Securing Firmware for Embedded Systems: Binding Firmware and Hardware

Royal Holloway

Luke Atherton

1st March 2019

Statement of Objectives

What I plan to achieve

- Produce a clear definition of "firmware";
- Review real-world reasons justifying the need for further research into secure firmware;
- Review existing solutions for binding hardware and software;
- Understand the feasible alterations/additions which can be made to widely used embedded system chips such as FPGAs and ARM Cortex-M Series. Understand the primitives which will be used for creating suggestions and improvements such as software control-flow graphs and PUFs;
- Provide suggestions for increasing firmware security though hardware to software binding.

Why I have chosen the project

As a software developer, neat and elegant solutions to problems appeal to me greatly. The computing constraints of embedded systems present the need for neat, elegant and efficient solutions. This cross-over is why the project interests me. To further my interest, embedded systems in the form of Internet of Things (IoT) and Operational Technology (OT) are becoming increasingly used and therefore their security is vital for consumers, businesses and the national critical infrastructure.

The project itself is of interest due to the lightweight use of encryption, examination and development of protocols and gaining a deeper understanding of embedded systems.

Methodology

How will I achieve the objectives of the above list

- Researching the following topics through journal articles, conference proceedings and books:
  - Binding of hardware and software
  - Secure software execution
  - Common attacks on embedded systems
  - Commercial cost of IP theft (through counterfeiting and other methods)
- Research, understand and critique primitives such as control-flow graphs and PUFs in line with my overall goal.
- Gain a deep enough understanding of FPGA and ARM Cortex-M to assess what is possible when it comes to hardware adjustments for the purpose of security.
- Experiment with implementing basic hardware to software binding, this will include:
  - Gaining access to either an FGPA or and ARM Cortex-M based development board
  - Understanding the processes behind writing software, provisioning and testing software as well as altering hardware.
  - Assess existing solutions for hardware to software binding for applicability for testing, e.g. is it possible, how much work would it be and what advantage would I get out of implementing it?
- Consider the design of a novel solution
  - If solution requires remote interaction (through a server) a basic understanding and implementation of a server is required, this will be implemented either through a laptop acting as a server or running up a cloud instance.

- Complete design and (if possible) implement a novel solution/s
- Assess novel solution/s and investigate the firmware transfer process required

What is my strategy for getting started

- Using the objectives set and predefined project chapters review literature which has already been read for the preliminary literature review, review literature which has already been gathered but not included in the literature review and review books for content which is applicable for this project.
- Gain an understanding of technologies used (FGPA and/or ARM), perhaps gain access to a corresponding developer board and write a simple application.
- Develop knowledge of control-flow graphs and the method of creating them.
- Gain a deeper understanding of C, assembly language and C compilers.

Work plan

December 2018 - January 2019

Meet with project supervisor to begin discussions. Discuss potential project directions with supervisor. Gather and assess reading material followed by studying highly applicable material. Refine ideas for project title.

February 2019

Meet with project supervisor to discuss project topic choice, once topic is decided commence literature review of literature already possessed and any further work which comes to light during the literature review. Fill out Project Description Form (PDF) and formalise Preliminary Literature Review (PLR) before submitting to supervisor for review. Once changes suggested from review are made make final submission to supervisor on 1st March 2019.

March 2019

Begin work on understanding FPGA and ARM Cortex-M. Complete review-based chapters and submit to supervisor for review. Discussion with supervisor on learnings and thoughts so far.

April 2019

As this period will be used mostly for exam preparation project work will be not proceed at a high pace. Practical understanding of FPGA and/or ARM Cortex-M will be gained, including writing and implementing software for said systems. Notes will be made during this phase.

May 2019

As this period will also be used for exam preparation/partaking project work will not proceed at a high pace. Control-flow graphs and PUFs will be studied.

Notes will be made during this phase.

June 2019

Notes on understanding of hardware and primitives will be formalised into chapters. Existing solutions will be studied and compared, creating a chapter. This will be submitted to supervisor mid June 2019 (aiming for 14th June 2019). Begin work on producing a novel method for binding hardware and firmware.

July 2019

Update chapters already submitted to supervisor with feedback provided. Meet with supervisor to discuss novel solution. Begin basic implementation of novel solution. Create chapter on firmware transfer. Submit first draft of project to supervisor at end of month (aim for 26th July 2019).

August 2019

Finalise implementation of solution. Discuss findings with supervisor and write up respective chapter for review (aim for 9th August 2019). Submit final draft to supervisor 16th August. Submit MSc Project by deadline of 22nd August 2019.

Draft Table of Contents

6. Proposed solution
    6.1 Founding principles
    6.2 Hardware requirements
    6.3 Programmer/compiler requirements and constraints
    6.4 Description of solution
7. Implementation
    7.1 Implementation steps
    7.2 Implementation findings
8. Analysis
    8.1 Assessment of solution against attacks
    8.2 Comparison with existing solutions
9. Remarks and Conclusion
    9.1 Conclusion