Computer Systems Science & Engineering
DOI:10.32604/csse.2022.017130
Article

Tech Science Press

Blockchain: Secured Solution for Signature Transfer in Distributed Intrusion Detection System

Shraddha R. Khonde[1,2,*] and Venugopal Ulagamuthalvi[1]

[1]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, 600119, India
[2]Department of Computer Engineering, M.E.S. College of Engineering, S. P. Pune University, Pune, 411001, India
*Corresponding Author: Shraddha R. Khonde. Email: khondeshraddha21@gmail.com
Received: 21 January 2021; Accepted: 08 April 2021

Abstract: Exchange of data in networks necessitates provision of security and confidentiality. Most networks compromised by intruders are those where the exchange of data is at high risk. The main objective of this paper is to present a solution for secure exchange of attack signatures between the nodes of a distributed network. Malicious activities are monitored and detected by the Intrusion Detection System (IDS) that operates with nodes connected to a distributed network. The IDS operates in two phases, where the first phase consists of detection of anomaly attacks using an ensemble of classifiers such as Random forest, Convolutional neural network, and XGBoost along with genetic algorithm to improve the performance of IDS. The novel attacks detected in this phase are converted into signatures and exchanged further through the network using the blockchain framework in the second phase. This phase uses the cryptosystem as part of the blockchain to store data and secure it at a higher level. The blockchain is implemented using the Hyperledger Fabric v1.0 and v2.0, to create a prototype for secure signature transfer. It exchanges signatures in a much more secured manner using the blockchain architecture when implemented with version 2.0 of Hyperledger Fabric. The performance of the proposed blockchain system is evaluated on UNSW NB15 dataset. Blockchain performance has been evaluated in terms of execution time, average latency, throughput and transaction processing time. Experimental evidence of the proposed IDS system demonstrates improved performance with accuracy, detection rate and false alarm rate (FAR) as key parameters used. Accuracy and detection rate increase by 2% and 3% respectively whereas FAR reduces by 1.7%.

Keywords: Blockchain; intrusion detection system; machine learning algorithms; secured communication; ensemble approach

1 Introduction

In current times, data exchange and sharing over the Internet is widespread and hence data security is a major concern. Most private and confidential data being exchanged requires a robust digital infrastructure to protect against attacks by intruders in the network. A secured intrusion detection system (IDS) is useful in

safeguarding the network from an attack. Most organizations, individuals, and businesses use IDS for network security. The IDS is a device or software that helps to monitor the network for abnormal activities. If any malicious activity is detected, an alarm is generated for the administrator. IDS identify malicious activities in two different ways: Signature based (SIDS) and Anomaly based detection (AIDS).

Signature based detection systems store signatures in a database and use them to match patterns with inbound traffic. A successful match is considered as an attack triggering an alarm. In case of no match found, the packet is treated as normal traffic and passed through the network. Anomaly based detection is based on behavioral analysis where normal behavior of all types of network traffic is stored as rules [1]. Behavior of each packet entering the network is verified against the stored rules and if the behavior matches with normal activities, then it is considered as normal traffic. Any deviation in traffic is considered as an attack. Due to missing signatures and novel nature of attacks, AIDS is preferred for traffic analysis in modern networks over SIDS. The efficacy of SIDS towards novel attack detection reduces if an outdated and old dataset is used. On the contrary, AIDS can generate false alarms because of pre-set prediction rules. Slight deviation from normal behavior can result in generating a false alarm for administrators. In the modern era, IDS should offer real-time, cross-platform and pre-host protection services so that network data can be secured from any type of attack [2].

In today's cyber world, we are witnessing an explosion of novel attacks across various platforms. It mandates the need for intelligent and innovative IDS that can collaborate with other systems in the network to exchange and transfer information. Data such as signature database, network traffic resources, data alerts and anomaly attack signatures can be shared among nodes connected at different sites in a distributed environment. However, it requires greater caution when a signature of a novel attack or a signature database is shared in the network. Since security is required for this data, while being transmitted through a distributed network. A potential hazard of providing security to information being exchanged is the risk of an intruder becoming a part of the network thereby having access to all information passing between the nodes. In case of signature transmission, an intruder can delete the signature and provide a malicious or inaccurate signature. Intruders can also harm the files used to save the signatures or signature database, leading to degradation in detection accuracy of the host.

An innovative blockchain technology is majorly used to record activities in a better and safer way. It allows replicating data at large number of nodes to avoid single node bottleneck [3–7]. Blockchain becomes a workable solution for distributed IDS nodes due to its built-in immutability to overcome trust issues and consensus building among various nodes of IDS [8]. Based on the blockchain framework, components of the general framework for alert sharing and consensus building can be used for multimedia and secured networks [9,10]. Blockchain framework is used by many systems and security applications due to its various advantages and improvised performance [11–14]. Internet of Things, Financial aids, IDS and many more applications uses blockchain framework due to its security aspects [15,16]. Blockchain has become an emerging area in the cloud as well as web application development [17].

In this paper, a blockchain based framework for distributed IDS is proposed to transfer novel signatures from one node to another. Anomaly based attack detection technique is used to detect the novel attacks. When a new attack is detected, a signature is created, validated, and then transferred into the network to provide information to other nodes. The transfer of signature is done using blockchain technology in a secured way, so that attackers cannot hamper the newly generated signatures. This signature can be used by other IDS nodes for attack detection. The architecture proposed in this work is the first kind of distributed IDS, employing blockchain technology for signature exchange. This architecture provides various features like distributed network, fault tolerance and maintenance of a single source of truth. In case of failure of a node, data can still be obtained as blockchains are replicated on multiple nodes. An attacker cannot tamper the signature as it is impossible to trace each node and alter the signature in the

blockchain. A unique feature of this architecture, that makes it one of the best solutions for distributed IDS networks, is that only trusted or authenticated nodes can transmit the signature.

