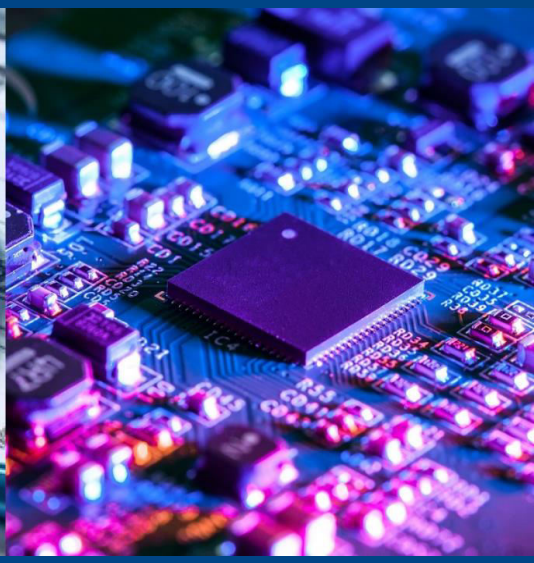
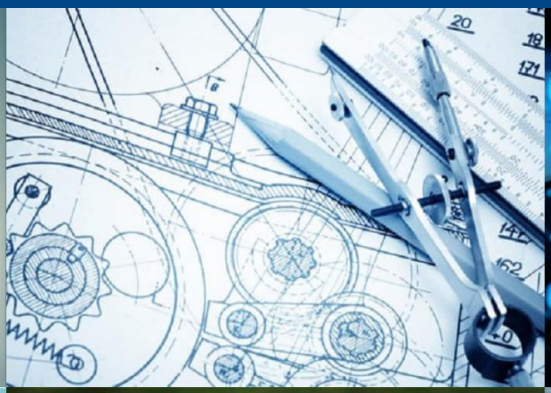
ISSN: 2582-7219

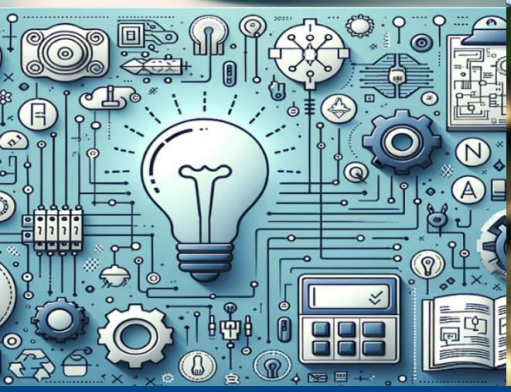




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 9, Issue 3, March 2026**

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI:10.15680/IJMRSET.2026.0903031

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

SecureAPI: Bridging Formal Language Theory and Web Security Using Grammar-Based Validation

Ms. D.Sterlin Rani, S Mohammad Rafeeq, A Prawin Balaji, R Murugan, K Prem Vignesh

Assistant Professor, Department of Computer Science and Engineering, R.M.D Engineering College, Chennai, India

Student, Department of Computer Science and Engineering, R.M.D Engineering College, Chennai, India

Student, Department of Computer Science and Engineering, R.M.D Engineering College, Chennai, India

Student, Department of Computer Science and Engineering, R.M.D Engineering College, Chennai, India

Student, Department of Computer Science and Engineering, R.M.D Engineering College, Chennai, India

ABSTRACT: Application Programming Interfaces (API) have become an essential component of the modern web system, which enables the exchange of data between different services through a structured manner. In spite of the benefits of API, the lack of proper validation of API requests has led to different security issues, such as malformed input attacks and structural manipulation attacks. In spite of the popularity of different techniques, such as the use of regular expressions, the API request validation has focused only on the surface level of the API request, which does not guarantee the proper validation of API requests. This paper presents a new API request validation system, "SecureAPI," based on the Formal Language Theory, which ensures the security of the API request validation process. By using the CFG, the proposed system can validate the API requests through a parser, which ensures the security of the API request processing system. It presents the application of the theoretical computation model for ensuring the security of the API request processing system, which has become an integral component of the modern web system.

