PhD (M/F): Model-Based Threat Modeling for Compartmentalized Systems

Gurvan Le Guernic

February 12, 2026

1 Project Summary

From an evaluation point of view, one way to analyze the robustness of systems to cyberattacks is to threat model it. Threat modeling [TC20] is a generic term encompassing various methodologies (STRIDE, PASTA, OCTAVE, TRIKE, …) and tools (Microsoft Threat Modeling Tools, OWASP's PyTM, OWASP's ThreatDragon, SPARTA, IriusRisk, …) whose end goal is to identify some threats to the security (confidentiality, integrity and availability) of the analyzed system or data it processes. Threat modeling is more commonly used for, and adapted to, the analysis of large distributed systems, rather than single programs or even embedded systems. During threat modeling, those systems are often abstracted as a set of processes exchanging various data and executed in various networks (or domains) with differing levels of trust. The system abstraction languages used are usually quite simple with only 2 levels (processes executed into domains); and generic elements (processes, data, domains) that are completed by unspecified attributes.

From a design point of view, architecture compartmentalization is an efficient last defense against cyber attacks [Shu+16; Lef+25]. However, it is seldom used, or not to its fullest extent, in the design of security-sensitive embedded systems (or even of the majority of IT systems).

> "Despite longstanding recognition within the academic sphere and proven effectiveness in seminal industry projects, the adoption of compartmentalization techniques in main-stream software remains inconsistent: compartmentalizing is still far from being a common engineering practice." Lefeuvre et al. [Lef+25]

One of the main reasons for this state of affairs is that, except for virtualization and dockerization (which are mostly deployed in large systems for other reasons than security), compartmentalization is not a concept mastered and sufficiently valued by embedded system architects, or even for the majority of security-sensitive system architects.

By providing a better evaluation of compartmentalized systems than monolithic ones, threat modeling can play a role in incentivizing the compartmentalization of embedded systems. The goal of this project is to propose a new tool-supported methodology to threat model compartmentalized embedded systems from a precise model of their architecture. The modeling language used to describe the architecture of the embedded system is based on a detailed ontology of compartmentalization to support a precise semantics of compartmentalization allowing the proposed semi-automated threat modeling algorithm to take into account the advantages and limitations of various compartmentalization techniques. The evaluation of the proposed tool-supported methodology consists in a comparison to existing tools on known well-compartmentalized software. If time permits, an evaluation will also be performed on a security-sensitive real-world embedded system from the US Naval Research Laboratory.

2 Project Goals

The main objective of the proposed project is to propose a semi-automated tool-supported threat modeling methodology for embedded systems that takes into account the level of compartmentalization of the analyzed system

architecture. To reach this goal, the following will be accomplished:

1. evaluate and determine the root cause of the limitations and shortcomings of existing threat modeling methods and tools with regard to the evaluation of compartmentalized embedded systems;
2. propose a formal ontology of compartmentalization methods that generalizes compartmentalization features and characteristics;
3. propose an Architecture Description [Domain Specific] Language (AD[DS]L) relying on the compartmentalization ontology to describe the architecture of compartmentalized embedded systems;
4. propose a methodology and associated tooling to evaluate (threat model) the compartmentalized embedded systems architectures described using the proposed ADDSL.

The originality of the proposed project lies in its focus on:

- the evaluation of embedded systems rather than large IT systems;
- the evaluation of compartmentalized architectures.

3 Limitations of the State of the Art or Practice

Current threat modeling methodologies and tools focus mainly on the evaluation of large IT systems. They are not adapted to the evaluation of compartmentalized embedded systems:

- They rely on high level artifacts for the description of the system to analyze. They mostly rely on Data Flow Diagrams augmented with a generic notion of Trust Boundaries that assign the generic "execution platforms" into different trust domains without the ability to describe deep compartmentalization nesting or different types of compartmentalization.
- The evaluation is mostly based on open attributes without precise semantics associated with the different elements used to describe the system evaluated.

In order to handle the impact of deeply nested compartmentalized architectures and specificities of different compartmentalization techniques, the proposed approach will rely on a more expressive Architecture Description (Domain Specific) Language (ADL / DSL) associated with a finer semantics based on a specifically developed ontology that distinguishes the different features and characteristics of various compartmentalization techniques.

