IEICE TRANS. INF. & SYST., VOL.E105–D, NO.2 FEBRUARY 2022

PAPER Special Section on Blockchain Systems and Applications

BlockCSDN: Towards Blockchain-Based Collaborative Intrusion Detection in Software Defined Networking*

Wenjuan LI[†,††a)], **Yu WANG**[†b)], **Weizhi MENG**[†,†††c)], **Jin LI**[†d)], ***Nonmembers***, ***and*** **Chunhua SU**[††††e)], ***Member***

SUMMARY To safeguard critical services and assets in a distributed environment, collaborative intrusion detection systems (CIDSs) are usually adopted to share necessary data and information among various nodes, and enhance the detection capability. For simplifying the network management, software defined networking (SDN) is an emerging platform that decouples the controller plane from the data plane. Intuitively, SDN can help lighten the management complexity in CIDSs, and a CIDS can protect the security of SDN. In practical implementation, trust management is an important approach to help identify insider attacks (or malicious nodes) in CIDSs, but the challenge is how to ensure the data integrity when evaluating the reputation of a node. Motivated by the recent development of blockchain technology, in this work, we design BlockCSDN — a framework of blockchain-based collaborative intrusion detection in SDN, and take the challenge-based CIDS as a study. The experimental results under both external and internal attacks indicate that using blockchain technology can benefit the robustness and security of CIDSs and SDN.

key words: *collaborative intrusion detection, blockchain technology, software defined networking, insider attack, challenge-based trust management*

1. Introduction

To identify various cyber threats, intrusion detection systems (IDSs) are an important and essential security mechanism by monitoring a target system and network [8], [15]. Typically, an IDS can be categorized into either rule-based or anomaly-based detection [32]. The former can detect an unfavorable event via signature matching [22], while the latter can figure out a malicious event by measuring the similarity between current profile and the pre-defined profile, i.e., building the profile via machine learning algorithms [37].

Manuscript received May 21, 2021.
Manuscript revised July 17, 2021.
Manuscript publicized September 16, 2021.

[†]The authors are with the Institute of Artificial Intelligence and Blockchain, Guangzhou University, China.
[††]The author is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, China.
[†††]The author is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark.
[††††]The author is with the University of Aizu, Aizuwakamatsu-shi, 965–8580, Japan.

*A preliminary version of this paper appears in Proc. 14th International Conference on Network and System Security (NSS 2020), pp.261–276, 2021 [1].

a) E-mail: wenjuan.li@polyu.edu.hk
b) E-mail: yuwang@gzhu.edu.cn
c) E-mail: weme@dtu.dk (Corresponding author)
d) E-mail: jinli@gzhu.edu.cn
e) E-mail: chsu@u-aizu.ac.jp

DOI: 10.1587/transinf.2021BCP0013

With the increasing architecture complexity of computer networks, collaborative intrusion detection systems (CIDSs) are believed to be a necessity, which enables the data shared among all parties for detection enhancement. For instance, Eskandari et al. [3] introduced Passban, an IDS that can be deployed on cheap Internet of Things (IoT) gateways and communicated with edge computing devices against malicious traffic. However, due to the rapid growth of IoT devices and the extension of cloud services, network management would become a difficult and error-prone task.

As a solution, software defined networking (SDN) can provide dynamic, automated and flexible network management by separating the control plane from the data plane, i.e., it can adjust network-wide traffic flow to meet practical needs [33]. The centralized controller allows administrators to handle network assets and update network infrastructure in real-time. According to the IDC report, SDN market would reach over $12 billion in 2022 [34]. In a SDN-based CIDS, SDN can ease the management complexity of detection, while a CIDS can also help protect the security of SDN. However, insider attacks are still a big threat, due to the distributed network infrastructure. An attacker can perform an internal exploit by compromising one of the underlying devices or Virtual Machines (VMs) [12]. In addition, insiders can spread manipulated data to degrade the detection performance, i.e., making faked flows to deceive SDN switches or controllers [33].

To defeat insider attacks, trust management is an important solution to check the reputation of a node in CIDSs or any other distributed environments. For instance, Liu et al. [14] introduced a trust-based detection scheme using the K-means classification algorithm to detect internal threats in an IoT environment. For most current trust-based schemes, it is still a challenge on how to ensure the integrity and authenticity of shared data and information. With the advent of blockchain technology, it can be a promising solution to check the integrity of shared data among different nodes without the need of a trusted third party [13].

Contributions. Motivated by the advantages of combining SDN-based CIDSs with blockchain, we propose *BlockCSDN*, a framework of blockchain-based collaborative intrusion detection in SDN. The framework can examine the integrity and the authenticity of exchanged data, which enables improving the robustness of trust management against insider attacks. The contributions of this work can be summarized as below.

