https://orcid.org/0000-0002-7476-5757
https://orcid.org/0000-0001-9622-3871
https://orcid.org/0000-0003-2334-2276
https://orcid.org/0000-0002-0530-1828

UDC 004.75

I.A. BURMAKA[*], V.V. LYTVYNOV[*], I.S. SKITER[**], S.V. LYTVYN[*]

EVALUATING A BLOCKCHAIN-BASED NETWORK PERFORMANCE FOR THE INTRUSION DETECTION SYSTEM

[*]Chernihiv National University of Technology, Chernihiv, Ukraine
[**]Institute of Mathematical Machines and Systems Problems National Academy of Science of Ukraine, Kyiv, Ukraine

Анотація. *Системи виявлення вторгнень є важливою частиною кожної корпоративної мережі. Але поточний обсяг мережевого трафіка в корпоративній мережі настільки великий, що централізовані системи виявлення вторгнень не можуть обробити таку кількість трафіка. Тому сучасним корпоративним мережам потрібна розподілена система виявлення вторгнень. Але для великих розподілених систем потрібен механізм встановлення довіри між вузлами. Блокчейн може використовуватися як такий механізм, але більшість випадків використання робочих блокчейнів пов'язані з криптовалютами, де блокчейн успішно використовується як децентралізована база даних, яка зберігає всю історію транзакцій і має механізм перевірки цілісності даних. Таким чином, протоколи блокчейна, які спочатку були розроблені для криптовалют, можуть мати проблеми із продуктивністю, коли вони будуть використовуватися для обміну інформацією про вторгнення в системи. В більшості випадків це відбувається через те, що обсяг даних у системах виявлення вторгнень для запису набагато більший, ніж у мережі криптовалют. Крім того, основна відмінність використання блокчейна для системи виявлення вторгнень і криптовалюти полягає в тому, що криптовалюта має в середньому стабільні обсяги даних для запису, коли система виявлення вторгнень має змінний обсяг даних, який збільшується у випадку аномальних ситуацій у мережі. У цій роботі представлені результати моделювання мережі на основі блокчейна для загальної оцінки продуктивності виявлення вторгнень у систему. Експеримент включає агентське моделювання динаміки параметрів, які мають найбільший вплив на продуктивність мережі. Результати моделювання допоможуть нам знайти оптимальні випадки використання для систем виявлення вторгнень на основі блокчейна. Також він покаже нам слабкі сторони протоколів блокчейна та шляхи покращення продуктивності блокчейна для цього випадку використання.*
Ключові слова: *система виявлення вторгнень, блокчейн, імітаційна модель, оцінка працездатності.*

Аннотация. *Системы обнаружения вторжений являются важной частью каждой корпоративной сети. Но текущий объем сетевого трафика в корпоративной сети настолько велик, что централизованные системы обнаружения вторжений не могут обработать такое количество трафика. Поэтому современным корпоративным сетям нужна распределенная система обнаружения вторжений. Но для больших распределенных систем нужен механизм установления доверия между узлами. Блокчейн может использоваться как такой механизм, но большинство случаев использования рабочих блокчейнов связаны с криптовалютой, где блокчейн успешно используется как децентрализованная база данных, которая хранит всю историю транзакций и имеет механизм проверки целостности данных. Таким образом, протоколы блокчейна, которые первоначально были разработаны для криптовалют, могут иметь проблемы с производительностью, когда они будут использоваться для обмена информацией о вторжении в системы. В большинстве случав это происходит из-за того, что объем данных в системах обнаружения вторжений для записи гораздо больше, чем в сети криптовалют. Кроме того, основное отличие использования блокчейна для системы обнаружения вторжений и криптовалюты заключается в том, что криптовалюта имеет в среднем стабильные объемы данных для записи, когда система обнаружения вторжений имеет переменный объем данных, который увеличивается в случае аномальных ситуаций в сети. В этой работе представлены результаты моделирования сети на основе блокчейна для общей оценки производительности обнаружения вторжений в систему. Эксперимент включает агентское моделирование динамики параметров, которые оказывают наибольшее влияние на производительность сети. Результаты моделирования помогут нам найти оптимальные случаи ис-*

