the on-chip domain. This was done by modeling an attack that aims to block resources of a NoC used as the communication structure of a SoC.

This is an ongoing work, and, as future work, we intend to simulate possible attacks to NoC-based SoCs by means of implementing these attacks on a virtual platform that emulates a SoC. By doing that, we will be able to measure the impacts of the attacks on the SoC.

5. REFERENCES

[1] S. Pasricha and P. Dutt, *On-Chip communication architectures: System-on-Chip interconnect.* Burlington: Elsevier, 2008.

[2] Y. Cheng, C. Chen, C. Chiang, J. Wang and C. Laih, *Generating attack scenarios with causal relationship*. In: IEEE International Conference On Granular Computing, Fremont, 2007. Proceedings**...** Piscataway: IEEE, 2007, p. 368.

[3] M. J. Sepúlveda, G. Gogniat, R. Pires, W. J. Chau, M. J. Strum, *Dynamic NoC-Based architecture for MPSoC security implementation*. In: 24th Symposium on Integrated Circuits and Systems Design, João Pessoa, 2011. Proceedings… New York: ACM, 2011, p. 192-202.

[4] S. Baron, M. S. Wangham, C. A. Zeferino, *Security mechanisms to improve the availability of a Network-on-Chip*. In: IEEE International Conference On Electronics, Circuits, And Systems, Abu Dhabi, 2013. Proceedings... New York: IEEE, 2013, p. 609-612.

[5] S. Baron, M. S. Wangham and C. A. Zeferino, *Segurança em Redes-em-Chip: Conceitos e Revisão do Estado da Arte*, Revista de Informática Teórica e Aplicada, Porto Alegre, v.21, n.1, p. 110-126, 2014. (*in Portuguese*)

[6] A. P. Moore, R. J. Ellison, and R. C. Linger, *Technical Note CMU/SEI-2001-TN-001: Attack modeling for information security and survivability*, Pittsburgh, 2001.