2 Backgrounds and Related Work

2.1 Blockchain Technology

Blockchain is a distributed ledger technology allowing exchange of information in a secured manner. It allows storage of data permanently in the form of a block for use by all nodes. Additionally, blockchain does not require any intermediaries like a third party for communication between nodes in a network. In the blockchain, information is stored on specific structures known as blocks. Each block contains several transactions stored per signature. All nodes can add or modify a block and validate each transaction. All blocks are connected with a cryptographic function consisting of hash functions. Every block is attached to blocks on either side with hash functions.

Since blockchain is considered as a shared secure distributed ledger, nodes can read information without any constraints, and if specific constraints are met, a write operation can be performed. Only authorized nodes are permitted to read, update, and add a block. Accessibility to blocks entirely depends on the type of blockchain used.

Depending on the permission granted to a node, various types of blockchains are available. Nodes connected in a blockchain network can add, read, or modify blocks in the existing chain [18]. Deletion operation is not allowed in the blockchain. There are following types of blockchains, namely, *Public Blockchain (Permission-less), Private Blockchain and Consortium Blockchain*. In a *public blockchain*, anyone can join as a participant and can read the blocks and embedded transactions. Nodes can participate as a reader or writer where they can read or add a block in the chain. Examples of public blockchains are Bitcoin, which is used for crypto currency [19] and Zerocash [20]. Ethereum is one of the best examples of public blockchain [21].

Private Blockchain is owned by an organization for their private network. It falls in the category of a permissioned blockchain where an organization has overall access control. Participant nodes are decided by the organization and they can read or write the block according to the permission granted by the organization. In this type of blockchain, intruders cannot tamper any block connected in chain. Hyperledger is an example of a permissioned blockchain [22]. *Consortium Blockchain* is initiated and maintained by multiple organizations rather than a single organization. As multiple organizations are involved, everyone has an shared responsibility to keep the blockchain secure [23]. Nodes in this blockchain are predefined by the organization and permissions are granted according to the requirement as a reader or writer. This blockchain is used to provide security to mobile devices [24].

In a blockchain network, all nodes must follow a consensus protocol. It is analogous to an agreement defining the block structure for data sharing. This protocol maintains valid data in the blockchain and provides a guarantee to the node with respect to uniform structure of the transaction and the block. Various types of consensus protocols are used such as proof-of-work (PoW), proof-of-stack (PoS), proof-of-burn (PoB), proof-of-capacity (PoC) and many more. Additional security to the block is provided by the cryptographic secured hash function. This allows nodes to add new block in the chain using the hash function. To summarize, blockchain is a framework providing a secure, immutable, distributed, and flexible architecture to share data in the network.

2.2 Hybrid Distributed Intrusion Detection System

Blockchain is emerging as a solution for every application where data security is of prime importance and needs continuous improvement. As per surveys, it is observed that only 3% of IDS utilizes blockchain for

malware detection. Most IDS currently available in the market use various classifiers for attack detection. Of these, many IDS make use of supervised, unsupervised, and semi-supervised algorithms to detect attacks. Approaches such as deep learning, data mining and cloud computing are used for signature creation and distribution [25]. An ensemble technique along with feature selection is used to improve the performance of IDS over individual classifiers [26–31]. Hybrid approach is used for attack detection in many IDS [32–34]. All hybrid architectures use various classifiers for signature as well as anomaly attack detection.

It is observed, from the literature survey, that most IDS do not share signatures detected by the anomaly approach. Some hybrid approaches share signatures in the network without considering any security aspects thereby necessitating the need for a new, intelligent and innovative IDS that uses a secured mechanism [35]. In the proposed architecture, IDS with Blockchain (IDSwBC) technology is employed. It uses anomaly detection along with a security mechanism to exchange the signature within the network. Signatures are exchanged with other nodes, so that they can be used for detection of attack by other nodes. Private blockchain framework is used to transfer the signature securely within the distributed network. The proposed system overcomes the limitations of existing IDS. Key highlights of the proposed IDSwBC architecture are given below:

- IDSwBC creates and distributes the signature in a network without tampering the data.
- IDSwBC uses blockchain technology to distribute the signature securely in the network.
- IDSwBC works in a distributed environment and the network is fault tolerant.

The next section provides a detailed illustration of all phases and methodologies used in the implementation of the proposed IDSwBC system.

3 Proposed Methodology

The proposed architecture is a novel approach to share signatures in the distributed environment. It uses an anomaly detection method to improve network security. IDSwBC operates in two phases, namely anomaly detection and signature transfer using blockchain. This model is proposed for the nodes connected in a distributed environment. Nodes will be able to detect an attack by analyzing each packet in the network. Packets are examined for malicious activity by the analysis unit (AU). This unit captures and detects the packet using anomaly detection phase. In this phase, a genetic algorithm is used to create a rule-based dataset. This dataset is created from the modern UNSW NB 15 dataset and used to train all classifiers. Classifiers used in this phase are XGBoost, Random Forest (RF) and Convolutional neural network (CNN). All these classifiers are grouped together to get a final predictive analysis using a majority voting algorithm. If the packet is analyzed as an attack, then it is directly rejected by the node, otherwise forwarded to the second phase. Fig. 1 presents the general architecture of the proposed distributed IDS.

