1. Record Nr. UNINA9910887000903321

Autore	Garcia-Alfaro Joaquin
Titolo	Computer Security – ESORICS 2024 : 29th European Symposium on Research in Computer Security, Bydgoszcz, Poland, September 16–20, 2024, Proceedings, Part III / / edited by Joaquin Garcia-Alfaro, Rafa Kozik, Micha Chora, Sokratis Katsikas
Pubbl/distr/stampa	Cham : , : Springer Nature Switzerland : , : Imprint : Springer, , 2024
ISBN	3-031-70896-2
Edizione	[1st ed. 2024.]
Descrizione fisica	1 online resource (461 pages)
Collana	Lecture Notes in Computer Science, , 1611-3349 ; ; 14984
Altri autori (Persone)	KozikRafa ChorasMicha KatsikasSokratis
Disciplina	005.8
Soggetti	Data protection Cryptography Data encryption (Computer science) Computer networks - Security measures Computer networks Computer systems Data and Information Security Cryptology Security Services Mobile and Network Security Computer Communication Networks Computer System Implementation
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Preface -- Organization -- Contents - Part III -- Software and Systems Security -- Scheduled Execution-Based Binary Indirect Call Targets Refinement -- 1 Introduction -- 2 Overview -- 3 Basic Block Execution -- 3.1 Basic Block Scheduler -- 3.2 State Manager -- 4 Targets Inference -- 4.1 Background: Data Flow Analysis -- 4.2 Handling Branch Convergence -- 4.3 Handling Loop-Related Memory Access -- 4.4 Handling Multiple Calls -- 5 Implementation -- 6

Evaluation -- 6.1 Manual Analysis -- 6.2 Precision and Completeness -- 6.3 Case Studies -- 6.4 CFI Evaluation -- 6.5 Performance Evaluation -- 7 Related Work -- 8 Conclusion -- References -- Companion Apps or Backdoors? On the Security of Automotive Companion Apps -- 1 Introduction -- 2 The System Model and the Threat Model -- 2.1 The System Model -- 2.2 The Threat Model -- 3 Vehicle Companion App Analysis -- 3.1 App Function Analysis -- 3.2 Static Source Code Analysis -- 3.3 Network Traffic Analysis -- 3.4 Vulnerability Identification -- 4 Vulnerability Analysis and Results -- 4.1 App Data Collection -- 4.2 Experiment Setup -- 4.3 CAN Control Messages and Vulnerability Evaluation -- 4.4 Vulnerability Assessment -- 4.5 Attack Summary -- 5 Discussions -- 6 Related Work -- 7 Conclusion -- A Summary of Vulnerabilities in Top Apps -- References -- A Study of Malicious Source Code Reuse Among GitHub, StackOverflow and Underground Forums -- 1 Introduction -- 2 Related Work -- 3 Taxonomy of Clones -- 4 Methodology -- 4.1 Data Collection -- 4.2 Pre-processing -- 4.3 Malicious Code Reuse Detection -- 4.4 Analysis -- 5 Evaluation -- 5.1 Evaluation Dataset -- 5.2 Evaluation of Methodology -- 5.3 Evaluation of Function-Call Extraction Techniques -- 6 Code Reuse Measurement -- 6.1 C and C++ -- 6.2 Java -- 6.3 Python -- 6.4 Findings -- 7 Discussion and Conclusions -- 7.1 Limitations -- 7.2 Key Takeaways.
7.3 Conclusion -- A Benign Datasets -- B Prominent Measurement Clusters -- B.1 C/C++ Clusters -- B.2 Java Clusters -- B.3 Python Clusters -- References -- Predicting Code Vulnerability Types via Heterogeneous GNN Learning -- 1 Introduction -- 2 Inter-procedural Compressed Code Property Graph -- 2.1 Building CCPGs -- 2.2 Processing Call Relationships -- 3 Heterogeneous GNN Learning for Multi-class Vulnerability Detection -- 3.1 Embeddings -- 3.2 Heterogeneous GNN Training -- 4 Experiments -- 5 Related Work -- 6 Conclusions -- References -- WASMixer: Binary Obfuscation for WebAssembly -- 1 Introduction -- 2 Background -- 2.1 WebAssembly (Wasm) -- 2.2 Obfuscation -- 3 WASMixer: Design and Challenges -- 3.1 Overview -- 3.2 Why Binary Obfuscator? -- 3.3 Challenges -- 4 Approach -- 4.1 Data Obfuscator -- 4.2 Code Obfuscator -- 5 Implementation and Evaluation -- 5.1 Implementation and Research Questions -- 5.2 RQ1: Semantic Consistency -- 5.3 RQ2: Effectiveness -- 5.4 RQ3: Overhead -- 6 Related Work -- 7 Threats of Validity -- 8 Conclusion -- A Appendix Figures and Tables -- References -- BloomFuzz: Unveiling Bluetooth L2CAP Vulnerabilities via State Cluster Fuzzing with Target-Oriented State Machines -- 1 Introduction -- 2 Motivation -- 2.1 Background -- 2.2 Technical Challenges -- 3 Design of BloomFuzz -- 3.1 State Machine Construction (P1) -- 3.2 Cluster-Based Packet Mutation (P2) -- 3.3 Crash Detection (P3) -- 4 Evaluation -- 4.1 Experimental Setup -- 4.2 Experiment on Crash Detection -- 4.3 Effectiveness of State Machine Generation -- 4.4 Efficiency of State Tracking and Packet Mutation -- 5 Discussion -- 6 Related Works -- 7 Conclusion -- A Discovered Crashes -- B Efficiency in Addressing Missing and Hidden States -- References -- TGRop: Top Gun of Return-Oriented Programming Automation -- 1 Introduction.
2 Shortcomings of State-of-the-Art Approaches -- 3 Methodology -- 3.1 Preprocessing: Blueprint Generation -- 3.2 Preprocessing: Analyzing Gadgets -- 3.3 Phase I: Achieving Sub-goals -- 3.4 Phase II: Resolving Dependencies -- 3.5 Phase III: Eliminating Side-Effects -- 3.6 Phase IV: Generating Final Chains -- 4 Implementation -- 5 Evaluation -- 5.1 Experiment Setup -- 5.2 Performance of TGRop -- 5.3 Ablation Analysis of TGRop -- 5.4 New Findings and Real-World Impact -- 6 Discussion -- 7 Conclusion -- A Appendix -- A.1 IRB Process --

