Characterizing and understanding security risks through Security-Aware Mutation Testing of security configuration in RESTful APIs

Report 2: Review of comments of the last meeting and progress

Carlos Andres Delgado Saavedra
carlos.andres.delgado@correounivalle.edu.co
Advisors:
Jesus A. Aranda, Universidad del Valle
Gills Perrouin, University of Namur, James Ortiz, Universite of Namur

December 3, 2024

1 Questions about the work

1. What are the existing mutation operators for security testing of restful APIs?
2. How effective are these mutation operators in detecting security vulnerabilities?
3. What are the limitations of the current mutation operators?
4. How can we detect these vulnerabilities currently?
5. What are the elements defined in restful services?
6. How are these vulnerabilities handled in the software development process?
7. What strategies are there for improving security practices in restful APIs?
8. What elements define common security misconfigurations in restful APIs?
9. How can security-aware mutation operators be designed to improve the coverage of security testing in the configuration of security policies in restful APIs?
10. What is the planning for the next two years for completing the tasks?

2 Answers to questions

2.1 Existing Mutation Operators for Security Testing of RESTful APIs

The field of security testing for RESTful APIs has seen several advancements with respect to mutation operators designed to uncover vulnerabilities. These mutation operators are crucial for generating test cases that can effectively assess the security posture of APIs. Below, we discuss notable mutation operators and relevant research:

1. **Real Coded Genetic Algorithm (RGA-MS):** Mishra et al. propose the RGA-MS, a novel algorithm aimed at

maximizing mutation coverage. This algorithm applies path coverage-based testing as a precursor to mutation testing, achieving high mutation scores by optimizing test data [1].

2. **Adaptive Hypermutation:** Zhang and Arcuri introduced an adaptive weight-based hypermutation operator specifically designed for REST API testing. This operator, integrated into the EvoMaster tool, adjusts mutation strategies based on gene fitness impact, enhancing coverage and enabling better test case generation [2].

3. **Pythia's Learning-Based Mutation:** Atlidakis et al. introduced Pythia, which utilizes learning-based mutation strategies for REST API fuzzing. By injecting slight deviations from learned usage patterns, Pythia creates valid test cases while prioritizing those likely to expose new bugs [3].

4. **Mutation Operators with Data Type Constraints:** Liu and Chen proposed seven mutation operators targeting input data type constraints for testing REST APIs, such as Constraint Value Replacement (CVR) and Logical Connectives Replacement (LCR). These operators optimize test data generation and improve coverage [4].

5. **Mutation over Execution Traces:** Paiva et al. focus on extending test suites by mutating test cases based on user execution traces. This involves operators that mimic potential real failures, enhancing the richness of the test cases [5].

6. **Security-Oriented Mutation Operators:** Salva and Sue present an approach with 17 specialized mutation operators aimed at enhancing security testing for RESTful APIs. These operators generate security-specific test cases, addressing vulnerabilities effectively [6].

The continued evolution and application of these mutation operators illustrate significant advancements in efficiently testing Restful APIs by improving test coverage and detecting security weaknesses.

References

[1] Mishra, D. B., Acharya, B., Rath, D., Gerogiannis, V. C., & Kanavos, A. (2022). A Novel Real Coded Genetic Algorithm for Software Mutation Testing. *Symmetry, 14*(8), 1525. https://doi.org/10.3390/sym14081525

[2] Zhang, M., & Arcuri, A. (2021).Adaptive Hypermutation for Search-Based System Test Generation: A Study on REST APIs with EvoMaster. *ACM Trans. Softw. Eng. Methodol., 31*(1), 52 pages. https://doi.org/10.1145/3464940

[3] Atlidakis, V., Geambasu, R., Godefroid, P., Polishchuk, M., & Ray, B. (2020). Pythia Grammar-based fuzzing of rest APIs with coverage-guided feedback and learning-based mutations. *arXiv preprint arXiv:2005.11498*.

[4] Liu, J., & Chen, W. (2017). Optimized Test Data Generation for RESTful Web Service. *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*. doi:10.1109/apsec.2017.85

[5] Paiva, A. C. R., Restivo, A., & Almeida, S. (2020). Test case generation based on mutations over user execution traces. *Software Quality Journal*. doi:10.1007/s11219-020-09503-4

[6] Salva, S., & Sue, J. Security Testing of RESTful APIs With Test Case Mutation

(n.d.). *LIMOS - UMR CNRS 6158, Clermont Auvergne University, UCA, France.*