Jurnal E-Komtek
Vol. 9, No. 1. (2025) pp. 88-97
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

The Application of Smart Contracts in Cybersecurity for Threat Detection and Response

Stefanus Eko Prasetyo[1], Gautama Wijaya[2], Kennedi[3]

[1-3]Department of Information Technology, Universitas Internasional Batam, Indonesia, 29426

2132057.kennedi@uib.edu

https://doi.org/10.37339/e-komtek.v9i1.2285

Published by Politeknik Piksi Ganesha Indonesia

Artikel Info
Submitted:
18-01-2025
Revised:
19-05-2025
Accepted:
26-06-2025
Online first :
30-06-2025

Abstract

Cybersecurity is becoming an increasingly important issue in the digital era due to the rise in threats and attacks on information systems. One innovative technology with great potential to enhance cybersecurity is smart contracts. This article discusses the application of smart contracts in detecting and responding to cybersecurity threats. With their automatic, transparent, and immutable nature, smart contracts can be used to manage threat responses in real-time, improve attack detection efficiency, and minimize the risk of human error. This study also explores use cases such as secure data access management, blockchain-based anomaly detection, and incident response automation. Furthermore, implementation challenges such as scalability, interoperability, and smart contract code vulnerabilities are also addressed to provide a comprehensive overview. Through the integration of smart contracts into cybersecurity, this article concludes that this technology holds great potential to strengthen information systems' resilience against increasingly complex and dynamic threats.

Keywords: Cybersecurity, Digital era, Threats and attacks

Abstrak

Keamanan siber menjadi isu yang semakin penting di era digital karena meningkatnya ancaman dan serangan terhadap sistem informasi. Salah satu teknologi inovatif yang memiliki potensi besar dalam meningkatkan keamanan siber adalah smart contract. Artikel ini membahas penerapan smart contract dalam mendeteksi dan merespons ancaman keamanan siber. Dengan sifatnya yang otomatis, transparan, dan tidak dapat diubah, smart contract dapat digunakan untuk mengelola respons terhadap ancaman secara real-time, meningkatkan efisiensi deteksi serangan, serta meminimalkan risiko kesalahan manusia. Studi ini juga mengeksplorasi kasus penggunaan, seperti pengelolaan akses data yang aman, deteksi anomali berbasis blockchain, dan otomatisasi respons insiden. Selain itu, tantangan implementasi, seperti skalabilitas, interoperabilitas, dan kerentanan kode smart contract, juga dibahas untuk memberikan gambaran yang menyeluruh. Melalui integrasi smart contract dalam keamanan siber, artikel ini menyimpulkan bahwa teknologi ini memiliki potensi besar untuk meningkatkan ketahanan sistem informasi terhadap ancaman yang semakin kompleks dan dinamis.

***Kata-kata kunci**: Keamanan siber, Era digital, Ancaman dan Serangan*

1. Introduction

This research aims to explore the application of smart contracts in enhancing threat detection in cybersecurity systems. As cyber threats become more sophisticated, traditional security measures are often insufficient to protect critical information and infrastructure. Smart contracts, powered by blockchain technology, offer a promising solution by automating responses to detected threats and providing a decentralized, secure framework for cybersecurity operations. The integration of smart contracts into cybersecurity systems has the potential to improve the speed and accuracy of threat detection while reducing the reliance on human intervention **[1] [2] [3]** .

The research methodology involves several stages, beginning with the identification of the scope and objectives, including the specific cybersecurity threats to be addressed, such as network monitoring or anomaly detection. This step is crucial as it helps to establish a clear understanding of what needs to be protected and the potential risks involved[4]. Stakeholders must also agree on the conditions that will trigger the smart contract execution, such as detecting unusual login attempts, abnormal traffic patterns, or unauthorized data access. These agreements are fundamental to ensuring that all parties are aligned and that expectations are met **[5]**.

