Optimization and Evaluation of Authentication System using Blockchain Technology

Imam Riadi [1*], Aulyah Zakilah Ifani [2], Ridho Surya Kusuma [2]

[1] *Department of Information System, Universitas Ahmad Dahlan, Yogyakarta 55164, Indonesia*

[2] *Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta 55164, Indonesia*

Abstract

User data security innovation is a particular concern in protecting one's privacy rights, which is one of the serious violations when an attacker can bypass the user authentication so that it looks like something legitimate and becomes legal. Based on these issues, the research aims at optimizing and evaluating the blockchain-based authentication systems to minimize data leakage, manipulate the data, and modify the data. Blockchain is one of the innovations that can solve this problem. Data or transactions in the blockchain are saved in hash form to make it difficult for hackers to break into them. The Blockchain implementation uses the Solidity programming language to build smart contracts and other tools such as MetaMask, Ganache, and Truffle. The Network Forensics Development Life Cycle (NFLDC) is used as a framework with the following five stages: Initiation, Acquisition, Implementation, Operation, and Disposition. Based on the research conducted, the attack strategy against blockchain-based systems consists of several scenarios covering the Burp Suite, XSS, SQL Injection, and DoS. The results show that the percentage of authentication optimization reaches a value of 90.1%, and 8.9% is the percentage for evaluating systems such as the possibility of cyberattack. Based on these results, this research has achieved its goals and may assist in further research.

1- Introduction

Internet development in this era is something required by internet users. The big challenge in using the internet is the existence of illegal attacks to gain access to systems [1]. Many research studies have begun to discuss wireless, for example, by using MAC Address Filter [2, 3], Service Set Identifier (SSID) [4], Extensible Authentication Protocol (EAP) [5], and even captive portal [6]. The process of modifying the login authentication system uses blockchain and innovative contracts. In this study, the Blockchain technology-based login system is the main subject of testing with various attack scenarios. The framework used is the Network Forensic Development Life Circle (NFDLC), while the testing uses XSS, Burp Suite, SQL Injection, and DoS. Systems that are not properly secured will be vulnerable to attack, either on the network side or directly from the system. In the testing, the research uses several attacks on computer network behavior through network traffic logs to reconstruct the early events with new engineering attacks [7].

The login authentication system requires to be considered as a security. Authentication is proof of identity [8]. This system usually uses an authentication process in the form of usernames and passwords. It requires systems that provide early warning when DoS, Burp Suite, SQL injection, and XSS attack the user's site or website. Cross-site-scripting

---

(XSS) is a gap in the system that may allow other people to enter by exploiting the system [9]. DoS, or commonly called Denial of Service, is one of the web attacks. This type of attack is generally conducted by blocking the network traffic through many attacks, and this may occur on many computers [10, 11].

Blockchain-based authentication occurs when a user logs into the system. Authentication cannot be used as a guideline to secure the system from the attacks of DoS, Burp Suite, and XSS, so this research uses blockchain technology and smart contracts in building the system. It will be difficult for hackers to corrupt the data, particularly by changing or modifying the data [12]. That is because all computers have the same data. Furthermore, blockchain technology has a decentralized nature, which means that it takes a long time for hackers to crack the code on each block [13].

Some previous researches using blockchain technology includes those on blockchain applications with smart contract integration in crowdfunding systems. The research explains that the system is centralized, not providing the complete data on embezzlement activities and complete transparency. The fundraising users and funders have indicated that the functional requirements of the system may implement the following use case designs [14]. The research on blockchain crypto currency technology aims at illustrating the opportunities obtained from the technology that plays a role in crypto currency [15]. The research on security and privacy uses blockchain [16]. The research on top-up transaction data security of school service uses blockchain technology. The results of this research obtain that the modification of cryptographic technology and blockchain may function properly on the system [17]. Various applications of blockchain technology are about health, Internet of Things (IoT), fundraising, digital asset management, education, and many others [18-20]. Blockchain can also be used in improving e-commerce systems [16] and e-voting system [21].

The login system requires authentication to secure the user's identity, so the research uses blockchain technology to secure the data from users and use the Network Forensic Development Life Circle (NFDLC) as the framework in supporting the system development and in overcoming the system weaknesses. Blockchain may provide trust to the third parties who oversee the process between sellers and buyers to confirm the authenticity of data and information [22]. The research aims at securing the login authentication system using blockchain technology from the attacks of Burp Suite, XSS, SQL Injection, and DoS based on the previous research.

2- Research Method

Various applications can use blockchain technoloy-based login authentication systems. Smart payment applications, crypto currencies, healthcare systems, and many others can use blockchain technology.

2-1- Blockchain Technology

The blockchain is strands of sustainable of data blocks that list the previous data blocks from the new data. Each block records a set of related metadata data transactions. The blockchain network cannot destroy and store the data blocks and the data in every computer participating in it. Satoshi Nakamoto first implemented blockchain in 2008 as a peer-to-peer money exchange system. Nakamoto referred to transactional tokens exchanged between clients in the system as Bitcoin [23, 24]. The first blockchain creates transactions between A and B records without the intermediaries. The cost of transactions with blockchain is much cheaper compared to traditional ways involving intermediaries. Transactions using blockchain are much more secure blocks [25].

Blockchain is not a stand-alone technology but is a configuration of many technologies, tools, and methods that overcome particular issues or use cases [26]. Blockchain technology separates itself from traditional centralized approaches that make it possible to securely manage the chain data across a network of distributed and interconnected nodes [27]. The bitcoin crypto currency is the first application of blockchain underlying the transaction recording mechanism [28]. Bitcoin uses the blockchain concept which is a solution to the problem of the lack of third parties or intermediaries from financial institutions. Distributed bookkeeping technology, commonly called Distributed Ledger Technology (DLT), defines the blockchain concept, which in its implementation means that every connected person in a network has the privilege to access the block [29]. The concept of distributed database is that data is stored and distributed to each network when the data or information is recorded. The technology describes the method of eliminating third parties (financial institutions) in crypto currencies. The blockchain concept may also prevent double transactions because if someone changes a block in the blockchain or manipulates a block (transaction duplication in this case), the hash value will become invalid. Furthermore, it does not store a valid value, i.e. the hash value of the previous block [30].