I. INTRODUCTION

In the context of modern web-based systems and applications, Application Programming Interfaces (APIs) have emerged as the primary interfaces for the exchange of data among various software modules and systems. APIs can vary from mobile application APIs to cloud-based system APIs. APIs also play an essential role in the exchange of structured data in formats such as JSON and XML. APIs handle sensitive information such as authentication details, financial information, and user information. Therefore, robust validation of API requests is now an essential requirement. The existing API validation methodologies generally rely on regular expression-based methodologies as well as rule-based filtering methodologies. These methodologies generally do not provide robust security for attacks such as nested structural manipulation and grammatical errors. Hackers can easily bypass these superficial validation methodologies for security breaches and vulnerability attacks.

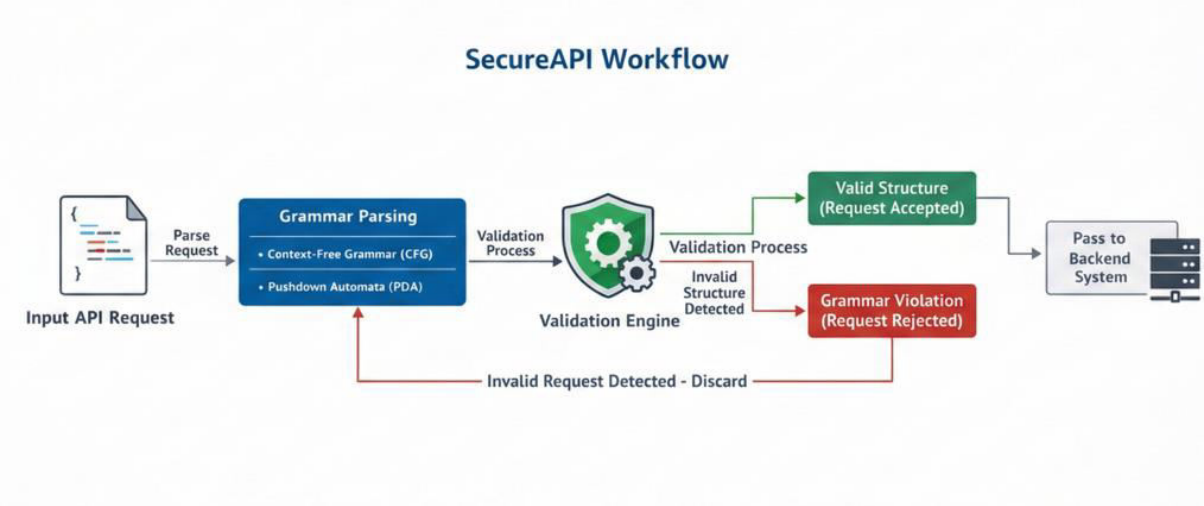
Formal language theory is an essential component of the theory of computation. Formal language theory generally introduces mathematical models for structured data and languages. Context-Free Grammars (CFG) can play an essential role in the context of API request validation. CFG can provide essential methodologies for modeling structured data such as JSON and XML. CFG can be combined with parser-based methodologies such as Pushdown Automata (PDA) for robustness.

© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI:10.15680/IJMRSET.2026.0903031

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



The formal language theory, which is one of the major concepts in the theory of computation, is used to create a theoretical model to define the structure of the language. It has the ability to define the structure of the language, which is in the form of the JSON or XML format used to present the data. It is possible to validate the correctness of the API request using the grammar validation mechanism through the use of the CFG and the parser validation techniques derived from the PDA model. The present paper proposes a new system known as the SecureAPI that is based on the grammar validation mechanism to bridge the gap between the theoretical computation model and web security. It is possible to ensure that the API request is strictly grammatical before the backend services process the request using the formal language theory to validate the API request and thereby increase the structural security of the web application and prevent the web application from malformed input attacks.

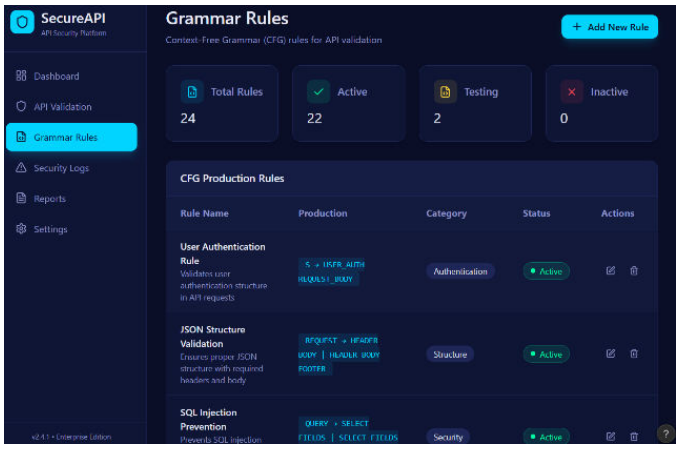
II. PROPOSED SYSTEM

1. Grammar-Based API Modeling

In the case of the SecureAPI tool, the format in which the API request is made is defined using a Context-Free Grammar (CFG). It is used to define the rules that need to be strictly followed in the case of valid requests made to the system.