Once the conditions are established, the next step involves coding the business logic into an executable form. Programming languages like Solidity or Go are used to create the code that automates the necessary actions, such as isolating compromised systems or notifying administrators of potential threats. This stage requires careful consideration of how the smart contract will interact with the network and the specific threats it is designed to address **[3] [5]**. Proper coding ensures that the smart contract operates efficiently and securely, reducing the risk of vulnerabilities in the system.

In addition to coding, the implementation of encryption and blockchain technology is essential for ensuring the confidentiality, integrity, and resilience of the smart contract. Blockchain offers a tamper-proof environment that protects the data from malicious actors, while encryption ensures that sensitive information remains private and secure. These technologies work together to provide a robust framework for smart contract execution, making it resistant to attacks and ensuring the reliability of cybersecurity systems **[6] [7] [8].**

Despite the many benefits, implementing smart contracts in cybersecurity comes with its challenges. Issues such as scalability, interoperability with existing infrastructure, and vulnerabilities in smart contract code must be addressed before widespread adoption can occur. The research will delve into these challenges, exploring potential solutions and offering insights into how smart contracts can be optimized for real-world applications. Ultimately, the goal is to demonstrate how smart contracts can be leveraged to enhance cybersecurity operations and respond more effectively to evolving threats.

2. Method

This research aims to explore the application of smart contracts in enhancing threat detection in cybersecurity systems. The research methodology includes several stages.

2.2. Stages of Implementing Smart Contracts

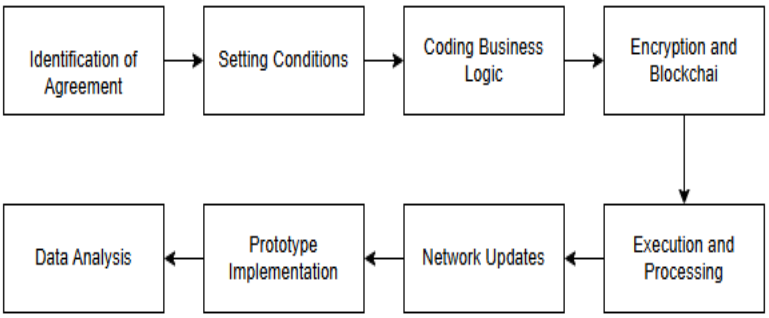

Figure 1. Implementing Smart Contracts

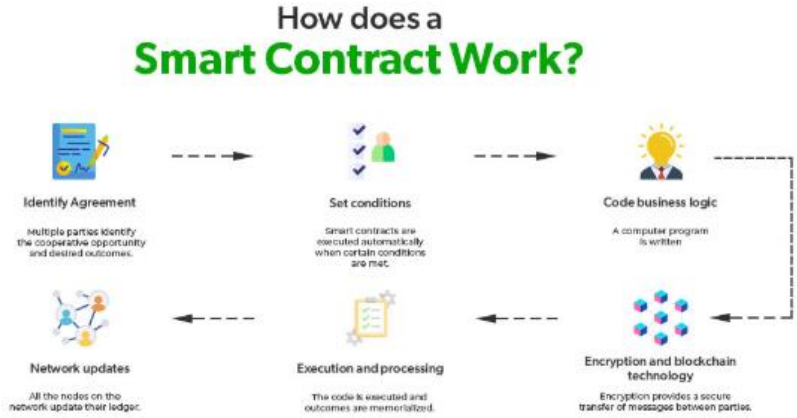

Figure 2. Smart Contract Work

a. Identification of Agreement

Define the scope and objectives, including specific threats to address (e.g., network monitoring or anomaly detection). Stakeholders establish clear agreements to align expectations and identify assets requiring protection **[3] [7].**

b. Setting Conditions

Specify conditions that trigger smart contract execution, such as detecting unusual login attempts, abnormal traffic patterns, or data access violations **[1] [6].**

c. Coding Business Logic:

Translate defined conditions into executable code using languages like Solidity or Go. The code automates actions such as isolating compromised systems or notifying administrators **[3] [5].**

d. Encryption and Blockchain

Utilize encryption to ensure the confidentiality and integrity of the contract. Blockchain provides secure, tamper-proof environment, safeguarding sensitive data and ensuring resilience against malicious actors **[6] [8].**

2.2 Static Analysis with Slither

Static analysis with Slither refers to the use of a specialized static analysis tool to analyze Ethereum smart contracts written in Solidity **[9]**. Slither is a popular and powerful tool designed for detecting vulnerabilities, improving code quality, and ensuring best practices in smart contract development **[10]**.

Slither is a static analysis tool specifically designed for Ethereum smart contracts, written in Solidity. Its benefits include:

a. Time-Saving : Automates the process of identifying common bugs.

b. Improved Security : Helps ensure that your smart contracts are less vulnerable to attacks.

c. Better Code Quality : Encourages developers to follow best practices in Solidity.

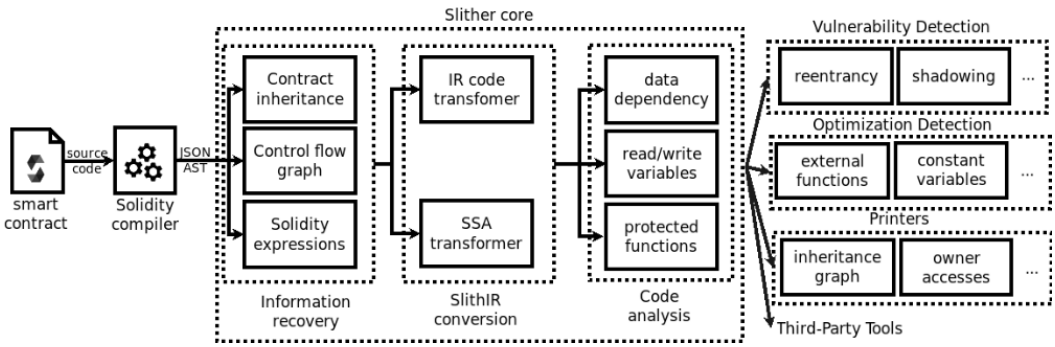

Figure 3. *Static Analyzer For Smart Contracts*

Slither operates as a static analysis tool that evaluates the code structure and logic of Ethereum smart contracts written in Solidity. It performs in-depth checks to uncover potential vulnerabilities and areas for optimization[11]. By analyzing contract components and their interactions, Slither helps developers pinpoint security risks, inefficiencies, and other issues that could compromise the integrity of the contract. The steps in its analysis process are :

- Parsing : Slither parses Solidity source code into an abstract syntax tree (AST) and constructs an intermediate representation (IR) of the contract.

Figure 4. Parsing Slither Code

- Analysis : It performs various static analysis techniques, such as data flow analysis and control flow analysis, to detect vulnerabilities and inefficiencies.
- Output Results : Slither generates a detailed report highlighting security risks, optimization opportunities, and contract design issues, providing actionable insights to help developers quickly address problems and improve the quality and security of their code.

2.3. Fuzz Testing with Echidna

Fuzz Testing with Echidna involves using the Echidna tool to test the security of smart contracts written in Solidity **[12]**. Fuzz testing sends random or unexpected inputs to the code to uncover vulnerabilities and ensure it handles unexpected conditions **[13]**. Echidna helps identify issues such as: security vulnerabilities, logical errors, and unexpected bugs

To provide a clear explanation of how fuzz testing works with Echidna, here are the general steps:

- Smart Contract Analysis : Echidna first takes your Solidity smart contract as input. The contract can have various functions and complex logic that may have vulnerabilities, such as reentrancy, overflows, or unexpected behavior due to edge cases .
- Generating Inputs : Echidna generates random inputs for the smart contract's functions. The goal is to trigger edge cases or unforeseen conditions that may not be apparent during manual testing.
- Execution & Monitoring : Echidna fuzzes smart contracts by generating random inputs and running them in a simulation (e.g., EVM). It checks for issues like assertion failures, logic errors, gas limit exhaustion, unexpected state changes, and security flaws[14].

