Indonesian Journal of Electrical Engineering and Computer Science
Vol. 22, No. 1, April 2021, pp. 485~496
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v22.i1.pp485-496

Designing consensus algorithm for collaborative signature-based intrusion detection system

Eko Arip Winanto[1], Mohd Yazid Idris[2], Deris Stiawan[3] ,Mohammad Sulkhan Nurfatih[4]
[1,2,4]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Malaysia
[3]Department of Computer Science, Universitas Sriwijaya, Indonesia

Article Info

Article history:

Received Mar 5, 2020
Revised Dec 5, 2020
Accepted Jan 11, 2021

Keywords:

Blockchain
CIDS
Consensus algorithm
IDS
Signature-based

ABSTRACT

Signature-based collaborative intrusion detection system (CIDS) is highly depends on the reliability of nodes to provide IDS attack signatures. Each node in the network is responsible to provide new attack signature to be shared with other node. There are two problems exist in CIDS highlighted in this paper, first is to provide data consistency and second is to maintain trust among the nodes while sharing the attack signatures. Recently, researcher find that blockchain has a great potential to solve those problems. Consensus algorithm in blockchain is able to increase trusts among the node and allows data to be inserted from a single source of truth. In this paper, we are investigating three blockchain consensus algorithms: proof of work (PoW), proof of stake (PoS), and hybrid PoW-PoS chain-based consensus algorithm which are possibly to be implemented in CIDS. Finally, we design an extension of hybrid PoW-PoS chain-based consensus algorithm to fulfill the requirement. This extension we name it as proof of attack signature (PoAS).

Corresponding Author:

Mohd Yazid Idris,
School of Computing, Faculty of Engineering
Universiti Teknologi Malaysia
Email: yazid@utm.my

Deris Stiawan
School of Computing, Faculty of Engineering
Universiti Teknologi Malaysia
Email: deris@unsri.ac.id

1. INTRODUCTION

Collaborative intrusion detection system (CIDS) has been designed to enhance the detection capability of IDS. CIDS allows IDS nodes to collect and exchange required information among the nodes [1]. Collecting traffic characteristics and attack signatures from different nodes create more sensitive detection capability compared to a single node IDS. CIDS framework is widely adopted and deployed in various organizations due to its detection capability. However, there are two major issues remain in CIDS which are consistency on data sharing and trust among the nodes [2-3].

Maintaining data consistency is a big challenge in CIDS [4]. Nodes are required to identify in which peer to collect the latest update of attack signatures. In some cases, several nodes provide different version of attack signatures that can lead the data inconsistency issue. This problem occurs due to lack of data versioning control such as in a centralized IDS system.

The second issue in CIDS is to manage trust among the nodes [5]. Trust management is important to prevent false or malicious data to be used in CIDS. Research in trust management has been discussed in [6-9]. In order to maintain the trust, all the data contributes by each node shall be validated by the other nodes. All these nodes can contribute data in a network where each node is considered benign. In actual case each node is exposed to malicious activities such as insider attacks. This attack can cause CIDS getting an invalid data and greatly degrade the security.

The same issues of maintaining data consistency and managing trust have been discussed in blockchain technology [9-10] due to its nature to prevent cryptocurrency transaction from being manipulated in distributed network. In CIDS, researchers are doing research to identify how blockchain can improve security in CIDS. One of the features exists in blockchain dealing with transaction data consistency and trust is a blockchain consensus mechanism. This consensus mechanism allows peers to validate transaction through its consensus proofing algorithm [11-13]. The algorithm requires nodes to compete to be elected as a trusted node for updating transaction in blockchain. Furthermore this algorithm can provide data consistency by having a single source of data update at one time.

Motivations: The promising solution of consensus mechanism has motivate us to extent the blockchain consensus algorithm for CIDS. Recent related research can be found in [14-18]. In [14-15] authors had discussed the important of blockchain in CIDS. On [16] authors introduced a framework to utilize blockchain to improve the performance of anomaly detection via consensus mechanism. Meanwhile in [17-18] authors had designed a collaborative blockchain signature-based IDS framework, which can incrementally establish a trusted signature database in a collaborative environment.

Contributions: Motivated by the recent achievement of blockchain and CIDS research. We have raised a concern to design an extension of consensus algorithm in CIDS. The extension means to provide trusted nodes and attack signature consistency to be stored in blockchain. The contributions of our work can be summarized:

– We investigate three blockchain consensus algorithms and design an extension to validate attack signature, maintaining data consistency and increase trust between nodes based on hybrid PoW-PoS Chain based consensus algorithm. This extension provides CIDS signature-based related definition of stakes and the measurement of node's credibility to take care of network from malicious activities.

This paper is organized in 6 sections. Section 2 are background and related work CIDS and blockchain research. In section 3, we introduce the blockchain consensus algorithm. Section 4 is designing proof of attack signature (PoAS) consensus algorithm and describe element and process of PoAS. For section 5 is our initial experiment and result section 6 we conclude our findings and suggest for future works.

2. BACKGROUND AND RELATED WORK

Collaborative intrusion detection systems **(**CIDS) are deployed in network based on three types of architectural design including centralized CIDS, decentralized CIDS and distributed CIDS such as in Figure 1. Centralized CIDS is depending on a single control point and require less effort to deploy [19]. Meanwhile the decentralized CIDS disperse a single control point to multiple interconnected control point [20-21] to avoid dependencies on single server. The third category of CIDS is distributed CIDS with no control point deployed. The detection tasks are distributed among nodes to request or receive data from peers [22-23]. Distributed CIDS had several identified issues including the data consistency and trust among the peers. Data are moving from one node to another node with peer to peer (P2P) communication. In distributed CIDS, each node can hold different set of data. The data distribute across the network with without a proper control.