For example, the rules that are included in the grammar may be as follows:

- Mandatory attributes
- Proper bracket balancing
- Nested object constraints
- Allowed data types



ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In the case of the SecureAPI tool, the format in which the API request is made is defined using a Context-Free Grammar (CFG). It is used to define the rules that need to be strictly followed in the case of valid requests made to the system. For example, the rules that are included in the grammar may be as follows:

- Mandatory attributes
- Proper bracket balancing

2. Parser-Driven Validation Engine

The idea behind the development of the validation engine is based on the concept of parsers, which is derived from the Pushdown Automata (PDA). It is because the structured format in which the API request is made, such as JSON, has a recursive nature. The parser in the validation engine is used to process the incoming API request and validate it according to the rules defined in the grammar. If the rules are strictly followed, the request is said to be structurally valid; otherwise, the system will report a grammar violation.

3. Security Enforcement Layer

The integration of the validation engine between the client requests and the processing layer is achieved through the SecureAPI, which ensures that:

- Faulty requests are rejected at the earliest possible time.
- Nested structural attacks can be detected.
- Unauthorized parameters can be filtered.
- Structural injection attacks can be prevented.
- Only valid and structurally correct requests can be allowed to proceed to the backend.

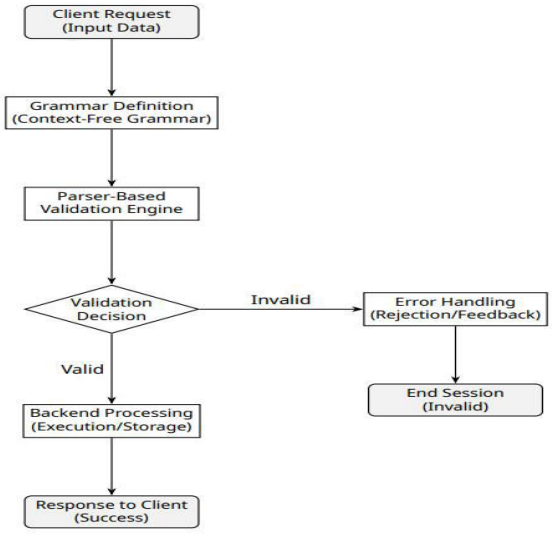
4. Comparative Validation Mechanism

In order to compare the effectiveness of the SecureAPI, it can be compared to other validation mechanisms, like

- Regular expression filtering
- Schema validation tools
- Use of the grammar-based model offers more control.
- Nested object constraints
- Allowed data types

5. System Architecture Overview

The overall workflow of SecureAPI consists of:



© 2026 IJMRSET | Volume 9, Issue 3, March 2026| DOI:10.15680/IJMRSET.2026.0903031

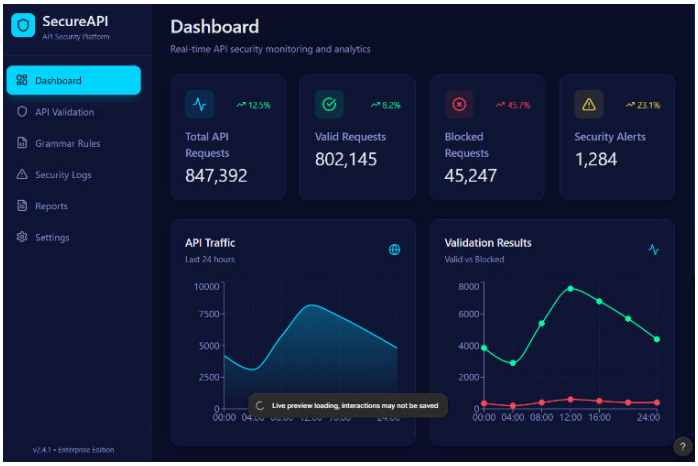
ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

