Eric Bodden

List of Publications by Year
in descending order

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	FlowDroid. ACM SIGPLAN Notices, 2014, 49, 259-269.	0.2	801
2	FlowDroid. , 2014, , .		659
3	IccTA: Detecting Inter-Component Privacy Leaks in Android Apps. , 2015, , .		258
4	A Machine-learning Approach for Classifying and Categorizing Android Sources and Sinks. , 2014, , .		213
5	Taming reflection. , 2011, , .		171
6	Automated API Property Inference Techniques. IEEE Transactions on Software Engineering, 2013, 39, 613-637.	5.6	152
7	Jumping through hoops. , 2016, , .		147
8	Mining Apps for Abnormal Usage of Sensitive Data. , 2015, , .		111
9	Temporal Assertions using AspectJ. Electronic Notes in Theoretical Computer Science, 2006, 144, 109-124.	0.9	106
10	Inter-procedural data-flow analysis with IFDS/IDE and Soot. , 2012, , .		89
11	Harvesting Runtime Values in Android Applications That Feature Anti-Analysis Techniques. , 2016, , .		83
12	SPL ^{LIFT}. , 2013, , .		65
13	Racer. , 2008, , .		58
14	Do Android taint analysis tools keep their promises?. , 2018, , .		57
15	Finding programming errors earlier by evaluating runtime monitors ahead-of-time. , 2008, , .		56
16	Efficient hybrid typestate analysis by determining continuation-equivalent states. , 2010, , .		51
17	Reviser: efficiently updating IDE-/IFDS-based data-flow analyses in response to incremental program changes. , 2014, , .		50
18	StubDroid. , 2016, , .		48

#	Article	IF	Citations
19	DroidForce: Enforcing Complex, Data-centric, System-wide Policies in Android. , 2014, , .		47
20	Context-, flow-, and field-sensitive data-flow analysis using synchronized Pushdown systems. , 2019, 3, 1-29.		45
21	CogniCrypt: Supporting developers in using cryptography. , 2017, , .		41
22	Just-in-time static analysis. , 2017, , .		40
23	PhASAR: An Inter-procedural Static Analysis Framework for C/C++. Lecture Notes in Computer Science, 2019, , 393-410.	1.3	39
24	Clara: A Framework for Partially Evaluating Finite-State Runtime Monitors Ahead of Time. Lecture Notes in Computer Science, 2010, , 183-197.	1.3	37
25	A Staged Static Program Analysis to Improve the Performance of Runtime Monitoring. Lecture Notes in Computer Science, 2007, , 525-549.	1.3	33
26	Variational Data Structures. , 2014, , .		32
27	Collaborative Runtime Verification with Tracematches. Journal of Logic and Computation, 2010, 20, 707-723.	0.8	29
28	Tracking load-time configuration options. , 2014, , .		29
29	Join point interfaces for safe and flexible decoupling of aspects. ACM Transactions on Software Engineering and Methodology, 2014, 23, 1-41.	6.0	29
30	SPL ^{LIFT}. ACM SIGPLAN Notices, 2013, 48, 355-364.	0.2	29
31	Aspect-Oriented Race Detection in Java. IEEE Transactions on Software Engineering, 2010, 36, 509-527.	5.6	28
32	FlowTwist: efficient context-sensitive inside-out taint analysis for large codebases. , 2014, , .		24
33	Collaborative Runtime Verification with Tracematches. , 2007, , 22-37.		24
34	Tracking Load-Time Configuration Options. IEEE Transactions on Software Engineering, 2018, 44, 1269-1291.	5.6	23
35	Instrumenting Android and Java Applications as Easy as abc. Lecture Notes in Computer Science, 2013, , 364-381.	1.3	23
36	Dependent advice. , 2009, , .		22

