

Eric Bodden

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/8842167/publications.pdf>

Version: 2024-02-01

125
papers

4,469
citations

471509

17
h-index

289244

40
g-index

125
all docs

125
docs citations

125
times ranked

1749
citing authors

#	ARTICLE	IF	CITATIONS
1	Explaining Static Analysis With Rule Graphs. IEEE Transactions on Software Engineering, 2022, 48, 678-690.	5.6	3
2	Identifying Challenges for OSS Vulnerability Scanners - A Study & Test Suite. IEEE Transactions on Software Engineering, 2022, 48, 3613-3625.	5.6	4
3	TaintBench: Automatic real-world malware benchmarking of Android taint analyses. Empirical Software Engineering, 2022, 27, 1.	3.9	11
4	Static data-flow analysis for software product lines in C. Automated Software Engineering, 2022, 29, .	2.9	5
5	Computation on Encrypted Data Using Dataflow Authentication. ACM Transactions on Privacy and Security, 2022, 25, 1-36.	3.0	0
6	Fluently specifying taint-flow queries with fluentTQL. Empirical Software Engineering, 2022, 27, .	3.9	4
7	CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs. IEEE Transactions on Software Engineering, 2021, 47, 2382-2400.	5.6	19
8	Qualitative and Quantitative Analysis of Callgraph Algorithms for Python. , 2021, , .		4
9	A Systematic Hardening of Java's Information Hiding. , 2021, , .		1
10	Automated cell header generator for Jupyter notebooks. , 2021, , .		2
11	SootFX: A Static Code Feature Extraction Tool for Java and Android. , 2021, , .		2
12	Into the Woods: Experiences from Building a Dataflow Analysis Framework for C/C++. , 2021, , .		1
13	SecuCheck: Engineering configurable taint analysis for software developers. , 2021, , .		3
14	Debugging Static Analysis. IEEE Transactions on Software Engineering, 2020, 46, 697-709.	5.6	7
15	ModGuard: Identifying Integrity & Confidentiality Violations in Java Modules. IEEE Transactions on Software Engineering, 2020, , 1-1.	5.6	1
16	A systematic literature review of model-driven security engineering for cyber-physical systems. Journal of Systems and Software, 2020, 169, 110697.	4.5	18
17	Security-Oriented Fault-Tolerance in Systems Engineering: A Conceptual Threat Modelling Approach for Cyber-Physical Production Systems. Advances in Intelligent Systems and Computing, 2020, , 1458-1469.	0.6	5
18	CogniCrypt _{GEN} : generating code for the secure usage of crypto APIs. , 2020, , .		7

#	ARTICLE	IF	CITATIONS
19	Heaps'n leaks. , 2020, , .		6
20	AuthCheck: Program-State Analysis for Access-Control Vulnerabilities. Lecture Notes in Computer Science, 2020, , 557-572.	1.3	0
21	PhASAR: An Inter-procedural Static Analysis Framework for C/C++. Lecture Notes in Computer Science, 2019, , 393-410.	1.3	39
22	The Impact of Developer Experience in Using Java Cryptography. , 2019, , .		11
23	Context-, flow-, and field-sensitive data-flow analysis using synchronized Pushdown systems. , 2019, 3, 1-29.		45
24	Architectural Runtime Verification. , 2019, , .		2
25	Explaining Static Analysis - A Perspective. , 2019, , .		5
26	SWAN_ASSIST: Semi-Automated Detection of Code-Specific, Security-Relevant Methods. , 2019, , .		1
27	A Qualitative Analysis of Android Taint-Analysis Results. , 2019, , .		15
28	Codebase-adaptive detection of security-relevant methods. , 2019, , .		12
29	ACMiner. , 2019, , .		11
30	Do Android taint analysis tools keep their promises?. , 2018, , .		57
31	Gamifying static analysis. , 2018, , .		3
32	The secret sauce in efficient and precise static analysis. , 2018, , .		14
33	Towards ensuring security by design in cyber-physical systems engineering processes. , 2018, , .		15
34	Model Checking the Information Flow Security of Real-Time Systems. Lecture Notes in Computer Science, 2018, , 27-43.	1.3	9
35	VISUFLOW. , 2018, , .		2
36	Tracking Load-Time Configuration Options. IEEE Transactions on Software Engineering, 2018, 44, 1269-1291.	5.6	23