After validation and consensus decision, each blockchain works [30] cryptographically on the previous block hash links. When the mining process succeeds in creating a new block, the data on the previous block will be difficult to change or manipulate. Data storage or transactions on the blockchain will be saved in the form of hashes. The hash form is hexadecimal except for storage, as a pointer that connects blocks using a hash that has a function. It can generate and validate new blocks [31]. Figure 1 is the format for each block.
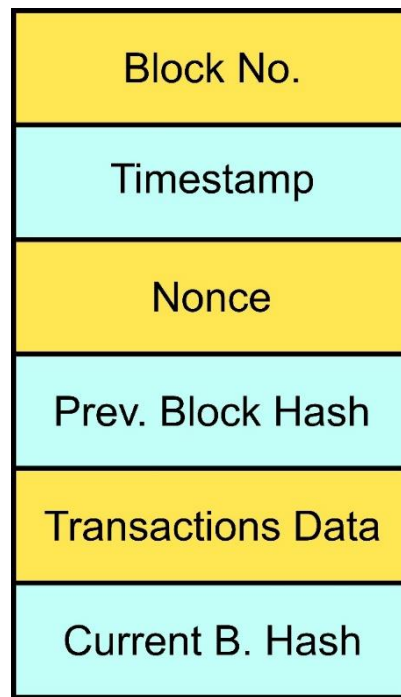
Figure 1. Block Format

Figure 1 indicates the contents of each block. Block No, which represents block numbers, identifies any block in the ledger. Timestamp is also an important factor in finding any block of the entire ledger. Nonce is for providing consensus among nodes as well as creating identical hashes from different ledgers. Previous block hash connects one block to another and creates a list of links from blocks in the ledger that makes the data irreversible. The amount of transaction data stored through the Markle tree concept can find any transaction from the block. The current block hash indicates the integrity of block hash functions such as MD5, SHA 256. The Genesis Block becomes the first block of a ledger without hash blocks and previous transactions [31].

2-2- *Smart Contract*

Smart Contract was Invented by Nick Szabo in the mid-1990s. Nick Szabo recommends converting contract clauses into code and putting them into software and hardware for automated execution, minimizing contract costs between the transacting parties and preventing unwanted errors and malicious behaviour during the contract processing [32]. A smart contract is a self-executing computer transaction record that facilitates and supports the validation of any contract. A smart contract has a code function consisting of a complete set of turning operations used to create contracts. After calling the contract, each contract will save the transaction into a decentralized and irreversable database [33]—an integrated contract procedure for databases operated by blockchain for managing and transferring digital assets. Blockchain runs database programs for managing and transferring digital assets. The programming language for building smart contracts is the solidity programming language [34].

The characteristic of smart contracts is that the program or code runs on a blockchain platform, and a machine can read that code. The smart contract is part of a special program in the application. After the Smart Contract is available, it can be distributed [35, 36]. Figure 2 is a smart contract mechanism.
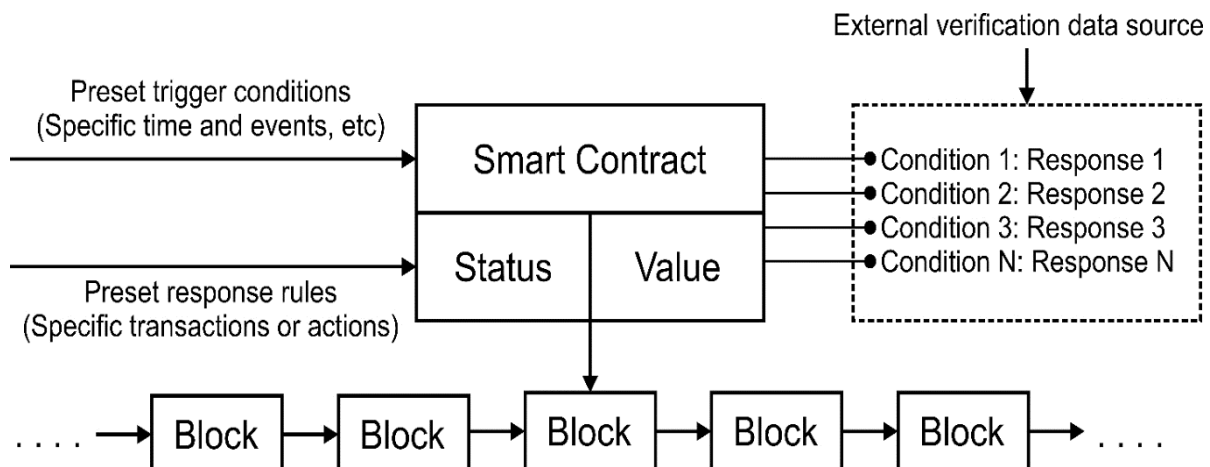

Figure 2. Smart Contract Mechanism [36]

Figure 2 indicates the smart contract process mechanism in which all parties have signed the smart contract. It is attached to the blockchain as a program code (such as a Bitcoin transaction), spread through the P2P network, verified by nodes, and registered. A smart contract consists of a predefined set of transition status and rules, There are some situations that trigger the execution of a contract, such as the inevitable occurrence in specific time or moment, a response in a particular situation, and any others. Blockchain monitors the real-time status of smart contracts and executes contracts when certain activity conditions are fulfilled [36].

2-3- Burp Suite

Burp Suite is an open-source tool for performing or testing the security on an application or system [37]. PortSwigger Company creates Burp Suite using the java programming language. The ability to intercept HTTP becomes a priority on the Burp Suite [38][39]. The primary function of the Burp Suite is to intercept and display HTTP messages in a structured manner. The Burp Suite provides the tester a brief overview of the target system, all messages and parameters sent. In addition, it also provides a GUI that has complete control over all messages – drop, forward, repeat, modify, send later, and so on. So the tester may design different attack scenarios and execute manually through the Burp Suite. The results of the game can be directly viewed and analyzed by the tester [40]. Burp Suite has a commercial version, such as XSS, brute force, and others [41]. The research uses a free version of Burp Suite. This research does not require professional features.

2-4- Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) is one that attacks web applications. XSS can steal information from the users [42]. The vulnerability's web application could allow cybercriminals to inject their malicious code into their TV displayed for end-users to receive. There are three XSS attacks: XSS, Reflected-XSS, and DOM-based XSS attacks [43].

These attacks can work well on the client-side, server, or both [44]. For the analyzing process there are three XSS-based defense approaches: static, dynamic, and hybrid [45].

(a) **Static analysis approach:** it is to notice the data control during the runtime before running the program;

(b) **Dynamic analysis approach:** it is done during the agreement at runtime after the software is released. This approach analyzes the data obtained during the program execution;

(c) **Hybrid approach:** it uses both static and dynamic approaches.

2-5- Denial-of-Service (DoS)

Denial-of-Service is a type of attack on a computer or server on an Internet network that depletes a computer's resources until the computer no longer functions properly, which indirectly prevents other users from accessing the computer services attacked. [46]. The most common DoS attack aims to deplete network bandwidth, CPU cycles, or memory on the target system to unavailable services for the legitimate users [47]. It also aims to flood the resources of the target system and to explore and exploit the weak points in the system [48].

The types of DoS attacks most widely used by the attackers are Syn-Spoofing, UDP-Flooding, HTTP-Flooding, Slowloris, Slow post, ICMP Echo, Brute Force.