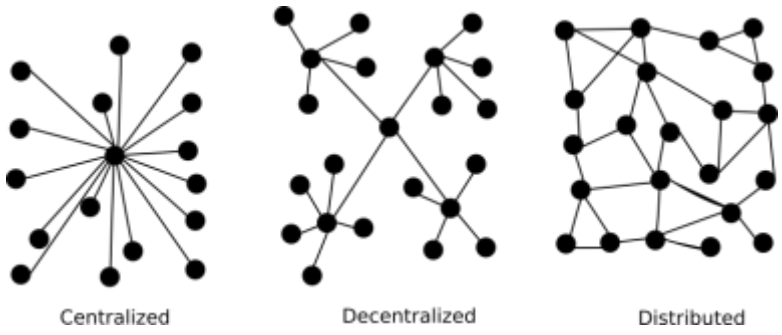

Figure 1. Overview of centralized, decentralized and distributed CIDS architectures

Distributed CIDS improves performance and eliminate a single/multiple point of failures in CIDS. Regarding the mitigation of data consistency and trust issues within distributed CIDS, various methods of trust management have been developed among CIDS nodes. In [9] suggested that the IDS nodes continuously monitor the behavior of their CIDS peers and evaluate the quality of the security-related information that they

share to estimate their credibility. Any data contributed by a CIDS node is taken into consideration depending on the nodes calculated credibility.

Apart from measuring the credibility of CIDS nodes, calculating and maintaining the trustworthiness of external IP sources have also been proposed [3]. By filtering their incoming packets according to a collaborative trust-based scheme, large-scale DDoS attacks can be mitigated. The packet filtering mechanism is relied on blacklisted (i.e. untrusted) IP sources table whose packets are immediately dropped, without further inspection and analysis, thus reducing the workload of detection units. The challenge in the case of distributed CIDS is that there is no central trusted authority to support the establishment of coordination between the peers [13]. In the sequel, we address this challenge by proposing a blockchain solution, to secure the credibility of the CIDS data.

Blockchain intrusion detection system: Blockchain was designed to protect a public distributed ledger of cryptocurrency in a trustless P2P network [24]. This distributed ledger located in a series of blocks and needs to be secured by deploying a hashing mechanism to provide an immutable ledger transaction history. Moreover, a blockchain through its consensus mechanism allows public nodes in distributed network to validate transaction. All nodes have a right to compete to push the transaction data to a blockchain block. This process requires blockchain network implementing a consensus strategy. This strategy is to avoid the same transaction updated by many nodes (avoiding data inconsistency), and ensuring data is valid and update by a trusted node through consensus.

Since the early stage of distributed CIDS research, data consistency and trust issues in distributed collaborative network get a lot of intentions. When blockchain technology has been emerged, the solution to fill the required security gap in distributed CIDS is quite promising. The first proposed CIDS system with blockchain technology can be found in [14], authors described a framework to demonstrate how blockchain can be implemented in CIDS. The author states intersection of CIDSs and blockchains. Particularly, it introduces the idea of utilizing blockchain technologies as a mechanism for improving CIDS. They argue that certain properties of blockchains can be of significant benefit for CIDS; namely for the improvement of trust between monitors, and for providing accountability and consensus. They show the related work and highlight the research gaps and challenges towards such a task. Finally, the author proposes a generic architecture for the incorporation of blockchains into the field of CIDS. On other hand [15] author also provided the review regarding the intersection of blockchain and intrusion detection, introduce the potential application of such combination. They indicated that blockchain can help enhance IDS in the aspects of data sharing, trust computation and alarm exchange.

In anomaly detection, [16] described a framework that utilizes the blockchain concept to perform distributed and collaborative anomaly detection for devices with limited resources. The framework uses blockchain to incrementally a trusted anomaly detection model via self-attestation and consensus among IoT devices. In this case, each device will create its own intelligence model under normal network conditions and periodically every t time elapses, an agent updates its report in the locally stored chain's partial block. The agent then broadcasts the entire locally stored chain to all neighbouring agents. The limitation is that nobody verifies the intelligence model created by the IDS node because the intelligence model is based on normal traffic in the local network.

By contrast, [17-18] demonstrated how to use blockchains to enhance the performance of collaborative signature-based IDS via building a verifiable attack signatures. They design this framework to consortium blockchain. Each IDS node in the consortium blockchain can monitor the network traffic, identify attacks, and periodically share a set of attack signatures with others. This set of attack signatures has to be signed by a private key and a public key, in order to understand the source of attack signatures. Other nodes will only accept these attack signatures by verifying them against their local database. In this case, the blockchain can be only expanded if the majority of nodes have verified that the received block contains trusted rules. The limitation is not all nodes can provide an attack signature because it uses the concept of a consortium blockchain.

3. BLOCKCHAIN CONSENSUS ALGORITHM

Blockchain consensus algorithm is a crucial part in blockchain technology. This algorithm maintains the data validity which recorded on the chaining block. The consensus algorithm requires a technique to ensure the public nodes are trusted and responsible for data updates. This algorithm follows the concept of game theory. It is a strategy for dealing with competitive situations which is depending on the actions of participated nodes. Nodes in the network will compete to spend their resources or to have a large of stakes to increase chances for updating data in a blockchain. In the first strategy of spending resources, it can be found in proof of work (PoW) consensus algorithm, meanwhile the second strategy has been implemented in proof of stake (PoS) algorithm.