- We first introduce a framework of blockchain-based collaborative intrusion detection in SDN, named BlockCSDN. The blockchain can examine the data integrity, figure out potentially malicious input, and enhance the robustness of establishing a trust relationship among different nodes in CIDSs. If any malicious data is found, the SDN controller can quickly inform all participating nodes according to the information on the chain.
- As a study, we consider a type of challenge-based CIDS and detail the framework implementation. The challenge-based trust management scheme measures the reputation of a node by sending challenges, and derives a trust value based on the received feedback.
- In the evaluation, we examine the scheme performance in both a simulated and a real environment, under external and internal attacks. The experimental results demonstrate the viability and the effectiveness of our BlockCSDN.

The remaining parts of this work are organized as follows. Section 2 introduces the related work regarding SDN, blockchain and CIDSs. Section 3 describes our proposed framework in detail and shows an example based on challenge-based CIDS. Section 4 evaluates our framework and analyzes the collected results. Finally, Sect. 5 concludes this work.

2. Related Work

(1) Trust-Based CIDS.

Collaborative intrusion detection aims to enhance the detection performance by exchanging the required information within a network system, but a big threat is insider attack, where an attacker can exploit network vulnerabilities inside the environment. For protection, building trust management is believed to be one of the necessary approaches.

Fung *et al.* [4] presented a challenge-based CIDS by sending a kind of message called *challenge* to examine the reputation of other nodes. Li *et al.* [9] introduced a sensitivity-based CIDS, which applies the notion of intrusion sensitivity for evaluating the detection sensitivity of a detector. To automatically assign the value of sensitivity, several machine learning classifiers can be used, like SVM and KNN [16]. By highlighting the importance of expert nodes, sensitivity-based CIDS can detect malicious nodes in a fast way, like pollution attack [10]. Veeraiah and Krishna [28] designed a trust-aware fuzzy clustering and fuzzy Naive Bayes system, by using several trust factors to predict the trust value of a node, such as direct trust, indirect trust, and the recent trust.

(2) Blockchain-Based CIDS.

Due to the potential merits of blockchain, many studies have started exploring its usage in intrusion detection. An early CIDS framework [2] was proposed through considering a set of alarms as transactions in a blockchain. Then, CIDS nodes can communicate and perform activities via consensus protocols. A more detailed analysis was given by a review [19], which discusses how to combine blockchain with IDS/CIDS, and what are the main limitations. Blockchain was believed to be helpful in the aspects of data sharing, trust management and alarm exchange (ensuring integrity).

A type of anomaly-based CIDS was introduced by Golomb *et al.* [5], which used blockchains to help enhance the performance of anomaly detection. A blockchain- and rule-based CIDS [11], [38] was then developed, which could use blockchain to help build a verifiable rule database. Meng et al. [23] focused on rule-based IDS, and designed a blockchain-enabled single character frequency-based matching scheme, which can build a verifiable database of malicious payloads via blockchains. Hu et al. [5] introduced a blockchain-based CIDS for multimicrogrid systems, while they only used the blockchain to store the final detection results. Kanth et al. [6] introduced an Ethereum blockchain-based CIDS by leveraging pluggable authentication modules.

(3) SDN-Based Intrusion Detection.

In SDN, Lamb and Heileman [7] presented a concept of trust-based CIDS that interacts among host, switches, controllers, repositories and applications. Yan *et al.* [40] introduced a trust framework for SDN, using a reputation management component to measure the trustworthiness of others. A system of TruSDn was introduced by Paladi and Gehrmann [30], which used Intel Software Guard Extensions (SGX) to enhance the trust in SDN. Meng *et al.* [17] introduced a trust-based security mechanism based on Bayesian inference to defeat insider attacks in a healthcare SDN environment. Their idea is to monitor the traffic status and identify malicious actions. Zhang *et al.* [41] explored the use of deep reinforcement learning to establish a trust relation for connected vehicles using SDN, i.e., the controller aims to communicate with vehicles at discrete time steps. Li *et al.* [12] studied the performance of challenge-based CIDS in SDN, and found its effectiveness against insider attacks.

(4) Blockchain, SDN and CIDS.

In the literature, there are not many relevant studies regarding the combination of blockchain, SDN and CIDS. Steichen *et al.* [36] presented an OpenFlow-based firewall named ChainGuard, which could secure blockchain-based SDN by detecting malicious events inside the network. A snort-based CIDS was introduced by Ujjan et al. [31], which used SDN to help enhance the detection performance.

The above studies combined SDN-based CIDS with blockchain, whereas none of them considered a trust-based CIDS. As aforementioned, trust management is an important solution to protect computer networks against insider attacks. In this work, we therefore aim to bridge this gap and introduce a framework for blockchain- and trust-based CIDS in SDN.

3. Our Framework

Since each of SDN, blockchain, CIDS and trust management can contribute to either network management or security, the combination of them should be able to complement each other.

3.1 BlockCSDN

Figure 1 depicts the framework of blockchain-based CIDS in SDN, called BlockCSDN.

