1. Record Nr. UNINA9910864193603321

Autore	Niu Weina
Titolo	Android Malware Detection and Adversarial Methods / / by Weina Niu, Xiaosong Zhang, Ran Yan, Jiacheng Gong
Pubbl/distr/stampa	Singapore : , : Springer Nature Singapore : , : Imprint : Springer, , 2024
ISBN	981-9714-59-1
Edizione	[1st ed. 2024.]
Descrizione fisica	1 online resource (xiv, 190 pages) : illustrations
Altri autori (Persone)	ZhangXiaosong <1968-> YanRan <active 2019-> GongJiacheng
Disciplina	005.8
Soggetti	Computer networks - Security measures Data protection Data protection - Law and legislation Machine learning Blockchains (Databases) Mobile and Network Security Data and Information Security Security Services Privacy Machine Learning Blockchain Cadena de blocs (Bases de dades) Protecció de dades Aprenentatge automàtic Llibres electrònics
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Intro -- Foreword -- Preface -- Acknowledgments -- Contents -- Part I The Overview of Android Malware Detection -- 1 Introduction of Android Malware Detection -- 1.1 Android Malware Family -- 1.1.1 Trojan Horse -- 1.1.2 Viruses -- 1.1.3 The Back Door -- 1.1.4 Zombies -- 1.1.5 Espionage -- 1.1.6 Intimidation -- 1.1.7 Extortion -- 1.1.8

Advertising -- 1.1.9 Tracking -- 1.2 History of Android Malware Detection -- 1.3 Android Malware Detection Overview -- 1.4 Challenges and Apps of Android Malware Detection -- 1.5 Domestic and International Android Malware Detection -- 1.5.1 Android Malware Detection Method Based on Static Analysis -- 1.5.2 Android Malware Detection Method Based on Dynamic Analysis -- 1.5.3 Android Malware Detection Method Based on Hybrid Analysis -- 1.6 Chapter Summary -- References -- Part II The General Android Malware Detection Method -- 2 Feature Code Based Android Malware Detection Method -- 2.1 Detection Based on Traditional Feature Codes -- 2.1.1 Introduction -- 2.1.2 DroidAnalyzer: A Case Study in Android Malware Analysis -- 2.1.2.1 Suspicious Android APIs and Keywords 3 -- 2.1.2.2 Main Algorithm of DroidAnalyzer -- 2.2 Detection Based on Semantic Feature Codes -- 2.2.1 Introduction -- 2.2.2 DroidNative: A Case Study in Android Malware Analysis -- 2.2.2.1 Static Analysis in DroidNative -- 2.2.2.2 System Design and Implementation -- 2.3 Chapter Summary -- References -- 3 Behavior-Based Detection Method for Android Malware -- 3.1 Privacy Disclosure -- 3.2 Permission Escalation -- 3.2.1 Permission Escalation Method -- 3.2.2 Authorization Based on Configuration Files -- 3.2.3 Code Analysis -- 3.2.4 Taint Analysis -- 3.3 Machine Learning Technology and Malicious Behavior of Android Software -- 3.4 Chapter Summary -- References -- 4 AI-Based Android Malware Detection Methods.
4.1 Detection Based on Permissions, APIs, and Components -- 4.1.1 Permissions in Android System -- 4.1.1.1 Permissions in Android System -- 4.1.1.2 Overview of Permission-Based Detection Methods -- 4.1.2 Detection Based on API -- 4.1.3 Component-Based Detection -- 4.1.3.1 Components of an Application -- 4.1.3.2 Overview of Component-Based Detection Methods -- 4.1.4 Specific Case: Drebin -- 4.1.4.1 Static Analysis of Applications -- 4.1.4.2 Embedding in Vector Space -- 4.1.4.3 Learning-Based Detection -- 4.1.4.4 Explanation -- 4.2 Detection Anchored in Dynamic Runtime Features -- 4.2.1 Dynamic Analysis and Runtime Features -- 4.2.2 Overview of Detection Methods Based on Dynamic Runtime Features -- 4.2.3 Specific Case: EnDroid -- 4.2.3.1 Training Phase -- 4.2.3.2 Detection Phase -- 4.3 Detection Through Semantic Code Analysis -- 4.3.1 Dalvik Bytecode -- 4.3.2 Overview of Code Semantic-Based Detection Methods -- 4.3.3 Specific Case: MviiDroid -- 4.3.3.1 Static Analysis Phase -- 4.3.3.2 Feature Generation Phase -- 4.3.3.3 Model Training Phase -- 4.4 Detection via Image Analysis -- 4.4.1 Overview of Image-Based Detection Methods -- 4.4.2 Specific Case: R2-D2 -- 4.5 Detection Through Graph Analysis -- 4.5.1 Overview of Homogeneous Graph-Based Detection Methods -- 4.5.2 Overview of Heterogeneous Graph-Based Detection Methods -- 4.5.3 Case Study: HAWK -- 4.5.3.1 Feature Engineering -- 4.5.3.2 Constructing Heterogeneous Information Network (HIN) -- 4.5.3.3 Constructing Application Graph from HIN -- 4.6 Chapter Summary -- References -- Part III The Adversarial Method for Android Malware Detection -- 5 Static Adversarial Method -- 5.1 Static Obfuscation -- 5.1.1 Code Obfuscation -- 5.1.2 Resource Obfuscation -- 5.1.3 Manifest File Obfuscation -- 5.1.4 Control Flow Obfuscation -- 5.2 Common APK Static Obfuscation Tools -- 5.2.1 Obfuscapk -- 5.2.2 ProGuard.
5.2.3 DexGuard -- 5.2.4 Allatori -- 5.2.5 DashO -- 5.2.6 Bangcle -- 5.2.7 Arxan -- 5.2.8 Comparative Analysis -- 5.3 Research on Static Obfuscation -- 5.3.1 Detection Methods Based on New Features -- 5.3.1.1 Static Detection Based on Perceptual Hashing -- 5.3.1.2 Static Detection Based on Semantic Feature Set -- 5.3.1.3 Static Detection Based on Static Data Streams -- 5.3.1.4 Static Detection Based on