Proof of work (PoW): In PoW the nodes in the blockchain network reach consensus by participating in hashing puzzle searching process, where each node must find a nonce value to generate a new block. The nonce is a value in a hash function to be identified to make the current hash data is partially identical. The input of hash function is a previous hash data, and to get the target nonce a high computational power is required until the target value of adjusted current hash data is achieved. The first node that be able to solve the puzzle will be authorised to create and insert data in the block, and broadcast the block along with the data to other nodes. In PoW, the higher computational power node might have higher chances to be the winner and entitle to receive the reward [25].

Proof of stake (PoS): PoS was developed as a consensus mechanism with the aim to reduce the computational requirements of PoW. In cryptocurrency blockchain, participants with higher coin age of the coin in-network and their holding time have higher chances to be selected. The difficulty level of hashing puzzle PoW can be reduced by consuming coin age. A complete PoS network, the hash searching puzzle is completely removed, and the block leaders are no longer selected by computational power. However due to the security issue researcher highlighted a combination of PoW and PoS is better to avoid stake manipulation.

There are several PoS algorithms including chain-based PoS [26], committee-based PoS [27], byzantine fault tolerance (BFT)-based PoS [28]-[29] and delegated PoS (DPoS) [30]. Among these four types of PoS, the Chain-based is relies on stakes and computational works to select a trusted node [26]. Meanwhile committee-based PoS committee-based PoS determining a committee of stakeholders based on their stakes and allowing the committee to generate blocks, the BFT-based PoS is a PoS created on top of BFT algorithm. On the other hand, DPoS is classified as a democratic selection form of trusted node via public stake delegation. Among these four algorithms, Chain-based PoS had implemented a hybrid PoW and PoS strategy. In this paper, we extent the Chain-based PoS to develop a consensus algorithm that has better security such as in PoW but reduce computational resources.

Chain-based PoS: Chain-based PoS is an enhanced block generation mechanism with PoW and PoS characteristic. Chain-based PoS inherits many of the components of the PoW consensus protocol such as information propagation, block validation, and block finalization (i.e. longest-chain rule), except that the block generation mechanism is replaced with PoS. PoS does not hinge on wasteful hashing to generate blocks like PoW. A minter can solve the hashing puzzle only once for a clock tick. Since the hashing puzzle difficulty decreases with the minter's stake value, the expected number of hashing attempts for a minter to solve the puzzle can be significantly reduced if he stake value is high. Therefore, PoS avoids the brute-force hashing competition that would occur had PoW been used, thus achieving a significant reduction in energy usage. The extension Chain-based PoS is future discussed in Section 4.

4. DESIGNING PROOF OF ATTACK SIGNATURE (PoAS)

PoAS is an extension of Chain-based PoS. PoS was invented to reduce computational power required in PoW for mining purposes. In PoS the selection of nodes is defined by its proportion ownership value, and thus the highest stake owner will always get a chance to validate the transaction. It has been argued in [31], even PoS solves issues in PoW such as high computational resources but PoS raise a concern on the way of its determination of stakes ownership where it can be dominated by a same group of nodes. Then, a hybrid PoS and PoW was introduced to increase credibility on selecting on both criterias based on computational power and stake such as in Chain-based PoS.

4.1. Chain-based PoS

The general procedure of Chain-based PoS is summarized in Figure 2 [26]. In Chain-based PoS the determination of stake based on the ownership cryptocurrency value and it's age. The higher value of stake indicates a high loyalty of nodes in the network. Chain-based PoS include the features of PoW to solve a hashing puzzle on its function. However the difficulty of hashing puzzle determine by the stake. The combination between stake and hashing puzzle locate in blockgen function. A nonce (C) in BlockGen function is an input value to form the target hash output. The first successful node will be able to create and insert data to a new block. This process is repeated once a new transaction is arrived.

4.2. Designing PoAS extension on chain-based PoS

In Figure 3 is proposed PoAS architecture. PoAS is designed with a public blockchain that allows who to join the network. In simple terms, PoAS ensures that attack signatures produced by nodes are reliable and consistent. The process is each of attack signature validated by all nodes on the network before store in the blockchain.

Figure 2. Pseudocode Chain-based PoS [26]

Pseudocode 1: Chain-based PoS

```
Join the network by connecting to known peers;
Deposit in the stake pool;
Start BlockGen();
/* Main loop */
while running do
    if BlockGen() returns block then
        Write block into blockchain;
        Reset BlockGen() to the current blockchain;
    /* Gossiping rule */
    Broadcast block to peers;
    end
    /* Longest-chain */
    if block received & is valid & extends the longest chain then
        Write block into blockchain;
        Reset BlockGen() to the current blockchain;
        Relay block to peers;
    end
end
/* PoS-based block generation */
Function BlockGen():
    Prepare a block header and blockchain information;
    /* PoS hashing puzzle */
    Set up a clock (whose tick interval is a constant) and check for the following condition per clock tick:
        Hash (C|clock time) < target×stake value
    return new block;
end
```