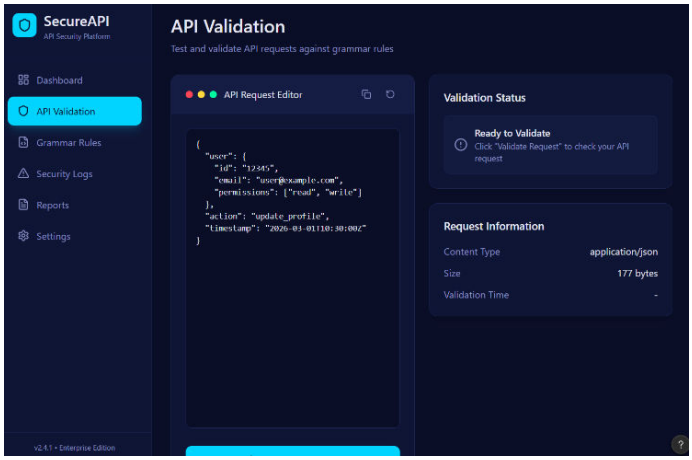
International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

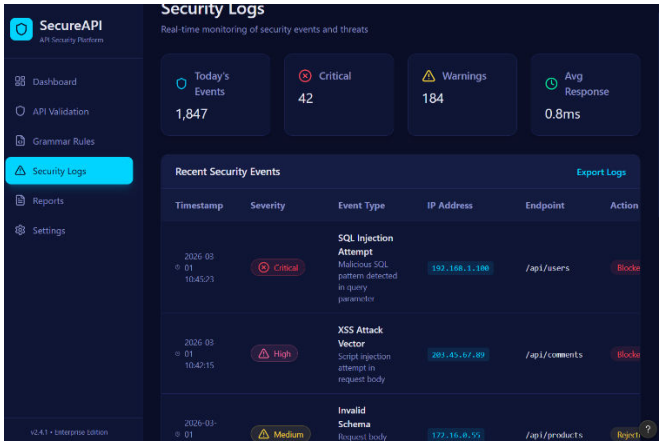
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. OPERATION AND SYSTEM FUNCTIONALITY

On the one hand, the purpose of the SecureAPI is to function as an intermediary validation tool between the client interface and the backend server. The purpose of the intermediary is to ensure that all requests to the backend server strictly adhere to certain predefined grammatical rules. When an entity requests data via an API, and the data is structured in some form, the system is designed to intercept the data before it is processed by the backend server. The system, therefore, attempts to validate the data based on certain predefined grammar. The data is parsed by a parser that is based on Context-Free Grammar (CFG). The data is checked to determine whether it is appropriately structured by checking whether the data is appropriately nested, whether the data has the appropriate number of opening and closing brackets, and whether the data is missing certain parameters. Additionally, the data is checked to determine whether the data has certain parameters that it should not have. The data is appropriately structured in a recursive manner, and therefore, the data is processed by a stack just like in the case of Pushdown Automata (PDA).







IV. LITERATURE SURVEY

Traditionally, API security is implemented through various approaches such as regular expression filtering, schema validation, and rule-based input validation. However, various studies on API security indicate that this is not a strong validation mechanism to prevent manipulations of API inputs or grammar-level inconsistencies. In this context, various studies on web security emphasize that pattern validation does not provide any formal assurance of correctness, particularly for data formats such as JSON or XML data structures. Formal language theory provides formal models of computation such as context-free grammars (CFG), which are used to define recursive data structures such as JSON or XML data structures. Although this is traditionally used for compiler design or syntax analysis, it is not traditionally used for API validation systems. Previous works on structured data validation indicate the need for strong structural validation

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

approaches to prevent injection vulnerabilities or malformed input attacks. However, few studies incorporate grammar-based approaches for API validation systems. The proposed SecureAPI framework is based on the formal theories of computation and incorporates a strong validation mechanism based on context-free grammars for web API validation systems.

V. ADVANTAGES OF THE PROPOSED SYSTEM

There are a number of advantages to using the SecureAPI framework over other validation methods for the API. First, there is a high level of structural validation, which means that the structure of the request to the API is defined according to a certain set of rules defined within a context-free grammar, or CFG. As a result, there is a level of validation of all the requests to the API. However, unlike other validation methods, such as the use of a regular expression, the use of a grammar provides a level of validation of hierarchical data structures, such as JSON or XML.