Grayscale Images -- 5.3.1.5 Static Detection Based on Permission Pairs -- 5.3.1.6 Static Detection Based on Static Sensitive Subgraphs -- 5.3.1.7 Static Detection Based on Malicious URLs -- 5.3.2 Detection Method Based on Binding Method -- 5.3.2.1 Static Detection Combined with Dynamic -- 5.3.2.2 Static Detection Combined with Machine Learning -- 5.3.2.3 Static Detection Combined with Deep Learning -- 5.4 Chapter Summary -- References -- 6 Dynamic Adversarial Method in Android Malware -- 6.1 Automatic Dynamic Analysis Evasion -- 6.1.1 Detection Dependent -- 6.1.1.1 Fingerprint -- 6.1.1.2 Reverse Turing Test -- 6.1.1.3 Target -- 6.1.2 Detection Independent -- 6.1.2.1 Stalling -- 6.1.2.2 Trigger-Based -- 6.1.2.3 Fileless Attack -- 6.2 Manual Dynamic Analysis Evasion -- 6.2.1 Direct Detection -- 6.2.1.1 Read PEB -- 6.2.1.2 Breakpoint Query -- 6.2.1.3 System Artifacts -- 6.2.1.4 Parent Process Detection -- 6.2.2 Deductive Detection -- 6.2.2.1 Trap -- 6.2.2.2 Time-Based Detection -- 6.2.3 Debugger Evasion -- 6.2.3.1 Control Flow Manipulation -- 6.2.3.2 Lockout Evasion -- 6.2.3.3 Debugger Identification -- 6.2.3.4 Fileless Malware -- 6.3 Related Research About Dynamic Analysis Evasion -- 6.3.1 Research About Improving Sandbox -- 6.3.1.1 The Droid is in the Details: Environment-Aware Evasion of Android Sandboxes -- 6.3.1.2 Morpheus: Automatically Generating Heuristics to Detect Android Emulators -- 6.3.2 Research About Detecting Dynamic Evasion. 6.3.2.1 CamoDroid: An Android App Analysis Environment Resilient Against Sandbox Evasion -- 6.3.2.2 Lumus: Dynamically Uncovering Evasive Android apps -- 6.4 Chapter Summary -- References -- 7 AI-Based Adversarial Method in Android -- 7.1 Introduction to Adversarial Examples -- 7.2 Classification of Adversarial Example Generation Methods -- 7.2.1 Gradient-Based Attacks -- 7.2.2 Optimization-Based Attacks -- 7.2.3 GAN-Based Attacks -- 7.2.4 Domain-Specific Attacks (Audio, Images, Text, etc.) -- 7.3 Black-Box Attacks -- 7.3.1 Introduction to Black-Box Attacks -- 7.3.2 Common Black-Box Attack Methods -- 7.3.3 Transfer Learning-Based Black-Box Attacks -- 7.3.4 Meta-Model Based Black-Box Attacks -- 7.3.5 Query-Based Attacks -- 7.3.6 Optimization-Based Attacks -- 7.4 White-Box Attacks -- 7.4.1 Optimization-Based Attacks -- 7.4.1.1 C&amp -- W Attack -- 7.4.1.2 PGD Attack -- 7.4.2 Gradient-Based Attacks -- 7.4.2.1 FGSM Attack -- 7.4.2.2 BIM Attack -- 7.4.3 App of Adversarial Attacks in Malware Detection -- 7.5 Chapter Summary -- References -- Part IV The Future Trends of Android Malware Detection -- 8 Future Trends in Android Malware Detection -- 8.1 Machine Learning And Deep Learning Techniques -- 8.1.1 Overview of Machine Learning and Deep Learning for Android Malware Detection -- 8.1.2 Challenges Faced -- 8.2 Integrated Solutions -- 8.2.1 Challenges Faced -- 8.3 Blockchain Technology -- 8.3.1 Introduction to Blockchain Technology -- 8.3.2 Examples of Blockchain Technology in the Field of Android Malware Detection -- 8.4 Hardware Technology -- 8.4.1 Advantages of Hardware Technology -- 8.4.2 Challenges to Hardware Technology -- 8.4.3 Examples of Hardware Technologies Applied in the Field of Android Malware Detection -- 8.5 BPF Technology -- 8.5.1 Development of BPF Technology -- 8.5.2 eBPF Technology Overview. 8.5.3 Examples of BPF Techniques in the Field of Android Malware Detection -- 8.6 Chapter Summary -- References.

Sommario/riassunto

The rise of Android malware poses a significant threat to users' information security and privacy. Malicious software can inflict severe harm on users by employing various tactics, including deception, personal information theft, and device control. To address this issue, both academia and industry are continually engaged in research and development efforts focused on detecting and countering Android

malware. This book is a comprehensive academic monograph crafted against this backdrop. The publication meticulously explores the background, methods, adversarial approaches, and future trends related to Android malware. It is organized into four parts: the overview of Android malware detection, the general Android malware detection method, the adversarial method for Android malware detection, and the future trends of Android malware detection. Within these sections, the book elucidates associated issues, principles, and highlights notable research. By engaging with this book, readers will gain not only a global perspective on Android malware detection and adversarial methods but also a detailed understanding of the taxonomy and general methods outlined in each part. The publication illustrates both the overarching model and representative academic work, facilitating a profound comprehension of Android malware detection.