Integrating Security Modeling in Embedded System Design

Jan Werner, Matthew Eby, Janos Mathe, Gabor Karsai, Yuan Xue, Janos Sztipanovits
{jwerner, meby, jmathe, gabor, yxue, janos}@isis.vanderbilt.edu
Institute for Software Integrated Systems
Vanderbilt University, Nashville, TN 37235

Abstract— **There is an increasing concern of security threats as embedded systems are moving towards networked applications. Model based approaches have proven to be an effective for embedded systems design. In this paper we show that existing modeling tools are not sufficient to meet the current and future security challenges of networked embedded systems. We propose a framework to integrate security modeling in embedded system design. We've developed an example co-design environment that addresses security, functionality and system architecture aspects of embedded systems.**

I. Introduction

Embedded systems play a crucial role in critical infrastructure, which is recognized in [1] as essential to national security, success and economic health [2]. There is increasing concern of the security threat on such embedded systems [3],[4]. Successful attacks have been reported on the US Power Grid [5] and the sewer system of Australia's Maroochy Shire Council [6]. Other security incidents such as worm [7] infection have affected the Davis-Besse Nuclear Power Plant and CSX Railroad Corp. [6]. To address such security threats we need to rethink to the embedded software design process.

Model Integrated Computing (MIC) [8] is gaining wide recognition in the field of embedded software design. Models represent embedded software, its deployment platform and its interactions with the physical environment. Models facilitate formal analysis, verification, validation and generation of embedded systems [9]. This approach is superior to handcrafting code, which is slow and error prone. Although, there is modeling tool support for analysis of functionality, performance, power consumption, safety, etc., currently available tools incorporate little if any support for security modeling. As a result, security is added only once the complete system has been built. At best, this approach of adding security is inefficient taking large amounts of effort to achieve only modest improvements in security. Engineers designing embedded systems usually do not have enough knowledge to address security issues and in many cases are not even aware of the issues [10]. Fixing security vulnerabilities involves releasing patches, which can introduce new problems such as viruses [11] or security vulnerabilities [12]. Still, systems designed without security in mind are intrinsically insecure. Patches can fix specific security vulnerabilities, but do not address poor system architecture.

Many times vulnerabilities are only discovered once they have been exploited. To address these unknown threats systems can be isolated in private corporate networks using firewalls and intrusion detection systems. Still, such perimeter defenses cannot protect against insider attacks [13]. In light of this situation, we need a modeling environment that incorporates security into the design phase of embedded systems.

One of the few modeling languages with security extensions is UML. Currently available extensions to UML provide: access control [14] [15], fair exchange, assumptions about secrecy, integrity etc [16]. There are existing tools that can guarantee some security properties using automated proof verifiers [16]. However, UML based Model Driven Security is not sufficient for design and analysis of embedded systems. There is no concept of hardware in UML which makes it unsuited for the diverse hardware architectures found in embedded systems. In many embedded applications system resources are scarce. Added overhead for security can have drastic effects on performance. An ideal embedded software development environment will allow the engineer to analyze security and performance tradeoffs based on the hardware environment in which will operate.

II. Security and Embedded System Co-Design Via Model Integrated Computing

MIC can meet the challenges of designing secure embedded systems. A key advantage of model based approach is abstraction of the application domain. This abstraction is facilitated through the use of domain specific modeling languages (DSML). A DSML provides a system designer a set of concepts that are specifically tailored for a certain application domain. In our case the domain is networked embedded real-time systems such as process control systems, automotive, avionics and robotics systems. A DSML with the proper level of abstraction hides the inconsequential details of a system while allowing the engineer to shift focus to more important details. There are numerous examples of the model based approach used successfully in the design of embedded

This work is supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422)

systems such as the Architectural Analysis and Design Language (AADL) [17]. We propose extending these DSMLs with the security concepts borrowed from UML extensions[14]. By extending embedded system DSMLs, we can add tool support for security analysis, validation, verification and generation. These security tools will extend the large tool chains that already exist for embedded system design.