#	ARTICLE	IF	CITATIONS
37	State of the systems security. , 2018, , .		5
38	Self-adaptive static analysis. , 2018, , .		7
39	IDE ^{<i>al</i>} : efficient and precise alias-aware dataflow analysis. , 2017, 1, 1-27.		12
40	Just-in-time static analysis. , 2017, , .		40
41	The Soot-Based Toolchain for Analyzing Android Apps. , 2017, , .		14
42	Cheetah: just-in-time taint analysis for android apps. , 2017, , .		5
43	Hardening Java™s Access Control by Abolishing Implicit Privilege Elevation. , 2017, , .		7
44	Time for Addressing Software Security Issues: Prediction Models and Impacting Factors. Data Science and Engineering, 2017, 2, 107-124.	6.4	18
45	CogniCrypt: Supporting developers in using cryptography. , 2017, , .		41
46	Toward an automated benchmark management system. , 2016, , .		5
47	An In-Depth Study of More Than Ten Years of Java Exploitation. , 2016, , .		22
48	ROPocop â€” Dynamic mitigation of code-reuse attacks. Journal of Information Security and Applications, 2016, 29, 16-26.	2.5	5
49	Towards a Comprehensive Model of Isolation for Mitigating Illicit Channels. Lecture Notes in Computer Science, 2016, , 116-138.	1.3	0
50	Information Flow Analysis for Go. Lecture Notes in Computer Science, 2016, , 431-445.	1.3	5
51	Towards cross-platform cross-language analysis with soot. , 2016, , .		7
52	Jumping through hoops. , 2016, , .		147
53	StubDroid. , 2016, , .		48
54	PSHAPE: Automatically Combining Gadgets for Arbitrary Method Execution. Lecture Notes in Computer Science, 2016, , 212-228.	1.3	8

#	ARTICLE	IF	CITATIONS
55	Investigating Users' Reaction to Fine-Grained Data Requests: A Market Experiment. , 2016, , .		8
56	Analyzing the Gadgets. Lecture Notes in Computer Science, 2016, , 155-172.	1.3	14
57	Harvesting Runtime Values in Android Applications That Feature Anti-Analysis Techniques. , 2016, , .		83
58	Towards secure integration of cryptographic software. , 2015, , .		18
59	Access-Path Abstraction: Scaling Field-Sensitive Data-Flow Analysis with Unbounded Access Paths (T). , 2015, , .		22
60	DroidSearch: A tool for scaling Android app triage to real-world app stores. , 2015, , .		10
61	Mining Apps for Abnormal Usage of Sensitive Data. , 2015, , .		111
62	Using targeted symbolic execution for reducing false-positives in dataflow analysis. , 2015, , .		15
63	Incorporating attacker capabilities in risk estimation and mitigation. Computers and Security, 2015, 51, 41-61.	6.0	22
64	How Current Android Malware Seeks to Evade Automated Code Analysis. Lecture Notes in Computer Science, 2015, , 187-202.	1.3	15
65	lccTA: Detecting Inter-Component Privacy Leaks in Android Apps. , 2015, , .		258
66	Dynamically Provisioning Isolation in Hierarchical Architectures. Lecture Notes in Computer Science, 2015, , 83-101.	1.3	2
67	Factors Impacting the Effort Required to Fix Security Vulnerabilities. Lecture Notes in Computer Science, 2015, , 102-119.	1.3	9
68	jÄ,,k: Using Dynamic Analysis to Crawl and Test Modern Web Applications. Lecture Notes in Computer Science, 2015, , 295-316.	1.3	22
69	FlowDroid. ACM SIGPLAN Notices, 2014, 49, 259-269.	0.2	801
70	Tracking load-time configuration options. , 2014, , .		29
71	Join point interfaces for safe and flexible decoupling of aspects. ACM Transactions on Software Engineering and Methodology, 2014, 23, 1-41.	6.0	29
72	A Machine-learning Approach for Classifying and Categorizing Android Sources and Sinks. , 2014, , .		213

#	ARTICLE	IF	CITATIONS
73	FlowTwist: efficient context-sensitive inside-out taint analysis for large codebases. , 2014, , .		24
74	Reviser: efficiently updating IDE-/IFDS-based data-flow analyses in response to incremental program changes. , 2014, , .		50
75	Denial-of-App Attack. , 2014, , .		6
76	Variational Data Structures. , 2014, , .		32
77	FlowDroid. , 2014, , .		659
78	DroidForce: Enforcing Complex, Data-centric, System-wide Policies in Android. , 2014, , .		47
79	TS4J. , 2014, , .		9
80	How to build the perfect Swiss army knife, and keep it sharp?. , 2014, , .		0
81	SPL ^{<sup>} LIFT</sup>. , 2013, , .		65
82	Automated API Property Inference Techniques. IEEE Transactions on Software Engineering, 2013, 39, 613-637.	5.6	152
83	A brief tour of join point interfaces. , 2013, , .		3
84	Easily instrumenting android applications for security purposes. , 2013, , .		4
85	Instrumenting Android and Java Applications as Easy as abc. Lecture Notes in Computer Science, 2013, , 364-381.	1.3	23
86	SPL ^{<sup>} LIFT</sup>. ACM SIGPLAN Notices, 2013, 48, 355-364.	0.2	29
87	Distributed Finite-State Runtime Monitoring with Aggregated Events. Lecture Notes in Computer Science, 2013, , 94-111.	1.3	1
88	Partially Evaluating Finite-State Runtime Monitors Ahead of Time. ACM Transactions on Programming Languages and Systems, 2012, 34, 1-52.	2.1	14
89	InvokeDynamic support in Soot. , 2012, , .		7
90	Inter-procedural data-flow analysis with IFDS/IDE and Soot. , 2012, , .		89