- *CIDS.* Each CIDS node can connect with each other and exchange required data or information. A node can contain several major components such as connection component (for physical connection), collaboration component (for information exchange), blockchain component (for communicating with the chain), and trust management component (for measuring nodes' reputation). Based on the requirements, both rule-based and anomaly-based detection approach can be deployed in a node.
- *Blockchain.* A blockchain can be established and updated via consensus protocol and smart contract agreed among all CIDS nodes. The consensus can be extended to SDN controller and applications. Based on the concrete schemes and requirements, various information can be chained, e.g., alarms, rules, messages. Intuitively, the blockchain ensures the data integrity and facilitates the information to be visible to other parties. For example, all SDN planes can access the chain for retrieving expected information. In practical usage, privacy-preserving techniques can be used to protect privacy.
- *SDN.* As introduced earlier, SDN has three layers (or called planes): application layer, control layer and data layer. The application layer can deploy customized applications and security mechanisms. For example, a trust management application can be deployed here to guide the controller how to retrieve the information from both the blockchain and CIDS, and then measure the network status and a node's reputation. The controller layer can enforce the security policies and react to malicious nodes and traffic. A CIDS often works at the data layer to detect various external or internal attacks, and share the information with both the chain and the controller.

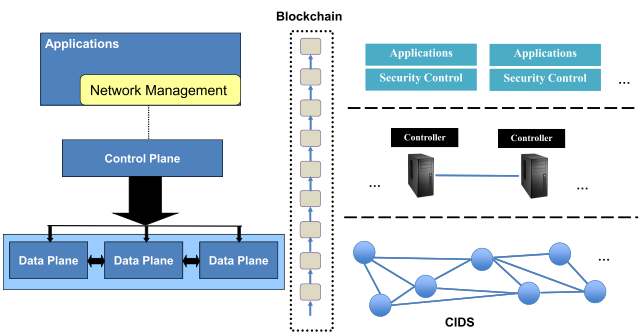


Fig. 1 BlockCSDN: the framework of blockchain-based CIDS in SDN.

Hence each of CIDS, blockchain and SDN can complement and work with each other, i.e., the trust management can be enhanced by retrieving information from the chain, and the SDN controller can enforce the policies. The framework can maintain the merits of SDN, blockchain and trust-based CIDS.

- *Data integrity.* This refers to the reliability and trustworthiness of shared data. Due to the nature design of blockchains, the chained data and information are inherently resistent to the modification (e.g., edition, deletion), as long as the data has been added to the chain.
- *Efficiency.* The framework can ensure the quality of information shared via consensus in the network and measure the reputation of each participant. The SDN controller can take actions immediately when malicious traffic or behavior are detected.
- *Dynamicity.* Participants (e.g., customers) can configure the software and applications on the controllers, and easily enforce their demands (e.g., on-demand services, access rules), without the need of understanding the underlying devices in the data plane.
- *Privacy.* To protect participants' privacy, the framework allows to implement privacy-preserving schemes and access control list. Therefore, the blockchain data can only be visible to authorized parties, who need to have credentials to recover the data and information.
- *Scalability.* The framework can scale to a large computer network. This is because most CIDSs are scalable, and the SDN itself is developed to help handle a large amount of network nodes. For blockchains, scalability (e.g., a high transaction per second) can be achieved by changing its consensus mechanism or adjusting some system parameters.
- *Security.* As the blockchain enables the integrity and the trustworthiness of data, it is more difficult for cyber-attackers to intrude the network and compromise the in-between channel. With trust management, the framework can be robust against insider attacks. When a malicious node is identified, SDN controller can take a quick response to mitigate the risk.

3.2 An Implementation Instance Based on Challenge-Based CIDS

To implement the framework, in this work, we consider a special kind of CIDS — challenge-based CIDS [4], [9], as it can evaluate the trustworthiness of other nodes via challenges (a type of message). Figure 2 depicts how to realize BlockCSDN with challenge-based CIDS.

Node expertise. Similar to previous studies [9], [12],

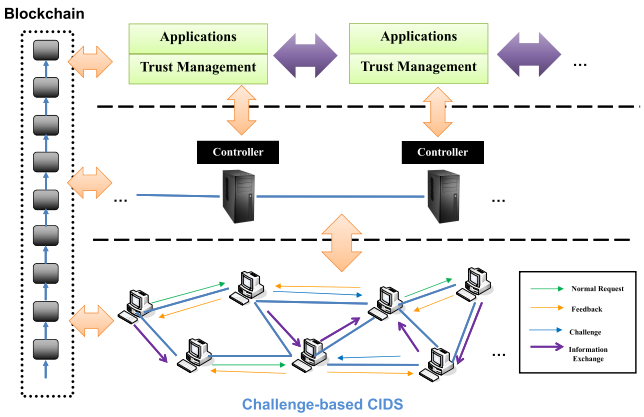


Fig. 2 Implementation instance of BlockCSDN with challenge-based CIDS.