(a) **SYN-Spoofing:** it attacks the target server tables that manage the connections to clients;

(b) **UDP-Flooding:** it targets a specific port on the victim's system or server and then floods that port with a UDP packet to overload that port and stop the provided port;

(c) **HTTP-Flooding:** it is a HTTP protocol attack, in which multiple bots flood the web server by assigning HTTP requests that deplete memory resources;

(d) **Slowloris:** it is an attack in which the attacker makes multiple connections to the server due to incomplete HTTP requests;

(e) **Slow post:** it is an HTTP attack that sends an HTTP request to the victim's server;

(f) **ICMP Echo:** it is an attack that floods the server with an echo request;

(g) **Brute Force:** it is the attack carried out on computer security systems by using passwords as authentication [49].

2-6- SQL Injection

SQL Injection is an attack technique against vulnerabilities or vulnerabilities owned by SQL [50]. It is a technique to discover vulnerabilities on websites that cause a hacker to influence SQL queries submitted through a database

website [51]. SQLI attacks usually use a single quote character (') or double quote character (-) or fence sign (#) at the end of the number parameter to find out if the website is vulnerable or not [50].

2-7- *Network Forensic Development Life Cycle (NFLDC)*

Network Forensic Development Life Cycle (NFDLC) is a combination of the Information Systems Development Life Circle (ISDLC) and Network Forensic Readiness (NFR) methods. The concept of NFR maximizes the ability to gather credible evidence while minimizing the cost of insid e response. The concept is a recommendation to increase the efficiency of the investigation. However, there is a little discussion on how to integrate NFR into a network of systems. NFR appears to investigate malicious online intruders. In this case, NFR is a case study. They design ISDLC to combine the security throughout the system development cycle. The ISDLC method of each phase is analyzed and modified to insert the additional steps that create digital forensic embedding. Specific ISDLC modifications result in NFLDC [52, 53]. The function of using this framework is to evaluate the authentication system in the use of blockchain technology. The use of NFDLC in research to evaluate the authentication system so that the results in network forensics serve as a reference in evaluating the authentication system and enabling the system's sustainability to be better.
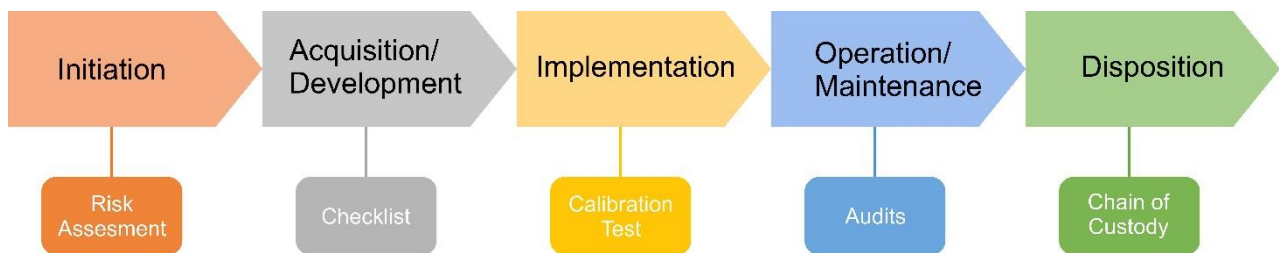

Figure 3. NFLDC Framework

Figure 3 indicates the NFLDC framework conducting the stages of research. The stages consist of five parts: Initiation, Acquisition, Implementation, Operation, and Disposition as follows:

(a) **Initiation:** This stage is called the preparation stage. This stage is the process of developing all the requirements that exist in this research. Preparation is required for the investigation process to run smoothly, including determining the software to be used in the investigation process to obtain effective results and planning the actions to be taken during the investigation.

(b) **Acquisition:** This is the stage of designing system development, flowchart design. Design, in this case, is used to support the creation of the system. The system design stage allocates the requirements of the hardware and software system to form the entire system architecture.

(c) **Implementation:** This is the stage of changing the design made on the system running on demand. This stage is to implement software design as a series of programs or program units. The testing involves verifying that each unit meets user specifications. Users perform tests to ensure that they are compatible with the software.

(d) **Operation/Maintenance:** This stage is the longest one. The system has been installed and used. Maintenance includes correcting errors missed in the previous stage based on the evaluation results. The system evaluation in this research uses Burp Suite, XSS, SQL Injection, and DoS attack scenarios. Maintenance includes fixing errors that are not found in the previous stage based on the evaluation results to optimize the system better.

(e) **Disposition:** This is the stage of ensuring the evidence obtained that is appropriately stored. It is required to be able to reuse it.

3- Result and Discussion

3-1- *Initiation*

Case simulations of this research use the Visual Studio Code tool. The result of the simulation stage is a system that is tested and run. The login system utilizes the Ethereum blockchain platform, which implements blockchain technology and smart contracts. As an application programming interface, Web3js connects the browser with an extension called a metamask, which acts as a bridge between the login system and the Ethereum blockchain. This metamask acts as an Ethereum wallet for information management. Meanwhile, smart contracts are built using the solidity programming language.

In this research stage, the user should be the member of blockchain network. Becoming the member, the user will have an Ethereum account. An Ethereum blockchain-based application must run a smart contract. So smart contracts

will be signed first by those who already have an Ethereum account to run an Ethereum-based application. If the user wants to log in, he should be registered as a smart contract signer. Login data from users will be saved as a hash to the blockchain via smart contract. The user requests access to the website; for example, a hash derived from the user is provided credentials compared to the hash stored in the Smart Contract. Therefore, when the user logs in and matches the data, he is authorized to access the web. On the contrary, if it does not fit, then the access is denied. Each user should be connected with an Ethereum address previously done in the registration process, because this address generates the user's login hash. The testing to discover the optimal level of the system against the cyberattack is done using several attacks such as Burp Suite, XSS, SQL Injection, and DoS. The testing uses multiple attack scenarios.

3-2- Acquisition

Flowchart is the first thing in doing this research because it is the research workflow to be built. The flowchart becomes a diagram in which it discusses the process, system, and algorithm that will be carried out in this research. Figure 4 is a system flowchart when performing operations. Figure 4 is a flowchart that indicates the existence of a login system to a blockchain-based system. This flowchart can easily help clarify the complex processes. More complexities help explain the relationships of the steps in each process.
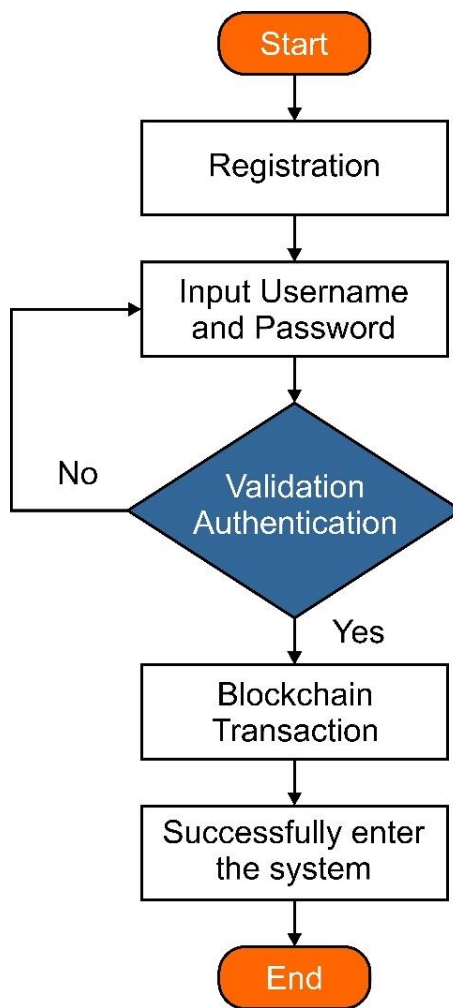

Figure 4. System flowchart

Figure 4 indicates the flowchart of the blockchain system, in which the user registers first and then inputs the username and password. The data entered is then authenticate the validation. The purpose is to discover the authenticity of data from the users. The system will verify, and then the user will perform blockchain transactions. The system ensures that the data on each transaction is correct, and then the system verifies the username and password. The process is complete and the user is successfully logged into the system. Figure 5 indicates the registration flow.