The second level of advantage to the SecureAPI framework is the level of detection of threats. As a result of the fact that the validation of the request to the backend of the API takes place prior to the actual processing, there is a level of detection of threats to the system, including the threat of injection attacks.

In addition, the parser validation method, which is based on the Pushdown Automata, can efficiently deal with data structures that exhibit recursion and nesting. This ensures that the framework can be scalable and flexible enough to accommodate the latest microservice-based and microservice-oriented architectures. By employing the theoretical computation model, the SecureAPI framework ensures the robustness of the systems and improves the security.

VI. DRAWBACKS OF THE PROPOSED SYSTEM

Despite all these, it has some limitations, which need to be addressed. Some of the limitations that can be identified with the use of SecureAPI are the computational cost, which is incurred, and this might be slightly on the higher side compared to other validation methods, like pattern matching. Another limitation, which can be identified, is the fact that the use of grammars to parse the API might incur some extra computational cost, as the complexity of the API structure might be a little on the higher side. In addition to this, the use of grammars might require some extra efforts to be put in so that the grammar rules are updated appropriately as and when the API structures change. As in the case of SecureAPI, the API structures are defined, and in case of any changes in the API structures, the CFG rules need to be updated accordingly. Finally, the SecureAPI ensures the grammatical correctness of the API, and this might be slightly on the vulnerable side in terms of the presence of a few logical errors, as in the case of SecureAPI, where the API structures are validated and the logical correctness of the API might be a challenge.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VII. APPLICATIONS OF THE SYSTEM

The application of the SecureAPI framework can also be extended to other areas of application where the secure communication of the API is of critical importance. For example, in the development of web applications, the SecureAPI framework can be applied to ensure that the exchange of structured data between the client and server strictly follows set grammar. This ensures that malformed request attacks are eliminated. Also, in the finance and banking sector, the SecureAPI framework can be applied to ensure that critical transactional data is validated before processing to eliminate the risk of structural manipulation and unauthorized parameter injection. Furthermore, in the development of cloud and microservices architecture, where multiple services communicate with each other using the API, the SecureAPI framework can be applied to strengthen the communication between the services. This is done by strictly validating the input data at any point of interaction. Finally, in the e-commerce sector, the SecureAPI framework can be applied to validate order requests, payment information, and credentials. Furthermore, the SecureAPI framework can also be applied in the development of government and enterprise applications that handle confidential information to strengthen the security of the system by eliminating potential vulnerabilities. The application of the formal language in real-world scenarios is essential in strengthening structured data validation in different sectors that rely on API communication.

VIII. CONCLUSION

This paper has proposed the SecureAPI framework, which is based on the application of the concepts of the Formal Language Theory for improving the security of the web. The proposed framework is based on the application of the Context-Free Grammar (CFG) and the application of the Pushdown Automata (PDA) validation technique. This would ensure the strict correctness of the structure of the API requests before the processing takes place. This is quite different from the traditional techniques of pattern validation, which are not necessarily strict.The proposed framework is based on the application of the concepts of the Theory of Computation. The application of the concepts of the Theory of Computation is effective in the proposed framework. The proposed framework has some limitations, which include the computational overhead. However, the proposed framework is effective in the validation of the structured data. The proposed framework can be enhanced by the application of the semantic validation techniques and the intelligent threat detection techniques.

REFERENCES

1. A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd ed., Boston, MA, USA: Pearson, 2006.
2. M. Sipser, *Introduction to the Theory of Computation*, 3rd ed., Boston, MA, USA: Cengage Learning, 2012.
3. W. Stallings, *Network Security Essentials: Applications and Standards*, 6th ed., Pearson Education, 2017.
4. OWASP Foundation, “OWASP API Security Top 10,” 2023. [Online]. Available: https://owasp.org/www-project-api-security/
5. G. McGraw, *Software Security: Building Security In*, Addison-Wesley Professional, 2006.
6. R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Doctoral Dissertation, University of California, Irvine, 2000.
7. S. Chandra and T. Reps, “Practical Structural Validation for Secure Web Services,” *IEEE Security & Privacy*, vol. 15, no. 3, pp. 45–53, 2017.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com