#	Article	IF	Citations
37	Access-Path Abstraction: Scaling Field-Sensitive Data-Flow Analysis with Unbounded Access Paths (T). , 2015, , .		22
38	Incorporating attacker capabilities in risk estimation and mitigation. Computers and Security, 2015, 51, 41-61.	6.0	22
39	An In-Depth Study of More Than Ten Years of Java Exploitation. , 2016, , .		22
40	jÄk: Using Dynamic Analysis to Crawl and Test Modern Web Applications. Lecture Notes in Computer Science, 2015, , 295-316.	1.3	22
41	Closure joinpoints. , 2011, , .		20
42	IDE 2.0. , 2010, , .		19
43	CrySL: An Extensible Approach to Validating the Correct Usage of Cryptographic APIs. IEEE Transactions on Software Engineering, 2021, 47, 2382-2400.	5.6	19
44	Join point interfaces for modular reasoning in aspect-oriented programs. , 2011, , .		18
45	The Clara framework for hybrid typestate analysis. International Journal on Software Tools for Technology Transfer, 2012, 14, 307-326.	1.9	18
46	Towards secure integration of cryptographic software. , 2015, , .		18
47	Time for Addressing Software Security Issues: Prediction Models and Impacting Factors. Data Science and Engineering, 2017, 2, 107-124.	6.4	18
48	A systematic literature review of model-driven security engineering for cyber–physical systems. Journal of Systems and Software, 2020, 169, 110697.	4.5	18
49	Reducing Configurations to Monitor in a Software Product Line. Lecture Notes in Computer Science, 2010, , 285-299.	1.3	16
50	Using targeted symbolic execution for reducing false-positives in dataflow analysis. , 2015, , .		15
51	How Current Android Malware Seeks to Evade Automated Code Analysis. Lecture Notes in Computer Science, 2015, , 187-202.	1.3	15
52	Towards ensuring security by design in cyber-physical systems engineering processes. , 2018, , .		15
53	A Qualitative Analysis of Android Taint-Analysis Results. , 2019, , .		15
54	Efficient trace monitoring. , 2006, , .		14

#	Article	IF	Citations
55	Partially Evaluating Finite-State Runtime Monitors Ahead of Time. ACM Transactions on Programming Languages and Systems, 2012, 34, 1-52.	2.1	14
56	The Soot-Based Toolchain for Analyzing Android Apps. , 2017, , .		14
57	The secret sauce in efficient and precise static analysis. , 2018, , .		14
58	Tracechecks: Defining Semantic Interfaces with Temporal Logic. Lecture Notes in Computer Science, 2006, , 147-162.	1.3	14
59	Analyzing the Gadgets. Lecture Notes in Computer Science, 2016, , 155-172.	1.3	14
60	MOPBox: A Library Approach to Runtime Verification. Lecture Notes in Computer Science, 2012, , 365-369.	1.3	14
61	A lightweight LTL runtime verification tool for java. , 2004, , .		13
62	Effective API navigation and reuse. , 2010, , .		13
63	IDE ^{al} : efficient and precise alias-aware dataflow analysis. , 2017, 1, 1-27.		12
64	Codebase-adaptive detection of security-relevant methods. , 2019, , .		12
65	The Impact of Developer Experience in Using Java Cryptography. , 2019, , .		11
66	ACMiner. , 2019, , .		11
67	TaintBench: Automatic real-world malware benchmarking of Android taint analyses. Empirical Software Engineering, 2022, 27, 1.	3.9	11
68	DroidSearch: A tool for scaling Android app triage to real-world app stores. , 2015, , .		10
69	Relational aspects as tracematches. , 2008, , .		9
70	TS4J. , 2014, , .		9
71	Model Checking the Information Flow Security of Real-Time Systems. Lecture Notes in Computer Science, 2018, , 27-43.	1.3	9
72	Factors Impacting the Effort Required to Fix Security Vulnerabilities. Lecture Notes in Computer Science, 2015, , 102-119.	1.3	9

#	Article	IF	Citations
73	PSHAPE: Automatically Combining Gadgets for Arbitrary Method Execution. Lecture Notes in Computer Science, 2016, , 212-228.	1.3	8
74	Investigating Users' Reaction to Fine-Grained Data Requests: A Market Experiment. , 2016, , .		8
75	InvokeDynamic support in Soot. , 2012, , .		7
76	Towards cross-platform cross-language analysis with soot. , 2016, , .		7
77	Hardening Java's Access Control by Abolishing Implicit Privilege Elevation. , 2017, , .		7
78	Self-adaptive static analysis. , 2018, , .		7
79	Debugging Static Analysis. IEEE Transactions on Software Engineering, 2020, 46, 697-709.	5.6	7
80	CogniCrypt _{<i>GEN</i>} : generating code for the secure usage of crypto APIs. , 2020, , .		7
81	Stateful breakpoints. , 2011, , .		6
82	Static flow-sensitive & context-sensitive information-flow analysis for software product lines. , 2012, , .		6
83	Denial-of-App Attack. , 2014, , .		6
84	Clara: Partially Evaluating Runtime Monitors at Compile Time. Lecture Notes in Computer Science, 2010, , 74-88.	1.3	6
85	Heaps'n leaks. , 2020, , .		6
86	Toward an automated benchmark management system. , 2016, , .		5
87	ROPocop â€" Dynamic mitigation of code-reuse attacks. Journal of Information Security and Applications, 2016, 29, 16-26.	2.5	5
88	Information Flow Analysis for Go. Lecture Notes in Computer Science, 2016, , 431-445.	1.3	5
89	Cheetah: just-in-time taint analysis for android apps. , 2017, , .		5
90	State of the systems security. , 2018, , .		5