Upon attack detection, the signature creation and transfer phase enables transfer of signatures securely from one node to another. The affected node is responsible for creating a signature of the detected novel attack. Once the signature is created, it is encrypted and added as a block in the blockchain. All nodes receive the signature and update their dataset, so that a similar attack can be identified within the network in the future. This reduces the detection and processing time for other nodes. The details of each phase are explained in the next section.

3.1 Phase 1: Anomaly Detection

Anomaly detection phase utilizes behavior-based approach of intrusion detection. In this approach, the classifiers are trained using rules that define the behavior of the packet and network. Genetic algorithm is used to establish the normal behavior patterns of the data entering the system. To find the rules; the

genetic algorithm works in a four-step process such as initial population generation, chromosome designing, fitness value calculation and genetic operator designing. Rule-based dataset is generated as an output of process. The standard dataset UNSW NB15 is used as input for this process. This dataset is used by the algorithm to find the optimal features that ultimately define the final rules. Fitness value is calculated for each feature set and those having a strong fitness value are considered for rule-based dataset generation. Eq. (1) is used to calculate the fitness value. Threshold considered for the fitness value is 0.90. All chromosomes with a value nearer to the threshold are considered for dataset generation.

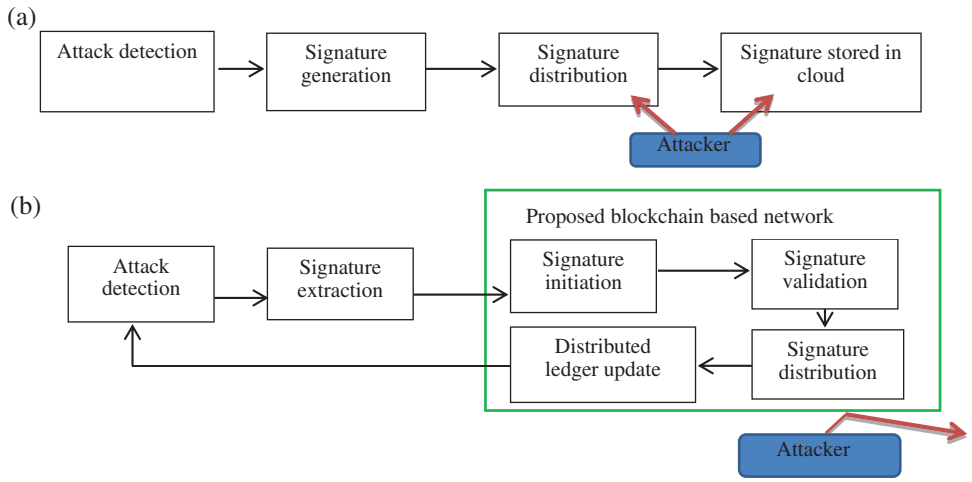

Figure 1: System architecture of IDSwBC using blockchain (a) Existing intrusion detection systems (b) Proposed distributed intrusion detection system

$$Fitness\ value = (a/A) - (b/B) \tag{1}$$

where, A: Total number of attacks.

a: Number of attack connections correctly classified by the individual classifier.

B: Normal connections in the population.

b: Number of normal connections correctly classified by classifiers.

Fig. 2 depicts the genetic algorithm initiating with the formulation of population and chromosome and subsequently evaluating each of them with a crossover and mutation operator to get the most important and specific chromosome. This individual chromosome is then converted into rules to create a dataset that further trains the classifiers to detect malicious activity in Phase 1.

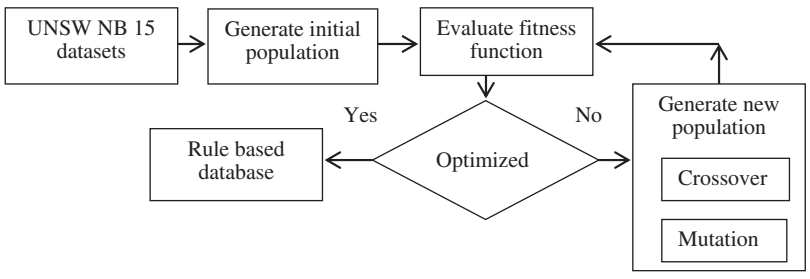

Figure 2: Rule-based dataset generation

The population size, crossover folds and mutation size can vary in case of benchmark datasets. The primary objective of this process is to get specified and accurate rules defining the normal behavior of data flowing through the network. Rule-based dataset is used to train classifiers like CNN, RF, and XG-boost. From the literature review, it is observed that all these classifiers show improved performance in a distributed environment. Classifiers are used in an ensemble approach to avoid biased prediction results by an individual classifier and improve the performance of IDS. If an attack is identified, an alert is generated and the packet is transferred to phase 2 that converts the packet into a signature to distribute among other nodes. In case of a normal data packet, it is allowed to transmit through the network. System architecture for phase 2 is presented in Fig. 3.

