Characterizing and understanding security risks through Secure-Aware Mutation Testing of RESTful APIs

Carlos Andres Delgado Saavedra
carlos.andres.delgado@correounivalle.edu.co

Universidad del Valle

UNIVERSITÉ DE NAMUR

AVISPA
AMBIENTES VISUALES DE PROGRAMACIÓN APLICATIVA

Colombia

October 14, 2023

Overview

1. Research Proposal
   - Specific
2. Literature Review
   - Challenges
3. Work plan

Universidad del Valle

Research Proposal

Problem

1. API-Restful is an architectural style for designing web services
2. RESTful APIs exchange sensitive information and private date
3. Top 10 vulnerabilities Application Security Project (OWASP) https://owasp.org/www-project-api-security/
4. Coverage of the security tests: penetration and policies
5. Opportunity for mutation testing

Research question

¿How to desing secure-aware mutation operators in the coverage of vulnerabilities in exchanging data in RESTful APis?

Objectives

Develop a collection of security-conscious mutation operators designed for safeguarding data integrity within Restful APIs.

Specific objective	Expected result
1. Selection of a common set of vulnerabilities and faults in data integrity in API-Restful	Selected vulnerabilities in API-Restful handle in this thesis
2. Describe a set of security-aware mutation operators for evaluating safeguarding data integrity in API-Restful	Description of the mutation operators
3. Develop a set of security-aware mutation operators for penetration testings in two Python API-Restful Frameworks	Source code of the secure-aware mutation operators
5. Evaluate the proposed security-aware mutation operators in RESTful APIs	Report about the performance of the create operators against tools from the literature.

Literature Review

Literature Review

1. RQ1: ¿What is current application of mutation testing in security?
2. RQ2: ¿Which are the challenges in security of RestFUL APIs?
3. RQ3: ¿Which are the testing security techniques in RestFUL APIs?
4. RQ4: ¿What are the most common security mistakes of the developers in the building of restful API?

Challenges

1. RESTFul APIs handle sensitive information that needs to be protected, software testing evaluates how they are handled, but because vulnerabilities are constantly being discovered, there is an opportunity for improvement in this area.
2. Mutation testing has proven to be a strategy for evaluating the security of applications, there has been a lot of work done related to specific applications in languages such as Java and Python, there is an opportunity to contribute to the development of RESTFul API.
3. Security is a challenge for software development today, and several recent studies have identified security gaps in many of them, which could be studied to provide a framework for the development of tools to assess data security and generate recommendations for improvement.

Work plan

Contribution selection

Working plan: Following the snowball methodology

- Review of vulnerabilities in RESTful APIs: Survey in the interception between mutation testing and security evaluation in Restful-API.
- Description of the mutation operators
- Prototype implementation and testing

Total: 3 years.

References I

[1] J. M. Hanks, “Testing cobol programs by mutation,” Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Georgia, 1980.

[2] Y. Jia and M. Harman, “An analysis and survey of the development of mutation testing,” *IEEE Transactions on Software Engineering*, vol. 37, no. 5, pp. 649–678, 2011, ISSN: 00985589. DOI: 10.1109/TSE.2010.62.

[3] A. J. Offutt and R. H. Untch, “Mutation 2000: Uniting the Orthogonal,” *Mutation Testing for the New Century*, pp. 34–44, 2001. DOI: 10.1007/978-1-4757-5939-6_7.

[4] A. T. Acree, “On mutation,” Ph.D. dissertation, Department of Computer Science, USA, 1980.

[5] L. J. Morell, “A theory of error-based testing,” Ph.D. dissertation, Department of Computer Science, USA, 1983.

[6] R. Geist, A. J. Offutt, and F. C. Harris, “Estimation and enhancement of real-time software reliability through mutation analysis,” *IEEE Transactions on Computers*, vol. 41, no. 5, pp. 550–558, 1992.

References II

[7] R. Lipton, “Fault diagnosis of computer programs. student report,” *Carnegie Mellon University*, vol. 2, p. 2, 1971.