#	Article	IF	Citations
91	Explaining Static Analysis - A Perspective. , 2019, , .		5
92	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
93	Static data-flow analysis for software product lines in C. Automated Software Engineering, 2022, 29, .	2.9	5
94	Easily instrumenting android applications for security purposes. , 2013, , .		4
95	Qualitative and Quantitative Analysis of Callgraph Algorithms for Python. , 2021, , .		4
96	Identifying Challenges for OSS Vulnerability Scanners - A Study & Test Suite. IEEE Transactions on Software Engineering, 2022, 48, 3613-3625.	5.6	4
97	Fluently specifying taint-flow queries with fluentTQL. Empirical Software Engineering, 2022, 27, .	3.9	4
98	A high-level view of Java applications. , 2003, , .		3
99	A brief tour of join point interfaces. , 2013, , .		3
100	Gamifying static analysis. , 2018, , .		3
101	Explaining Static Analysis With Rule Graphs. IEEE Transactions on Software Engineering, 2022, 48, 678-690.	5.6	3
102	Object representatives: a uniform abstraction for pointer information. , 0, , .		3
103	SecuCheck: Engineering configurable taint analysis for software developers. , 2021, , .		3
104	VISUFLOW. , 2018, , .		2
105	Architectural Runtime Verification. , 2019, , .		2
106	Automated cell header generator for Jupyter notebooks. , 2021, , .		2
107	Dynamically Provisioning Isolation in Hierarchical Architectures. Lecture Notes in Computer Science, 2015, , 83-101.	1.3	2
108	SootFX: A Static Code Feature Extraction Tool for Java and Android. , 2021, , .		2

#	Article	IF	Citations
109	Continuation equivalence. , 2011, , .		1
110	SWAN_ASSIST: Semi-Automated Detection of Code-Specific, Security-Relevant Methods. , 2019, , .		1
111	ModGuard: Identifying Integrity &Confidentiality Violations in Java Modules. IEEE Transactions on Software Engineering, 2020, , 1-1.	5.6	1
112	A Systematic Hardening of Java's Information Hiding. , 2021, , .		1
113	Transforming Timeline Specifications into Automata for Runtime Monitoring. Lecture Notes in Computer Science, 2008, , 249-264.	1.3	1
114	Challenges for Refinement and Composition of Instrumentations: Position Paper. Lecture Notes in Computer Science, 2012, , 86-96.	1.3	1
115	Dynamic Anomaly Detection for More Trustworthy Outsourced Computation. Lecture Notes in Computer Science, 2012, , 168-187.	1.3	1
116	Distributed Finite-State Runtime Monitoring with Aggregated Events. Lecture Notes in Computer Science, 2013, , 94-111.	1.3	1
117	Into the Woods: Experiences from Building a Dataflow Analysis Framework for C/C++. , 2021, , .		1
118	The design and implementation of formal monitoring techniques. , 2007, , .		0
119	Towards a Comprehensive Model of Isolation for Mitigating Illicit Channels. Lecture Notes in Computer Science, 2016, , 116-138.	1.3	0
120	Aspects and Data Refinement. Lecture Notes in Computer Science, 2006, , 5-9.	1.3	0
121	Challenges in Defining a Programming Language for Provably Correct Dynamic Analyses. Lecture Notes in Computer Science, 2012, , 4-18.	1.3	0
122	Delta-Oriented Monitor Specification. Lecture Notes in Computer Science, 2012, , 162-177.	1.3	0
123	How to build the perfect Swiss army knife, and keep it sharp?. , 2014, , .		0
124	AuthCheck: Program-State Analysis for Access-Control Vulnerabilities. Lecture Notes in Computer Science, 2020, , 557-572.	1.3	0
125	Computation on Encrypted Data Using Dataflow Authentication. ACM Transactions on Privacy and Security, 2022, 25, 1-36.	3.0	0