Towards Reliable Things: Formal Verification of IoT Software with Frama-C

Allan Blanchard*, Nikolai Kosmatov† and Frédéric Loulergue‡
* *Inria Lille – Nord Europe, France*
Allan.Blanchard@inria.fr
† *CEA, List, Software Reliability and Security Lab, PC 174, 91191 Gif-sur-Yvette, France*
Nikolai.Kosmatov@cea.fr
‡ *School of Informatics Computing and Cyber Systems, Northern Arizona University, USA*
Email: Frederic.Loulergue@nau.edu
Phone: +1-928-523-5044

1. The Topic

Among distributed systems, connected devices and services, also referred to as the Internet of Things (IoT), have proliferated very quickly in the past years. There are now billions of interconnected devices, and this number is growing. It is anticipated that by 2021, about 46 billions of devices will be in use.

Some of these devices are in service in safety and security critical domains, and even in domains that are not necessarily critical, privacy issues may arise with devices collecting and transmitting a lot of personal information. Moreover insufficiently secured devices may be used for example for massive distributed denial of service attacks. This raises important security challenges. Formal methods have been used successfully for years in highly critical domains, now they can help to bring security into the IoT field.

While the correctness of an implementation with respect to a formal functional specification provides the strongest form of guarantee, it can be very costly to achieve. In practice it is therefore more common to rely on a combination of formal methods to achieve an appropriate degree of guarantee: static analyses to guarantee the absence of runtime errors, deductive verification of functional correctness, dynamic verification for parts that cannot be proved using deductive verification.

Frama-C[1] is a source code analysis platform that aims at conducting verification of industrial-size programs written in ISO C99 source code. Frama-C fully supports the combination of formal methods approach, by providing to its users with a collection of plug-ins that perform static and dynamic analysis for safety and security critical software. Moreover collaborative verification across cooperating plug-ins is enabled by their integration on top of a shared kernel, and their compliance to a common specification language: ACSL [1].

Recently Frama-C has been applied to the verification of software in the context of the Internet of Things, more specifically the verification of modules of Contiki [2], an open source operating system for the IoT.

1. https://frama-c.com

2. Summary of the Tutorial Format

The proposed tutorial is 180 minutes long: about 20 minutes for the introduction including an overview of Frama-C and Contiki, 50 minutes for each of the 3 main parts, and 10 minutes for conclusion.

Each part consists of a presentation using slides and live demonstration, and a session of exercises. To work on the exercises, the attendees will be provided a virtual machine image containing all the tools ready to use.

In more details, the tutorial will be structured as follows:

1) Introduction
   a) IoT and verification challenges
   b) An overview of Frama-C
   c) The Contiki operating system
2) Verification of the absence of runtime errors using the plug-in EVA
   - Presentation of EVA
   - An application to Contiki
   - Exercise
   - Limitations and how to deal with them
3) Deductive verification using the plug-in WP
   - Presentation of WP
   - An application to Contiki
   - Exercise
   - Limitations and how to deal with them
4) Runtime verification with the plug-in E-ACSL
   - Presentation of E-ACSL
   - An application to Contiki
   - Exercise
5) Conclusion and Further References

Bibliography

- An overview of Frama-C: [3]
- Introduction to deductive verification with Frama-C: [4]
- Application of Frama-C to the IoT: [5], [6], [7]
- Previous Frama-C tutorial papers: [8], [9], [10], [11]

3. Expected Audience and Learning Outcomes

This tutorial will be of interest for both researchers and practitioners in the area of IoT software and software verification in general, as well as students in Software Engineering. The tutorial only assumes knowledge of the C programming language.

Participants will learn how to use the different Frama-C analyzers and how to combine them. Several examples and use cases presented during the tutorial will give them a clear practical vision of possible usages of the underlying static and dynamic analyses in their everyday work. The presented code fragments are part of Contiki [2], a real-world lightweight operating system for the IoT.

4. Prior Tutorials on Similar Topics

While there have already been several tutorials about some parts of Frama-C, a complete similar tutorial, entitled "Frama-C, a Collaborative Framework for C Code Verification" was given by only at:

- the 16th international conference on Runtime Verification (RV 2016, Madrid, Spain), https://rv2016.imag.fr/?page_id=349,
- the 28th IEEE International Symposium on Software Reliability Engineering (ISSRE 2017, Toulouse, France), where the tutorial was an IEEE certified tutorial, http://2017.issre.net/Tutorials#t3.

Both tutorials were presented by Nikolai Kosmatov and Julien Signoles to about 25 attendees.

The focus on program verification for the Internet of Things is completely new and is related to recent research work conducted by the presenters. A similar tutorial was only presented (to about 15 attendees) at

- the Zooming Innovation in Consumer Electronics International Conference (ZINC), May 30-31, 2018, Novi Sad, Serbia, http://www.gozinc.org/#information.

Other presentations are planned in the future.

5. Biographies

Allan BLANCHARD obtained his PhD in Computer Science from the University of Orléans in 2016. He prepared his PhD at the Software Reliability Laboratory of the CEA LIST.