Figure 5. Registration Flow System

Figure 5 is the registration flow system. The workflow of registration for users has no an account yet. Deploy commands will be sent to the user, then the contract is forwarded to the browser. The metamask browser will detect the contract. Metamask will disseminate the contract after confirming, then the identification of the contract is reversed when used. Identification appears on the website after the identifier is detected, then it returns to the metamask extension. Metamask stores the contract identifiers, and now the data can be used. When the user has already had an account, the user will log in. Figure 6 indicates the flow of the login.

Figure 6. Login Procedure Flow

Figure 6 indicates the flow of the login procedure. After the registration process is complete, the user will go to the login page in chrome. Chrome storage identifies the contract from the metamask extension storage. Furthermore, the background page contains contract, timestamp and timestamp flags to the content script. Figure 7 indicates the security scenario of the login system.
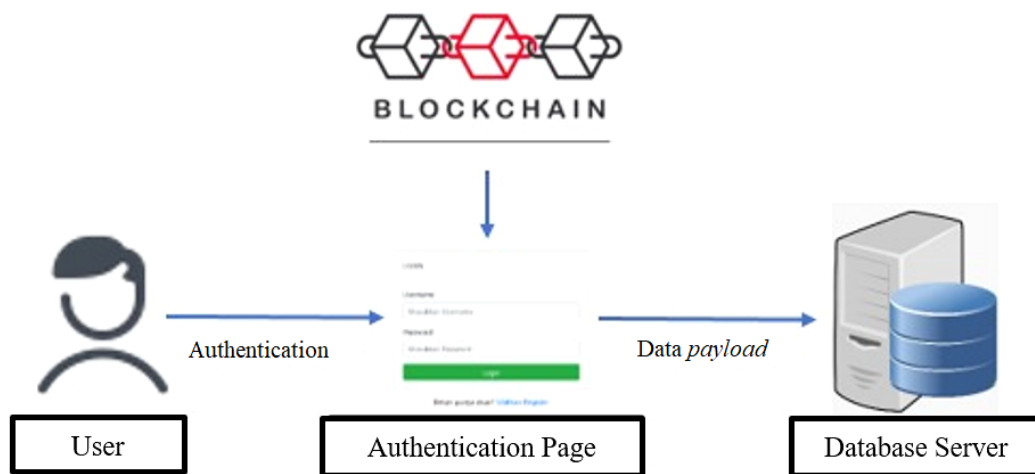

Figure 7. Login System Security Scenario

Figure 7 is a security scenario in which the user performs the process of filling in the username and password, and then the system performs the authentication process. If the user wants to log in, he should be registered as a smart contract signer. The user will save the login data as a hash derived from the user's credentials, then compare it with the hash in the smart contract. When the user logs into the system and the data is appropriate, the user is authorized to access the web. Otherwise, the access is denied, such as testing the login system against cyber-attacks using several attack strategies against the Ethereum blockchain system consisting of several scenarios Burp Suite, XSS, SQL Injection, and DDoS as shown in Figure 8.
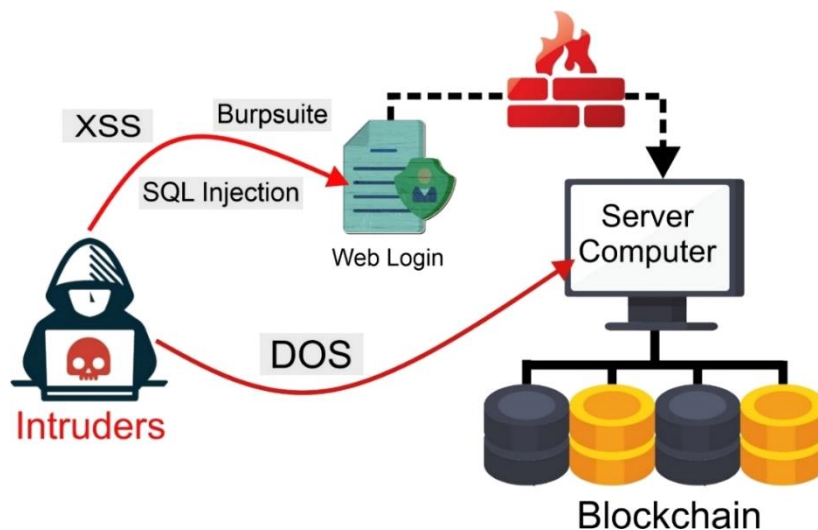

Figure 8. Scenario of Cyber Attack against Website Login

Figure 8 provides an overview of the flow of cyberattack on this system testing. The attacks of Burp Suite, XSS, and SQL injection aim to obtain any security breaches that may occur, while Denial of Service (DOS) attack is to test the capabilities of the server infrastructure against indiscriminate attacks, thus degrading the system performance. The attack strategy in this research is formulated in Equation 1.

$$D^*_p \in \arg\max_{D_p \in \varphi(D_p)} \mathcal{A}(D_p, \theta) \tag{1}$$

Equation 1, variable $D_p \in \varphi(D_p)$, is a set of cyber threats used by an attacker. Variable $A(D_p, \theta)$ is a function to represent the attacker's objectives [40]. The system test seeks to find system optimization problems by considering each vulnerability and several cyber threat models.

3-3- Implementation

This section describes the implementation of blockchain in which this research uses the Ethereum blockchain platform that implements blockchain technology and smart contracts. As an application programming interface, web3js connects the browser with an extension called MetaMask, which acts as a bridge between the login system and the Ethereum blockchain. This MetaMask acts as an Ethereum wallet for information management. Meanwhile, the following smart contract code uses the solidity programming language in Table 1.