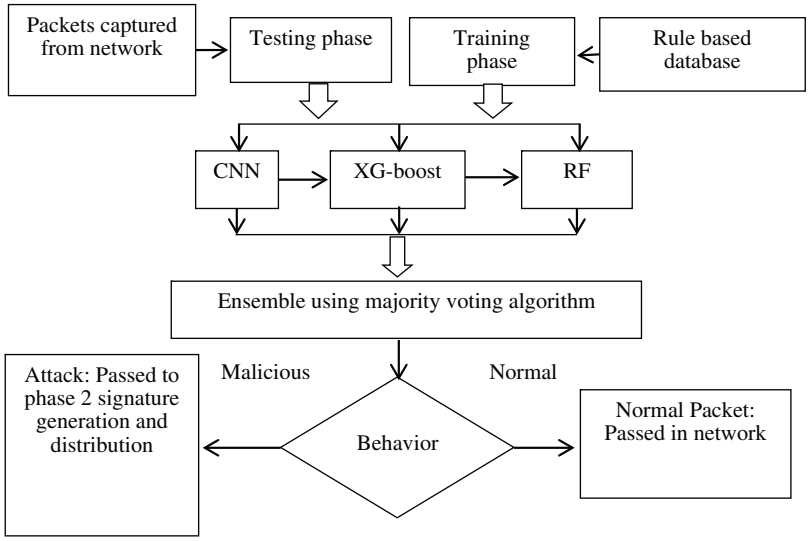

Figure 3: Architecture model of phase 2 for IDSwBC

3.2 Phase 2: Signature Extraction and Distribution Using Blockchain Framework

In IDSwBC, secure distribution of signatures through the network is the primary objective of phase 2. Input packets received in this phase have already been predicted and classified as an attack by the anomaly detection phase [36]. This phase constitutes three sub-activities namely, signature creation for the received packet, signature upload and verification and signature distribution. Fig. 4 presents individual node structure used in blockchain network.

All nodes in the network are connected in a distributed manner, following one of the consensus protocols. Blockchain can be implemented on various platforms such as Ethereum [37] and Hyperledger [38]. As per the requirement of the IDSwBC, a private and permissioned blockchain is developed with the help of Hyperledger, utilizing the consensus protocol as a proof of stack (PoS). Hyperledger is a platform, used to build customized applications on the permissioned blockchain. Permissioned blockchain consists of authorized nodes exclusively. Each node is responsible for extracting the signature, creating a block and distributing among all authorized nodes within the network. IDSwBC consists of two types of nodes, namely Initiator node and Validator node. Initiator nodes are responsible for signature creation whereas validator nodes are tasked with validating the signature and converting it into a block for distribution within the network. The validator node can also double up as an initiator node.

Each node consists of an analysis unit (AU) and a distribution unit (DU). The analysis unit evaluates all packets entering the node through the network. This unit uses anomaly detection to identify malicious activities. It also helps to update the dataset with new signatures that were considered as attacks in the

anomaly detection phase. Along with the analysis and distribution unit, each node consists of the blockchain complete ledger. The structure used to carry the data is referred to as a block. The transactions are signatures extracted from the packet. Signature extraction, block upload, and signature distribution are explained in the next section.

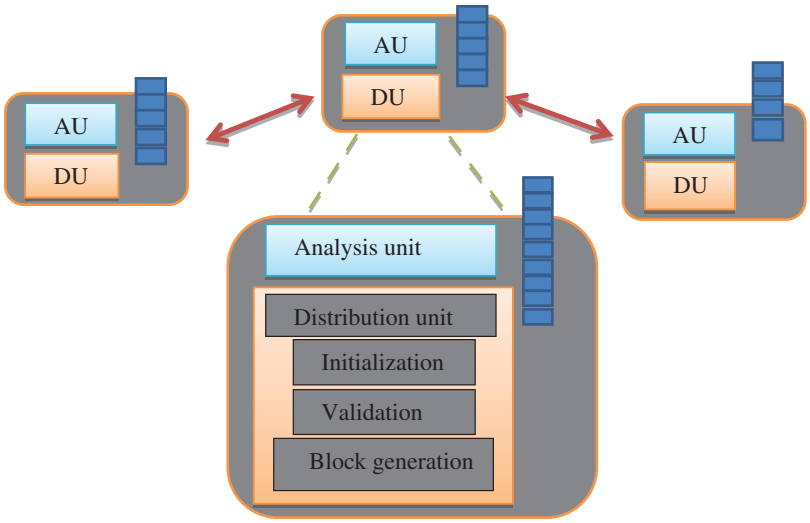

Figure 4: Blockchain network of IDSwBC

3.2.1 Signature Extraction

Signature extraction is performed on the packets received as an input from the anomaly detection phase. Features are selected and extracted from the packet to create a signature as per the format of the UNSW NB15 dataset. UNSW NB15 dataset uses 49 features for the detection of various types of attacks. A script is written which takes the packet as an input and generates its signature equivalent to the features of the UNSW NB15 dataset. The standard format for signature creation is as follows;

{MAC address, IP address, Private Key, Type, Features}

where, MAC address –Address of the node responsible for signature extraction.

IP address – Node address which is responsible for signature extraction.

Private Key – Key of the node responsible for signature extraction from a pair of Public-private keys.

Type – Type of attack whose signature is extracted by the node.

Features – UNSW NB15 dataset features extracted from packet.

Having extracted the features from the packets, the initiator node executes the signature creation Algorithm 1 to convert the extracted features into the prescribed signature format. All initiator nodes follow smart contracts to create the signatures in the required format. Smart contracts are a set of rules prescribed and stored in the system, used by blockchain whenever a certain action is performed. The created signature is encrypted by a private key of the initiator node and sent to the validator node.

3.2.2 Signature Validation

Signature validation is carried out by the validator node and is an important step in this phase to confirm all signatures are created by initiator nodes. The validator node's responsibility is to check the validity, authorization, and significance of the signature before adding to the blockchain.