[8] R. A. DeMilli and A. J. Offutt, “Constraint-based automatic test data generation,” *IEEE Transactions on Software Engineering*, vol. 17, no. 9, pp. 900–910, 1991.

[9] R. A. DeMillo, E. W. Krauser, and A. P. Mathur, “Compiler-integrated program mutation,” in *[1991] Proceedings The Fifteenth Annual International Computer Software Applications Conference*, 1991, pp. 351–356.

[10] J. C. Maldonado, M. E. Delamaro, S. C. P. F. Fabbri, *et al.*, “Proteum: A family of tools to support specification and program testing based on mutation,” in *Mutation Testing for the New Century*, Springer US, 2001, pp. 113–116. DOI: 10.1007/978-1-4757-5939-6_19. [Online]. Available: https://doi.org/10.1007/978-1-4757-5939-6_19.

[11] A. J. Offutt and R. A. Demillo, “Automatic test data generation,” Ph.D. dissertation, Department of Computer Science, USA, 1988.

References III

[12] A. J. Offutt and K. N. King, "A fortran 77 interpreter for mutation analysis," in *Papers of the Symposium on Interpreters and interpretive techniques 87*, ACM Press, 1987. DOI: 10.1145/29650.29669. [Online]. Available: https://doi.org/10.1145/29650.29669.

[13] Y. Wang, J. Yao, and X. Yu, "Information security protection in software testing," in *2018 14th International Conference on Computational Intelligence and Security (CIS)*, 2018, pp. 449–452.

[14] T. Nelson, N. Danas, T. Giannakopoulos, and S. Krishnamurthi, "Synthesizing mutable configurations: Setting up systems for success," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)*, 2019, pp. 81–85.

[15] I. Kravets and D. Tsafrir, "Feasibility of mutable replay for automated regression testing of security updates," in *Runtime Environments/Systems, Layering, & Virtualized Environments workshop (RESoLVE)*, 2012.

References IV

[16] T. Mouelhi, Y. L. Traon, and B. Baudry, “Mutation analysis for security tests qualification,” in *Testing: Academic and Industrial Conference Practice and Research Techniques - MUTATION (TAICPART-MUTATION 2007)*, IEEE, Sep. 2007. DOI: 10.1109/taic.part.2007.21. [Online]. Available: https://doi.org/10.1109/taic.part.2007.21.

[17] L. Thomas, W. Xu, and D. Xu, “Mutation analysis of magento for evaluating threat model-based security testing,” in *2011 IEEE 35th Annual Computer Software and Applications Conference Workshops*, IEEE, Jul. 2011. DOI: 10.1109/compsacw.2011.40. [Online]. Available: https://doi.org/10.1109/compsacw.2011.40.

[18] T. Loise, X. Devroey, G. Perrouin, M. Papadakis, and P. Heymans, “Towards security-aware mutation testing,” in *2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, IEEE, Mar. 2017. DOI: 10.1109/icstw.2017.24. [Online]. Available: https://doi.org/10.1109/icstw.2017.24.

References V

[19] Z. Zhang, H. Zhu, M. Wen, Y. Tao, Y. Liu, and Y. Xiong, "How do python framework APIs evolve? an exploratory study," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, IEEE, Feb. 2020. DOI: 10.1109/saner48275.2020.9054800. [Online]. Available: https://doi.org/10.1109/saner48275.2020.9054800.

[20] M. Kechagia, X. Devroey, A. Panichella, G. Gousios, and A. van Deursen, "Effective and efficient API misuse detection via exception propagation and search-based testing," in *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis - ISSTA 2019*, ACM Press, 2019. DOI: 10.1145/3293882.3330552. [Online]. Available: https://doi.org/10.1145/3293882.3330552.

[21] O. Ozdemir, T. S. Ingec, T. Erdinc, A. Roy, and M. Durran, *Upgrade verification tool*, Mar. 2019.