this work considers three expert levels of a node: low (0.1), medium (0.5) and high (0.95). A beta function is used to model the expertise of an IDS node:

$$f(p'|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} p'^{\alpha-1}(1-p')^{\beta-1}$$
$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \quad (1)$$

where p' ($\in [0,1]$) indicates the probability of an attack checked by the IDS. $f(p'|\alpha,\beta)$ indicates the probability that a node with expertise level l responses with a value of p' to an attack of difficulty level of d ($\in [0,1]$). A bigger value of l indicates a higher probability of correctly detecting an attack and a bigger value of d indicates that an attack is more difficult to find. The setting of α and β can refer to [4], [9]:

$$\alpha = 1 + \frac{l(1-d)}{d(1-l)} r$$
$$\beta = 1 + \frac{l(1-d)}{d(1-l)} (1-r) \quad (2)$$

where $r \in \{0,1\}$ indicates the expected output. For a fixed level of detection difficulty, the node with a higher level of expertise can achieve higher probability of correctly detecting an attack.

Trust evaluation at nodes. To measure the trustworthiness of a node, a testing node can send a *challenge* to another node using a random generation process, and then check its satisfaction level by comparing the received feedback with the expected feedback. We can derive the reputation of a node i according to node j as follows:

$$T_i^j = \left(w_s \frac{\sum_{k=0}^{n} F_k^{j,i} \lambda^{tk}}{\sum_{k=0}^{n} \lambda^{tk}} - T_s \right) (1-x)^d + T_s \quad (3)$$

where $F_k^{j,i} \in [0,1]$ indicates the score of the received feedback k and n is the total number of feedback. λ is a *forgetting factor* that assigns less weight to older feedback. w_s means a *significant weight* relying on the total number of received feedback, if there is only a few feedback under a certain minimum m, then $w_s = \frac{\sum_{k=0}^{n} \lambda^{tk}}{m}$; otherwise $w_s = 1$. x is the percentage of "don't know" answers during a time period. d is a positive incentive parameter to control the severity of punishment to "don't know" replies.

Satisfaction evaluation. The satisfaction level can be measured based on an expected feedback ($e \in [0,1]$) and an actual received feedback ($r \in [0,1]$). A function F ($\in [0,1]$) is built to reflect the satisfaction by measuring the difference between the received answer and the expected answer as follows.

$$F = 1 - \left(\frac{e-r}{max(c_1 e, 1-e)} \right)^{c_2} \quad e > r \quad (4)$$
$$F = 1 - \left(\frac{c_1(r-e)}{max(c_1 e, 1-e)} \right)^{c_2} \quad e \le r \quad (5)$$

where c_1 controls the degree of penalty for wrong estimates and c_2 controls the sensitivity of estimation. This work sets $c_1 = 1.5$ and $c_2 = 1$.

4. Evaluation

This work considers two experiments to study the performance under a simulated and a real network environment, respectively.

4.1 Experiment-1

We first established the SDN environment by means of Open vSwitch [27] and POX controller [29], and then constructed a challenge-based CIDS with 55 nodes (based on Snort [35]), which were randomly distributed in a 15×15 grid region. The challenge-based CIDS encourages nodes to connect and share information like alarms with each other. Each node maintains a *partner list* and the reputation of newcomers would be $T_s = 0.5$.

For sending challenges, we suppose that each node can send a challenge randomly to its partners with an average rate of ε, according to the previous work [4], [9], [12]. Two special request frequencies — low and high are adopted. The request frequency of ε_l is low for a highly trusted or highly untrusted node, since their feedback should be very confident. On the other hand, the request frequency of ε_h should be high for others with a trust value around the threshold. The consortium blockchain was deployed in a mid-end computer with Intel(R) Core (TM) i5-6300HQ Processor, CPU 2.5 GHz and 500 GB storage. There is a need for 2/3 nodes in the network to sign a block to be appended to the blockchain. Table 1 summarizes the simulation parameters. To avoid performance bias, we repeat each experiment for 10 times.

External attack. The purpose of this experiment is to evaluate the framework performance against external attacks such as flooding attack. We used NetScanTools [25] to create packets and flood our environment, through randomly manipulating the IP sources. Figure 3 shows the impact of flooding attacks on the bandwidth between our framework

Table 1 Experimental setup with simulation parameters.