He is interested in the analysis of concurrent code using formal methods and more precisely deductive verification. His current work, in the EU H2020 VESSEDIA project, is to apply formal verification to the Contiki microkernel and its libraries, mostly to show the absence of runtime errors. He mostly uses FRAMA-C with the EVA and WP plugins.

He is the author of an online tutorial on deductive verification with FRAMA-C and its WP plugin[2].

Web site: https://allan-blanchard.fr

2. available at https://allan-blanchard.fr/teaching.html

Nikolai KOSMATOV got a PhD in Mathematics in 2001 jointly from Saint-Petersburg State University and University of Besançon. Nikolai's research interests include software verification and testing, constraint solving and combinations of various software verification techniques. Nikolai gives theoretical courses and exercise sessions on software verification and testing since 2009. He is the main author of the online testing service pathcrawler-online.com.

Nikolai co-organized over a dozen of successful tutorials at various international events, e.g. TAP, TAROT, ASE, QSIC, iFM, SAC, RV, ZINC, ISSRE.

Web site: http://nikolai.kosmatov.free.fr/

Frédéric LOULERGUE obtained his PhD in Computer Science from the University of Orléans in 2000 and his Habilitation in Computer Science from Université Paris Val-de-Marne in 2004. He is currently a full professor at Northern Arizona University, Flagstaff, USA. His research interest are the practical and formal aspects of the design, implementation and application, in particular to large-scale data-intensive software, of structured parallel programming languages and libraries, as well as applied formal methods and cyber security in this broad context. Software associated to his research work include Bulk Synchronous Parallel ML (BSML) and the SYDPACC framework for the systematic development of programs for scalable computing.

He co-organized several international workshops on High-Level Parallel Programming and Applications (HLPP) and on Practical Aspects of High-Level Parallel Programming (PAPP), and the PAPP ACM SAC Track in 2016 and 2017. He co-chaired the Formal Approaches to Parallel and Distributed System (4PAD) symposium in 2016 and 2018. He is a member of the editorial board of Scalable Computing: Practice and Experience. He was associate director of the Laboratory of Algorithms, Complexity and Logic (LACL), and associate director of the Laboratoire d'Informatique Fondamentale d'Orléans (LIFO). He founded and lead the Logic Modeling and Verification (LMV) research team at LIFO (2015-16).

Web site: http://frederic.loulergue.eu

References

[1] P. Baudin, P. Cuoq, J. C. Filliâtre, C. Marché, B. Monate, Y. Moy, and V. Prevosto, *ACSL: ANSI/ISO C Specification Language*. [Online]. Available: http://frama-c.com/acsl.html

[2] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - A lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE Conference on Local Computer Networks (LCN 2004)*. IEEE Computer Society, 2004, pp. 455–462.

[3] F. Kirchner, N. Kosmatov, V. Prevosto, J. Signoles, and B. Yakobowski, "Frama-C: A software analysis perspective," *Formal Asp. Comput.*, vol. 27, no. 3, pp. 573–609, 2015.

[4] A. Blanchard, "Introduction to C program proof using Frama-C and its WP plugin," december 2017. [Online]. Available: https://allan-blanchard.fr/publis/frama-c-wp-tutorial-en.pdf

[5] A. Blanchard, N. Kosmatov, and F. Loulergue, "Ghosts for Lists: A Critical Module of Contiki Verified in Frama-C," in *Nasa Formal Methods*, ser. LNCS, vol. 10811. Springer, 2018, to appear.

[6] A. Peyrard, S. Duquennoy, N. Kosmatov, and S. Raza, "Towards formal verification of Contiki: Analysis of the AES–CCM* modules with Frama-C," in *2nd International Workshop on Recent Advances in Secure Management of Data and Resources in the IoT (RED-IoT 2017) co-located with the International Conference on Embedded Wireless Systems and Networks (EWSN 2018)*. ACM, 2018, to appear.

[7] F. Mangano, S. Duquennoy, and N. Kosmatov, "A memory allocation module of Contiki formally verified with Frama-C. A case study," in *11th International Conference on Risks and Security of Internet and Systems (CRiSIS 2016)*, ser. LNCS, vol. 10158. Springer, 2016, pp. 114–120.

[8] N. Kosmatov, V. Prevosto, and J. Signoles, "A lesson on proof of programs with Frama-C," in *International Conference on Tests and Proofs (TAP'13)*, ser. LNCS. Springer, 2013.

[9] N. Kosmatov and J. Signoles, "A lesson on runtime assertion checking with Frama-C," in *International Conference on Runtime Verification (RV 2013)*, ser. LNCS, vol. 8174. Springer, 2013, pp. 386–399.

[10] ——, "Runtime assertion checking and its combinations with static and dynamic analyses – tutorial synopsis," in *International Conference on Tests and Proofs (TAP 2014)*, ser. LNCS. Springer, 2014, pp. 165–168.

[11] ——, "Frama-C, A collaborative framework for C code verification: Tutorial synopsis," in *International Conference on Runtime Verification (RV 2016)*, ser. LNCS, vol. 10012. Springer, 2017, pp. 92–115.