Algorithm 1: Signature creation

1.Procedure: Convert features extracted from packets into signature according to standard format
2.Inputs: Features retrieved from packets (F_{var}) and values of Features retrieved (F_{value})
3.Output: Signature in prescribed format
4.Mandatory features: (protocol, source IP, source port, dest. IP, dest. Port, type of service, duration)
5. Read F_{var}
6.If any mandatory features missing:
7. return error
8. exit
9.else:
10. for each F_{var}
11. Read F_{value} for F_{var}
12. Store F_{value} into equivalent feature from standard format
13. Ignore feature from standard format if F_{value} not available
14. End for loop
15. End if
16.Insert default F_{value} for all features whose value not extracted
17.End procedure

The validation process is followed as a smart contract by each initiator and validator node before confirming the addition of signature into the blockchain. Once the signature is validated, the node initiates the process of block creation. Algorithm 2 is used by the validator node to complete the validation process. A signature is verified by the validator if all conditions are satisfied. Otherwise, it is refused by the validator leading to a signature drop. If the same signature is created by another initiator node, then it will be ignored by the validator. The validated signature is further considered for block creation.

Algorithm 2: Signature validation

1.Procedure: Verification (Signature, MAC address, IP address)
2.Inputs: Signature in standard format, MAC address of the Initiator node, IP address of the Initiator node
3.Output: Validated / Refused
4.If (Signature is in standard format) and (IP is valid IP) and (MAC is valid MAC) and (public key verifies private key of Initiator):
5. Return Signature Validated
6. Push Signature for block creation
7.Elseif: Signature already present
8. Ignore the Signature
9.Else:
10. Return Signature Refused
11. Drop Signature
12.End if
13.End Procedure

3.2.3 Creation of Block for Signature

After validation of the signature, a block is created and added according to the Hyperledger format. According to the system architecture, if any novel attack is detected by the node, it would be converted into a signature. This signature can be used by other nodes for future use. The block structure used in IDSwBC is as shown in Fig. 5.

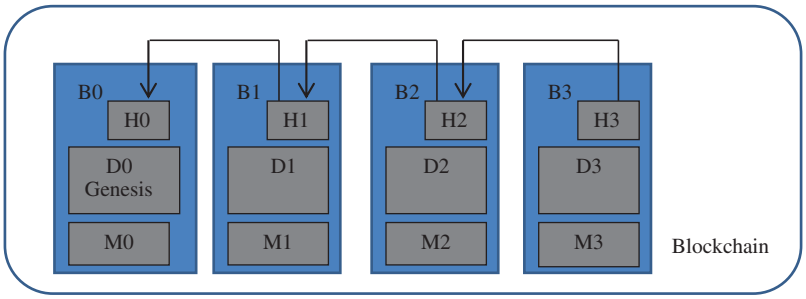

Figure 5: Structure of each block connected in Hyperledger blockchain

As shown in Fig. 5, the block is divided into three sections as header, data, and metadata. An explanation of all sections is given below.

a. Header: It provides following information about the block.

- *Block number*: It is an identification assigned to each newly created block by the validator node. This is used to access a block in the blockchain during future communication.
- *Previous block hash value*: 256 bits previous block hash value computed using SHA256.
- *Current block hash value*: 256 bits current block hash value in hexadecimal format.

In Hyperledger, a chain of blocks is created to connect each block to the next and previous block. Blocks are connected using a common link called block hash value. Secure Hashing Algorithm (SHA256) is used as the default cryptographic algorithm. It is a successor of SHA-1 and SHA-2 hash algorithms. It generates a unique 256 bit (32 bytes) hash value for each block. Hash values are represented in hexadecimal format. SHA256 algorithm has not been compromised in any manner till date. This is the main reason of using SHA256 as a default hash algorithm in Hyperledger.

b. Data: This section consists of an actual signature created in the standard format.

c. Metadata: It contains information about the block like timestamp, consensus protocols, and private key of the initiator, validator and signature details.

3.2.4 Signature Distribution

Upon creation, a block is distributed and added to the ledger. The validator node adds the block to the existing blockchain. This information is broadcasted in the network to all nodes. Once the validated block is received, the individual ledger gets updated. The updated ledger can be used for further processing. The block is permanently attached and the blockchain is committed once the operation for each node is finished. In the IDSwBC, nodes save a new block in the ledger and update the signature in the UNSW NB15 dataset. Updating the dataset helps nodes for further analysis of packets entering the network.

Blockchain provides several features to the IDSwBC such as *Immutability*, *Decentralization, Enhanced security, Distributed ledgers*, *Consensus* and *Fault tolerance*. The next sections explain the experiments, results, and discussion of the IDSwBC.

4 Results and Analysis

In this section, the performance of IDSwBC is elaborated. The IDSwBC is implemented using Hyperledger Fabric versions 1.0 and 2.0. The performance parameters used for evaluation are execution time, average latency, throughput, and transaction processing time. System performance is evaluated in two parts: the performance of blockchain and IDS performance with and without blockchain. Based on the experiments conducted, Hyperledger Fabric version 2.0 provides better results compared to 1.0. The accuracy and detection rate of IDS improves with the help of blockchain.

4.1 Performance Evaluation of Blockchain

Many approaches are presented to evaluate the performance of blockchain [39,40]. The performance of blockchain in IDSwBC is evaluated according to the number of nodes in the network and the execution time. Parameters used to evaluate the performance of the initiator and validator nodes are given below.

a. Evaluation of Execution Time

The execution time is evaluated for the two platforms by varying the frequency of transactions in the network [41]. Execution time is analyzed for different functions, such as the time required for executing a simple query, the initiation and validation process. Query processing execution time increases with increase in the number of transactions. Execution time taken by Fabric v2.0 is less than Fabric v1.0 for the simple query function, as shown in Fig. 6a. Figs. 6b–6c demonstrates the execution time required for the initiation and validation process for both implementations as Fabric v1.0 and v2.0. As the dataset size increases, the execution difference between both versions keeps on increasing. When the number of transactions increases, it is found that the execution time for initiation and validation function is less in Fabric v2.0. With a small dataset, the execution time for both versions is similar.