Parameters	Value	Description
λ	0.9	Forgetting factor
ε_l	10/day	Low request frequency
ε_h	20/day	High request frequency
r	0.8	Trust threshold
T_s	0.5	Trust value for newcomers
m	10	Lower limit of received feedback
d	0.3	Severity of punishment

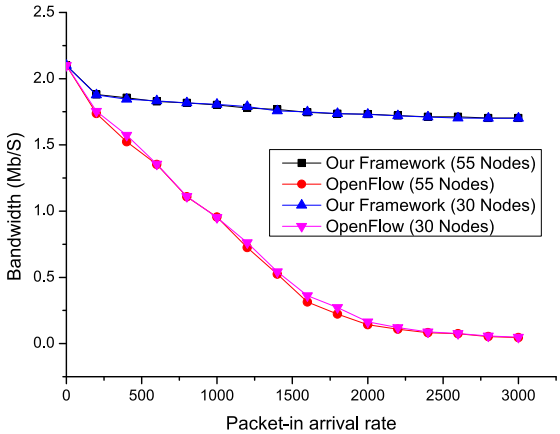


Fig. 3 The impact of flooding attacks on the bandwidth (Experiment-1).

and OpenFlow (normal condition), with 30 nodes and 55 nodes.

It is found that similar to previous work [1], the bandwidth under OpenFlow was reduced very fast during the flooding period. When the packet-in arrival rate achieved at around 1200 packets/s and 1400 packets/s, the bandwidth decreased to below 0.76 and 0.5, respectively. The network function was compromised (the bandwidth is below 0.1) when the packet-in arrival rate reached 2400 packets/s. In the comparison, our framework could mitigate the negative impact and maintain the bandwidth after a small decrease to around 1.7. This is because our framework is able to figure out malicious traffic and make a reaction quickly, through SDN controllers and blockchain technology.

Insider attack. To examine the framework performance against insider attacks, we randomly selected three expert nodes to perform a betrayal attack, in which a trusted node turns into a malicious one, i.e., delivering malicious packets or sending fake alarms. During alarm aggregation, our framework can examine the received alarms by leveraging the chain, which have to be checked by other nodes.

Figures 4 and 5 demonstrate the average trust value of malicious nodes and alarm aggregation errors under insider attack, with 30 nodes and 55 nodes respectively. It is identified that our framework could detect malicious nodes faster than the original challenge-based CIDS. This is because the SDN controller can quickly take actions based on the policy and emphasize the penalty on malicious events. On the other hand, our framework could greatly reduce the alarm aggregation errors to below 8%, as compared with over 20% for

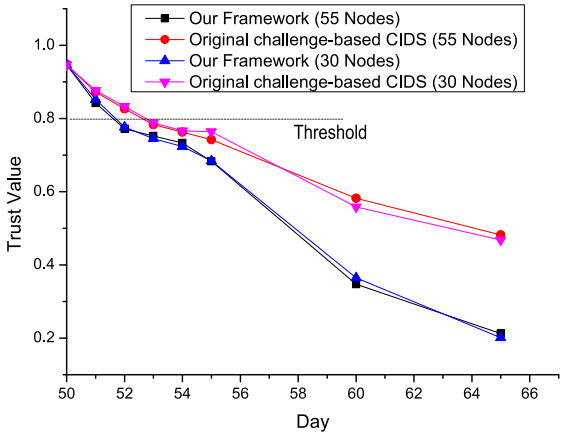


Fig. 4 The average trust value of malicious nodes (Experiment-1).

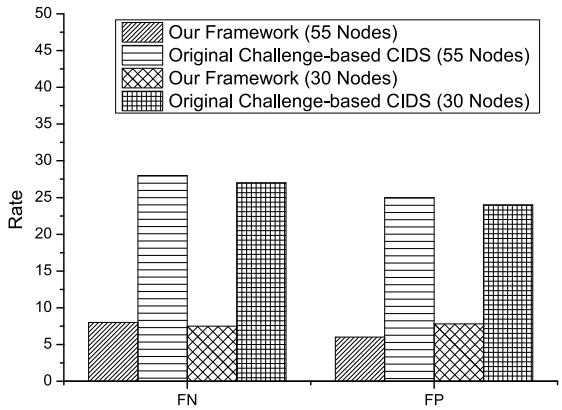


Fig. 5 Alarm aggregation errors under insider attacks (Experiment-1).

the original scheme. The main reason is that our approach utilizes blockchain technology to check any malicious content during the consensus process.

4.2 Experiment-2

In this experiment, we collaborated with an IT organization to further explore and validate the performance of our system in a real CIDN environment. There are a total of 60 nodes in the CIDN, while we randomly selected 35 of them in our evaluation, based on the organization policy. A Demilitarized Zone (DMZ) was set between the Internet and the internal environment, and Snort was deployed in each node as IDS detector.

In addition, we adopted the same settings in Table 1, and the consortium blockchain was deployed in a mid-end computer with Intel(R) Core (TM) i5-6300HQ Processor, CPU 2.5 GHz and 500 GB storage. There is a need for 2/3 nodes in the network to sign a block to be appended to the blockchain.