© Burmaka I.A., Lytvynov V.V., Skiter I.S., Lytvyn S.V., 2020

пользования для систем обнаружения вторжений на основе блокчейна. Также он покажет нам слабые стороны протоколов блокчейна и пути улучшения производительности блокчейна для этого случая использования.
Ключевые слова: *система обнаружения вторжений, блокчейн, имитационная модель, оценка работоспособности.*

Abstract. *Intrusion detection systems are the important part of every corporate network. But current amount of network traffic in corporate network is so big that centralized intrusion detection systems can not process such amounts of traffic. So modern corporate networks need distributed intrusion detection system. But big distributed systems need a mechanism of setting trust between the nodes. A blockchain can be used as such kind of mechanism, but most of working blockchains use cases are related to cryptocurrencies, where blockchain is successfully used as a decentralized database which saves all history of transactions and has a mechanism for checking data integrity. So the blockchain protocols which were initially designed for cryptocurrencies can have a performance issues when they will be used for the intrusion detection system information exchange. Mostly because the amounts of data in intrusion detection systems for recording are much bigger then in cryptocurrency network. Also the main difference between blockchain usage for intrusion detection system and cryptocurrency is that cryptocurrency has in average stable amounts of data for recording when intrusion detection system has variable amount of data which is increasing in case of abnormal situations in network. In this paper, the results of the modeling of blockchain-based network to evaluate performance for a collaborative intrusion detection system are presented. The experiment includes the agent-based modeling of the dynamics of parameters which have the biggest impact on the network performance. Modeling results will help us to find optimal use cases for the blockchain-based intrusion detection systems. Also it will show us weak points of blockchain protocols and ways to improve blockchain performance for this use case.*
Keywords: *Intrusion detection system, blockchain, simulating model, performance evaluation.*

DOI: 10.34121/1028-9763-2020-1-99-109

1. Introduction

Cyberattacks are becoming more advanced and complicate every year. To detect network intrusions in time and prevent their critical damage, intrusion detection systems are widely implemented. Intrusion detection systems (IDSs) can be classified as host-based IDS and network-based IDS depending on the deployed location [1]. The first one monitors the local system, when the second one takes information from all the network by monitoring all traffic and analyzes network protocols for malicious events.

Also IDSs can be classified as signature-based IDS and anomaly-based IDS by detection mechanism [2]. The main advantage of anomaly-based detection is a higher rate of detection without frequent signatures updates, but this also can cause a high false positive rate. When the signature based IDS has much lower false positive rate, but needs a frequent signature updated.

Old intrusion detection systems, even the network, were based on a single machine, but in our time the amounts of traffic are very high, so to protect a big corporate network the distributed and collaborative network intrusion detection systems are used. But in big networks it's hard to setup a trust between huge amounts of nodes, because some of the nodes can be modified and can sent some untrusted and irrelevant data then all distributed nodes can work incorrectly.

As a solution of this problem researchers proposed to use a blockchain technology to set up trusted data exchange between nodes [3, 4]. The blockchain, as a continuously growing list of linked records can help to prevent the modification of already accepted data. But blockcain systems as usual have a limited performance, which can be a problem for a system with a lot of nodes. So the main goal is to evaluate a blockchain performance for the blockchain-based intrusion detection systems by modeling the blockchain transaction processing with different system load.

The rest of the paper has been structured as follows. Section 2 discusses the related works. Section 3 outlines the general performance requirements for the blockchain-based intrusion detection system. Section 4 describes the proposed blockchain model and experimental results for a different network size. Conclusions are provided in Section 5.