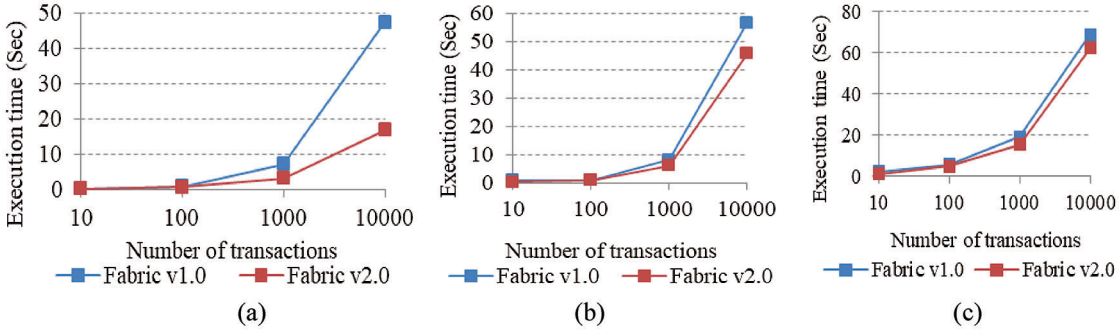

Figure 6: Execution time (a) query (b) initialization process (c) validation process

b. Evaluation of Average Latency

The average latency is evaluated with different number of transactions for both versions. Latency average is calculated for the execution of a simple query, initialization and validation process. The comparison of the average latency values between both versions is shown in Figs. 7a–7c. It is observed that as the number of transactions increases, the latency time taken by v1.0 is more as compared to v2.0. The average latency considered in the proposed system is same as the block time.

c. Evaluation of Throughput

The throughput of the system is the number of transactions executed by the node per unit time. The average throughput of a network depends on the number of nodes available in the network. Figs. 8a and 8b demonstrates the throughput of the system with different block sizes and the number of nodes in the

network, respectively. Throughput depends on a number of features, such as the block size allowed in the blockchain or the number of nodes in the network. As the number of nodes (initiator and validator) increases, the throughput of the system also increases.

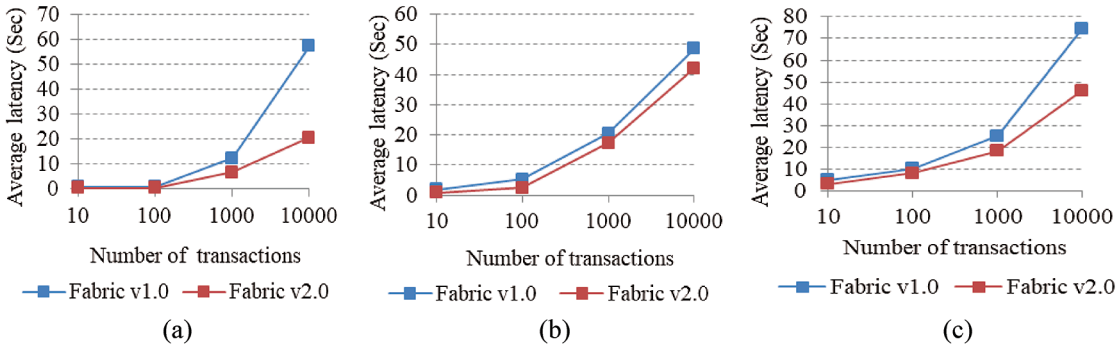

Figure 7: Average latency (a) query (b) initialization process (c) validation process

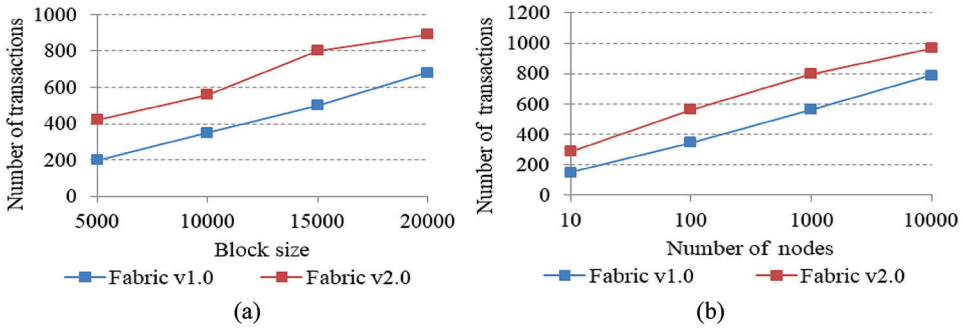

Figure 8: Throughput (a) different block size (b) different number of nodes

d. Evaluation of Transaction Processing Time

The transaction processing time is the time taken from the point of transaction initiation to the validation, completion, and addition of blocks in the blockchain. In most evaluation parameters, the transaction processing time is the same as latency, the time taken by the network to add a block in a blockchain [42]. Whereas we assumed that the latency and time required to add a block in the blockchain is included in the transaction processing time. Figs. 9a and 9b shows the transaction processing time in consideration of different block sizes and number of nodes in the network, respectively. By increasing the degree of parallelism, the transaction processing time can be improved. Average transaction processing time is calculated by the execution time required for each transaction and the total transaction processing time divided by the number of transactions.