#	ARTICLE	IF	CITATIONS
91	Static flow-sensitive & context-sensitive information-flow analysis for software product lines. , 2012, , .		6
92	The Clara framework for hybrid typestate analysis. International Journal on Software Tools for Technology Transfer, 2012, 14, 307-326.	1.9	18
93	MOPBox: A Library Approach to Runtime Verification. Lecture Notes in Computer Science, 2012, , 365-369.	1.3	14
94	Challenges in Defining a Programming Language for Provably Correct Dynamic Analyses. Lecture Notes in Computer Science, 2012, , 4-18.	1.3	0
95	Challenges for Refinement and Composition of Instrumentations: Position Paper. Lecture Notes in Computer Science, 2012, , 86-96.	1.3	1
96	Delta-Oriented Monitor Specification. Lecture Notes in Computer Science, 2012, , 162-177.	1.3	0
97	Dynamic Anomaly Detection for More Trustworthy Outsourced Computation. Lecture Notes in Computer Science, 2012, , 168-187.	1.3	1
98	Taming reflection. , 2011, , .		171
99	Continuation equivalence. , 2011, , .		1
100	Stateful breakpoints. , 2011, , .		6
101	Join point interfaces for modular reasoning in aspect-oriented programs. , 2011, , .		18
102	Closure joinpoints. , 2011, , .		20
103	Aspect-Oriented Race Detection in Java. IEEE Transactions on Software Engineering, 2010, 36, 509-527.	5.6	28
104	Collaborative Runtime Verification with Tracematches. Journal of Logic and Computation, 2010, 20, 707-723.	0.8	29
105	IDE 2.0. , 2010, , .		19
106	Efficient hybrid typestate analysis by determining continuation-equivalent states. , 2010, , .		51
107	Effective API navigation and reuse. , 2010, , .		13
108	Clara: A Framework for Partially Evaluating Finite-State Runtime Monitors Ahead of Time. Lecture Notes in Computer Science, 2010, , 183-197.	1.3	37

#	ARTICLE	IF	CITATIONS
109	Reducing Configurations to Monitor in a Software Product Line. Lecture Notes in Computer Science, 2010, , 285-299.	1.3	16
110	Clara: Partially Evaluating Runtime Monitors at Compile Time. Lecture Notes in Computer Science, 2010, , 74-88.	1.3	6
111	Dependent advice. , 2009, , .		22
112	Relational aspects as tracematches. , 2008, , .		9
113	Finding programming errors earlier by evaluating runtime monitors ahead-of-time. , 2008, , .		56
114	Racer. , 2008, , .		58
115	Transforming Timeline Specifications into Automata for Runtime Monitoring. Lecture Notes in Computer Science, 2008, , 249-264.	1.3	1
116	The design and implementation of formal monitoring techniques. , 2007, , .		0
117	A Staged Static Program Analysis to Improve the Performance of Runtime Monitoring. Lecture Notes in Computer Science, 2007, , 525-549.	1.3	33
118	Collaborative Runtime Verification with Tracematches. , 2007, , 22-37.		24
119	Temporal Assertions using AspectJ. Electronic Notes in Theoretical Computer Science, 2006, 144, 109-124.	0.9	106
120	Efficient trace monitoring. , 2006, , .		14
121	Tracechecks: Defining Semantic Interfaces with Temporal Logic. Lecture Notes in Computer Science, 2006, , 147-162.	1.3	14
122	Aspects and Data Refinement. Lecture Notes in Computer Science, 2006, , 5-9.	1.3	0
123	A lightweight LTL runtime verification tool for java. , 2004, , .		13
124	A high-level view of Java applications. , 2003, , .		3
125	Object representatives: a uniform abstraction for pointer information. , 0, , .		3