

Contents

- 1 Introduction 1**
 - 1.1 Problem Statement 3
 - 1.2 Research Question and Methodology 4
 - 1.3 Impact of Thesis Contributions 5
 - 1.4 Structure of the Thesis 6
- 2 Technical Background, Preliminaries and Assumptions 9**
 - 2.1 The Rootkit Evolution 9
 - 2.2 Typical x86-Based System Architecture 12
 - 2.3 Intel x86 Based Host Central Processing Unit 14
 - 2.4 Direct Memory Access 16
 - 2.5 Bus Master 17
 - 2.6 Input/Output Memory Management Units 18
 - 2.7 Trust and Adversary/Attacker Model 18
- 3 Related Work 21**
 - 3.1 DMA Attacks 21
 - 3.1.1 Devices Connectable from the Outside 21
 - 3.1.2 Devices Firmly Established Inside the Platform Chassis. 23
 - 3.2 Countermeasure Approaches 24
 - 3.2.1 Measured Firmware 24
 - 3.2.2 Signed Firmware 25
 - 3.2.3 Software/Latency-Based Attestation 25
 - 3.2.4 Monitoring Approaches 26
 - 3.2.5 Bus Snooping Approaches 26
 - 3.2.6 Sensitive Data Protection 27
 - 3.2.7 Input/Output Memory Management Unit 27

3.3	Secure Communication Channels Considering Platform State Reporting	27
3.3.1	Trusted Platform Module Based Approaches	28
3.3.2	Co-processor and Smart Card Based Approaches	30
4	Study of a Stealthy, Direct Memory Access Based Malicious Software	33
4.1	DMA Malware Definition	34
4.2	DMA Malware Core Functionality	35
4.3	Design and Implementation of DAGGER	36
4.3.1	General Design	36
4.3.2	Implementation Based on Intel's ME Environment	37
4.3.3	Attack Implementation Details for Linux and Windows Targets.	39
4.4	Evaluation.	43
4.4.1	DMA Malware Fulfillment	43
4.4.2	Effectiveness and Efficiency	45
4.4.3	ME Firmware Condition.	47
4.4.4	I/OMMU	47
4.5	Countermeasures Considerations	47
4.5.1	I/OMMU Issues	48
4.5.2	Detection Approach Based on DMA Side Effects	49
4.6	Chapter Summary	51
5	A Primitive for Detecting DMA Malware	53
5.1	General Detection Model	55
5.2	An Implementation of the Detection Model.	56
5.2.1	Bus Master Analysis	57
5.2.2	Bus Agent Runtime Monitor.	62
5.3	Evaluation of the Detection Model Implementation	64
5.3.1	Tolerance Value \mathcal{T}	64
5.3.2	Performance Overhead When Permanently Monitoring.	65
5.3.3	A Use Case to Demonstrate BARM's Effectiveness.	66
5.4	Limitations of Current BARM Implementation	68
5.5	Chapter Summary	69
6	Authentic Reporting to External Platforms	71
6.1	Implementation Independent Model	73
6.1.1	Negotiating an Authentic Reporting Channel.	74
6.2	Implementation of the Authentic Reporting Channel for BARM	76
6.2.1	Bus Master Analysis: Ethernet Controller	76
6.2.2	Implementation Based on OpenSSL.	80

6.3	Evaluation.	86
6.3.1	Expected Bus Activity Validation	87
6.3.2	Network Performance Overhead Evaluation	89
6.3.3	Test with DAGGER.	90
6.4	Security Considerations	92
6.5	Chapter Summary	93
7	Conclusions and Future Work	95
	References.	99

Detecting Peripheral-based Attacks on the Host
Memory

Stewin, P.

2015, XV, 108 p. 35 illus., 34 illus. in color., Hardcover

ISBN: 978-3-319-13514-4