Table 1. Building a Smart Contract

Program Code
contract Authentication {
uint256 public nbOfUsers;
struct user {
string signature hash;
address user address;
contract Authentication {
}
mapping(address => User) private user;
constructor() {
nbOfUsers = 0;
}
function register(string memory _signature) public {
require(
user[msg.sender].userAddress ==
address(0x0000000000000000000000000000000000000000),
"already registered"
);
user[msg.sender].signatureHash = _signature;
user[msg.sender].userAddress = msg.sender;
nbOfUsers++;
}
function getSignatureHash() public view returns (string memory) {
return user[msg.sender].signatureHash;}
function getUserAddress() public view returns (address) {
return user[msg.sender].userAddress; }}

Table 1 is the source code for creating smart contracts. After executing the contract on the blockchain, the contract can provide services to use. The contract transaction will be directed to the contract-related address when the user registers, as in the registration function. In this case, the MSG variable is a place to save all data information. The contract then processes the recipient's data with the process written in the registration function. At the time of registration, the user should be the member of blockchain network. First, the user creates an account before accessing

the network. Ganache becomes the place to get network. Ethereum blockchain-based applications must run smart contracts. Smart contracts will be signed first by those who already have an Ether ID to run Ethereum-based applications. Ganache already provides 10 default accounts, and each account has a balance of 100 eth. This account is for transactions on the Ganache blockchain. Figure 9 indicates the Ethereum address in the Ganache.

Figure 1. Ethereum Address in Ganache

Ganache has already provided 10 default accounts, and each account has a balance of 100 eth (Figure 9). This account is for transactions on the Ganache blockchain, in which one network is for one registration process. Ganache is a local blockchain that normally has many nodes (computers), which in this study it can be deployed with localhost or on Ganache. Ganache is only on the personal computer and other people cannot access it because it is not on the real network. However, people may access it using a wallet called MetaMask. So if the user wants to register a new account to enter the login system, the user may take one Ethereum network on the Ganache. If the user wants to log in, he should have been registered as a signer of the smart contract. Before he logs into the system page, he first connects the Ganache to MetaMask. This MetaMask is a bridge for users to log into the web browser. This MetaMask allows the users to run Ethereum Dapps directly in the browser. Gas price is gas consumption for transaction delivery. Gas fees are transaction fees that users pay to miners on the blockchain protocol to have their transactions included in the block. The following is a login screen via the Truffle program, as shown in Figure 10.

Figure 2. Contract Deployment Results

The migration script goes into the migration folder. Migrations are files that help deploy new changes in contracts to the Ethereum blockchain, and these migration contracts help track which migrations have been executed. Truffles execute scripts in those folders using lexicographic sequences. The command to run Truffle is Struffle migrate. This command will run all migrations located in the project directory. Migration is only a set of managed deployment scripts if the previous migration is successfully managed or executed. This folder, by default, contains 1_initiation_migration.js scripts, which migrates Migration.sol contracts, which is useful for Truffles (Figure 10). Blockchain will store blogs that have been registered and carry out transactions until the process is complete. Figure 11 indicates a record of successful transactions on the blockchain.

Figure 11. Successful Transactions Recorded on the Blockchain

Note that on the Blockchain (Figure 11), users can see the gas used. After the transaction is successful, users may log into the system using the correct username, password. In this blockchain, the cost of computing varies depending on the type of contract. The Ganache platform in this research identifies two contracts: call contracts and contract creation. The routine of making contracts through new agreements involves two parties, buyers and suppliers. The routine of contract calls is a daily process that is run based on the user specified when needed. The permanence of blockchain can be seen on this contract call, which does not undergo the changes that previously occurred in the contract-making routine. Data from users cannot be deleted, edited, or even deleted without the permission of others. After that, the user is managed to get into the system. Table 2 is the record of the resulting tx hash.

Table 2. Hash Transaction

Block	Tx Hash	Gas Used
1	TX 0x3028abd60f6b9dffbe944f1358109e074437fbb6626c7bbb2bd6b444c552f606	605211
2	TX 0x37e11c23c9eadfeb60c9624d89b160f2b55456f735bce9a8eb307729991abada	14796
3	TX 0xfd4ff9b826779977439ec0b58f53e280040378d2e96178184bf4278e2a70869b	132976
4	TX 0x983fc20847ca06847630824a63cb7ccd4da0ae7c8859c49442ce9e27fe0993b8	132976
5	TX 0xc336ccfe700354cc5ba74a7b30f957b21e3ae0264bc6359fc12f28abc342f1d2	132976
6	TX 0x83c97365d2214f08d050477e019052e1f92e8cf75c2152738f88561214161a84	132976
7	TX 0x6478248e0961131ac3fb20205e726843d7315d3217933a92be0570e71f22de9f	132976
8	TX 0xe0cae91fa0fe27c09c219da6e07567a316674d0d297991a2a31ba4b81bd85fbc	132976
9	TX 0x472232d8e68a717382f33a660c55be7fe4aa3a095e9c752b30cc00eaf970fb07	132976
10	TX 0xcd938e4c334109fe7747dd47125e5a1c86ee3d0ff113782db99204bbcb179414	132976
	132976	

Table 2 indicates the tx hash and gas generated after entering the username and password on the system. The registration process generates a hash value on the Ganache. Ganache is a place to store interconnected blocks. TX data contains bytecode for smart contracts, the Ganache indicates gas and gas limits for blockchain. Each transaction sets a gas price as the highest amount of ETH that the transaction is willing to pay for each gas unit. The transaction also sets the gas limit, which is the maximum amount of gas equal to the gas limit and is the maximum amount that the transaction is willing to pay.

3-4- Operation

This section describes the test results using Burp Suite, XSS, SQL Injection, and DDoS. The login page in this research only uses blockchain without any security code. Here are the results of the attack scenario.

- Burp Suite attack scenario in testing this system using version 2021.8.2 community edition has not made a significant impact.

- Scenario SQL Injection attack attempts to violate security by entering the database system on the website. This attack attempt only provides an "Incorrect Login" response repeatedly.

- The XSS attack scenario in this research uses the injection code <script src="https://10.10.10.8:3000/hook.js"></script>, and injects the code into any URL or website elements. The injection code attack on the URL address is not successful, so the website page only provides a refresh response display. The results of the XSS injection attack in the login form section can be seen in Figure 12.

Figure 12. XSS Injection Attack on Website Login

Figure 12 provides information on the results of the XSS attack. The attack is managed to make the website page into an error due to an undefined AccountAddress. Based on these results, applying secure code in the login form section is necessary to propagate or verify that the input value entered is appropriate.