III. Case Study of the Architectural Analysis and Design Language

We have created a prototype security co-design environment. In this environment we use access control as an example security mechanism and incorporate it in the embedded system design process. We implement the co-design environment in the Generic Modeling Environment (GME) [9]. GME is a metaprogramable tool which facilitates the graphical implementation of domain specific modeling languages (DSML) through the use of metamodels.

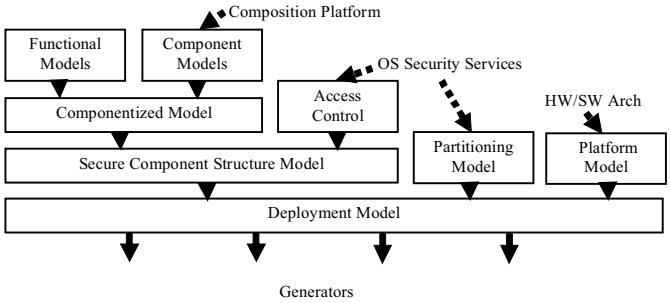

Fig. 1 Design flow in a security co-design environment

Fig. 1 illustrates the process of building models in the co-design environment. We begin with functional models of embedded systems built and verified in modeling environments such as Matlab/Simulink and Stateflow. Next, we model the architecture of the system with AADL. AADL is a SAE Aerospace Standard that is intended to provide a standard interface and environment for system designers to model, analyze and generate embedded system code. This is accomplished by modeling the binding between software and hardware components within a system.

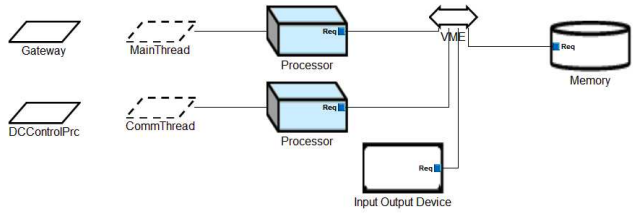

Fig. 2 GUI for specifying AADL models

We have extended AADL to support modeling of Role Based Access Control (RBAC). With RBAC an engineer can specify what permissions a component has to access other components in the system. The security policies expressed in the model are enforced through the use of underlying OS security services such as kernel partitioning [18]. Figure two shows an example AADL model created in our codesign environment using GME. We have built code generators for the co-design environment that can generate secure C code and textual AADL from graphical models. The textual representation of an AADL model can be ported to other tool environments to make use of existing analysis tools. Adding security mechanisms such as access control and time/space partitioning can prevent security threats such as a denial of service attack. For example, a system that has a network component and process control component can protect the process control from packet flooding by placing the two components in separate partitions.

This prototype co-design environment is intended as an example for the potential of MIC in the development of embedded software for security critical infrastructure. We do not intend to limit our research to AADL or the specific security mechanisms discussed here. We want to extend other concept such as encryption and authentication to AADL and other DSMLs.

IV. Related Work

AADL has been proposed for mission critical software development [19], and used to perform scheduling and memory requirements analysis [20]. Security for model driven design was proposed in [14,16]. Formal methods of security model verification was also proposed by Jürjens in[16].

V. Conclusion

Model-Integrated Computing has proved itself to be a valuable tool in embedded systems design process. Model driven security approaches have been successfully used in various industrial, governmental and financial applications. We have shown the potential for applying these security modeling concepts to DSMLs gear toward embedded systems. There is much work that needs to be done to incorporate these security aspects into the large tool chains that currently exist for embedded systems.

References

[1] Fernandez, J. D. and Fernandez, A. E. 2005. *SCADA systems: vulnerabilities and remediation. J. Comput. Small Coll.* 20, 4 (Apr. 2005), 160-168.