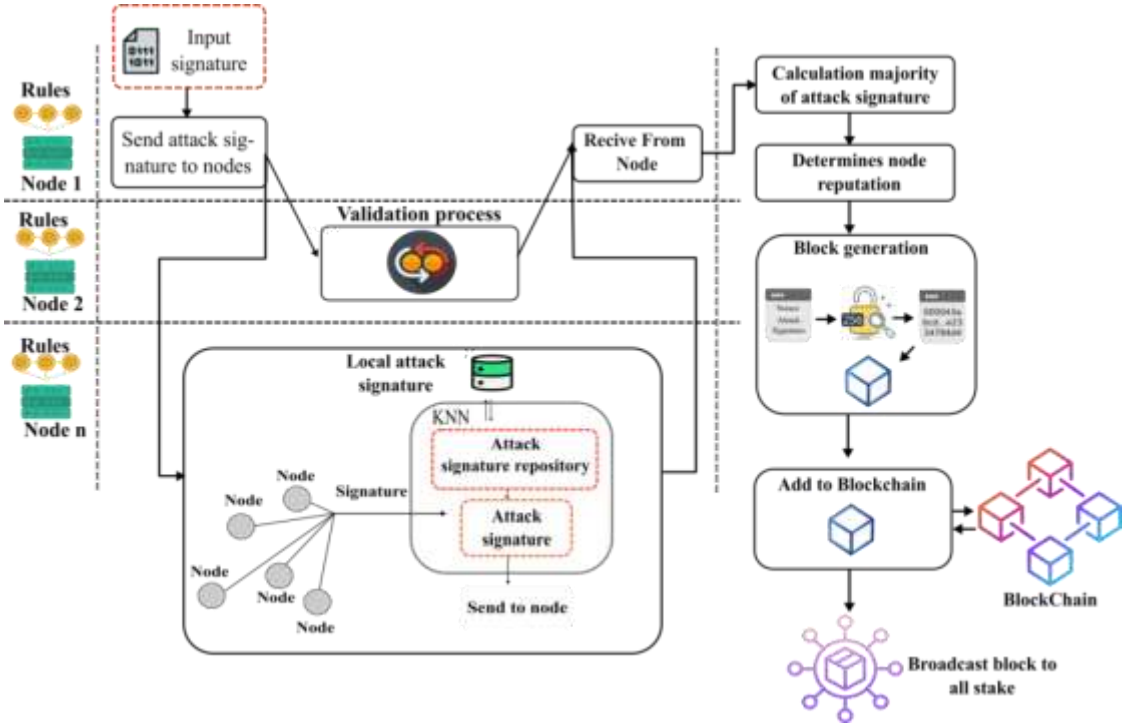

Figure 3. The process of Proof of Attack Signature

Pseudocode 1 is extent to fulfill the requirement of Chain-based PoS consensus algorithm in CIDS. It requires extensions in several parts of the algorithm including stake value definition, attack signature validation process, determining node reputation and block generation. Figure 4 shows PoAS pseudocode extension.

Figure 4. PoAS Pseudocode

Pseudocode 2: PoAS Consensus Algorithm

```
Join the network by connecting to known peers;
Conduct attack signature validation to get initial reputation;
Received attack_signature;
Write reputation;
Start Validate(attack_signature);
/* Main loop */
while running do
    if Validate() returns class attack then
        Calculate majority of validation process;
        if majority reach more than50% of stake and correct then
            Add initial reputation + 0.2 into reputation;
            Start BlockGen(attack_signature);
        end
        else
            Proses end;
        end
    end
    if BlockGen() returns block then
        Write block into blockchain;
        Add reputation + 1 into reputation;
        Reset BlockGen() to the current blockchain;
    /* Gossiping rule */
    Broadcast block to peers;
    end
    /* Longest-chain */
    if block received & is valid & extends the longest chain then
        Write block into blockchain;
        Reset BlockGen() to the current blockchain;
        Relay block to peers;
    end
end
/* PoS-based block generation */
Function BlockGen(attack_signature):
    Prepare a block header attack signature and blockchain information;
    /* PoS hashing puzzle */
Set up a clock (whose tick interval is a constant) and check for the following condition per clock tick:
            Hash (C|clock time) < target × reputation;
    return new block;
end
/* Validate attack signature */
Function Validate(attack_signature):
    Read attack signature repository into S;
    Write x and call function StringtoNumeric() to convert attack_signature;
    Write y and call function StringtoNumeric() to convert S;
    Calculate with KNN algorithm (more detail in pseudocode 3);
    return class attack;
end
/* Convert attack signature to numerical */
Function StringtoNumeric(attack_signature):
    Write x array to store result convert attack signature;
    for Si, Є attack_signature do
        Write Z
        if Si not number then
            Convert Si to ASCII into Z;
            Sum Z;
            Append Z into x;
        end
        else
            Append Si into x;
        end
    end
    return x;
end
```

Stake value definition. Stake in cryptocurrency blockchain is defined in a form of currencies value, which reflects the amount owned by a user. However, in the context of CIDS with blockchain, stake is defined as a value of reputation of a node on detecting attacks. This reputation is measured from the detection rate. In order to participate in the network, nodes must be able to produce valid attack signature. In PoAS the

number of valid attack signatures will be used as a threshold to indicate the reputation score. The reputation score will determine the difficulty of hashing puzzle as shown in Table 2.

Attack signature validation process. This process is to validate attack signature proposed by a node. The validation process extension is required for CIDS to enable nodes working collaboratively and having consent by its peer. In PoAS validation of attack signature is done by deploying K-nearest neighbors (KNN). KNN measures the distance of n-feature instances where the value of x =(x_1, x_2... xn) representing selected traffic header fields. Meanwhile is y= (y1, y2... yn) representing the traffic header in local storage. The distance is compute as in the following equation:

$$dist(x,y) = \sqrt{(y1-x1)^2 + (y2-x2)^2 + \cdots + (yn-xn)^2}$$

Pseudocode 3 shows the validation process in PoAS. The process runs on every node to validate attack signatures which is proposed by a node. Attack signature validation process as shown in Figure 5.

