

THESIS

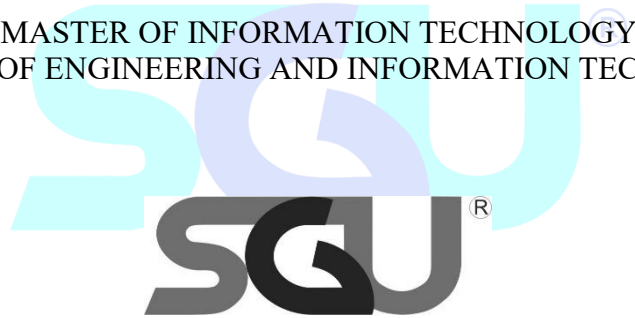
CONTRACT TESTING: A FRAMEWORK FOR SECURITY EVALUATION IN
gRPC

By

MUHAMAD ZAENUL HASAN BASRI

MASTER'S DEGREE
in

MASTER OF INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY



SWISS GERMAN UNIVERSITY
The Prominence Tower
Jalan Jalur Sutera Barat No. 15, Alam Sutera
Tangerang, Banten 15143 - Indonesia

February 2025

STATEMENT BY THE AUTHOR

I hereby declare that this thesis is my own work and to the best of my knowledge, this thesis does not contain any material that has been published or written by anyone else, nor any material that has been accepted for a degree or diploma at any educational institution, unless the source is stated in the thesis.

MUHAMAD ZAENUL HASAN BASRI

Student

Date

Approved By

Dr. Ir. Charles Lim BSc., MSc.

Thesis Advisor

Date

Kalpin Erlangga Silaen, S.Si. M.Kom, Co-Advisor

Thesis Advisor

Date

Dr. Maulahikmah Galinium, S.Kom., M.Sc.

Dean of Faculty of Engineering and Information
Technology

Date

MUHAMAD ZAENUL HASAN BASRI

ABSTRACT

CONTRACT TESTING: A FRAMEWORK FOR SECURITY EVALUATION IN gRPC

By

MUHAMAD ZAENUL HASAN BASRI
Dr. Ir. Charles Lim BSc., MSc., Advisor
Kalpin Erlangga Silaen, S.Si. M.Kom, Co-Advisor

SWISS GERMAN UNIVERSITY

The growth of APIs, including SOAP, REST, and gRPC, has made security a critical priority, with incidents such as those in the 2023 Paloalto report highlighting the financial losses resulting from API breaches. While existing tools focus on REST APIs, gRPC remains underserved, requiring time-consuming manual testing. This research addresses this gap by proposing a security testing framework tailored to gRPC, integrating automated methods and DevSecOps can use to improve efficiency. gRPC, built on HTTP/2, uses a binary message format and client stubs generated from proto files, creating unique challenges for testing. By analyzing gRPC components and adapting common API security practices, the framework identifies vulnerabilities, streamlines testing, and reduces manual effort. It automates processes such as payload generation and stub generation, enabling faster and more reliable testing compared to traditional methods. Additionally, the framework can be combined with DevSecOps to enable continuous security validation, ensuring vulnerabilities are identified early in the development lifecycle. This approach not only improves testing efficiency by significantly reducing time, but also sets a benchmark for secure API development. Ultimately, this research serves as a reference for security professionals looking to improve gRPC API testing, offers practical solutions to address security gaps, and paves the way for further advancements in gRPC security practices.

Keywords: gRPC, framework, API, REST, testing



DEDICATION

This thesis is dedicated to everyone who has supported and inspired me. To my family, whose love and encouragement have been my foundation, and to someone who supported me before and my mentors, whose guidance has shaped my path.

To the professionals and researchers in cybersecurity, whose efforts protect our digital world and inspire this research.

Finally, to future cybersecurity experts—may this work encourage you to innovate, explore, and build a safer digital future.



ACKNOWLEDGEMENT

First of all, I would like to express my deepest gratitude to my supervisor, Dr. Ir. Charles Lim BSc., MSc., for his invaluable guidance, encouragement, and expertise throughout this journey. His support was crucial in the completion of this thesis.

I would also like to thank my co-supervisor, Kalpin Erlangga Silaen, S.Si., M.Kom., for his constructive feedback, insightful suggestions, and constant encouragement, which have greatly improved the quality of this work.

I would also like to express my sincere appreciation to my family and friends, who have been a source of unwavering support and motivation.

Lastly, I would like to thank everyone who contributed to this research directly or indirectly. Your help and encouragement are greatly appreciated.®



TABLE OF CONTENTS

	Page
STATEMENT BY THE AUTHOR.....	2
ABSTRACT.....	3
DEDICATION.....	5
ACKNOWLEDGEMENT.....	6
TABLE OF CONTENTS.....	7
LIST OF FIGURES.....	10
LIST OF TABLES.....	11
LIST OF EQUATION.....	12
CHAPTER 1 - INTRODUCTION.....	13
1.1. Background.....	13
1.2. Research Problem.....	16
1.3. Research Objective.....	17
1.4. Research Question.....	19
1.5. Hypothesis.....	19
1.6. Research Scope and Limitation.....	20
1.7. Significance of Study.....	21
1.8. Thesis Structure.....	22
CHAPTER 2 - LITERATURE REVIEW.....	23
2.1. Business Transformation.....	23
2.1.1. Business Challenges in API Transactions.....	23
2.1.2. Business Optimization in API-Driven Transactions.....	24
2.1.3. Common API Issue.....	24
2.1.4. Balancing Performance and Security.....	25
2.2. API Evolution.....	25
2.2.1. gRPC.....	26
2.2.1.1. Type of gRPC Service.....	29
2.2.1.2. Object Type In gRPC.....	31
2.2.1.3. Status Code In gRPC.....	32
2.2.2. REST.....	33
2.2.3. SOAP.....	34
2.2.4. GraphQL.....	34
2.2.5. Comparison of gRPC and Other API.....	34
2.2.5.1. gRPC API Criteria.....	36
2.3. API Security Testing.....	37
2.3.1. API Vulnerabilities.....	37