- Feedback and Results : If Echidna identifies any problematic behavior, it reports it, along with the specific inputs that caused the failure. This allows the developer to debug and fix potential vulnerabilities.
- Customization : You can configure Echidna to focus on certain properties, such as ensuring that certain invariants hold or testing specific parts of the contract. This customization makes Echidna useful for testing complex smart contracts.

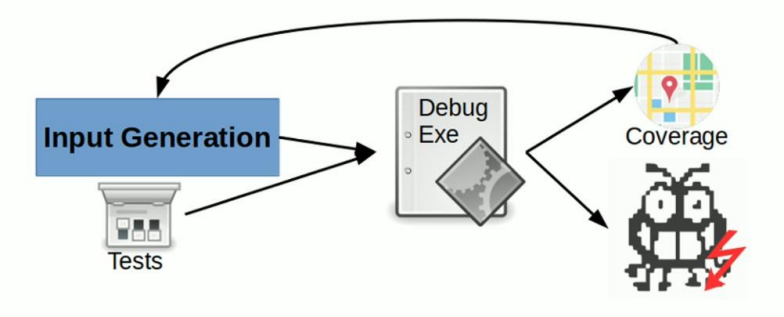

Figure 5. Fuzzing Testing

2.4. Symbolic Execution with MythX

Symbolic execution with MythX is a security analysis technique used to examine Ethereum smart contracts. It involves using symbolic values as inputs instead of actual data, enabling the exploration of all possible execution paths within the contract's code. This method allows for the identification of potential vulnerabilities such as reentrancy attacks and integer overflows, without having to run the contract with every possible input combination[15]. MythX leverages symbolic execution, along with other analysis techniques, to provide in-depth security assessments of smart contracts written in Solidity, helping to uncover issues before deployment on the Ethereum blockchain **[16].**

Figure 6. Security Analysis Tool For Smart Contracts

This research investigates the application of smart contracts in enhancing cybersecurity systems, particularly in threat detection and automated response. Smart contracts, powered by blockchain technology, are self-executing programs that automatically perform actions when predefined conditions are met. This innovation offers potential benefits for cybersecurity by enhancing the speed, efficiency, and transparency of security operations. Scholars have noted the increasing importance of such technologies in addressing the limitations of traditional cybersecurity systems, especially as cyber threats evolve and become more sophisticated.

a. Identification of Agreement

One of the main stages in implementing smart contracts for cybersecurity is identifying the agreement. This stage involves defining the scope and objectives, such as monitoring networks or detecting anomalies. Clearly identifying specific threats and establishing agreements among stakeholders are critical for setting expectations and safeguarding valuable assets. Research highlights that aligning these objectives with actionable, automated responses ensures that security protocols can operate seamlessly and effectively in real-time, reducing the burden on human intervention.

b. Setting Conditions

Following the identification phase, setting conditions that trigger the execution of the smart contract is an essential step. Conditions such as unusual login attempts, abnormal traffic patterns, or data access violations can trigger the contract to execute predetermined actions, such as isolating compromised systems or notifying administrators. The use of smart contracts in such contexts has been proven to improve efficiency by automating complex security tasks, thus mitigating human error and enhancing response times. Previous studies emphasize how these conditions enable smart contracts to provide fast and effective protection against a wide range of cyber threats

c. Coding Business Logic

Once conditions are defined, the next stage involves coding the business logic of the smart contract, typically using programming languages like Solidity or Go. Solidity, widely used for Ethereum-based smart contracts, ensures the execution of the logic that automatically manages threat detection and response actions. Researchers have recognized the importance of secure and efficient coding practices in preventing vulnerabilities. The coding of business logic ensures that smart contracts can be executed reliably and efficiently in response to cybersecurity threats.