Figure 5. Attack signature validation process

Pseudocode 3: KNN algorithm in PoAS

```
1   Read attack signature from other stakes into x;
2   Read attack signature repository into y;
3   Write z array to store distance
    /* Main loop */
4   for i ∈ y do
        for j∈ i do
5           Compute dist(x,y)
6           Append dist into z
7   end for
8   Sort the z distances by increasing order;
9   Count the number of occurrences of class among k nearest neighbors;
10  Assign to x the most frequent into class attack;
```

In order validate attack signature all the required field packet header are converted into numeric as shown in Table 1. The process of this conversion done in StringtoNumeric() function in pseudocode 3. All string value converted to numeric by ASCII mapping except protocol which represent as integer number.

Table 1. Example result of convert attack signature

Attack signature before convert									Attack signature after convert							
packet size	flags					TCP/UDP segment length	protocol	header length	packet size	flags				TCP/UDP segment length	protocol	header length
	U	A	R	S	F					U	A	R	S	F			
1800	-	-	R	-	F	1760	tcp	40	1800			82		70	1760	1	40
340	-	-	-	S	-	300	tcp	40	340				83		300	1	40
510	-	-	-	-	-	480	udp	40	510			-			480	0	40

Determining node reputation. In order to determine the node reputation all the validation results from other nodes have been collected. It will proceed to solve the hashing puzzle if majority of nodes (more than 50%) agree that the provided signature is valid. On the other hand when the majority of nodes reject the process will be stop. For a successful node it will be rewarded a certain reputation score. The collective score will determine the difficulty of hashing puzzle as shown in Table 2. In this example a predetermined zero digit in each hash value require node to identify nonce. As predetermined zero digit increase the level to find nonce become more difficult. This rule shows high node reputation requires low computation while low node reputation requires high computation to solve the hashing puzzle.

Block generation. In this stage, the successful nodes will generate a new block for adding data in a block. The validation process of new attack signature and solving the hashing puzzle will continue. A node with high reputation has a better chance to win the competition and thus it will increase trust of reputable node. Meanwhile the consistency of data can be preserved once the data has been input in the block.

Table 2. Reputation threshold value

id	Reputation	Sample target with initial null digit determination
1	0-20	0000015783b76425....48f46a33fe9297cf
2	21-40	000015783b764257....48f46a33fe9297cf
3	41-60	00015783b7642578....48f46a33fe9297cf
4	61-80	0015783b76425789....48f46a33fe9297cf
5	81-100	015783b764257890....48f46a33fe9297cf

5. EXPERIMENT AND INITIAL RESULT

5.1. Testbed environment setup

In this section, we evaluate the performance of PoAS under some adversarial scenarios in a simulated environment consist of fourteen nodes. Eleven nodes are attack-free nodes, meanwhile the other three nodes are hosting simulated insider attacks (malicious nodes) as shown in Figure 6. These nodes are virtual machines setup in Ubuntu operating system container with a blockchain platform and intrusion detection system. Each node has their own local attack signature database and attack signature data stored in blockchain. Both attack signature data are required to demonstrate the ability of PoAS to maintain attack signature correctness and consistency in blockchain. In this experiment we extract CICIDS2017 intrusion detection evaluation dataset consist of raw network traffic packet captured in five consequence days started at 9 a.m. on Monday, 3 July and ended at 5 p.m. on Friday 7 July 2017. CICIDS2017 contains benign and attack traffic data including FTP and SSH Brute Force, Denial of Service, Heartbleed, Port Scan, Botnet, DDoS and Web Attack. There are two categories of CICIDS2017 data which are labeled and unlabeled traffic data. The labeled traffic data consist of specific attack information for validation process.

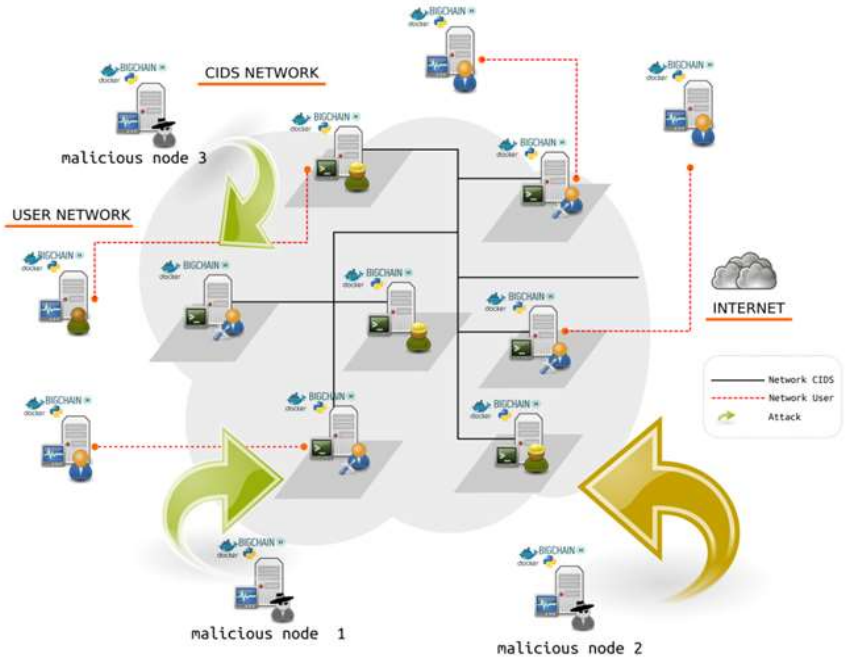

Figure 6. The high-level architecture of environments collaborative intrusion detection system with blockchain

Attack signatures from labeled CICIDS2017 dataset are extracted into 14 parts. Each part stored in local attack signature database in every node. Three malicious nodes simulate insider attacks that attempt to modify the signature data. Meanwhile, on the blockchain process, PoAS will implement consensus validation and hashing puzzle solution before the data can be inserted in the blockchain. Then, we observe the result of implementing PoAS on every node and their implication of detection rate in CIDS compared with the usage of normal attack signature database.