External attack. Similar to Experiment-1, we used the NetScanTools to launch the flooding attack, and Fig. 6 shows the impact of flooding attacks. It is seen that the bandwidth decreased significantly under OpenFlow during

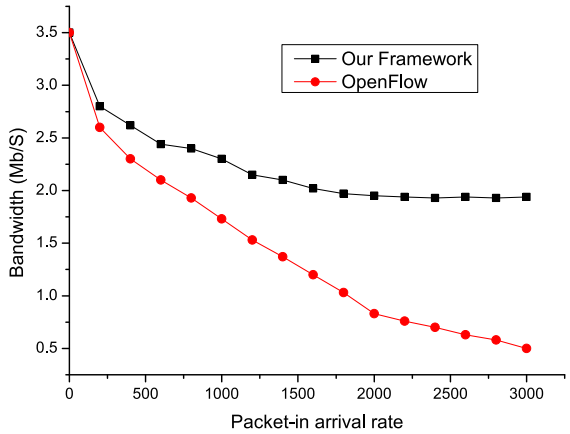


Fig. 6 The impact of flooding attacks in the real environment (Experiment-2).

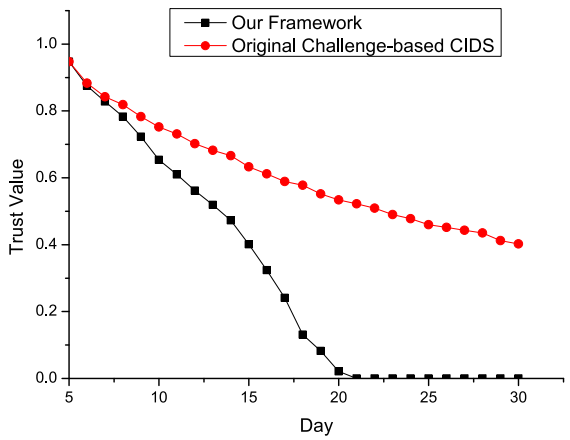


Fig. 7 The average trust value of malicious nodes (Experiment-2).

the flooding attack, while our system can maintain the bandwidth over 2 Mb/s. The results validated that OpenFlow-based environment would be compromised when the packet-in arrival rate reached 3000, and that our system could mitigate the impact of flooding attack, due to the involvement of SDN controllers and blockchain.

Insider attack. Similarly, we chosen three expert nodes in a random manner and launched a betrayal attack. We have the following observations.

- Figure 7 presents the average trust value of malicious nodes in the real environment. It is observed that our approach could identify malicious nodes quickly and reduce the reputation sharply. This validates the effectiveness of SDN when enforcing the policies and rules.
- Figure 8 depicts the errors caused during alarm aggregation. It is found that our framework could control both error rates below 7.5%, as compared with 25% in the original challenge-based scheme.

Our results validate the use of blockchain in figuring out untruthful alarms, and our approach could provide much better performance than the original scheme.

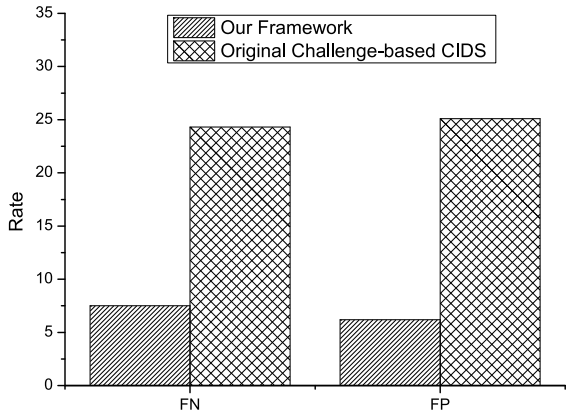


Fig. 8 Alarm aggregation errors under insider attacks (Experiment-2).

4.3 Discussion

Our experimental results are positive in a practical environment, some more properties and features could be considered.

- *Communication workload.* To make all components and parties collaborated with each other, our framework may cause some additional communication load to each CIDS node and the whole network environment. This is an important topic that can be explored in our future study.
- *Advanced attack evaluation.* This work mainly considers some intuitive attacks to test the viability of our framework. One of our future directions is to explore the framework performance against advanced external and internal attacks. Furthermore, we plan to provide a large comparison with similar approaches and platforms.
- *Privacy issue.* Due to the GDPR enforcement in Europe, privacy issue has received much more attention. For example, an IDS may monitor the whole environment and collect as much information as possible to identify malicious events. How to ensure the proper usage of such collected information should be considered in future.

5. Conclusion

In this work, we advocate the advantage by combining blockchain, SDN and CIDS, and then design a general framework of blockchain-based collaborative intrusion detection in SDN, shortly BlockCSDN. We show an implementation instance with challenge-based CIDS, and performed two experiments to investigate its performance in a simulated and a real network environment, respectively. The experimental results demonstrate the viability and the effectiveness of our approach in defeating insider attacks, maintaining network bandwidth and enhancing the robustness of alarm aggregation.

Acknowledgments

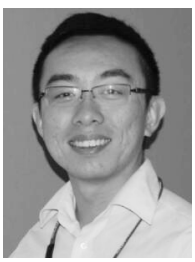
This work was partially supported by National Natural Science Foundation of China (No. 61802077).