References VI

[22] P. G. Frankl and S. N. Weiss, "An experimental comparison of the effectiveness of the all-uses and all-edges adequacy criteria," in *Proceedings of the symposium on Testing, analysis, and verification - TAV4*, ACM Press, 1991. DOI: 10.1145/120807.120821. [Online]. Available: https://doi.org/10.1145/120807.120821.

[23] P. Frankl and S. Weiss, "An experimental comparison of the effectiveness of branch testing and data flow testing," *IEEE Transactions on Software Engineering*, vol. 19, no. 8, pp. 774–787, 1993. DOI: 10.1109/32.238581. [Online]. Available: https://doi.org/10.1109/32.238581.

[24] P. Anbalagan and T. Xie, "Automated generation of pointcut mutants for testing pointcuts in AspectJ programs," in *2008 19th International Symposium on Software Reliability Engineering (ISSRE)*, IEEE, Nov. 2008. DOI: 10.1109/issre.2008.58. [Online]. Available: https://doi.org/10.1109/issre.2008.58.

References VII

[25] P. Delgado-Pérez, S. Segura, and I. Medina-Bulo, "Assessment of c object-oriented mutation operators: A selective mutation approach," *Software Testing, Verification and Reliability*, vol. 27, no. 4-5, e1630, Mar. 2017. DOI: 10.1002/stvr.1630. [Online]. Available: https://doi.org/10.1002/stvr.1630.

[26] A. Derezinska and K. Kowalski, "Object-oriented mutation applied in common intermediate language programs originated from c," in *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, IEEE, Mar. 2011. DOI: 10.1109/icstw.2011.54. [Online]. Available: https://doi.org/10.1109/icstw.2011.54.

[27] F. C. Ferrari, J. Maldonado, and A. Rashid, "Mutation testing for aspect-oriented programs," in *2008 International Conference on Software Testing, Verification, and Validation*, IEEE, Apr. 2008. DOI: 10.1109/icst.2008.37. [Online]. Available: https://doi.org/10.1109/icst.2008.37.

References VIII

[28] A. Estero-Botaro, F. Palomo-Lozano, and I. Medina-Bulo, "Quantitative evaluation of mutation operators for WS-BPEL compositions," in *2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, IEEE, Apr. 2010. DOI: 10.1109/icstw.2010.36. [Online]. Available: https://doi.org/10.1109/icstw.2010.36.

[29] J. Boubeta-Puig, I. Medina-Bulo, and A. García-Domínguez, "Analogies and differences between mutation operators for WS-BPEL 2.0 and other languages," in *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, IEEE, Mar. 2011. DOI: 10.1109/icstw.2011.52. [Online]. Available: https://doi.org/10.1109/icstw.2011.52.

References IX

[30] A. J. Offutt, J. Pan, K. Tewary, and T. Zhang, "An experimental evaluation of data flow and mutation testing," *Software: Practice and Experience*, vol. 26, no. 2, pp. 165–176, Feb. 1996. DOI: 10.1002/(sici)1097-024x(199602)26:2<165::aid-spe5>3.0.co;2-k. [Online]. Available: https://doi.org/10.1002/(sici)1097-024x(199602)26:2%3C165::aid-spe5%3E3.0.co;2-k.

[31] P. G. Frankl, S. N. Weiss, and C. Hu, "All-uses vs mutation testing: An experimental comparison of effectiveness," *Journal of Systems and Software*, vol. 38, no. 3, pp. 235–253, Sep. 1997. DOI: 10.1016/s0164-1212(96)00154-9. [Online]. Available: https://doi.org/10.1016/s0164-1212(96)00154-9.

[32] P. G. Frankl and O. Iakounenko, "Further empirical studies of test effectiveness," in *Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering - SIGSOFT 98/FSE-6*, ACM Press, 1998. DOI: 10.1145/288195.288298. [Online]. Available: https://doi.org/10.1145/288195.288298.

References X

[33] S. Kakarla, S. Momotaz, and A. S. Namin, "An evaluation of mutation and data-flow testing: A meta-analysis," in *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, IEEE, Mar. 2011. DOI: 10.1109/icstw.2011.51. [Online]. Available: https://doi.org/10.1109/icstw.2011.51.