5.2. Experimental result

In this experiment we measure the detection rate of each node using normal attack signature database and PoAS attack signature blockchain. Second, we measure the data consistency in blockchain process, before and after data enter the block, and after signature distribute to the other node in CIDS. Figure 7 shows comparison of the detection rate results with PoAS and without PoAS. It shows detection rate with PoAS consensus algorithm obtained higher accuracy. The attempt of insider attack changing the signatures is successful on normal signature database, meanwhile PoAS is able to prevent the changes from entering blockchain and distribute the wrong attack signatures to the other nodes in CIDS. The attack signature stored in blockchain has to overcome trust issues between nodes. On the other hand, CIDS without implementing PoAS is not able to prevent insider attacks from changing the attack signatures.

On the second experiment, we compare the attack signature during implementing PoAS in three stages. First, the attack signature in a node, second, the attack signature on validation process, and third, the attack signature in the block. This process is to find out the list of attack signatures from the initial process until block created are remain consistent. Figure 8 is a comparison of attack features in two nodes. First node implements PoAS and the second node using attack signature database. The graph shows attack signatures are consistent even in the existence of insider attacks with PoAS. In the second node attack signatures are inconsistent since insider attackers can make changes on attack signature and this lead the node contribute wrong signatures to the other node in CIDS.

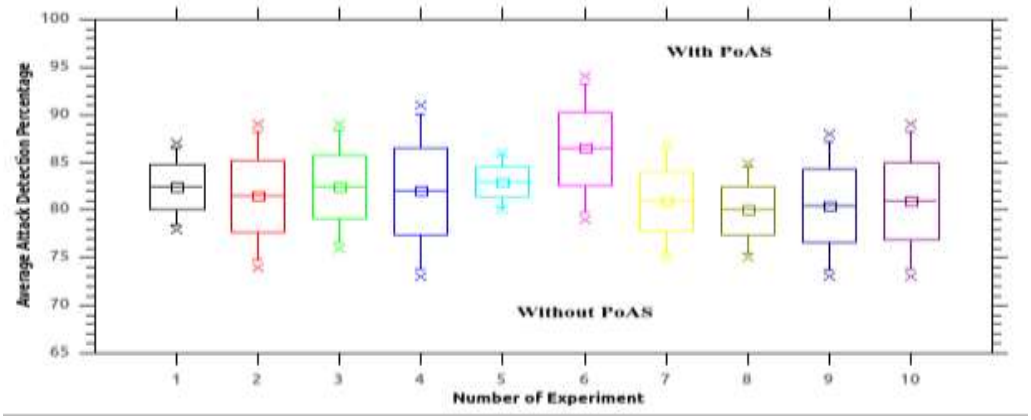

Figure 7. Comparison of Attack Detection Rate (with and without PoAS)

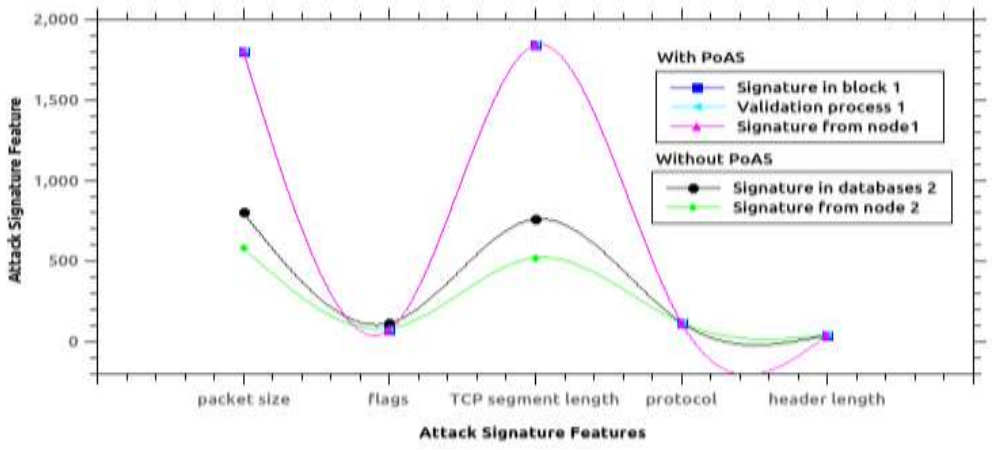

Figure 8. Data Consistency on PoAS

Figure 9 shows the time difference for PoAS validation process on each node. The outcomes indicate validation time is not the same for each validation process which influences by the size of dataset and validation method used in each node. Figure 10 shows the result of time comparison to solve hashing

puzzle by nodes with a reputation score. The results of the application of reputation show that the value of reputation affects the performance of the node to create a new block.