References

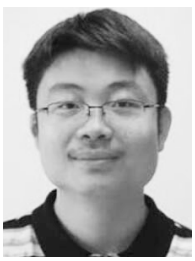
[1] W. Li, J. Tan, and Y. Wang, "A framework of blockchain-based collaborative intrusion detection in software defined networking," Network and System Security, NSS 2020, LNCS 12570, pp.261–276, Springer, 2020.
[2] N. Alexopoulos, E Vasilomanolakis, N.R. Ivánkó, and M. Mühlhäuser, "Towards blockchain-based collaborative intrusion detection systems," Proc. 12th International Conference on Critical Information Infrastructures Security, LNCS 10707, pp.107–118, Springer, 2017.
[3] M. Eskandari, Z.H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," IEEE Internet Things J., vol.7, no.8, pp.6882–6897, 2020.
[4] C.J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "Trust management for host-based collaborative intrusion detection," F. De Turck, W. Kellerer, and G. Kormentzas (eds.), Managing Large-Scale Service Deployment, DSOM 2008, LNCS 5273, pp.109–122, Springer, 2008.
[5] T. Golomb, Y. Mirsky, and Y. Elovici, "CIoTA: Collaborative IoT anomaly detection via blockchain," Proc. Workshop on Decentralized IoT Security and Standards (DISS), pp.1–6, 2018.
[6] V. Kanth, A. McAbee, M. Tummala, and J.C. McEachen, "Collaborative intrusion detection leveraging blockchain and pluggable authentication modules," Proc. HICSS 2020, pp.1–7, 2020.
[7] C.C. Lamb and G.L. Heileman, "Towards robust trust in software defined networks," GLOBECOM Workshops, pp.166–171, 2014.
[8] W. Lee, J.B.D. Cabrera, A. Thomas, N. Balwalli, S. Saluja, and Y. Zhang, "Performance adaptation in real-time intrusion detection systems," Recent Advances in Intrusion Detection, RAID 2002, LNCS 2516, pp.252–273, Springer, 2002.
[9] W. Li, W. Meng, and L.-F. Kwok, "Design of intrusion sensitivity-based trust management model for collaborative intrusion detection networks," Proc. 8th IFIP WG 11.11 International Conference on Trust Management (IFIPTM), pp.61–76, Springer, 2014.
[10] W. Li and W. Meng, "Enhancing collaborative intrusion detection networks using intrusion sensitivity in detecting pollution attacks," Information and Computer Security, vol.24, no.3, pp.265–276, Emerald, 2016.
[11] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," Future Generation Computer Systems, vol.96, pp.481–489, 2019.
[12] W. Li, Y. Wang, Z. Jin, K. Yu, J. Li, and Y. Xiang, "Challenge-based collaborative intrusion detection in software-defined networking: An evaluation," Digital Communications and Networks, vol.7, no.2, pp.257–263, 2021.
[13] W. Li, W. Meng, Z. Liu, and M.-H. Au, "Towards blockchain-based software-defined networking: Security challenges and solutions," IEICE Trans. Inf. & Syst., vol.E103-D, no.2, pp.196–203, Feb. 2020.
[14] L. Liu, J. Yang, and W. Meng, "Detecting malicious nodes via gradient descent and support vector machine in Internet of Things," Comput. Electr. Eng., vol.77, pp.339–353, 2019.
[15] Y.-X. Meng, "The practice on using machine learning for network anomaly intrusion detection," Proc. 2011 International Conference on Machine Learning and Cybernetics (ICMLC 2011), pp.576–581, IEEE, 2011.
[16] W. Meng, W. Li, and L.-F. Kwok, "Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection," Security and Communication Networks, vol.8, no.18, pp.3883–3895, Wiley, 2015.
[17] W. Meng, K.-K.R. Choo, S. Furnell, A.V. Vasilakos, and C.W. Probst, "Towards Bayesian-based trust management for insider attacks in healthcare software-defined networks," IEEE Trans. Netw. Service Manag., vol.15, no.2, pp.761–773, 2018.
[18] W. Meng, J. Wang, X. Wang, J. Liu, Z. Yu, J. Li, Y. Zhao, and S.S.M. Chow, "Position paper on blockchain technology: Smart contract and applications," The 12th International Conference on Network and System Security (NSS), LNCS 11058, pp.474–483, Springer, 2018.
[19] 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.
[20] W. Meng, W. Li, L.T. Yang, and P. Li, "Enhancing challenge-based collaborative intrusion detection networks against insider attacks using blockchain," International Journal of Information Security, vol.19, no.3, pp.279–290, Springer, 2020.
[21] W. Meng, W. Li, and L. Zhu, "Enhancing medical smartphone networks via blockchain-based trust management against insider attacks," IEEE Trans. Eng. Manag., vol.67, no.4, pp.1377–1386, 2020.
[22] Y. Meng and W. Li, "Adaptive character frequency-based exclusive signature matching scheme in distributed intrusion detection environment," Proc. 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp.223–230, 2012.
[23] W. Meng, W. Li, S. Tug, and J. Tan, "Towards blockchain-enabled single character frequency-based exclusive signature matching in IoT-assisted smart cities," Journal of Parallel and Distributed Computing, vol.144, pp.268–277, 2020.
[24] Y. Mu, F. Rezaeibagha, and K. Huang, "Policy-driven blockchain and its applications for transport systems," IEEE Trans. Serv. Comput., vol.13, no.2, pp.230–240, 2020.
[25] NetScanTools, https://www.netscantools.com/nstpro_packet_generator.html (access on July 2020).
[26] OpenFlow Switch Specification - Open Networking Foundation, https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf
[27] Open vSwitch, an open virtual switch, http://openvswitch.org/ (access on June 2020).
[28] N. Veeraiah and B.T. Krishna, "Trust-aware FuzzyClus-Fuzzy NB: Intrusion detection scheme based on fuzzy clustering and Bayesian rule," Wirel. Netw., vol.25, no.7, pp.4021–4035, 2019.
[29] The POX Controller, https://github.com/noxrepo/pox/ (access on March 2020).
[30] N. Paladi and C. Gehrmann, "Bootstrapping trust in software defined networks," EAI Endorsed Trans. Security Safety, vol.4, no.11, e5, 2017.
[31] R.M.A. Ujjan, Z. Pervez, and K. Dahal, "Snort based collaborative intrusion detection system using blockchain in SDN," Proc. SKIMA, pp.1–8, 2019.
[32] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST Special Publication 800-94, 2007.
[33] R. Sahay, W. Meng, and C.D. Jensen, "The application of software defined networking on securing computer networks: A survey," Journal of Network and Computer Applications, vol.131, pp.89–108, 2019.
[34] What is SDN and where software-defined networking is going, https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html (access on 1 Sept. 2020).
[35] Snort, An open source network intrusion prevention and detection system (IDS/IPS), Homepage: http://www.snort.org/
[36] M. Steichen, S. Hommes, and R. State, "ChainGuard — A firewall for blockchain applications using SDN with OpenFlow," Proc. International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm), pp.1–8, 2017.
[37] K.M.C. Tan, K.S. Killourhy, and R.A. Maxion, "Undermining an