[34] M. Papadakis, M. Kintis, J. Zhang, Y. Jia, Y. L. Traon, and M. Harman, *Mutation Testing Advances: An Analysis and Survey*, 1st ed. Elsevier Inc., 2019, vol. 112, pp. 275–378, ISBN: 9780128151211. DOI: 10.1016/bs.adcom.2018.03.015. [Online]. Available: http://dx.doi.org/10.1016/bs.adcom.2018.03.015.

[35] I. IEC, "9126-1 (2001). software engineering product quality-part 1: Quality model," *International Organization for Standardization*, p. 16, 2001.

[36] ISO/IEC, *Iso/iec 25010: 2011 systems and software engineering–systems and software quality requirements and evaluation (square)–system and software quality models*, 2011.

References XI

[37] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, “State of the art: Automated black-box web application vulnerability testing,” in *2010 IEEE Symposium on Security and Privacy*, IEEE, 2010. DOI: 10.1109/sp.2010.27. [Online]. Available: https://doi.org/10.1109/sp.2010.27.

[38] C. Mitre, *Common vulnerabilities and exposures*, 2005.

[39] G. Tassey, *Nist: The economic impacts of inadequate infrastructure for software testing*, 2002.

[40] M. V. HAYDEN, *Committee on national security systems national manager*, 2004.

[41] G. Tian-yang, S. Yin-Sheng, and F. You-yuan, “Research on software security testing,” *World Academy of science, engineering and Technology*, vol. 70, pp. 647–651, 2010.

[42] Y.-S. Ma, J. Offutt, and Y.-R. Kwon, “MuJava,” in *Proceeding of the 28th international conference on Software engineering - ICSE 06*, ACM Press, 2006. DOI: 10.1145/1134285.1134425. [Online]. Available: https://doi.org/10.1145/1134285.1134425.

References XII

[43] R. Just, "The major mutation framework: Efficient and scalable mutation analysis for java," in *Proceedings of the 2014 International Symposium on Software Testing and Analysis ISSTA 2014*, ACM Press, 2014. DOI: 10.1145/2610384.2628053. [Online]. Available: https://doi.org/10.1145/2610384.2628053.

[44] H. Coles, T. Laurent, C. Henard, M. Papadakis, and A. Ventresque, "PIT: A practical mutation testing tool for java (demo)," in *Proceedings of the 25th International Symposium on Software Testing and Analysis - ISSTA 2016*, ACM Press, 2016. DOI: 10.1145/2931037.2948707. [Online]. Available: https://doi.org/10.1145/2931037.2948707.

[45] D. Rodriguez-Baquero and M. Linares-Vásquez, "Mutode: Generic JavaScript and node.js mutation testing tool," in *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis - ISSTA 2018*, ACM Press, 2018. DOI: 10.1145/3213846.3229504. [Online]. Available: https://doi.org/10.1145/3213846.3229504.

References XIII

[46] M. Kusano and Chao Wang, “Ccmutator: A mutation generator for concurrency constructs in multithreaded c/c++ applications,” in *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2013, pp. 722–725.

[47] M. Wen, Y. Liu, R. Wu, X. Xie, S.-C. Cheung, and Z. Su, “Exposing library API misuses via mutation analysis,” in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, IEEE, May 2019. DOI: 10.1109/icse.2019.00093. [Online]. Available: https://doi.org/10.1109/icse.2019.00093.

[48] J. Viega, J. Bloch, Y. Kohno, and G. McGraw, “ITS4: A static vulnerability scanner for c and c++ code,” in *Proceedings 16th Annual Computer Security Applications Conference (ACSAC00)*, IEEE Comput. Soc, 2000. DOI: 10.1109/acsac.2000.898880. [Online]. Available: https://doi.org/10.1109/acsac.2000.898880.

