Characterizing and understanding security risks through Fuzzing Secure-Aware Mutation Testing of RESTFul-API services

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

Universidad del Valle

UNIVERSITÉ DE NAMUR

AVISPA
Ambientes VISuales de Programación Aplicativa

Colombia

February 2nd, 2024

Overview

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 design fuzzed secure-aware mutation operators in the coverage of the vulnerabilities in the configuration of security policies in RESTFul APIs?

Objectives

Develop a collection of security-aware mutation operators designed for safeguarding the configuration of security policies within RESTFul API services.

Specific

Specific objective	**Expected result**
1. Identification of the elements of the security policies in API-RESTFul services	Characteristics of the security policies in API-Restful services
2. Describe a set of fuzzed security-aware mutation operators for testing of security policies in API-RESTFul services	Description of the mutation operators according to the elements of security policies in API-Restful services
3. Develop the set of security-aware mutation operators for testing in Django Rest and Flask Frameworks in Python	Source code of the secure-aware mutation operators
4. Evaluate the proposed security-aware mutation operators in REST-Ful API services	Report about the performance of the created operators against tools from the literature.

Table 1: Specific objectives and expected results

Literature Review

Strategy

1. Questions about the current state of art in the configuration security policies of RESTFul APIs.
2. Window of time from 2000 to 2024. https://doi.org/10.1515/itit-2013-1035
3. Emphasis in the last 5 years. https://doi.org/10.1145/3617175, https://journal.ijresm.com/index.php/ijresm/article/view/970 the rise of the RESTFul APIs.

Research questions

1. RQ1: What are the elements of the security configuration policies in the RESTFul API Services?
2. RQ2: What are the current challenges about the security policies of RESTFul API Services?
3. RQ3: What are the most common configuration security mistakes of the developers in the building of RESTFul API Services?
4. RQ4: What are the current testing techniques and tools for the testing of configuration policies of RESTFul API Services based on Python?
5. RQ5: What experiences have been reported in the literature about the use of mutation testing for the security testing of RESTFul API Services?

RQ1: Elements of security configuration policies

1. Authentication: Methods for the identification of the user.
2. Authorization: Methods for the access control.
3. Encryption: Protocol SSL/TLS.
4. Data masking: Hide sensitive data in logs and responses.
5. Input validation and sanitization: Prevent injection attacks (SQL, XSS).
6. Thottling: Number of requests per time.
7. API Keys: Each user with their own key.
8. Login level: Detailed and security monitoring.

RQ1: References I

Kellezi, D., Boegelund, C., & Meng, W. (2019). Towards secure open banking architecture: An evaluation with owasp. In *Lecture notes in computer science* (pp. 185–198). Springer International Publishing. https://doi.org/10.1007/978-3-030-36938-5_11

Luo, Y., Puyang, T., Luo, W., Shen, Q., Ruan, A., & Wu, Z. (2016). Multipol: Towards a multi-policy authorization framework for restful interfaces in the cloud. In *Lecture notes in computer science* (pp. 214–226). Springer International Publishing. https://doi.org/10.1007/978-3-319-50011-9_17

Madden, N. (2021, February). *API security in action*. Manning Publications.

Siriwardena, P. (2020). *Advanced api security: Oauth 2.0 and beyond*. Apress. https://doi.org/10.1007/978-1-4842-2050-4

Subramanian, H., & Raj, P. (2019, January). *Hands-On RESTful API design patterns and best practices*. Packt Publishing.

RQ2: Current challenges

1. Keep the data integrity in RESTFul API Services is a challenge that changes every day.
2. Several recent studies have identified security gaps in many of them.
3. One of the most problems about software vulnerabilities is the configuration security policies of RESTFul APIs
4. Testing methods and tools are not enough to cover all the vulnerabilities.

RQ2: References

Bakhtin, A., Al Maruf, A., Cerny, T., & Taibi, D. (2022). Survey on tools and techniques detecting microservice api patterns. *2022 IEEE International Conference on Services Computing (SCC)*. https://doi.org/10.1109/scc55611.2022.00018

Idris, M., Syarif, I., & Winarno, I. (2022). Web application security education platform based on OWASP API security project. *EMIT. Int. J. Eng. Technol.*, 246–261.

Khoda Parast, F., Sindhav, C., Nikam, S., Izadi Yekta, H., Kent, K. B., & Hakak, S. (2022). Cloud computing security: A survey of service-based models. *Computers and Security*, *114*, 102580. https://doi.org/10.1016/j.cose.2021.102580

Martin-Lopez, A., Segura, S., & Ruiz-Cortés, A. (2022). Online testing of restful apis: Promises and challenges. https://doi.org/10.5281/ZENODO.6941292

Zhang, M., & Arcuri, A. (2023). Open problems in fuzzing restful apis: A comparison of tools. *ACM Transactions on Software Engineering and Methodology*, *32*(6), 1–45. https://doi.org/10.1145/3597205

Universidad del Valle

RQ3: Common configuration mistakes

1. Lack of input validation.
2. Insecure deserialization.
3. Lack of proper authentication and authorization.
4. Insecure direct object references.
5. Lack of proper logging and monitoring.
6. Insecure communication with untrusted components.