4.2 Performance Evaluation of IDSwBC

The IDSwBC performance is checked based on accuracy, detection rate, and false alarm rate parameters. IDSwBC uses ensemble technique and genetic algorithm in two phases to improve the accuracy and detection rate with reduced false alarm rate. The performance is evaluated with and without blockchain. It is observed that the detection phase shows improvement in detection rate and accuracy, when used along with blockchain compared to without blockchain. Tab. 1 shows the performance of each phase of IDSwBC with and without blockchain.

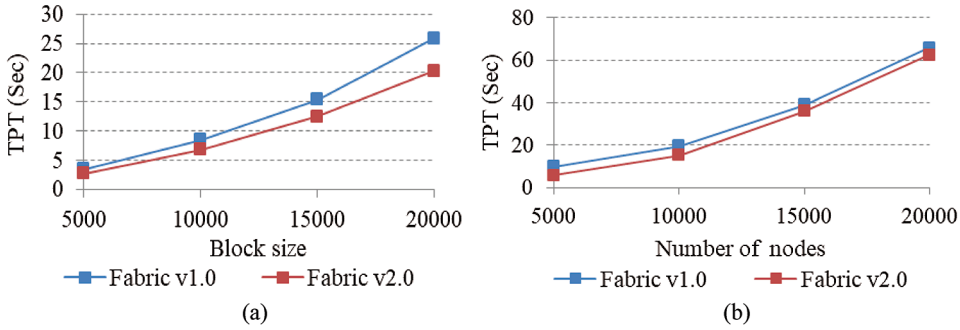

Figure 9: Transaction processing time (a) different block size (b) different number of nodes

Table 1: Performance of IDS with and without blockchain

Performance parameters	IDS without blockchain	IDS with blockchain (IDSwBC)
Accuracy	95.7%	97.8%
Detection Rate	95.5%	98.2%
False alarm rate	3.2%	1.5%

From Tab. 1, it can be observed that Phase 1 performance is increased due to signature update in the dataset using blockchain. IDSwBC shows an improved accuracy and detection rate. Accuracy increased by approximately 2%. We observed an increase in detection rate approximately by 3% and the false alarm rate dropped by 1.7%. Fig. 10 shows the comparison of performance parameters such as accuracy, detection rate, and false alarm rate. It is observed that blockchain technology helps in improving the performance of conventional IDS. Significant improvement in signature detection is observed when used along with blockchain in terms of accuracy and detection rate. Updating the dataset using blockchain tends towards the reduction of false alarms generated by system. Blockchain improves the overall performance of IDS compared to the conventional systems.

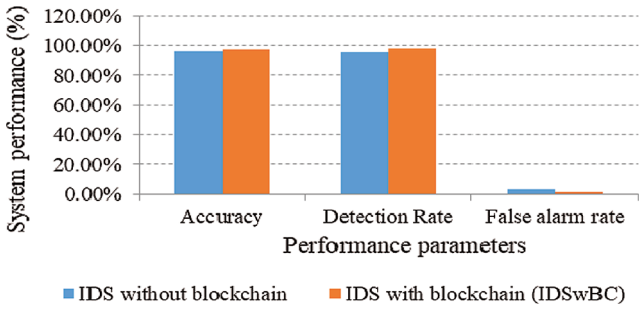

Figure 10: IDS performance with and without blockchain (IDSwBC)

5 Conclusion and Future Works

A novel IDS system is presented in this paper as IDSwBC, which is one of the first Intrusion Detection System implemented using blockchain framework. The system is a unique solution to exchange the signature securely within the network. IDSwBC is implemented and executed in two phases, such as anomaly detection and signature creation with distribution. Both phases work together as a single system to

provide security. Blockchain performance is evaluated on execution time, average latency, throughput and transaction processing time. According to the results presented, Hyperledger Fabric v2.0 provides improved performance compared to v1.0. IDSwBC is also evaluated with parameters accuracy, detection rate, and false alarm rate. IDSwBC provides improvised performance if used along with blockchain compared to without blockchain. Use of blockchain drastically increases the performance of signature detection. IDSwBC provides an improvement in accuracy by 2%, 3% in detection rate and 1.7% reduction in false alarm rate. IDSwBC is a blockchain based unique and innovative IDS, which improvises the performance of conventional IDS systems. However, the bandwidth overhead and processing time can become a limitation of this system. In the future, IDS can be implemented using public blockchains so that more common users can use it to secure data from intruders.

Acknowledgement: The authors are thankful to the Dr. S.H. Gawande of M.E.S. College of Engineering, Pune, India for continuous support and encouragement.

Funding Statement: This work is not supported fully or partially by any funding organization or agency.

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. Ahmed, A. N. Mahmood and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, no. 1, pp. 19–31, 2016.

[2] J. Manan, A. Ahmed, I. Ullah, L. M. Boulahia and D. Gaiti, "Distributed intrusion detection scheme for next generation networks," *Journal of Network and Computer Applications*, vol. 147, pp. 102422, 2019.

[3] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi and K. R. Choo, "A systematic literature review of blockchain cyber security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 147–156, 2020.

[4] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang *et al.,* "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.

[5] T. Dinh, R. Liu, M. Zhang, G. Chen, B. Ooi *et al.,* "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[6] M. Miraz and M. Ali, "Applications of blockchain technology beyond crypto currency," *Annals of Emerging Technologies in Computing*, vol. 2, no. 1, pp. 1–6, 2018.

[7] D. Berdik, S. Otoum, N. Schmidt, D. Porter and Y. Jararweh, "A survey on blockchain for information systems management and security," *Information Processing & Management*, vol. 58, no. 1, pp. 102397, 2021.