References XIV

[49] S. Mirshokraie, A. Mesbah, and K. Pattabiraman, "JSEFT: Automated javascript unit test generation," in *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, IEEE, Apr. 2015. DOI: 10.1109/icst.2015.7102595. [Online]. Available: https://doi.org/10.1109/icst.2015.7102595.

[50] R. Scandariato, J. Walden, and W. Joosen, "Static analysis versus penetration testing: A controlled experiment," in *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*, IEEE, Nov. 2013. DOI: 10.1109/issre.2013.6698898. [Online]. Available: https://doi.org/10.1109/issre.2013.6698898.

[51] F. Tambon, F. Khomh, and G. Antoniol, "A probabilistic framework for mutation testing in deep neural networks," *Information and Software Technology*, vol. 155, p. 107 129, Mar. 2023. DOI: 10.1016/j.infsof.2022.107129. [Online]. Available: https://doi.org/10.1016/j.infsof.2022.107129.

References XV

[52] M. Ojdanic, A. Garg, A. Khanfir, R. Degiovanni, M. Papadakis, and Y. L. Traon, "Syntactic versus semantic similarity of artificial and real faults in mutation testing studies," *IEEE Transactions on Software Engineering*, vol. 49, no. 7, pp. 3922–3938, Jul. 2023. DOI: 10.1109/tse.2023.3277564. [Online]. Available: https://doi.org/10.1109/tse.2023.3277564.

[53] R. Pitts, "Mutant selection strategies in mutation testing," in *2023 International Conference on Code Quality (ICCQ)*, IEEE, Apr. 2023. DOI: 10.1109/iccq57276.2023.10114663. [Online]. Available: https://doi.org/10.1109/iccq57276.2023.10114663.

[54] M. R. Naeem, T. Lin, H. Naeem, F. Ullah, and S. Saeed, "Scalable mutation testing using predictive analysis of deep learning model," *IEEE Access*, vol. 7, pp. 158 264–158 283, 2019. DOI: 10.1109/access.2019.2950171. [Online]. Available: https://doi.org/10.1109/access.2019.2950171.

References XVI

[55] D. Mao, L. Chen, and L. Zhang, "An extensive study on cross-project predictive mutation testing," in *2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST)*, IEEE, Apr. 2019. DOI: 10.1109/icst.2019.00025. [Online]. Available: https://doi.org/10.1109/icst.2019.00025.

[56] Z. Tian, J. Chen, Q. Zhu, J. Yang, and L. Zhang, "Learning to construct better mutation faults," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ACM, Oct. 2022. DOI: 10.1145/3551349.3556949. [Online]. Available: https://doi.org/10.1145/3551349.3556949.

[57] C. Noy, "Sampling knowledge: The hermeneutics of snowball sampling in qualitative research," *International Journal of Social Research Methodology*, vol. 11, no. 4, pp. 327–344, Oct. 2008. DOI: 10.1080/13645570701401305. [Online]. Available: https://doi.org/10.1080/13645570701401305.

References XVII

[58] M. Büchler, "Security testing with fault-models and properties," in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013, pp. 501–502. DOI: 10.1109/ICST.2013.74.

[59] A. Arcuri, "Restful api automated test case generation with evomaster," *ACM Transactions on Software Engineering and Methodology*, vol. 28, no. 1, Jan. 2019, ISSN: 1049-331X. DOI: 10.1145/3293455. [Online]. Available: https://doi.org/10.1145/3293455.

[60] S. Segura, J. A. Parejo, J. Troya, and A. Ruiz-Cortés, "Metamorphic testing of restful web apis," in *Proceedings of the 40th International Conference on Software Engineering*, ser. ICSE '18, Gothenburg, Sweden: Association for Computing Machinery, 2018, p. 882, ISBN: 9781450356381. DOI: 10.1145/3180155.3182528. [Online]. Available: https://doi.org/10.1145/3180155.3182528.

References XVIII

[61] A. Arcuri, "Restful api automated test case generation," in *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2017, pp. 9–20. DOI: 10.1109/QRS.2017.11.