- The DoS attack scenario in this research seeks to degrade the performance of blockchain servers. The attack process uses FLOOD SYN by executing the command "hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 10.10.10.6", DoS attack runs for 30 seconds with a total network traffic log of 2,372,850 log lines. The graph of DOS attack on the webserver blockchain can be seen in Figure 13.
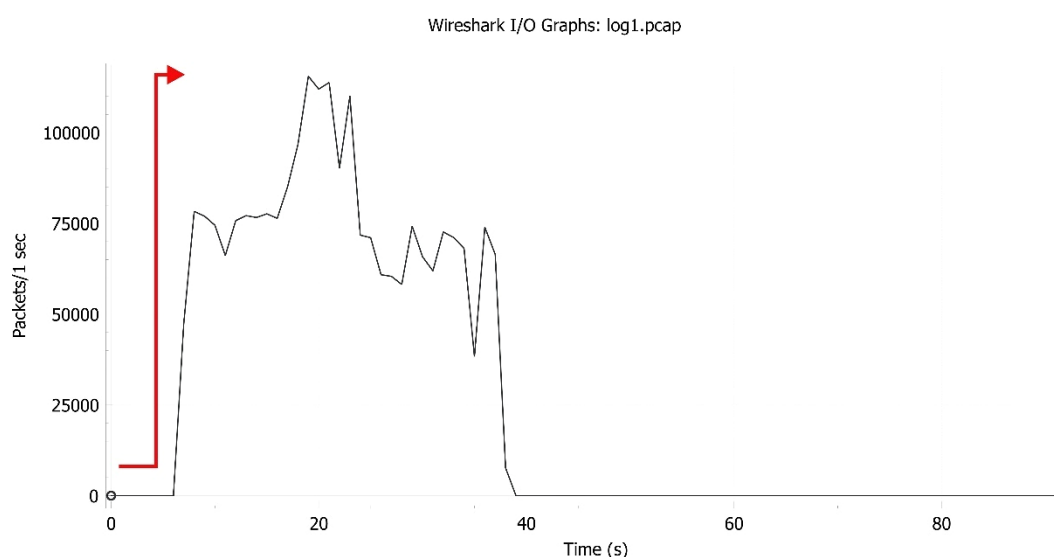

Figure 13. DoS Attack on Blockchain Web Server

Figure 13. It indicates the pattern of network behaviour that is surging due to DoS cyberattacks. The details of DoS attack can be in Figure 14.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
˅ Frame	100.0	2372859	100.0	412870946	35 M	0	0	0
˅ Ethernet	100.0	2372859	8.0	33220026	2883 k	0	0	0
˅ Internet Protocol Version 4	100.0	2372811	11.5	47456220	4119 k	0	0	0
˅ User Datagram Protocol	0.0	1	0.0	8	0	0	0	0
˅ NetBIOS Datagram Service	0.0	1	0.0	216	18	0	0	0
˅ SMB (Server Message Block Protocol)	0.0	1	0.0	134	11	0	0	0
˅ SMB MailSlot Protocol	0.0	1	0.0	25	2	0	0	0
Microsoft Windows Browser Protocol	0.0	1	0.0	48	4	1	48	4
Transmission Control Protocol	100.0	2372806	80.5	332192840	28 M	2372806	332192840	28 M
Internet Control Message Protocol	0.0	4	0.0	256	22	4	256	22
Address Resolution Protocol	0.0	48	0.0	1380	119	48	1380	119

Figure 14. Details of DoS Attack on Blockchain Web Server

Figure 14 provides detailed information from the NETWORK BEHAVIOR DOS attack on the web server. The red box indicates that the network protocol used is the Transmission Control Protocol (TCP). DoS attacks result in decreased web server performance due to full system memory, so the server must restart to operate again.

3-5- Disposition

Blockchain technology and smart contracts successfully built a login authentication system. User login data is saved as a hash on the blockchain through a smart contract. Smart contracts will be signed first by those who already have an Ether ID to run Ethereum-based applications. Hackers cannot change and manipulate the data contained in each block because every participant or connected computer stores all that data.

The testing of this system uses Wireshark tools. One relevant form of implementation applied to support documentation is using the external metadata approach .pcap to store Wireshark files to secure data for the users. This disposition stage indicates that the application of blockchain technology to the login authentication system can secure the user's data. User data becomes accurate because it encrypts and converts it into hash form.

The testing for attack scenarios using multiple scenarios indicates that the Burp Suite's attack scenario has not worked. Burp Suite has not displayed significant data. Unlike SQL Injection, XSS and DoS attack scenarios have a significant impact.

Table 3. Comparison of Security of the Proposed Scheme

Attacks	System Authentication Blockchain (current research)	Kareem et al. [51]	Shajina and Varalaksa [54]	Yang et al. [55]
Password guessing attack	Yes	No	No	No
Prevent replay attack	Yes	No	Yes	Yes
Prevent insider attack	Yes	No	No	Yes
Prefer impersonation attack	Yes	No	Yes	No
Attack with injection code	Yes	Yes	No	No
DoS attack decreases user server performance	Yes	No	No	No
Cyber-attack on login website	Yes	Yes	No	Yes

Table 3 indicates the solutions that guarantee key security requirements. Blockchain authentication is very efficient in its operation.

4- Discussion

After implementing blockchain technology, the data becomes secure as it turns into a hash form to calculate the results of monitoring the number of logs on the blockchain system using Equation 2.

$$E = 100 - \left(\frac{U_{BC} * NoS}{\sqrt{ReDB}} \right) \tag{2}$$

where U_BC is the number of blockchain users, NoS is the average number of blockchain logs. ReDB is the average number of overall logs [54]. The results obtained can be seen in the Table 4.

Table 4. Log List

Block	Log Line (NoS)	Overall Log Row (ReDB)
1	24	3734
2	52	2181
3	24	870
4	26	895
5	22	2153
6	22	658
7	20	715
8	25	731
9	20	709
10	22	741

Table 4 presents log results after using blockchain. NoS has an average number of log lines of 51.4 logs, while ReDB has an average number of overall logs of 2,677 logs. Therefore, the percentage of results obtained as much as 90.1%.

5- Conclusion

The results of this research show that blockchain technology and smart contracts have succeeded in building a login authentication system. The Login authentication system was developed using blockchain technology. The login system uses the PHP programming language to build the system interface, while the blockchain implementation uses the Solidity programming language. Solidity is used to build smart contracts. The Ethereum blockchain-based application must run a smart contract so that it can be signed first by those who already have an Ethereum account to run the application. Meanwhile, the Network Forensic Life Cycle (NFDLC) framework is used because the steps in it are easily implemented and have the integration obtained in the forensic process. A login authentication system using blockchain may secure the data, which is proven by Wireshark testing based on the resulting log data, with a percentage of 90.1% of the test results showing a relatively high level of system security. The test applying multiple attack scenarios makes the Burp Suite attack scenarios unsuccessful. Burp Suite has not been able to display the significant data yet. Several other scenarios, such as SQL Injection, Cross-Site Scripting (XSS), and Denial of Service (DoS) attack scenarios, have a significant impact. This research presents the attack implementation in detail. The scenario uses SQL Injection attacks with repeated incorrect log-in responses. The XSS attack scenario is successful in making the page display on the website an error. Meanwhile, the DoS scenario uses FLOOD SYN, which makes server performance decrease..

6- Declarations

6-1- Author Contributions

Conceptualization, I.R., A.Z.I. and R.S.K.; methodology, I.R.; software, A.Z.I.; validation, I.R., A.Z.I and R.S.K.; formal analysis, I.R.; investigation, R.S.K.; resources, I.R.; data curation, A.Z.I. and R.S.K.; writing—original draft preparation, I.R.; writing—review and editing, A.Z.I. and R.S.K.; visualization, I.R.; supervision, I.R.; project administration, A.Z.I. and R.S.K.; funding acquisition, I.R. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available in article.