References -- Formal Hardware/Software Models for Cache Locking Enabling Fast and Secure Code -- 1 Introduction -- 2 Hypotheses and Background -- 3 Memory Interface and Models of Cache -- 3.1 Software Cache Model -- 3.2 Hardware Cache Models -- 4 Evaluation -- 5 Observational Non-Interference with Attacker -- 5.1 Semantics of Instructions and Processes -- 5.2 ONI Preservation Principle with Attacker -- 5.3 Simulation and Indistinguishability -- 5.4 Discussion -- 6 Related Work -- 7 Conclusion -- A Evaluation of Algorithms with Input Dependent Locks -- B Semantics of Instructions -- C Proof of Theorem 1 -- References -- SerdeSniffer: Enhancing Java Deserialization Vulnerability Detection with Function Summaries -- 1 Introduction -- 1.1 Motivation -- 1.2 Research Contributions -- 1.3 Structure of the Paper -- 2 Framework Overview -- 3 Algorithms -- 3.1 Bottom-Up Information Flow Summary (BIFSum) -- 3.2 Data Processing -- 4 Experiments -- 4.1 Experimental Setup -- 4.2 Test Dataset -- 4.3 Effectiveness -- 4.4 Vulnerability Discovery -- 5 Discussion -- 6 Related Work -- 7 Conclusion -- A Appendix -- A.1 Clojure Command Execution -- References -- Interp-flow Hijacking: Launching Non-control Data Attack via Hijacking eBPF Interpretation Flow -- 1 Introduction -- 2 eBPF Background -- 2.1 eBPF Interpreter.
2.2 eBPF Programs and Maps -- 3 Threat Model and Assumptions -- 4 eBPF Interpretation Flow Hijacking -- 4.1 Overview -- 4.2 Identifying Hijack Targets -- 4.3 Tailcall Trampoline -- 5 Exploitability Evaluation -- 5.1 CVE Capability Requirement Analysis -- 5.2 Pivoting General CVE Capability -- 5.3 CVE Summary -- 6 Mitigation -- 6.1 Design -- 6.2 Implementation -- 6.3 Performance Evaluation -- 7 Related Work -- 7.1 eBPF and Bytecode Security -- 7.2 Common Kernel Attacks -- 8 Conclusion -- A Analysis Results -- B Arbitrary Kernel Code Execution -- References -- Applied Cryptopgraphy -- Fully Homomorphic Training and Inference on Binary Decision Tree and Random Forest -- 1 Introduction -- 2 Backgrounds -- 2.1 Notation -- 2.2 Binary Decision Tree -- 2.3 CKKS (Cheon-Kim-Kim-Song) Scheme -- 3 Related Work -- 4 Models -- 4.1 System Setting and Protocol Overview -- 4.2 Problem Definition -- 5 Homomorphic Binary Decision Tree (HBDT) -- 5.1 HBDT-Training Algorithm -- 5.2 HBDT-Inference Algorithm -- 6 Extending to Homomoprhic Random Forests (HRF) -- 7 Experimental Results -- 7.1 CKKS and Subroutines -- 7.2 Performance of Inference -- 7.3 Performance of HRF -- 8 Discussion -- 8.1 System Model Without KM -- 8.2 Discussion on Meeting the Privacy Requirements -- 9 Conclusion -- References -- Constant-Size Unbounded Multi-hop Fully Homomorphic Proxy Re-encryption from Lattices -- 1 Introduction -- 1.1 Our Contributions -- 1.2 Related Works -- 2 Preliminaries -- 2.1 Notations -- 2.2 Gaussian Distributions -- 2.3 Cyclotomic Rings -- 2.4 (Ring) Learning with Errors Problem -- 2.5 BD and P2 Algorithms -- 2.6 LWE Public Key Encryption -- 2.7 Key Switching -- 2.8 Modulus Switching -- 3 Homomorphic Computation and Bootstrapping -- 3.1 Homomorphic NAND Gate Evaluation -- 3.2 Bootstrapping -- 4 Fully Homomorphic Proxy Re-Encryption (FHPRE) -- 5 FHPRE Scheme.
6 Security Proof -- 7 Multi-user Computation System Based on FHPRE -- 8 Performance Analysis -- 9 Conclusions and Future Works -- A Homomorphic Gates Evaluation -- References -- Key Recovery Attack on CRYSTALS-Kyber and Saber KEMs in Key Reuse Scenario -- 1 Introduction -- 1.1 Background -- 1.2 Related Work -- 1.3 Our Contribution -- 1.4 Organization -- 2 Preliminaries -- 2.1 Notation -- 2.2 Kyber -- 2.3 Saber -- 3 Attacks at Asiacrypt 2021 -- 4 Generalized Scenario of the Key Mismatch Attack -- 5 Attack Against CCA-Secure Kyber KEM -- 6 Experiments -- A Linear Programming Method --