2. Related work

A separate intrusion detection system usually does not have information about the network which it tries to protect. This leaves opportunity for attackers to bypass IDS examination. In this case, there is a great need for a collaborative system or IDS network to leverage the detection performance of a single IDS [5]. One of the earliest prototypes for a collaborative intrusion detection was a distributed intrusion detection system (DIDS) [6]. It could utilize a distributed monitoring with a centralized data analysis to analyze the heterogeneous network. The next step in DIDS evolution was DOMINO architecture [7] which enables DNIDS deployed at diverse locations to securely share intrusion information between the heterogeneous nodes. All this solutions help to achieve better detection performance, but, as Li et al. [8] figured out, the main problem of these solutions is that most of distributed intrusion detection systems were relaying on the centralized data processing and a distributed part includes only some set of sensors. So they proposed CIDS based on the emerging decentralized location and routing infrastructure. But this mechanism requires that all peers in the network must be trusted. But in a field of collaborative intrusion detection, attacks from inside are the most vulnerable, because by injecting a compromised peer, the intruder can get a control over the whole system. Some researchers think that artificial immune system can improve the collaborative IDS detection accuracy and increase system robustness to internal attacks [9], but for now it's mostly a theoretical concept.

Li et al. [10] identified that different IDS nodes may have different levels of sensitivity in detection of different types of intrusions. They proposed a notion of intrusion sensitivity (IS) that measures the detection sensitivity of an IDS in detection of different kinds of intrusions. Accordingly, they proposed an intrusion sensitivity-based trust management model [11] that could allocate the values of IS by means of a machine learning classifiers. Meng and Kwok proposed the idea of improvement of this idea by decreasing a false alarm level with using a method of voted ensemble selection [12].

But Li et al. [13–15] found that there are few types of attacks which allow to compromise the challenge mechanism and send malicious feedbacks in some situations or to some specific nodes (Passive message fingerprint attack and special On-Off attack).

So unlike standalone IDS, a collaborative intrusion detection system uses information collected from some set of IDS nodes. And one of the main requirements to collaborative IDS is keeping trust between the nodes. On the other hand, information exchange between the nodes must be optimized, because huge amounts of information travelling between the nodes can overload network and make it unusable.

One of the ways to solve a trust problem in decentralized system is to use a blockchain. The name blockchain stems from its technical structure – a chain of blocks. Each block is linked to the previous block with a cryptographic hash. A block is a data structure which allows storing a list of transactions. Transactions are created and exchanged by peers of the blockchain network and modify the state of the blockchain. As such, the transactions can exchange monetary amounts, but are not restricted to financial transactions only and for example allow executing an arbitrary code within so-called smart contracts [16].

How to apply blockchains in the field of intrusion detection is an interesting and important topic. Many studies have started researching in this area. Alexopoulos et al. [17] described a framework of a blockchain-based CIDS, where they considered a set of raw alarms produced by each IDS as transactions in a blockchain. Then, all collaborating nodes employed a consensus protocol to ensure the validity of the transactions before delivering them in a block. This can guarantee the stored alerts are tamper resistant in the blockchain, but they did not implement and evaluate their method in practice.

Menget et al. [4] says that the blockchain technology will help to improve the distributed intrusion detecting mostly in aspects of data sharing, alarm exchange and trust computation. Sharmaet et al. [18] proposed DistBlockNet, a distributed secure software defined networking architecture for the internet of things by integrating the blockchain technology, allowing a node to interact with others without the need of a trusted central controller.

3. Architecture and basic requirements

It is very important on the first step of a system design to clearly articulate the basic requirements of the goal system. In accordance with related works [17, 19, 20] we can highlight the most basic requirements for a collaborative intrusion detection system.

– Accountability: Participating parties should be held accountable for their actions.

– Integrity: The integrity of the alert data is very important for detection attacks over time.

– Resilience: The system should not have SPoFs and should not depend on small groups of participants.

– Consensus: The system should be able to reach consensus on the quality of individual alert data and on the trust worthiness of each participant.

– Scalability: The system should be able to scale to a large number of participants/monitors and also handle churn.

– Minimum Overhead: The communication and computation overhead should be kept as low as possible.

These requirements will guide us on choosing the architecture of CIDS. Of course there are some tradeoffs between these requirements, so we cannot improve all parameters together. But here we will pay attention to the parameters related to a blockchain architecture solution.

3.1. Do we need a blockchain in IDS?

One of the main requirements in our case is a consensus requirement. The classical approach to implement it is the usage of centralized third-party trusted nodes. But using a centralized information exchange for a distributed IDS is not a good idea because increasing the number of distributed nodes will increase the load on a central trusted node and this node will be the SPoF of the system.