[2] Amin, M. North America's electricity infrastructure: are we ready for more perfect storms? Security & Privacy Magazine, IEEE Volume 1, Issue 5, Sept.-Oct. 2003 Page(s):19 - 25

[3] Kocher, P., Lee, R., McGraw, G., and Raghunathan, A. 2004. Security as a new dimension in embedded system design. In *Proceedings of the 41st Annual Conference on Design Automation* (San Diego, CA, USA, June 07 - 11, 2004). DAC '04. ACM Press, New York, NY, 753-760.

[4] Ravi, S., Raghunathan, A., Kocher, P., and Hattangady, S. 2004. Security in embedded systems: Design challenges. *Trans. on Embedded Computing Sys.* 3, 3 (Aug. 2004), 461-491.

[5] F. Schneider, editor. *Trust in* .National Academy Press, Washington, DC, 1999. Available at http://www.nap.edu/readingroom/books/trust/

[6] Andrew Hildick-Smith, *Security for Critical Infrastructure SCADA systems.* August 24 2005. SANS Institute. Available at http://www.sans.org/rr/whitepapers/warfare/1644.php

[7] Weaver, N., Paxson, V., Staniford, S., and Cunningham, R. 2003. *A taxonomy of computer worms*. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode* (Washington, DC, USA, October 27 - 27, 2003). WORM '03. ACM Press, New York, NY, 11-18

[8] Sztipanovits, J.; Karsai, G. *Model-integrated computing,* Computer Volume 30, Issue 4, April 1997 Page(s):110 – 111

[9] Karsai, G., Sztipanovits, J., Ledeczi, A., Bapty, T.: *"Model-Integrated Development of Embedded Software,"* Proceedings of the IEEE, Vol. 91, No.1., pp. 145-164, January, 2003

[10] Tsipenyuk, K.; Chess, B.; McGraw, G.; *Seven pernicious kingdoms: a taxonomy of software security errors*
Security & Privacy Magazine, IEEE
Volume 3, Issue 6, Nov.-Dec. 2005 Page(s):81 - 84

[11] Microsoft TechNet. Information About Virus-Infected Hotfixes. April 25 2001. Available at http://www.microsoft.com/technet/security/alerts/info/vihotfix.mspx

[12] Microsoft TechNet. Microsoft Security Bulletin MS06-007. February 14 2006. Available at http://www.microsoft.com/technet/security/Bulletin/MS06-007.mspx

[13] Kevin Poulsen, *Slammer worm crashed Ohio nuke plant network,* August 19 2003. Available at http://www.securityfocus.com/news/6767

[14] Jürjens, J. 2005. *Sound methods and effective tools for model-based security engineering with UML*. In *Proceedings of the 27th international Conference on Software Engineering* (St. Louis, MO, USA, May 15 - 21, 2005). ICSE '05. ACM Press, New York, NY, 322-331.

[15] Basin, D., Doser, J., and Lodderstedt, T. 2003. *Model driven security for process-oriented systems*. In *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies* (Como, Italy, June 02 - 03, 2003). SACMAT '03. ACM Press, New York, NY, 100-109.

[16] Jan Jürjens, *Secure Systems Development with UML*, Springer-Verlag, 2004

[17] *Architecture Analysis & Design Language*, SAE Aerospace Standard AS5506, September 2004

[18] John Rushby. The design and verification of secure systems. In Eight ACM Symposium on Operating System Principles, pages 12-21, Asilomar, CA, December 1981.

[19] P. Dissaux. Using the AADL for mission critical software development. *2nd European Congress ERTS, EMBEDDED REAL TIME SOFTWARE - 21, 22 and 23 January 2004, Toulouse*.

[20] Singhoff, F., Legrand, J., Nana, L., and Marcé, L. 2005. Scheduling and memory requirements analysis with AADL. In *Proceedings of the 2005 Annual ACM Sigada international Conference on Ada: the Engineering of Correct and Reliable Software For Real-Time &Amp; Distributed Systems Using Ada and Related Technologies* (Atlanta, GA, USA, November 13 - 17, 2005). SigAda '05. ACM Press, New York, NY, 1-10.