Contents

Cover 1

Acknowledgements 3

Abstract 5

Resumen 7

Resum 9

Contents 14

List of Figures 15

List of Tables 17

List of Listings 19

1 Introduction 21

1.1 Secure partitioned systems . . . 21

1.1.1 Secure hypervisors . . . 21

1.1.2 Validation and verification of secure hypervisors . . . 22

1.1.3 Challenges on the validation of secure hypervisors . . . 24

1.1.4 Terminology . . . 26

1.2 Motivation and main goals . . . 26

1.3 Contributions of this thesis . . . 26

1.4 Outline of this thesis . . . 27

1.5 Research Context . . . 28

2 Secure Hypervisor Verification **29**
2.1 Formal methods in secure hypervisors . . . 29
2.1.1 Deductive verification . . . 29
2.1.2 Theorem provers . . . 30
2.1.3 Static code analysis . . . 31
2.2 Formal methods in safety standards . . . 32
2.2.1 RTCA Standards . . . 32
2.2.2 IEC-61508 Standards . . . 33
2.2.3 ECSS Standards . . . 33
2.2.4 Common Criteria Framework . . . 34
2.3 Summary . . . 35

3 XtratuM foundations: A formalisation approach. **37**
3.1 Introduction . . . 37
3.2 XtratuM Overview . . . 38
3.2.1 XtratuM Architecture . . . 38
3.3 Trustability enforcement . . . 39
3.3.1 Interrupt Model . . . 41
3.3.2 Fault Management Model . . . 42
3.3.3 System specification . . . 43
3.4 Hypervisor model . . . 45
3.4.1 Hypervisor state variables . . . 47
3.4.2 General properties . . . 48
3.4.3 Spatial isolation properties . . . 48
3.4.4 Temporal isolation properties . . . 49
3.4.5 Hypervisor state management . . . 50
3.4.6 Hypervisor pre- and post-conditions . . . 50
3.5 Conclusion . . . 51

4 Formal Validation of XtratuM Components **53**
4.1 Introduction . . . 53
4.1.1 XtratuM Hypervisor core . . . 54
4.2 Deductive Formal Methods . . . 54
4.3 Proposed Approach . . . 55

4.3.1 Method . . . 55
4.3.2 Contract Specification . . . 56
4.3.3 The Frama-C Framework . . . 56
4.4 Approach Evaluation . . . 57
4.4.1 Code Refactor . . . 57
4.4.2 Contract Annotation . . . 57
4.4.3 Proof Verification . . . 58
4.4.4 Proof Results . . . 58
4.4.5 Results . . . 58
4.5 Conclusions . . . 59

5 Analysing the Impact and Detection of Kernel Stack Infoleaks 61
5.1 Introduction . . . 61
5.2 Related Work . . . 62
5.2.1 Protection Mechanisms . . . 62
5.2.2 Protection Techniques . . . 63
5.3 Classification of Information Disclosure Vulnerabilities . . . 64
5.3.1 The Anatomy Of An Infoleak . . . 64
5.3.2 Targets of Infoleaks . . . 65
5.3.3 Infoleaks Bug Causes . . . 65
5.3.4 Infoleaks Data Sources . . . 66
5.4 Analysis on the Impact of Stack Infoleaks . . . 67
5.4.1 The Anatomy of An Attack . . . 67
5.4.2 The Contents of the Kernel Stack . . . 68
5.4.3 Infoleak Based Attacks . . . 69
5.5 The Detection of Information Leak vulnerabilities . . . 70
5.5.1 Infoleak Vulnerability Model . . . 70
5.5.2 Semantic Patch Preparation . . . 71
5.5.3 Filter and Rank of matches . . . 71
5.5.4 Infoleak Code Review and Correction . . . 71
5.6 Experimental Evaluation of the Detection Technique . . . 72
5.6.1 Existing Infoleak Detection . . . 72
5.6.2 Discovery of Vulnerabilities . . . 72
5.6.3 Applications and Limitations of our Approach . . . 73
5.7 Conclusions and Further Work . . . 73

6 Conclusions and open research lines **75**

6.1 Conclusions 75

6.2 Research Lines 76

6.3 Publications related to this thesis 76

Bibliography **79**

Acronyms **87**

Glossary **89**