6-3- Funding and Acknowledgements

The authors would like to express appreciation and gratitude to Universitas Ahmad Dahlan for funding this research.

6-4- Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Kim, W., Jeong, O.-R., Kim, C., & So, J. (2011). The dark side of the Internet: Attacks, costs and responses. Information Systems, 36(3), 675–705. doi:10.1016/j.is.2010.11.003.

[2] Subekti, Z. M., & Subandri, S. (2020). Implementasi Metode Per Connection Queue Dengan Access User Direct Mac Filtering Pada Jaringan Wireless. INOVTEK Polbeng - Seri Informatika, 5(2), 240. doi:10.35314/isi.v5i2.1472.

[3] Tian, Y., Zheng, N., Chen, X., & Gao, L. (2021). Wasserstein Metric-Based Location Spoofing Attack Detection in WiFi Positioning Systems. Security and Communication Networks, 2021. doi:10.1155/2021/8817569.

[4] Teferi, F., & Nixon, J. S. (2019). A Security Mechanism to Mitigate DDoS Attack on Wireless Local Area Network (WLAN) using MAC with SSID. International Journal of Computer Sciences and Engineering, 7(4), 864–869. doi:10.26438/ijcse/v7i4.864869.

[5] Hidayat, T. N., & Riadi, I. (2021). Optimation Wireless Security IEEE 802.1X using the Extensible Authentication Protocol-Protected Extensible Authentication Protocol (EAP-PEAP). International Journal of Computer Applications, 174(11), 25–30. doi:10.5120/ijca2021920988.

[6] Marques, N., Zúquete, A., & Barraca, J. P. (2019). Integration of the Captive Portal paradigm with the 802.1 X architecture. arXiv preprint arXiv:1908.09927.

[7] Umar, R., Riadi, I., & Kusuma, R. S. (2021). Mitigating sodinokibi ransomware attack on cloud network using software-defined networking (SDN). International Journal of Safety and Security Engineering, 11(3), 239–246. doi:10.18280/ijsse.110304.

[8] Rahardja, U., Harahap, E. P., & Christianto, D. D. (2019). Pengaruh Teknologi Blockchain Terhadap Tingkat Keaslian Ijazah. Technomedia Journal, 4(2), 211–222. doi:10.33050/tmj.v4i2.1107.

[9] Cui, Y., Cui, J., & Hu, J. (2020). A Survey on XSS Attack Detection and Prevention in Web Applications. ACM International Conference Proceeding Series, al, 443–449. doi:10.1145/3383972.3384027.

[10] Vimala, S. T., & Dhas, J. P. M. (2018). SDN based DDoS attack detection system by exploiting ensemble classification for cloud computing. International Journal of Intelligent Engineering and Systems, 11(6), 282–291. doi:10.22266/IJIES2018.1231.28.

[11] El-Sofany, H. F. (2020). A new cybersecurity approach for protecting cloud services against DDoS attacks. International Journal of Intelligent Engineering and Systems, 13(2), 205–215. doi:10.22266/ijies2020.0430.20.

[12] Wang, Z., Yang, L., Wang, Q., Liu, D., Xu, Z., & Liu, S. (2019). ArtChain: Blockchain-enabled platform for art marketplace. Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019, 447–454. doi:10.1109/Blockchain.2019.00068.

[13] Salman, T., Zolanvari, M., Erbad, A., Jain, R., & Samaka, M. (2019). Security services using blockchains: A state of the art survey. IEEE Communications Surveys and Tutorials, 21(1), 858–880. doi:10.1109/COMST.2018.2863956.

[14] Aprialim, F., Adnan, & Paundu, A. W. (2021). Penerapan Blockchain dengan Integrasi Smart Contract pada Sistem Crowdfunding. Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), 5(1), 148–154. doi:10.29207/resti.v5i1.2613.

[15] Noorsanti, R., Yulianton, H., & Hadiono, K. (2018). Blockchain - Teknologi Mata Uang Kripto (Crypto Currency). Proceeding SENDI_U. Available online: https://www.unisbank.ac.id/ojs/index.php/sendi_u/article/view/5999 (accessed on May 2021).

[16] Zhang, R., Xue, R., & Liu, L. (2019). Security and privacy on blockchain. ACM Computing Surveys 52, (3), 1-34. doi:10.1145/3316481.

[17] Fadlil, A., Riadi, I., & Nugrahantoro, A. (2020). Data Security for School Service Top-Up Transactions Based on AES Combination Blockchain Technology. Lontar Komputer : Jurnal Ilmiah Teknologi Informasi, 11(3), 155. doi:10.24843/lkjiti.2020.v11.i03.p04.

[18] Ismanto, L., Ar, H. S., Fajar, A. N., Sfenrianto, & Bachtiar, S. (2019). Blockchain as E-Commerce Platform in Indonesia. Journal of Physics: Conference Series, 1179(1). doi:10.1088/1742-6596/1179/1/012114.

[19] Rizky, A., Kurniawan, S., Gumelar, R. D., Kurniawan, V., & Prakoso, M. B. (2021). Use Of blockchain technology in implementing information system security on education. Journal of Biology Education Sains & Technology, 4(1), 62–70.

[20] Zheng, Y., Li, Y., Wang, Z., Deng, C., Luo, Y., Li, Y., & Ding, J. (2019). Blockchain-based privacy protection unified identity authentication. Proceedings - 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2019, 42–49. doi:10.1109/CyberC.2019.00017.

[21] Alam, A., Zia Ur Rashid, S. M., Abdus Salam, M., & Islam, A. (2018). Towards Blockchain-Based E-voting System. 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET). doi:10.1109/iciset.2018.8745613.

[22] Shorman, S., Allaymoun, M., & Hamid, O. (2019). Developing the E-Commerce Model a Consumer To Consumer Using Blockchain Network Technique. International Journal of Managing Information Technology, 11(02), 55–64. doi:10.5121/ijmit.2019.11204.

[23] Schintler, L. A., & McNeely, C. L. (2022). Encyclopedia of Big Data. Springer Nature Switzerland doi:10.1007/978-3-319-32010-6.

[24] Efanov, D., & Roschin, P. (2018). The all-pervasiveness of the blockchain technology. Procedia Computer Science, 123, 116–121. doi:10.1016/j.procs.2018.01.019.

[25] Abas Sunarya, P., Henderi, Sulistiawati, Khoirunisa, A., & Nursaputri, P. (2020). Blockchain family deed certificate for privacy and data security. 5th International Conference on Informatics and Computing, ICIC 2020. doi:10.1109/ICIC50835.2020.9288528.

[26] Rejeb, A., Sűle, E., & Keogh, J. G. (2018). Exploring new technologies in procurement. Transport & Logistics: The International Journal, 18(45), 76–86.