In the distributed intrusion detection system the blockchain technology can be used for a secure and trusted peer-to-peer data exchange between the nodes. But using a blockchain is not a solution of all our problems and in some cases a blockchain can be a problem.

There are two types of a blockchain: a permissionless blockchain and a permissioned blockchain. A permissioned blockchain also can be separated on public and private (permissionless is a public by default).

A permissionless blockchain is used in most of cryptocurrencies and the main feature of a permissionless blockchain is that any peer can join or leave the network at any time and there is no central entity which can manage the membership of the nodes. Also there is no way to exclude some nodes from writing to a blockchain [16].

Permissioned blockchains have been proposed to only authorize a limited set of readers and writers. Here, a central entity decides and attributes the right to individual peers to participate in the write or read operations of the blockchain. To provide encapsulation and privacy, a reader and writer could also run in separated parallel blockchains that are interconnected. In private blockchain information could be read only by authorized peers, public blockchain could be read and verified by anyone [16, 21].

A blockchain can be updated via a consensus protocol, which ensures all participating entities to agree on a uniform view of the ledger. A consensus protocol can be dependent on a specific blockchain implementation and threat model [4].

– Proof of work. This method allows a node to successfully accept a block, when a predefined amount of computational resources (known as “work”) can be proved to spent [22]. This type of a consensus protocol is not suitable for IDS because it needs spending a lot of computational resources on specific mining nodes (in most of cases mining nodes can’t do anything else because of a high load of processors) which means high spend of energy, so this system will be very expensive.

– Proof of stake. This method ensures a consensus to be achieved by considering both a random selection and the influence (known as “stake”) of the participating entities. It is assumed that entities would guarantee the integrity of blocks when they have a large stake in the blockchained network [23]. This type of a consensus protocol is one of the most suitable for IDS but needs to be adopted for this case, because we need to choose something to stake (can be some virtual currency). This type of consensus protocol does not need a permanent high computation load.

– Proof of elapsed time. This method ensures a consensus to be achieved by requesting every potential verifier to share a secure and random waiting time from a trusted execution environment. Each user, once generating a block, also needs to generate a proof for the waiting activity with the assistance of SGX hardware, which is submitted together with the block. The proof of elapsed time also looks suitable for IDS because it is based on waiting, so it also doesn’t need a high computation power. But Lin Chen et al. showed vulnerabilities of a current protocol design [24].

In [16], Wust and Gervais outline a very general decision process that allows determining whether – and if yes, which type of – a blockchain makes sense for your case.

The main criteria are whether the application needs to store state, number and trust to writers. In general if we don’t need to store the state or we have no more than one writer we don’t need blockchain, because it will be much slower than using a simple database. On the other hand if we have multiple writers, blockchain can help to keep the integrity of records, replacing trusted third party which must be always online. But a blockchain is still not needed if all our writers are known and trusted.

If all writers are known but not necessarily trusted, we can use a permissioned blockchain. Which kind of a permissioned blockchain (public or private) will depend on the question whether we need public verifiability of the records? If this is the case – then anyone can join the network as a reader, implying a public permissioned blockchain. Otherwise we need to use a private permissioned blockchain. But we need to keep in mind that even a public permissioned or permissionless blockchain needs encryption or hashing for records protecting.

If set of writers is not known or can fluctuate greatly – a permissionless blockchain will be a good solution.

If we map these cases on plain of intrusion detection solutions we will have two cases when we can use a blockchain. First of all it’s a private network intrusion detection system inside some company, or between a few companies which trust each other and can have some private information. In this case we will use a permissioned blockchain. Another case is an open intrusion detection network, which can be joined by anyone. In this case the best solution will be a perissionless blockchain.

4. Blockchain performance

So a blockchain helps us to solve a trust problem, but we need to keep in mind that blockchain has an impact on the information exchange speed. So we need to evaluate the performance of a blockchain and hardware resources usage with different network parameters.

4.1. How blockchain works

First of all let's take a look on how a blockchain process our transactions (here we mean that transaction is any portion of information for exchange between nodes which will be written to the blockchain). Usually blocks in a blockchain are generated with some stable frequency which is defined in blockchain rules, so transactions cannot be written into block immediately.