[8] W. Meng, E. Tischhauser, Q. Wang, Y. Wang and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, no. 1, pp. 10179–10188, 2018.

[9] S. Aggarwal and N. Kumar, "Core components of blockchain," *Advances in Computers*, vol. 121, pp. 193–209, 2021.

[10] M. Jan, J. Cai, X. Gao, F. Khan, S. Mastorakis *et al.,* "Security and blockchain convergence with Internet of Multimedia Things: Current trends, research challenges and future directions," *Journal of Network and Computer Applications*, vol. 175, pp. 102918, 2021.

[11] A. Ramachandran and M. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," in *arXiv preprint arXiv*: 1709.10000, 2017.

[12] K. Toyoda, P. Mathiopoulos, I. Sasase and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.

[13] B. T. Rao, V. L. Narayana, V. Pavani and P. Anusha, "Use of blockchain in malicious activity detection for improving security," *International Journal of Advanced Science and Technology*, vol. 29, no. 03, pp. 9135–9146, 2020.

[14] A. R. Mathew, "Cyber security through blockchain technology," *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 3821–3824, 2019.

[15] J. Sengupta, S. Ruj and S. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT," *Journal of Network and Computer Applications*, vol. 149, no. 6, pp. 102481, 2020.

[16] C. Liang, B. Shanmugam, S. Azam, A. Karim, A. Islam *et al.,* "Intrusion detection system for the internet of things based on blockchain and multi-agent systems," *Electronics*, vol. 9, pp. 1–27, 2020.

[17] N. Agarwal and S. Hussain, "A closer look at intrusion detection system for web applications," *Security and Communication Networks*, vol. 2018, no. 2, pp. 1–27, 2018.

[18] K. Wüst and A. Gervais, "Do you need a blockchain?," in *Proc. Crypto Valley Conf. on Blockchain Technology (CVCBT)*, Zug, pp. 45–54, 2018.

[19] S. Nakamoto, "BitCoin: A peer-to-peer electronic cash system," 2008. [Online]. Available at: http://bitcoin.org/bitcoin.pdf.

[20] E. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers *et al.,* "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. IEEE Sym. on Security and Privacy*, Berkeley, CA, USA, pp. 459–474, 2014.

[21] G. Wood, "Ethereum: A secure decentralized generalized transaction ledger," document EIP-150 Revision, 2016.

[22] Linux Foundation, "Hyperledger blockchain for business," Accessed: Oct. 1, 2017. [Online]. Available at: https://www.hyperledger.org.

[23] Y. Chen, S. Chen, J. Liang, L. Feagan, W. Han *et al.,* "Decentralized data access control over consortium blockchains," *Information Systems*, vol. 94, no. 3, pp. 101590, 2020.

[24] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang *et al.,* "Consortium blockchain-based malware detection in mobile devices," *IEEE Access*, vol. 6, pp. 12118–12128, 2018.

[25] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-nemrat *et al.,* "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 14525–41550, 2019.

[26] A. Aburomman and M. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, no. 6, pp. 135–152, 2017.

[27] Q. Qassim, A. Zin and M. Aziz, "Anomalies classification approach for network—based intrusion detection system," *International Journal of Network Security*, vol. 18, no. 6, pp. 1159–1172, 2016.

[28] P. Tao, Z. Sun and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624–13631, 2018.

[29] Y. Xiao, C. Xing, T. Zhang and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.

[30] X. Li, W. Chen, Q. Zhang and L. Wu, "Building auto-encoder intrusion detection system based on random forest feature selection," *Computers & Security*, vol. 95, no. 1, pp. 101851, 2020.

[31] Y. Zhou, G. Cheng, S. Jiang and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, no. 8, pp. 107247, 2020.

[32] C. Guo, Y. Ping, N. Liu and S. S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[33] S. R. Khonde and V. Ulagamuthalvi, "Hybrid architecture for distributed intrusion detection system," *Ingenierie des Systemes d'Information*, vol. 24, no. 1, pp. 19–28, 2019.

[34] K. Wu, Z. Chen and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.

[35] Z. A. Baig, S. M. Sait and A. R. Shaheen, "GMDH-based networks for intelligent intrusion detection," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1731–1740, 2013.

[36] M. Signorini, M. Pontecorvi, W. Kanoun and R. Pietro, "BAD: A blockchain anomaly detection solution," *IEEE Access*, vol. 8, pp. 173481–173490, 2020.

[37] X. Wang, J. He, Z. Xie, G. Zhao and S. Cheung, "Contractguard: Defend ethereum smart contracts with embedded intrusion detection," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 314–328, 2020.

[38] P. Gaži, A. Kiayias and D. Zindros, "Proof-of-stake sidechains," in *2019 IEEE Sym. on Security and Privacy (SP)*, pp. 139–156, 2019.

[39] C. Fan, S. Ghaemi, H. Khazaei and P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," *IEEE Access*, vol. 8, pp. 126927–126950, 2020.

[40] S. Smetanin, A. Ometov, M. Komarov, P. Masek and Y. Koucheryavy, "Blockchain evaluation approaches: State-of-the-art and future perspective," *Sensors*, vol. 20, no. 12, pp. 3358, 2020.

[41] S. Pongnumkul, C. Siripanpornchana and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," in *2017 26th Int. Conf. on Computer Communications and Networks, ICCCN*, pp. 1–6, 2017.

[42] Q. Nasir, I. Qasse, M. Talib and A. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 2018, no. 1, pp. 1–14, 2018.