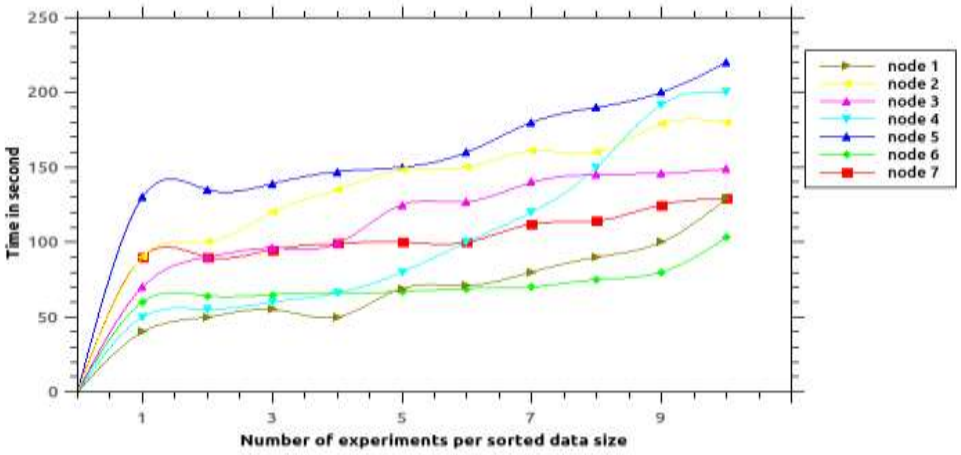

Figure 9. Comparison of validation time

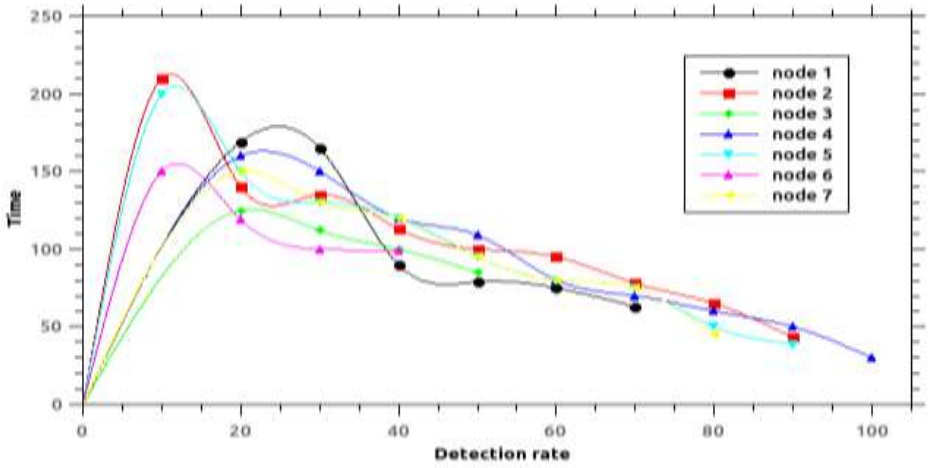

Figure 10. Comparison of hashing puzzle solving time

6. CONCLUSION

CIDS is an important solution to increase detection rate compared with a single node IDS. In CIDS the collaborative nodes can share and validate attack signature. However, each node is subject to malicious activities which can create invalid signatures and has a potential to form untrusted network. Furthermore, there is no single point of control in distributed CIDS to maintain data consistency. Due to the nature of cryptocurrency blockchain on preserving transaction in distributed environments, the implementation of blockchain consensus algorithm in CIDS is very promising. In this work, we have design an extension of Chain-based PoS consensus algorithm known as PoAS to be implemented in CIDS. The design consists of four elements PoAS including stake value definition, attack signature validation process, determining node reputation and block generation. We run an experiment using CICIDS2017 data and measure the detection rate, consistency and performance. The result has shown that PoAS can guarantee the consistency of attack signature which provide by node IDS. Moreover, PoAS can enhance the trust of the attack signature generated by the node which affects accuracy detection in node CIDS through identifying untruthful inputs and reducing error rates.

ACKNOWLEDGEMENTS

We are gratified of all research members. The research is supported by the Ministry of Higher Education (MoHE) under grant MRUN and Research Management Center (RMC) of Universiti Teknologi Malaysia, vote No: R.J130000.7851.4L872 and Communication Network and Information Security (COMNETS UNSRI) for supporting this research work.

REFERENCES

[1] W. Li, W. Meng, and L. F. Kwok, "Investigating the influence of special on-off attacks on challenge-based collaborative intrusion detection networks," *Futur. Internet*, vol. 10, no. 1, pp. 1-16, 2018, doi:10.3390/fi10010006.

[2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in *Proceedings-2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, pp. 557-564, 2017, doi:10.1109/BigDataCongress.2017.85.

[3] W. Meng, W. Li, and L. F. Kwok, "Towards Effective Trust-Based Packet Filtering in Collaborative Network Environments," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 1, pp. 233-245, 2017, doi:10.1109/TNSM.2017.2664893.

[4] O. Ajayi, O. Igbe, and T. Saadawi, "Consortium Blockchain-Based Architecture for Cyber-attack Signatures and Features Distribution," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2019*, pp. 0541–0549, 2019, doi:10.1109/UEMCON47517.2019.8993036.

[5] C. G. Cordero *et al.*, "Sphinx: A Colluder-Resistant Trust Mechanism for Collaborative Intrusion Detection," *IEEE Access*, vol. 6, pp. 72427-72438, 2018, doi:10.1109/ACCESS.2018.2880297 .

[6] W. Li, W. Meng, Y. Wang, L. F. Kwok, and R. Lu, "Identifying Passive Message Fingerprint Attacks via Honey Challenge in Collaborative Intrusion Detection Networks," *Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018*, pp. 1208-1213, 2018, doi:10.1109/TrustCom/BigDataSE.2018.00167.