[62] H. Wu, L. Xu, X. Niu, and C. Nie, "Combinatorial testing of restful apis," in *Proceedings of the 44th International Conference on Software Engineering*, ser. ICSE '22, Pittsburgh, Pennsylvania: Association for Computing Machinery, 2022, pp. 426–437, ISBN: 9781450392211. DOI: 10.1145/3510003.3510151. [Online]. Available: https://doi.org/10.1145/3510003.3510151.

[63] E. Viglianisi, M. Dallago, and M. Ceccato, "Resttestgen: Automated black-box testing of restful apis," in *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, 2020, pp. 142–152. DOI: 10.1109/ICST46399.2020.00024.

References XIX

[64] P. Godefroid, D. Lehmann, and M. Polishchuk, “Differential regression testing for REST APIs,” in *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ACM, Jul. 2020. DOI: 10.1145/3395363.3397374. [Online]. Available: https://doi.org/10.1145/3395363.3397374.

[65] A. Arcuri, “Automated black- and white-box testing of restful apis with evomaster,” *IEEE Software*, vol. 38, no. 3, pp. 72–78, 2021. DOI: 10.1109/MS.2020.3013820.

[66] S. Karlsson, A. Čaušević, and D. Sundmark, “Quickrest: Property-based test generation of openapi-described restful apis,” in *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, 2020, pp. 131–141. DOI: 10.1109/ICST46399.2020.00023.

[67] B. De, *API Testing Strategy*. Apress, 2017, pp. 153–164. DOI: 10.1007/978-1-4842-1305-6_9. [Online]. Available: https://doi.org/10.1007/978-1-4842-1305-6_9.

References XX

[68] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API testing methodologies: Rationale, challenges, and solution directions," *Applied Sciences*, vol. 12, no. 9, p. 4369, Apr. 2022. DOI: 10.3390/app12094369. [Online]. Available: https://doi.org/10.3390/app12094369.

[69] Sattam J Alharbi and T. Moulahi, "Api security testing: The challenges of security testing for restful apis," *International Journal of Innovative Science and Research Technology*, 2023. DOI: 10.5281/ZENODO.7988409. [Online]. Available: https://zenodo.org/record/7988409.

[70] H. Riggs, S. Tufail, I. Parvez, *et al.*, "Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure," *Sensors*, vol. 23, no. 8, p. 4060, Apr. 2023. DOI: 10.3390/s23084060. [Online]. Available: https://doi.org/10.3390/s23084060.

[71] P. Siriwardena, *Advanced API Security*. Apress, 2020. DOI: 10.1007/978-1-4842-2050-4. [Online]. Available: https://doi.org/10.1007/978-1-4842-2050-4.

References XXI

[72] C. Cheh and B. Chen, "Analyzing openapi specifications for security design issues," in *2021 IEEE Secure Development Conference (SecDev)*, 2021, pp. 15–22. DOI: 10.1109/SecDev51306.2021.00019.

[73] R. Sun, Q. Wang, and L. Guo, "Research towards key issues of api security," in *Cyber Security*, W. Lu, Y. Zhang, W. Wen, H. Yan, and C. Li, Eds., Singapore: Springer Nature Singapore, 2022, pp. 179–192.

[74] A. Golmohammadi, M. Zhang, and A. Arcuri, "Testing restful apis: A survey," *ACM Trans. Softw. Eng. Methodol.*, Aug. 2023, ISSN: 1049-331X. DOI: 10.1145/3617175. [Online]. Available: https://doi.org/10.1145/3617175.

[75] L. Williams, E. M. Maximilien, and M. Vouk, "Test-driven development as a defect-reduction practice," in *14th International Symposium on Software Reliability Engineering, 2003. ISSRE 2003.*, IEEE, 2003, pp. 34–45.

References XXII

[76] M. Idris, I. Syarif, and I. Winarno, "Development of vulnerable web application based on owasp api security risks," in *2021 International Electronics Symposium (IES)*, 2021, pp. 190–194. DOI: 10.1109/IES53407.2021.9593934.

