Paper Title:

Automatic Detection of API Access Control Vulnerabilities in Decentralized Web3 Applications

Paper link:

https://ieeexplore.ieee.org/document/10237022

1 Summary

1.1 Motivation

Web3 is a blockchain based web technology that enables decentralized applications (dApps) to interact with external data sources using web APIs. However, these APIs are often vulnerable to access control attacks, such as broken function-level authorization (BFLA), broken authentication (BA) and broken object-level authorization (BOLA), which can compromise the robustness of DApps. Therefore, to detect and prevent such attacks, an automated monitoring system is required which can monitor the security in real time.

1.2 Contribution

The paper proposes a novel approach called Access Behavior Learning (ABL) which can improve the robustness and security of decentralized apps. The ABL approach continuously monitors the different activities among smart contracts, APIs and oracles and then identifies potential vulnerabilities and attack vectors. The ABL approach also uses the Open API Specification (OAS) standard to define and establish the authentication and authorization schemes of the APIs and oracles. The ABL approach aims to provide decentralized application system administrators with a real-time API security monitoring system which can alert them of any malicious activities.

1.3 Methodology

The paper describes the design and implementation of the ABL system, consisting of four phases: data preprocessing and classification, training, detection, alert and message notification. The paper also explains the algorithms and formulas used to perform the various tasks in each phase, such as feature selection, data transformation, risk analysis, security scheme definition, attack detection, and alert generation. The paper analyzes the performance of the ABL system using various DApps and APIs with different levels of vulnerabilities in access control.

1.4 Conclusion

The paper demonstrates that the ABL system is capable of effectively detecting and preventing access control attacks in decentralized applications, such as BFLA, BA and BOLA in real-time. The paper also shows that the ABL approach can achieve high accuracy in identifying attack schemes and normal conditions. The paper concludes that the ABL system can enhance the security and reduce the risk of threats for decentralized applications.

2 Limitations

2.1 First Limitation

Assumption of OAS Compliance: The ABL approach relies on the OAS standard to define and verify the authentication and authorization schemes of the APIs and oracles. However, not all APIs and oracles may comply with the OAS standard or provide accurate and complete specifications. This may limit the applicability and effectiveness of the ABL approach in some cases, as it may not be able to detect or prevent attacks that exploit non-compliant or poorly specified APIs and oracles.

2.2 Second Limitation

Lack of Evaluation on Real-World Data: The paper evaluates the ABL approach using simulated data and scenarios based on open-source Web3 applications and APIs. However, this may not reflect the actual complexity and diversity of the real-world Web3 ecosystem, where different types of dApps, APIs, oracles, and attackers may exist. Therefore, the paper may not provide sufficient evidence or validation of the ABL approach's robustness and scalability in a realistic setting, where it may encounter various challenges and limitations.

3 Synthesis

The paper also opens up new avenues for future research and development in the field of Web3 API security. For instance, future work could explore how to extend the ABL approach to support other types of API security vulnerabilities, such as injection attacks, denial-of-service attacks, or insecure data exposure. Future work could also investigate how to integrate the ABL approach with other security tools and frameworks, such as blockchain analysis, smart contract verification, or API testing, to provide a comprehensive and holistic security solution for Web3 applications.