[7] W. Li and L. F. Kwok, "Challenge-based collaborative intrusion detection networks under passive message fingerprint attack: A further analysis," *J. Inf. Secur. Appl.*, vol. 47, pp. 1-7, 2019, doi:10.1016/j.jisa.2019.03.019.

[8] W. Li, W. Meng, L. F. Kwok, and H. H. S. IP, "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model," *J. Netw. Comput. Appl.*, vol. 77, no, pp. 135–145, 2017. October 2016, doi:10.1016/j.jnca.2016.09.014.

[9] N. Kolokotronis, S. Brotsis, G. Germanos, C. Vassilakis, and S. Shiaeles, "On blockchain architectures for trust-based collaborative intrusion detection," *Proc.-2019 IEEE World Congr. Serv. Serv. 2019*, pp. 21–28, 2019, doi:10.1109/SERVICES.2019.00019.

[10] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Futur. Gener. Comput. Syst.*, 2017, doi:10.1016/j.future.2017.08.020.

[11] Baliga, "Understanding Blockchain Consensus Models," *Whitepaper*, no. April, pp. 1-14, 2017.

[12] S. J. Alsunaidi and F. A. Alhaidari, "A survey of consensus algorithms for blockchain technology," *2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019*, pp. 1-6, 2019, doi:10.1109/ICCISci.2019.8716424.

[13] Z. Yu, X. G. Liu, and G. Wang, "A Survey of Consensus and Incentive Mechanism in Blockchain Derived from P2P," *Proc. Int. Conf. Parallel Distrib. Syst.-ICPADS*, vol. 2018-December, pp. 1010-1015, 2019, doi:10.1109/PADSW.2018.8645047.

[14] N. Alexopoulos and E. Vasilomanolakis, "Towards Blockchain-Based Collaborative Intrusion Detection Systems," *Crit. Inf. Infrastructures Secur.*, vol. 10707, pp. 107-118, 2018, doi:10.1007/978-3-319-99843-5_10.

[15] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179-10188, 2018, doi:10.1109/ACCESS.2018.2799854.

[16] T. Golomb, Y. Mirsky, and Y. Elovici, "CIoTA: Collaborative Anomaly Detection via Blockchain," in *Proceedings 2018 Workshop on Decentralized IoT Security and Standards*, no. February, 2018.

[17] S. Tug, W. Meng, and Y. Wang, "CBSigIDS: Towards Collaborative Blockchained Signature-Based Intrusion Detection," in *Proceedings-IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/Gree*, pp. 1228–1235, 2018, doi:10.1109/Cybermatics_2018.2018.00217.

[18] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Futur. Gener. Comput. Syst.*, vol. 96, pp. 481-489, 2019, doi:10.1016/j.future.2019.02.064.

[19] R. Jin, X. He, and H. Dai, "Collaborative IDS Configuration: A Two-Layer Game-Theoretic Approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 803-815, 2018.

[20] C. Vincent Zhou, C. Leckie, and S. Karunasekera, "Decentralized multi-dimensional alert correlation for collaborative intrusion detection," *J. Netw. Comput. Appl.*, vol. 32, no. 5, pp. 1106-1123, 2009, doi:10.1016/j.jnca.2009.02.010.

[21] E. Vasilomanolakis, S. Karuppayah, M. Muhlhauser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1-33, 2015, doi:10.1145/2716260.

[22] Y. Wang, W. Meng, W. Li, J. Li, W. X. Liu, and Y. Xiang, "A fog-based privacy-preserving approach for distributed signature-based intrusion detection," *J. Parallel Distrib. Comput.*, vol. 122, pp. 26-35, Dec. 2018, doi:10.1016/j.jpdc.2018.07.013.

[23] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *J. Netw. Comput. Appl.*, vol. 66, pp. 1-16, 2016, doi:10.1016/j.jnca.2016.03.011.
[24] W. Gao, W. G. Hatcher, and W. Yu, "A survey of blockchain: Techniques, applications, and challenges," *Proc.-Int. Conf. Comput. Commun. Networks, ICCCN*, vol. 2018-July, no. i, pp. 1-11, 2018, doi:10.1109/ICCCN.2018.8487348.
[25] P. Zhang, D. C. Schmidt, J. White, and A. Dubey, "Consensus mechanisms and information security technologies," in *Advances in Computers, 1st ed., Elsevier Inc.*, pp. 1-29, 2019, doi:10.1016/bs.adcom.2019.05.001.
[26] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," *IEEE Commun. Surv. Tutorials*, no. November, pp. 1-1, 2020, doi:10.1109/COMST.2020.2969706.
[27] Y. Liu, J. Liu, Z. Zhang, and H. Yu, "A fair selection protocol for committee-based permissionless blockchains," *Comput. Secur.*, vol. 91, 2020, doi:10.1016/j.cose.2020.101718.
[28] K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain," *Proc. Int. Conf. Parallel Distrib. Syst.-ICPADS*, vol. 2018-December, pp. 604-611, 2019, doi:10.1109/PADSW.2018.8644933.
[29] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine Fault-Tolerant ordering service for the hyperledger fabric blockchain platform," *Proc.-48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks*, DSN 2018, no. 1, pp. 51-58, 2018, doi:10.1109/DSN.2018.00018.
[30] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong, and M. Zhou, "Delegated Proof of Stake With Downgrade: A Secure and Efficient Blockchain Consensus Algorithm With Downgrade Mechanism," *IEEE Access*, vol. 7, pp. 118541-118555, 2019, doi:10.1109/ACCESS.2019.2935149.
[31] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," *IEEE Access*, vol. 7, pp. 85727-85745, 2019, doi:10.1109/ACCESS.2019.2925010.