Smart contract platforms like Solidity also support encryption and blockchain technologies, which guarantee the confidentiality and integrity of contracts. These mechanisms enhance cybersecurity by safeguarding sensitive data and providing a tamper-proof environment for smart contract execution.

d. Encryption and Blockchain

Blockchain ensures the security of smart contracts by providing a decentralized, immutable platform that prevents tampering. Combined with encryption, it protects data confidentiality and integrity. This combination helps prevent data breaches and unauthorized access, enhancing cybersecurity. Blockchain also offers transparency and accountability, enabling stakeholders to track every action performed by the smart contract

Tools like Slither, Echidna, and MythX are crucial for improving smart contract security. Slither identifies vulnerabilities in Ethereum smart contracts, Echidna uses fuzz testing to find unforeseen issues, and MythX performs symbolic execution to explore all execution paths and detect potential threats. These tools are essential for ensuring smart contracts are secure and reliable in cybersecurity systems.

3. Result and Discussion

In this research, we explored the use of smart contracts in enhancing cybersecurity, particularly in detecting and responding to security threats. Through various tools and methodologies such as static analysis with Slither, fuzz testing with Echidna, and symbolic execution with MythX, we found that smart contracts can play a crucial role in automating the process of threat detection and improving the efficiency of response actions.

The implementation of smart contracts in cybersecurity begins with the identification of specific threats, followed by setting conditions for automatic execution. By using platforms like Solidity and Go for coding, and integrating encryption and blockchain for data protection, smart contracts ensure the confidentiality, integrity, and resilience of cybersecurity systems. These features reduce the dependence on human intervention, thus minimizing the risk of human error and accelerating the response time to security incidents.

However, challenges remain in fully integrating smart contracts into cybersecurity systems. Scalability, interoperability with existing systems, and vulnerabilities in smart contract code need to be addressed before widespread adoption. Despite these hurdles, the promise of

smart contracts lies in their ability to streamline and secure critical cybersecurity operations such as anomaly detection, secure data access management, and incident response.

4. Conclusion

The integration of smart contracts in cybersecurity represents a revolutionary step towards automating and securing digital systems. Their decentralized nature and ability to operate transparently without intermediaries offer a promising solution to modern security challenges. The use of smart contracts for real-time threat detection, anomaly response, and secure data access management has the potential to significantly enhance the resilience of information systems against sophisticated cyber threats.

While there are notable challenges, such as scalability and code vulnerabilities, ongoing research and development are crucial to optimizing these systems for practical, real-world applications. As the technology matures, the application of blockchain and smart contracts in cybersecurity is expected to play an increasingly significant role in safeguarding data and ensuring the integrity of digital ecosystems.

References

[1] M. G. Cains, L. Flora, D. Taber, Z. King, and D. S. Henshel, "Defining Cyber Security and Cyber Security Risk within a Multidisciplinary Context using Expert Elicitation," *Risk Analysis*, vol. 42, no. 8, pp. 1643–1669, Aug. 2022, doi: 10.1111/risa.13687.

[2] Andini Eka Budiyanto, "ANALISIS YURIDIS PENGGUNAAN SMART CONTRACT DALAM PERSPEKTIF ASAS KEBEBASAN BERKONTRAK," *JOURNAL SAINS STUDENT RESEARCH*, vol. 1, no. 1, pp. 815–827, Oct. 2023, doi: 10.61722/jssr.v1i1.402.