4 Proposed Design and Evaluation Methodologies

The evaluation and root cause analyses of the limitations and shortcomings of the existing tools will rely on a comparative experimental evaluation of those tools on 2 architectural variants of known compartmentalized systems (among which Qmail [HJA04; Ber07], Postfix [HJ08] and OpenSSH [PFH03]): their original compartmentalized architecture, and a monolithic variant that relies on a (potentially multi-threaded) single process executed on a monocore CPU with a single shared unpartitioned memory.

The development of the compartmentalization ontology will rely on systematic study techniques starting from snowballing from 2 reviews of compartmentalization techniques [Shu+16; Lef+25].

The development of the Architecture Description DSL will rely on the experience of the DiverSE team in the domain.

The evaluation of the proposed method and tools will be achieved by rerunning the initial experimentation on variants of known compartmentalized systems. A more ambitious evaluation on the architectures of the Network Pump and DSD's Serial Data Diode Device [Mal03; MW05; RF05; RFW06; MFC12] would be desired. However, due to the complexity and work load required to fully evaluate those architectures, it does not seem realistic to plan it for this thesis. However, depending on the thesis results, it is planned to propose a new project for a post-doc or engineer to extensively evaluate the proposed methodology and tools on those systems architectures, and take into account those extensive evaluation results to improve the methodology and tools.

5 State of the Art

The state of the art already contains work reviewing different techniques and technologies for compartmentalization [Shu+16; Lef+25]. However, those are focused on an analysis of the existing techniques on a bounded domain, and not a synthesis generalizing the concepts implemented by those techniques with regard to the cybersecurity robustness they provide. Existing work is therefore more of an excellent starting point rather than a solution to satisfy our need for a compartmentalization ontology serving as foundation in the definition of the semantics of ADL for threat modeling of compartmentalized systems. On related subjects, Miller presents a technique for modeling trust boundaries derived from securable objects in Windows [Mil08], while Juglaret et al. characterize guarantees of compartmentalization under an attacker model, by extending notions of full abstraction [Jug+16]. In complement, Lefeuvre et al. emphasize the importance of also taking into account the compartment interfaces in the evaluation [Lef+23].

Similarly, enhancing threat modeling, in particular with ontologies, is already a type of work explored by the community. Välja et al. propose an ontology framework to support automation of threat modeling [Väl+20]. While De Rosa et al. present ThreMA, an ontology-based threat modeling metamodel designed for ICT infrastructures [De +22]. In his master's degree thesis, Compierchio builds an ontology for threat modeling IoT systems [Com24]. However, those works do not focus on the threat modeling of compartmentalized systems, and so do not enhance threat modeling specifically for this aspect.

References

[TC20] Izar Tarandach and Matthew J. Coles. *Threat Modeling: A Practical Guide for Development Teams*. O'Reilly, Dec. 2020.

[Shu+16] Rui Shu et al. "A Study of Security Isolation Techniques." In: *ACM Computing Surveys* 49.3 (Oct. 2016). DOI: 10.1145/2988545.

[Lef+25] Hugo Lefeuvre et al. "SoK: Software Compartmentalization." In: *Proc. Symp. Security and Privacy*. 2025, pp. 3107–3126. DOI: 10.1109/SP61157.2025.00075.

[HJA04] Munawar Hafiz, Ralph Johnson, and Raja Afandi. "The security architecture of qmail." In: *Proc. Conference on Patterns Language of Programming*. 2004.

[Ber07] Daniel J. Bernstein. "Some Thoughts on Security After Ten Years of qmail 1.0." In: *Proc. Work. Computer Security Architecture*. Association for Computing Machinery (ACM), 2007, pp. 1–10. ISBN: 9781595938909. DOI: 10.1145/1314466.1314467.

[HJ08] Munawar Hafiz and Ralph E. Johnson. "Evolution of the MTA architecture: the impact of security." In: *Software: Practice and Experience* 38.15 (2008), pp. 1569–1599. DOI: 10.1002/spe.880.