anomaly-based intrusion detection system using common exploits," Recent Advances in Intrusion Detection, RAID 2002, LNCS 2516, pp.54–73, Springer, 2002.
[38] S. Tug, W. Meng, and Y. Wang, "CBSigIDS: Towards collaborative blockchained signature-based intrusion detection," Proc. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp.1228–1235, 2018.
[39] K. Wüst and A. Gervais, "Do you need a blockchain?," CVCBT, pp.45–54, 2018.
[40] Z. Yan, P. Zhang, and A.V. Vasilakos, "A security and trust framework for virtualized networks and software-defined networking," Security and Communication Networks, vol.9, no.16, pp.3059–3069, 2016.
[41] D. Zhang, F.R. Yu, R. Yang, and H. Tang, "A deep reinforcement learning-based trust management scheme for software-defined vehicular networks," DIVANet@MSWiM 2018, pp.1–7, 2018.

Wenjuan Li received her Ph.D. degree in Computer Science from the City University of Hong Kong (CityU), Hong Kong. She received both Research Tuition Scholarships and Outstanding Academic Performance Award at CityU. Her research interests include network management and security, intrusion detection, spam detection, trust management, blockchain and E-commerce security.



Yu Wang received his Ph.D. degree in computer science from Deakin University, Victoria, Australia. He is currently an associate professor with the School of Computer Science, Guangzhou University, China. His research interests include network traffic analysis, mobile networks, social networks, and cyber security.



Weizhi Meng is currently an Associate Professor in the DTU Compute, Technical University of Denmark (DTU), Denmark. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong (CityU). His primary research interests are cyber security and intelligent technology in security, including intrusion detection, smartphone security, biometric authentication, HCI security, trust management, and blockchain.

Jin Li received the Ph.D. degree in information security from Sun Yat-sen University, Guangzhou, China, in 2007. He is currently a Professor with Guangzhou University, China. His current research interests include applied cryptography and cloud computing. He was selected as one of Youth Distinguished Scholars of China, Youth Yangzi-River Scholars of China, and New Stars of Science and Technology in Guangdong Province.

Chunhua Su received the B.S. degree from Beijing Electronic and Science Institute in 2003 and received his M.S. and Ph.D. in computer science from the Faculty of Engineering, Kyushu University in 2006 and 2009, respectively. He is currently working as a Senior Associate Professor in the Division of Computer Science, University of Aizu. His research interests include cryptanalysis, cryptographic protocols, privacy-preserving technologies in machine learning, IoT security & privacy.