1.	Record Nr.	UNINA9910492141603321
	Titolo	Android malware detection using machine learning : data-driven fingerprinting and threat intelligence / / ElMouatez Billah Karbab [and three others]
	Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2021] ©2021
	ISBN	3-030-74664-X
	Descrizione fisica	1 online resource (212 pages)
	Collana	Advances in Information Security ; ; v.86
	Disciplina	005.8
	Soggetti	Malware (Computer software) - Prevention Computer security - Standards
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Nota di contenuto	Intro -- Contents -- List of Figures -- List of Tables -- 1 Introduction -- 1.1 Motivations -- 1.2 Objectives -- 1.3 Research Contributions -- 1.4 Book Organization -- References -- 2 Background and Related Work -- 2.1 Background -- 2.1.1 Android OS Overview -- 2.1.1.1 Android Apk Format -- 2.1.1.2 Android Markets -- 2.1.2 Android Security -- 2.1.2.1 Android Security Threats -- 2.1.2.2 Design Challenges of Malware Detection Systems -- 2.2 Android Malware Detection Overview -- 2.3 Taxonomy of Android Malware Detection Systems -- 2.3.1 Malware Threats -- 2.3.2 Detection System Deployment -- 2.4 Performance Criteria for Malware Detection -- 2.4.1 Feature Selection -- 2.4.2 Detection Strategy -- 2.5 General Malware Threat Detection -- 2.5.1 Workstation-Based Solutions -- 2.5.2 Mobile-Based Solutions -- 2.5.3 Hybrid Solutions -- 2.5.4 Discussions -- 2.6 Specific Malware Threat Detection -- 2.6.1 Workstation-Based Solutions -- 2.6.2 Mobile-Based Solutions -- 2.6.3 Hybrid Solutions -- 2.6.4 Discussions -- 2.7 Android Malware Detection Helpers -- 2.7.1 Discussions -- 2.8 Summary -- References -- 3 Fingerprinting Android Malware Packages -- 3.1 Approximate Static Fingerprint -- 3.1.1 Fingerprint Structure -- 3.1.2 Fingerprints Generation -- 3.1.2.1 N-grams -- 3.1.2.2 Feature Hashing -- 3.1.2.3 Fingerprint Computation Process -- 3.1.2.4 Compute Fingerprints Similarity -- 3.2 Malware

Detection Framework -- 3.2.1 Peer-Fingerprint Voting -- 3.2.2 Peer-Matching -- 3.2.2.1 Family-Fingerprinting -- 3.3 Experimental Results -- 3.3.1 Testing Setup -- 3.3.2 Evaluation Results -- 3.3.2.1 Family-Fingerprinting Results -- 3.3.2.2 Peer-Matching Results -- 3.3.2.3 Peer-Voting vs Merged Fingerprints -- 3.3.3 Discussion -- 3.4 Summary -- References -- 4 Robust Android Malicious Community Fingerprinting -- 4.1 Threat Model -- 4.2 Usage Scenarios -- 4.3 Clustering Process.
4.4 Static Features -- 4.4.1 N-grams -- 4.4.1.1 Classes.dex Byte N-grams -- 4.4.1.2 Assembly Opcodes N-grams -- 4.4.2 Native Library N-grams -- 4.4.2.1 APK N-grams -- 4.4.3 Manifest File Features -- 4.4.4 Android API Calls -- 4.4.5 Resources -- 4.4.6 APK Content Types -- 4.4.7 Feature Preprocessing -- 4.5 LSH Similarity Computation -- 4.6 Community Detection -- 4.7 Community Fingerprint -- 4.8 Experimental Results -- 4.8.1 Dataset and Test Setup -- 4.8.1.1 App Detection Metrics -- 4.8.1.2 Community Detection Metrics -- 4.8.2 Mixed Dataset Results -- 4.8.3 Results of Malware-Only Datasets -- 4.8.4 Community Fingerprint Results -- 4.9 Hyper-Parameter Analyses -- 4.9.1 Purity Analysis -- 4.9.2 Coverage Analysis -- 4.9.3 Number of Communities Analysis -- 4.9.4 Efficiency Analysis -- 4.10 Case Study: Recall and Precision Settings -- 4.11 Case Study: Obfuscation -- 4.12 Summary -- References -- 5 Android Malware Fingerprinting Using Dynamic Analysis -- 5.1 Threat Model -- 5.2 Overview -- 5.2.1 Notation -- 5.3 Methodology -- 5.3.1 Behavioral Reports Generation -- 5.3.2 Report Vectorization -- 5.3.3 Build Models -- 5.3.4 Ensemble Composition -- 5.3.5 Ensemble Prediction Process -- 5.4 MalDy Framework -- 5.4.1 Machine Learning Algorithms -- 5.5 Evaluation Results -- 5.5.1 Evaluation Datasets -- 5.5.2 Effectiveness -- 5.5.2.1 Classifier Effect -- 5.5.2.2 Effect of the Vectorization Technique -- 5.5.2.3 Effect of Tuning Hyper-Parameters -- 5.5.3 Portability -- 5.5.3.1 MalDy on Win32 Malware -- 5.5.3.2 MalDy Train Dataset Size -- 5.5.4 Efficiency -- 5.6 Summary -- References -- 6 Fingerprinting Cyber-Infrastructures of Android Malware -- 6.1 Threat Model -- 6.2 Usage Scenarios -- 6.3 Methodology -- 6.3.1 Threat Communities Detection -- 6.3.2 Action Prioritization -- 6.3.2.1 PageRank Algorithm -- 6.3.3 Security Correlation.
6.3.3.1 Network Enrichment Using Passive DNS -- 6.3.3.2 Threat Network Tagging -- 6.4 Experimental Results -- 6.4.1 Android Malware Dataset -- 6.4.2 Implementation -- 6.4.3 Drebin Threat Network -- 6.4.4 Family Threat Networks -- 6.5 Summary -- References -- 7 Portable Supervised Malware Fingerprinting Using Deep Learning -- 7.1 Threat Model -- 7.2 Usage Scenarios -- 7.3 Methodology -- 7.3.1 MalDozer Method Embedding -- 7.3.2 MalDozer Neural Network -- 7.3.3 Implementation -- 7.4 Evaluation -- 7.4.1 Datasets -- 7.4.2 Malware Detection Performance -- 7.4.2.1 Unknown Malware Detection -- 7.4.2.2 Resiliency Against API Evolution Over Time -- 7.4.2.3 Resiliency Against Changing the Order of API Methods -- 7.4.3 Family Attribution Performance -- 7.4.4 Runtime Performance -- 7.4.4.1 Model Complexity Evaluation -- 7.5 Summary -- References -- 8 Resilient and Adaptive Android Malware Fingerprinting and Detection -- 8.1 Methodology -- 8.1.1 Approach -- 8.1.2 Android App Representation -- 8.1.3 Malware Detection -- 8.1.3.1 Fragment Detection -- 8.1.3.2 Inst2Vec Embedding -- 8.1.3.3 Classification Model -- 8.1.3.4 Dataset Notation -- 8.1.3.5 Detection Ensemble -- 8.1.3.6 Confidence Analysis -- 8.1.3.7 PetaDroid Adaptation Mechanism -- 8.1.4 Malware Clustering -- 8.1.4.1 InstNGram2Vec -- 8.1.4.2 Deep Neural Auto-Encoder and Digest Generation -- 8.1.4.3 Malware Family Clustering -- 8.1.5 Implementation -- 8.2 Evaluation