[PFH03] Niels Provos, Markus Friedl, and Peter Honeyman. "Preventing Privilege Escalation." In: *Proc. USENIX Security Symposium*. USENIX Association, 2003, pp. 231–241.

[KM93] Myong H Kang and Ira S Moskowitz. "A Pump for Rapid, Reliable, Secure Communication." In: *Conf. Computer and Communications Security*. Association for Computing Machinery (ACM). 1993, pp. 119–129.

[KML96] Myong H. Kang, Ira S. Moskowitz, and Daniel C. Lee. "A Network Pump." In: *Transactions on Software Engineering* 22.5 (1996), pp. 329–338.

[97] *Network Pump Protocol*. Tech. rep. NRL-PUMP-PRO-97-001. Naval Research Laboratory, Oct. 16 Oct. 1997.

[Moo00] Andrew P Moore. *Network Pump (NP) Security Target*. Tech. rep. NAVAL RESEARCH LAB WASHINGTON DC, 2000.

[KMC05] Myong H Kang, Ira S Moskowitz, and Stanley Chincheck. "The Pump: A Decade of Covert Fun." In: *Annual Computer Security Applications Conf.* Institute of Electrical and Electronics Engineering (IEEE). 2005, 7–pp.

[Mal03] S. Mallen. *Serial Data Diode Device – Operation Manual*. Tech. rep. Defense Signal Directorate, 2003.

[MW05] T. McComb and L. P. Wildman. “SIFA: a Tool for Evaluation of High-Grade Security Devices.” In: *Proc. Australasion Conference on Information Security and Privacy*. 2005.

[RF05] Andrew Rae and Colin J. Fidge. “Information Flow Analysis for Fail-Secure Devices.” In: *The Computer Journal* 48.1 (Jan. 2005), pp. 17–26. DOI: 10.1093/comjnl/bxh056.

[RFW06] Andrew Rae, Colin Fidge, and Luke Wildman. “Fault Evaluation for Security-Critical Communications Devices.” In: *The Computer Journal* 39.5 (May 2006), pp. 61–68. ISSN: 0018-9162. DOI: 10.1109/MC.2006.161. URL: http://eprints.qut.edu.au/5319/.

[MFC12] Chris Mills, Colin J. Fidge, and Diane Corney. “Tool-Supported Dataflow Analysis of a Security-Critical Embedded Device.” In: *Proc. Australasian Information Security Conference*. Ed. by Josef Pieprzyk and Clark Thomborson. Vol. 125. Conferences in Research and Practice in Information Technology. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., Jan. 2012, pp. 59–70. ISBN: 978-1-921770-06-7. URL: http://dl.acm.org/citation.cfm?id=2512113.2512121.

[Mil08] Matt Miller. “Modeling the Trust Boundaries Created by Securable Objects.” In: *Proceedings of the 2nd Conference on USENIX Workshop on Offensive Technologies*. WOOT’08. San Jose, CA: USENIX Association, 2008.

[Jug+16] Yannis Juglaret et al. “Beyond Good and Evil: Formalizing the Security Guarantees of Compartmentalizing Compilation.” In: *IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016*. IEEE Computer Society, 2016, pp. 45–60. DOI: 10.1109/CSF.2016.11. URL: https://doi.org/10.1109/CSF.2016.11.

[Lef+23] Hugo Lefeuvre et al. “Assessing the Impact of Interface Vulnerabilities in Compartmentalized Software.” In: *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023. URL: https://www.ndss-symposium.org/ndss-paper/assessing-the-impact-of-interface-vulnerabilities-in-compartmentalized-software/.

[Väl+20] Margus Välja et al. “Automating Threat Modeling using an Ontology Framework: Validated with Data from Critical Infrastructures.” In: *Cybersecurity* 3.19 (Oct. 2020).

[De +22] Fabio De Rosa et al. “ThreMA: Ontology-Based Automated Threat Modeling for ICT Infrastructures.” In: *IEEE Access* 10 (2022), pp. 116514–116526. DOI: 10.1109/ACCESS.2022.3219063.

[Com24] Elena Francesca Compierchio. “Ontology-driven Threat Modeling for IoT Systems.” Master of science program in Computer Engineering. Politecnico di Torino, M, 2024.