[77] M. Idris, I. Syarif, and I. Winarno, "Web application security education platform based on owasp api security project," *EMITTER International Journal of Engineering Technology*, vol. 10, no. 2, pp. 246–261, Dec. 2022. DOI: 10.24003/emitter.v10i2.705. [Online]. Available: https://emitter.pens.ac.id/index.php/emitter/article/view/705.

[78] C. Cheh and B. Chen, "Analyzing openapi specifications for security design issues," in *2021 IEEE Secure Development Conference (SecDev)*, 2021, pp. 15–22. DOI: 10.1109/SecDev51306.2021.00019.

References XXIII

[79] B. Modi, U. Chourasia, and R. Pandey, "Design and implementation of RESTFUL API based model for vulnerability detection and mitigation," *IOP Conference Series: Materials Science and Engineering*, vol. 1228, no. 1, p. 012 010, Mar. 2022. DOI: 10.1088/1757-899x/1228/1/012010. [Online]. Available: https://doi.org/10.1088/1757-899x/1228/1/012010.

[80] A. Munsch and P. Munsch, "The future of API (application programming interface) security: The adoption of APIs for digital communications and the implications for cyber security vulnerabilities," *Journal of International Technology and Information Management*, vol. 29, no. 3, pp. 24–45, Jan. 2021. DOI: 10.58729/1941-6679.1454. [Online]. Available: https://doi.org/10.58729/1941-6679.1454.

[81] Sattam J Alharbi and T. Moulahi, "Api security testing: The challenges of security testing for restful apis," *International Journal of Innovative Research in Science Engineering and Technology*, 2023. DOI: 10.5281/ZENODO.7988410. [Online]. Available: https://zenodo.org/record/7988410.

References XXIV

[82] L. Zhong, *A survey of prevent and detect access control vulnerabilities*, 2023. DOI: 10.48550/ARXIV.2304.10600. [Online]. Available: https://arxiv.org/abs/2304.10600.

[83] D. Corradini, M. Pasqua, and M. Ceccato, "Automated black-box testing of mass assignment vulnerabilities in restful apis," in *International Conference on Software Engineering (ICSE 2023)*, arXiv, 2023. DOI: 10.48550/ARXIV.2301.01261. [Online]. Available: https://arxiv.org/abs/2301.01261.

[84] N. Auricchio, A. Cappuccio, F. Caturano, G. Perrone, and S. P. Romano, "An automated approach to web offensive security," *Computer Communications*, vol. 195, pp. 248–261, Nov. 2022. DOI: 10.1016/j.comcom.2022.08.018. [Online]. Available: https://doi.org/10.1016/j.comcom.2022.08.018.

References XXV

[85] B. O. EMEKA, S. HIDAKA, and S. LIU, "A practical model driven approach for designing security aware RESTful web APIs using SOFL," *IEICE Transactions on Information and Systems*, vol. E106.D, no. 5, pp. 986–1000, May 2023. DOI: 10.1587/transinf.2022edp7194. [Online]. Available: https://doi.org/10.1587/transinf.2022edp7194.

[86] M. C. Ghanem and T. M. Chen, "Reinforcement learning for efficient network penetration testing," *Information*, vol. 11, no. 1, p. 6, Dec. 2019. DOI: 10.3390/info11010006. [Online]. Available: https://doi.org/10.3390/info11010006.

[87] Z. Hu, R. Beuran, and Y. Tan, "Automated penetration testing using deep reinforcement learning," in *2020 IEEE European Symposium on Security and Privacy Workshops*, IEEE, Sep. 2020. DOI: 10.1109/eurospw51379.2020.00010. [Online]. Available: https://doi.org/10.1109/eurospw51379.2020.00010.

References XXVI

[88] J. Schwartz and H. Kurniawati, *Autonomous penetration testing using reinforcement learning*, 2019. DOI: 10.48550/ARXIV.1905.05965. [Online]. Available: https://arxiv.org/abs/1905.05965.