References -- Secure Keyless Multi-party Storage Scheme -- 1 Introduction -- 2 Technical Overview -- 3 Generic Model -- 3.1 Multi-party Storage Scheme -- 3.2 KMPS Security Model -- 4 KMPS Instantiations -- 4.1 KAPRE - Upload Using Proxy Re-encryption -- 4.2 KAME - Upload Using Multikey Encryption -- 4.3 Common Download -- 5 Security Analysis -- 6 Instantiation and Experimental Results -- 7 Conclusion -- A Appendix -- References -- LLRing: Logarithmic Linkable Ring Signatures with Transparent Setup -- 1 Introduction -- 2 Technical Overview -- 2.1 Attack on DualDory -- 2.2 LLRing-P Linkable Ring Signature Scheme -- 2.3 LLRing-DL Linkable Ring Signature Scheme -- 3 Preliminaries and Models -- 4 DualDory -- 4.1 Malleability Attack on DualDory -- 5 LLRing-DL Linkable Ring Signature Scheme -- 6 LLRing-P Linkable Ring Signature Scheme -- 7 Empirical Evaluation -- 8 Conclusion -- A Additional Definitions -- References -- In Search of Partitioning Oracle Attacks Against TLS Session Tickets -- 1 Introduction -- 2 Background -- 2.1 Authenticated Encryption -- 2.2 Partitioning Oracle Attacks -- 2.3 TLS and Session Tickets -- 3 Partitioning Oracle Attacks on TLS -- 4 Library Evaluation -- 4.1 Methodology -- 4.2 Results -- 4.3 Attack Performance -- 5 Large-Scale Evaluation.
5.1 Library Identification.

Sommario/riassunto

This four-volume set LNCS 14982-14985 constitutes the refereed proceedings of the 29th European Symposium on Research in Computer Security, ESORICS 2024, held in Bydgoszcz, Poland, during September 16–20, 2024. The 86 full papers presented in these proceedings were carefully reviewed and selected from 535 submissions. They were organized in topical sections as follows: Part I: Security and Machine Learning. Part II: Network, Web, Hardware and Cloud; Privacy and Personal Datat Protection. Part III: Software and Systems Security; Applied Cryptopgraphy. Part IV: Attacks and Defenses; Miscellaneous.