[3] "AI-DRIVEN THREAT DETECTION AND RESPONSE: A PARADIGM SHIFT IN CYBERSECURITY Asad Yaseen." [Online]. Available: https://www.orangemantra.com/blog/wp-content/uploads/2022/08/AI-in-cybersecurity-scaled.jpg

[4] M. Raflie Akbar and T. Sutabri, "IJM: Indonesian Journal of Multidisciplinary Implementasi Teknologi AI Dalam Deteksi dan Pencegahan Serangan Malware pada Jaringan Komputer Perusahaan," *IJM: Indonesian Journal of Multidisciplinary*, vol. 2, 2024, [Online]. Available: https://journal.csspublishing/index.php/ijm

[5] J. Van Der Ham, "Toward a Better Understanding of 'Cybersecurity,'" *Digital Threats: Research and Practice*, vol. 2, no. 3, Jun. 2021, doi: 10.1145/3442445.

[6] F. Al Fikry, J. Mualimin, and S. Nurhasanah, "DIGITAL MARKETING, CYBERCRIME, AND ISLAMIC BUSINESS ETHICS A CASE STUDY IN INDONESIA," *AB-JOIEC: Al-Bahjah Journal of Islamic Economics*, vol. 1, no. 2, pp. 90–102, Dec. 2023, doi: 10.61553/abjoiec.v1i2.68.

[7] A. G. Lase and M. P. Pratiwi, "PERANCANGAN E-VOTING BERBASIS SMART CONTRACT MENGGUNAKAN KEAMANAN ALGORITMA KONSENSUS PROOF-OF-STAKE," *JURNAL COMASIE*, vol. 11, no. 01, 2024.

[8] M. ; Jurnal, W. Wasriyono, D. Apriliasari, and B. A. P. S. Seno, "Inovasi Pemanfaatan Blockchain dalam Meningkatkan Keamanan Kekayaan Intelektual Pendidikan," *Jurnal MENTARI: Manajemen, Pendidikan Dan Teknologi Informasi*, vol. 1, no. 1, pp. 68–76, 2022, [Online]. Available: https://journal.pandawan.id/mentari/article/view/142

[9] "Analisis+Peran+Teknologi+Blockchain+terhadap+Keamanan+Siber+Pada+Aset+Digital".

[10] R. M. Muria, A. Muntasa, M. Yusuf, and A. Hamzah, "Studi Litelatur: Peningkatan Kinerja Digital Forensik Dan Pencegahan Cyber Crime," 2022.
[11] "PERLINDUNGAN HUKUM TERHADAP KERUGIAN PELAKU USAHA."
[12] A. S. Putra and Y. Prayudi, "Jurnal Sistem dan Teknologi Informasi Implementasi Multi Smart Contract pada Bukti Digital dan Chain of Custody dalam Meningkatkan Keamanan dan Integritas Bukti Digital," vol. 6, no. 2, 2021, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO
[13] S. Bentuk Perkembangan Kecerdasan Buatan, P. Studi, and K. S. Program Magister, "TINJAUAN YURIDIS PENERAPAN SMART CONTRACT DI INDONESIA."
[14] Shofie Azizah, Zava Nurruzzuhrotil Ula, Dwi Mutiara, and Michelle Prajna Prameswari, "Keamanan siber sebagai fondasi pengembangan aplikasi keuangan mobile: Studi literatur mengenai cybercrime dan mitigasinya," *Akuntansi dan Teknologi Informasi*, vol. 17, no. 2, pp. 221–237, Oct. 2024, doi: 10.24123/jati.v17i2.6409.
[15] A. Agustina, "Kajian Penerapan Smart Contract dalam Pendaftaran Jaminan Fidusia Secara Elektronik," *Jurnal Cendekia Ilmiah*, vol. 3, no. 6, 2024.
[16] D. Perwej, S. Qamar Abbas, J. Pratap Dixit, N. Akhtar, and A. Kumar Jaiswal, "A Systematic Literature Review on the Cyber Security," *International Journal of Scientific Research and Management*, vol. 2021, no. 12, pp. 669–710, doi: 10.18535/ijsrm/v9i12.ec04ï.