[27] Milkovic, M., Samardžija, J., & Ognjan, M. (2020). Application of Blockchain Technology in Media Ecology. Medijska Istrazivanja, 26(1), 29–52. doi:10.22572/mi.26.1.2.

[28] Sartipi, F. (2021). Publicizing construction firms by cryptocurrency. Journal of Construction Materials, 2(3), 1–8, doi:10.36756/jcm.v2.3.1.

[29] Rejeb, A., & Rejeb, K. (2020). Blockchain and supply chain sustainability. Logforum, 16(3), 363–372. doi:10.17270/j.log.2020.467.

[30] Choi, S.-Y., & Whinston, A. B. (2000). The Future of the Digital Economy. Handbook on Electronic Commerce, 25–52. doi:10.1007/978-3-642-58327-8_2.

[31] Arse, M., & Dubey, J. (2020). A Survey of Internet of Things node's transactions Secure through Blockchain Technology. International Journal of Computer Applications, 175(25), 33–37. doi:10.5120/ijca2020920796.

[32] Zou, W., Lo, D., Kochhar, P. S., Le, X. B. D., Xia, X., Feng, Y., ... & Xu, B. (2019). Smart contract development: Challenges and opportunities. IEEE Transactions on Software Engineering, 1-20.

[33] Chaniago, N., Sukarno, P., & Wardana, A. A. (2021). Electronic document authenticity verification of diploma and transcript using smart contract on ethereum blockchain. Register: Jurnal Ilmiah Teknologi Sistem Informasi, 7(2), 149–163. doi:10.26594/REGISTER.V7I2.1959.

[34] Bragagnolo, S., Rocha, H., Denker, M., & Ducasse, S. (2018). SmartInspect: Solidity smart contract inspector. In 2018 IEEE 1st International Workshop on Blockchain Oriented Software Engineering, IWBOSE 2018 - Proceedings (Vols. 2018), 9–18. doi:10.1109/IWBOSE.2018.8327566.

[35] Mohanta, B. K., Panda, S. S., & Jena, D. (2018). An Overview of Smart Contract and Use Cases in Blockchain Technology. 2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018, 1–4. doi:10.1109/ICCCNT.2018.8494045.

[36] Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F. Y. (2018). An Overview of Smart Contract: Architecture, Applications, and Future Trends. IEEE Intelligent Vehicles Symposium, Proceedings, 2018-June, 108–113. doi:10.1109/IVS.2018.8500488.

[37] Sai Kiran, K. V. V. N. L., Devisetty, R. N. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques. Procedia Computer Science, 171, 2372–2379. doi:10.1016/j.procs.2020.04.257.

[38] Pallavi, C., Girija, R., & Jayalakshmi, S. L. (2021). An Analysis on Network Security Tools and Systems. SSRN Electronic Journal. doi:10.2139/ssrn.3833455.

[39] Gitanjali Simran T, and Sasikala D (2019). Vulnerability Assessment of Web Applications using Penetration Testing. In International Journal of Recent Technology and Engineering 8(4), 1552–1556. doi:10.35940/ijrte.b2133.118419.

[40] Sikos, L. F. (Ed.). (2019). AI in Cybersecurity. Intelligent Systems Reference Library. Springer Nature Switzerland. doi:10.1007/978-3-319-98842-9.

[41] Patel, K. (2019). A survey on vulnerability assessment penetration testing for secure communication. Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019, 320–325. doi:10.1109/ICOEI.2019.8862767.

[42] Mokbal, F. M. M., Dan, W., Imran, A., Jiuchuan, L., Akhtar, F., & Xiaoxi, W. (2019). MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique. IEEE Access, 7, 100567–100580. doi:10.1109/ACCESS.2019.2927417.

[43] Gupta, S., & Gupta, B. B. (2017). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. International Journal of Systems Assurance Engineering and Management, 8, 512–530. doi:10.1007/s13198-015-0376-0.

[44] Khera, Y., Kumar, D., Sujay, S., & Garg, N. (2019). Analysis and Impact of Vulnerability Assessment and Penetration Testing. Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019, 525–530. doi:10.1109/COMITCon.2019.8862224.

[45] Sarmah, U., Bhattacharyya, D. K., & Kalita, J. K. (2018). A survey of detection methods for XSS attacks. Journal of Network and Computer Applications, 118, 113–143. doi:10.1016/j.jnca.2018.06.004.

[46] Shurman, M. M., Khrais, R. M., & Yateem, A. A. (2019). IoT denial-of-service attack detection and prevention using hybrid IDS. Proceedings - 2019 International Arab Conference on Information Technology, ACIT 2019, 252–254. doi:10.1109/ACIT47987.2019.8991097.

[47] El-Sofany, H. F., El-Seoud, S. A., & Taj-Eddin, I. A. T. F. (2019). A case study of the impact of denial of service attacks in cloud applications. Journal of Communications, 14(2), 153–158. doi:10.12720/jcm.14.2.153-158.

[48] Syed, N. F., Baig, Z., Ibrahim, A., & Valli, C. (2020). Denial of service attack detection through machine learning for the IoT. Journal of Information and Telecommunication, 4(4), 482–503. doi:10.1080/24751839.2020.1767484.

[49] Abushwereb, M., Mustafa, M., Al-Kasassbeh, M., & Qasaimeh, M. (2020). Attack based DoS attack detection using multiple classifier. arXiv preprint arXiv:2001.05707..

[50] Chen, D., Yan, Q., Wu, C., & Zhao, J. (2021). SQL Injection Attack Detection and Prevention Techniques Using Deep Learning. Journal of Physics: Conference Series, 1757(1). doi:10.1088/1742-6596/1757/1/012055.

[51] Kareem, F. Q., Ameen, S. Y., Salih, A. A., Ahmed, D. M., Kak, S. F., Yasin, H. M., ... & Omar, N. (2021). SQL injection attacks prevention system technology. Asian Journal of Research in Computer Science, 13, 32.

[52] Endicott-Popovsky, B. E., & Frincke, D. A. (2006). Embedding forensic capabilities into networks: Addressing inefficiencies in digital forensics investigations. Proceedings of the 2006 IEEE Workshop on Information Assurance, 2006, 133–139. doi:10.1109/iaw.2006.1652087.

[53] Endicott-Popovsky, B., Frincke, D. A., & Taylor, C. A. (2007). A theoretical framework for organizational network forensic readiness. Journal of Computers, 2(3), 1–11. doi:10.4304/jcp.2.3.1-11.

[54] Shajina, A. R., & Varalakshmi, P. (2017). A novel dual authentication protocol (DAP) for multi-owners in cloud computing. Cluster Computing, 20(1), 507–523. doi:10.1007/s10586-017-0774-y.

[55] Yang, X., Chen, Y., & Chen, X. (2019). Effective scheme against 51% attack on proof-of-work blockchain with history weighted information. Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain 2019, 261–265. doi:10.1109/Blockchain.2019.00041.