RQ3: References

Hussain, F., Hussain, R., Noye, B., & Sharieh, S. (2020). Enterprise api security and gdpr compliance: Design and implementation perspective. *IT Professional*, *22*(5), 81–89. https://doi.org/10.1109/mitp.2020.2973852

Jin, Z., Xing, L., Fang, Y., Jia, Y., Yuan, B., & Liu, Q. (2022). P-verifier: Understanding and mitigating security risks in cloud-based iot access policies. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. https://doi.org/10.1145/3548606.3560680

Votipka, D., Fulton, K. R., Parker, J., Hou, M., Mazurek, M. L., & Hicks, M. (2020). Understanding security mistakes developers make: Qualitative analysis from build it, break it, fix it. *29th USENIX Security Symposium (USENIX Security 20)*, 109–126. https://www.usenix.org/conference/usenixsecurity20/presentation/votipka-understanding

Universidad del Valle

RQ4: Testing techniques and tools

1. OWASP ZAP: Penetration testing.
2. Postman: API testing.
3. Burp Suite: Penetration testing.
4. Nessus: Vulnerability assessment.
5. Nmap: Network scanning.
6. Metasploit: Penetration testing.

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

Bakhtin, A., Al Maruf, A., Cerny, T., & Taibi, D. (2022). Survey on tools and techniques detecting microservice api patterns. *2022 IEEE International Conference on Services Computing (SCC)*. https://doi.org/10.1109/scc55611.2022.00018

Hussain, F., Hussain, R., Noye, B., & Sharieh, S. (2020). Enterprise api security and gdpr compliance: Design and implementation perspective. *IT Professional*, *22*(5), 81–89. https://doi.org/10.1109/mitp.2020.2973852

Idris, M., Syarif, I., & Winarno, I. (2022). Web application security education platform based on OWASP API security project. *EMIT. Int. J. Eng. Technol.*, 246–261.

Jin, Z., Xing, L., Fang, Y., Jia, Y., Yuan, B., & Liu, Q. (2022). P-verifier: Understanding and mitigating security risks in cloud-based iot access policies. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. https://doi.org/10.1145/3548606.3560680

Universidad del Valle

References II

Kellezi, D., Boegelund, C., & Meng, W. (2019). Towards secure open banking architecture: An evaluation with owasp. In *Lecture notes in computer science* (pp. 185–198). Springer International Publishing. https://doi.org/10.1007/978-3-030-36938-5_11

Khoda Parast, F., Sindhav, C., Nikam, S., Izadi Yekta, H., Kent, K. B., & Hakak, S. (2022). Cloud computing security: A survey of service-based models. *Computers and Security*, *114*, 102580. https://doi.org/10.1016/j.cose.2021.102580

Luo, Y., Puyang, T., Luo, W., Shen, Q., Ruan, A., & Wu, Z. (2016). Multipol: Towards a multi-policy authorization framework for restful interfaces in the cloud. In *Lecture notes in computer science* (pp. 214–226). Springer International Publishing. https://doi.org/10.1007/978-3-319-50011-9_17

Madden, N. (2021, February). *API security in action*. Manning Publications.

Martin-Lopez, A., Segura, S., & Ruiz-Cortés, A. (2022). Online testing of restful apis: Promises and challenges. https://doi.org/10.5281/ZENODO.6941292

Universidad del Valle

References III

Roth, S., Barron, T., Calzavara, S., Nikiforakis, N., & Stock, B. (2020). Complex security policy? a longitudinal analysis of deployed content security policies. *Proceedings 2020 Network and Distributed System Security Symposium*. https://doi.org/10.14722/ndss.2020.23046

Siriwardena, P. (2020). *Advanced api security: Oauth 2.0 and beyond*. Apress. https://doi.org/10.1007/978-1-4842-2050-4

Subramanian, H., & Raj, P. (2019, January). *Hands-On RESTful API design patterns and best practices*. Packt Publishing.

Votipka, D., Fulton, K. R., Parker, J., Hou, M., Mazurek, M. L., & Hicks, M. (2020). Understanding security mistakes developers make: Qualitative analysis from build it, break it, fix it. *29th USENIX Security Symposium (USENIX Security 20)*, 109–126. https://www.usenix.org/conference/usenixsecurity20/presentation/votipka-understanding

Universidad del Valle

References IV

Zhang, M., & Arcuri, A. (2023). Open problems in fuzzing restful apis: A comparison of tools. *ACM Transactions on Software Engineering and Methodology*, *32*(6), 1–45. https://doi.org/10.1145/3597205

heading=bibintoc