

# Hee Beng Kuan Tan

## List of Publications by Citations

**Source:** <https://exaly.com/author-pdf/11363758/hee-beng-kuan-tan-publications-by-citations.pdf>

**Version:** 2024-04-28

This document has been generated based on the publications and citations recorded by exaly.com. For the latest version of this publication list, visit the link given above.

The third column is the impact factor (IF) of the journal, and the fourth column is the number of citations of the article.

44  
papers

521  
citations

11  
h-index

21  
g-index

55  
ext. papers

675  
ext. citations

2.5  
avg, IF

4.02  
L-index

#	Paper	IF	Citations
44	Measuring design complexity of semantic web ontologies. <i>Journal of Systems and Software</i> , <b>2010</b> , 83, 803-814	3.3	70
43	Web Application Vulnerability Prediction Using Hybrid Program Analysis and Machine Learning. <i>IEEE Transactions on Dependable and Secure Computing</i> , <b>2015</b> , 12, 688-707	3.9	51
42	Automated removal of cross site scripting vulnerabilities in web applications. <i>Information and Software Technology</i> , <b>2012</b> , 54, 467-478	3.4	45
41	Mining SQL injection and cross site scripting vulnerabilities using hybrid program analysis <b>2013</b> ,		45
40	Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns. <i>Information and Software Technology</i> , <b>2013</b> , 55, 1767-1780	3.4	37
39	Heuristics-based infeasible path detection for dynamic test data generation. <i>Information and Software Technology</i> , <b>2008</b> , 50, 641-655	3.4	33
38	Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities <b>2012</b> ,		21
37	Testing input validation in Web applications through automated model recovery. <i>Journal of Systems and Software</i> , <b>2008</b> , 81, 222-233	3.3	21
36	Covering code behavior on input validation in functional testing. <i>Information and Software Technology</i> , <b>2009</b> , 51, 546-553	3.4	20
35	Defending against Cross-Site Scripting Attacks. <i>Computer</i> , <b>2012</b> , 45, 55-62	1.6	18
34	Applying static analysis for automated extraction of database interactions in web applications. <i>Information and Software Technology</i> , <b>2008</b> , 50, 160-175	3.4	18
33	Checking enforcement of integrity constraints in database applications based on code patterns. <i>Journal of Systems and Software</i> , <b>2011</b> , 84, 2253-2264	3.3	11
32	Automated elicitation of functional dependencies from source codes of database transactions. <i>Information and Software Technology</i> , <b>2004</b> , 46, 109-117	3.4	11
31	Has this bug been reported? <b>2012</b> ,		9
30	An Empirical Study of Class Sizes for Large Java Systems <b>2007</b> ,		8
29	Buffer Overflow Vulnerability Prediction from x86 Executables Using Static Analysis and Machine Learning <b>2015</b> ,		7
28	An approach for the maintenance of input validation. <i>Information and Software Technology</i> , <b>2008</b> , 50, 449-461	3.4	7

27	Automated verification and test case generation for input validation <b>2006</b> ,		7
26	Exploring into programs for the recovery of data dependencies designed. <i>IEEE Transactions on Knowledge and Data Engineering</i> , <b>2002</b> , 14, 825-835	4.2	7
25	Auditing buffer overflow vulnerabilities using hybrid staticdynamic analysis. <i>IET Software</i> , <b>2016</b> , 10, 54-61	1	6
24	Binary Code Analysis. <i>Computer</i> , <b>2013</b> , 46, 60-68	1.6	6
23	Has this bug been reported? <b>2013</b> ,		6
22	Extraction of Attribute Dependency Graph from Database Applications <b>2011</b> ,		6
21	An Empirical Study of Class Sizes for Large Java Systems. <i>Proceedings of the Asia Pacific Software Engineering Conference</i> , <b>2007</b> ,		6
20	Detection of Infeasible Paths: Approaches and Challenges. <i>Communications in Computer and Information Science</i> , <b>2013</b> , 64-78	0.3	5
19	Defending against Buffer-Overflow Vulnerabilities. <i>Computer</i> , <b>2011</b> , 44, 53-60	1.6	4
18	An Approach to Aid the Understanding and Maintenance of Input Validation. <i>Conference on Software Maintenance, Proceedings of the</i> , <b>2006</b> ,		4
17	Automated elicitation of inclusion dependencies from the source code for database transactions. <i>Journal of Software: Evolution and Process</i> , <b>2003</b> , 15, 379-392		4
16	On formalization of the whole-part relationship in the Unified Modeling Language. <i>IEEE Transactions on Software Engineering</i> , <b>2003</b> , 29, 1054-1055	3.5	4
15	Software cost estimation through conceptual requirement <b>2003</b> ,		4
14	An approach for extracting code fragments that implement functionality from source programs. <i>Journal of Software: Evolution and Process</i> , <b>2001</b> , 13, 53-75		4
13	Predicting Buffer Overflow Vulnerabilities through Mining Light-Weight Static Code Attributes <b>2014</b> ,		3
12	Detecting infeasible branches based on code patterns <b>2014</b> ,		3
11	Aiding Maintenance of Database Applications Through Extracting Attribute Dependency Graph. <i>Journal of Database Management</i> , <b>2013</b> , 24, 20-35	2.2	3
10	Automated Insertion of Exception Handling for Key and Referential Constraints <b>2012</b> ,		2

9	Faceted Bug Report Search with Topic Model <b>2014</b> ,		1
8	Detection of Buffer Overflow Vulnerabilities in C/C++ with Pattern Based Limited Symbolic Evaluation <b>2012</b> ,		1
7	A method for the recovery of inclusion dependencies from data-intensive business programs. <i>Information and Software Technology</i> , <b>1997</b> , 39, 27-34	3-4	1
6	Empirical-based recovery and maintenance of input error-correction features. <i>Journal of Software: Evolution and Process</i> , <b>2007</b> , 19, 419-450		1
5	Components reuse for data-intensive business programs through an object-oriented architecture. <i>Journal of Systems and Software</i> , <b>1996</b> , 34, 3-20	3-3	1
4	Automated verification and testing of user-interactive undo features in database applications. <i>Software Testing Verification and Reliability</i> , <b>2012</b> , 22, 245-265	0-9	
3	Supporting the adaptation of open-source database applications through extracting data lifecycles. <i>IET Software</i> , <b>2013</b> , 7, 213-221		1
2	Reuse of components in data-intensive business programs through interface separation. <i>Information and Software Technology</i> , <b>1998</b> , 40, 15-25	3-4	
1	Automated Insertion of Exception Handling for Key and Referential Constraints. <i>Journal of Database Management</i> , <b>2013</b> , 24, 1-19	2-2	