2.3.2. Black-box	40
2.3.3. Gray-box.....	40
2.3.4. CVSS	40
2.3.5. DevSecOps	42
2.3.6. Tools.....	43
2.4. Related Work and Gap.....	45
2.4.1. Previous Study Related Framework	45
2.4.2. Previous Study Related Method	51
2.4.3. Proposed Framework.....	56
CHAPTER 3 - RESEARCH METHODS.....	59
3.1. Research Methodology	59
3.2. Research Framework	59
3.2.1. Extraction of Payload	60
3.2.2. Code Generation.....	61
3.2.3. Test Case Creation.....	62
3.2.4. Execution.....	62
3.3. Scope of Test.....	64
3.4. Evaluation	64
3.5. Validation.....	65
CHAPTER 4 - RESULT AND DISCUSSION.....	66
4.1. Introduction.....	66
4.2. Security Testing Execution	66
4.2.1. Manual Security Testing.....	67
4.2.2. GSTF Security Testing	71
4.2.2.1. Extraction of Payload	72
4.2.2.2. Code Generation	72
4.2.2.3. Test Case Creation	72
4.2.2.4. Execution	72
4.2.3. Comparison Between Manual and GSTF testing Based on Execution Time.....	75
4.2.4. Comparison of GSTF Security Testing Results with Other Tools	77
4.3. Evaluation	78
4.4. Validation.....	79
CHAPTER 5 - CONCLUSION AND RECOMMENDATIONS.....	84
5.1. Conclusion	84
5.2. Recommendations.....	84
REFERENCES	86
APPENDIX.....	90
A1. GSTF Script	90
A2. GSTF Result.....	103

CURRICULUM VITAE..... 119



LIST OF FIGURES

Figures	Page
Figure 1.1 API Security Incident 2023 (Melinda Marks, 2023).....	13
Figure 1.2 Fishbone Diagram	16
Figure 2.1 gRPC API Architecture (Anthony Giretti, 2022)	27
Figure 2.2 Sample proto file (Anthony Giretti, 2022)	28
Figure 2.3 Unary type process (Anthony Giretti, 2022)	29
Figure 2.4 Server-streaming type process (Anthony Giretti, 2022).....	29
Figure 2.5 Client-streaming type process (Anthony Giretti, 2022)	30
Figure 2.6 Bidirectional streaming type process (Anthony Giretti, 2022)	30
Figure 2.7 REST API Architecture (Anthony Giretti, 2022; Bohdan Marii and Ivan Zholubak, 2022).....	33
Figure 2.8 Protoc command in python.....	43
Figure 2.9 gRPCurl create a request (Anthony Giretti, 2022)	44
Figure 2.10 API Message-Driven Regression Testing (MDRT) Framework Process (Emine Dumlu Demircio ̇glu and Oya Kalipsiz, 2022)	48
Figure 2.11 GSTF proposed framework	56
Figure 3.1 Research methodology	59
Figure 3.2 GSTF (gRPC Security Testing Framework)	60
Figure 3.3 Phase 1 - Extraction of payload.....	60
Figure 3.4 Phase 2 - Code generation	61
Figure 3.5 Phase 3 - Test case creation.....	62
Figure 3.6 Phase 4 - Execution	63
Figure 4.1 API Architecture in manual testing using postman.....	67
Figure 4.2 Initialize import auth proto file.....	69
Figure 4.3 Sample security testing request in gRPC and validate manually for XPATH Injection	69
Figure 4.4 Sample security testing request in gRPC and validate manually for SQL Injection	70
Figure 4.5 Time execution chart for manual testing	71
Figure 4.6 API Architecture in GSTF Testing	71
Figure 4.7 Pseudocode for GSTF tool	73
Figure 4.8 Sample GSTF testing results	74
Figure 4.9 Time execution chart for GSTF testing	75
Figure 4.10 Comparison time execution between GSTF and manual testing.....	76
Figure 4.11 Sample request for manual verification - XSS	80
Figure 4.12 Sample request for manual verification - SQL Injection.....	81
Figure 4.13 Sample request for manual verification - XPath Injection	82

LIST OF TABLES

Table 1.1 gRPC aspect and research problem matrix (Hitesh Pattanayak, 2024)	16
Table 2.1 gRPC Status Code (Anthony Giretti, 2022).....	32
Table 2.2 Comparison between gRPC and other API based on API Component	35
Table 2.3 GSTF Compared with Other Approach	49
Table 2.4 Data type-based generation payload in a model test case & response validation compared with another method.....	55
Table 4.1 Time execution for manual security testing in gRPC	68
Table 4.2 Time execution using GSTF security testing in gRPC	74
Table 4.3 Improvement Execution Time	77
Table 4.4 Comparison execution time of GSTF vs other tools which support gRPC .	77
Table 4.5 Success and error rate in testing	78
Table 4.6 Status code in testing	79
Table 4.7 GSTF test result coverage with Vulnerable status only.....	83



LIST OF EQUATION

(3.1) Calculate efficiency of execution time.....63