At the first step, when the transaction created, the node stores it in memory pool and broadcasts to peers. On receiving a transaction each node also stores it in memory pool and broadcast. So transaction will be propagated over the whole network.

When the new block is generated the node which generated it writes transactions from the memory pool to block. Then the node broadcasts created block to peers. After receiving a new block each node verifies that the block satisfies all protocol rules, and that all transactions in the block follow the protocol and application rules. After verifying a received block, the node updates its view of the blockchain, clears included transactions from the memory pool, and broadcasts the block further [25].

But one of the problems of the blockchain is that blocks in most cases have constant number of transaction records. That's why there are two factors which define how many transactions can be processed. First one is a maximum number of transactions in the block. The second one is a block generating period. So average number of transactions per block time must be smaller than a system can process, or memory pool queue will constantly grow and system will crash.

4.2. Creating blockchain model

To evaluate a blockchain performance we decided to create a simulation model with any logic system. The best way to simulate a blockchain work is to use an agent based model where each network node will be represented by a separate agent. Of course the nodes are connected into a peer-to-peer network. In most blockchain networks, each node is connected to some limited number of neighbors (for bitcoin it's 8 outgoing connections to peers and up to 125 connections in general [26, 27]). To model this connection strategy any logic has a standard network type which is called “Small world”. This type of a network means that agents are connected with neighbors and form a ring, but some connections are established to far nodes, so that this can increase a network message propagation speed [28].

Our model will be a little simplified, so we will test a blockchain work in a normal system state and a system under attack. In a normal state modeling we define the alert frequency about 1 alert in 33 seconds, because as we know most of systems which have a direct access to the Internet and public IP address are constantly under small attacks like port scanners or some kind of brute force. Of course it happens not so frequently, so the rate which we choose for an alert generation will describe a real malicious activity in a normal state correctly.

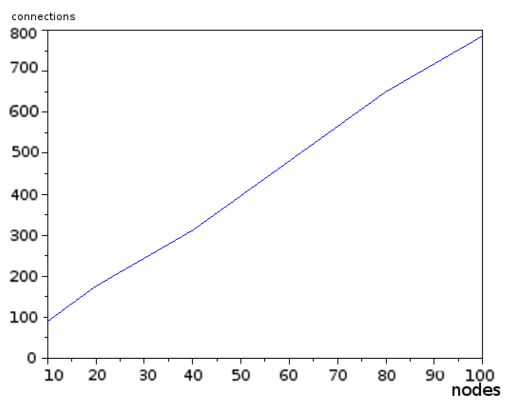
As for an abnormal system state we choose an alert rate as one alert per second. For now it's hard to say if it is enough, but of course we cannot create a separate event for every malicious packet, because it will cause a huge system overload. If we have a lot of malicious events we can put a few closes in time events into a single alert. So for now we are using this rate to represent a general system behavior under attack.

Alerts on the network nodes are generated randomly with exponential distribution which is suitable for setting timing for incoming calls or messages. Generated alert is added to memory pool queue (memory pool queue in our case are separate for each node so that we can explore message distribution between the nodes) and distributed to all connected nodes as in a real blockchain. Each node after receiving message also adds it to the memory pool queue and redistribute to neighbors.

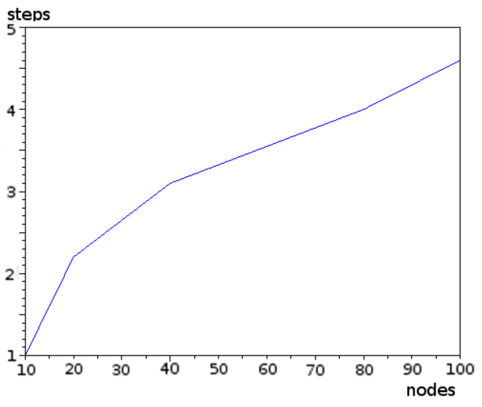
Another task of our blockchain model is simulating blocks generation. Here we are using a standard for most blockchains scheme – blocks are generated with almost constant timeout (in our case constant). For model simplification let's suggest that all nodes have the same probabilities to generate a new block (node choice does not have a significant impact on a general blockchain performance). Then a node which was chosen for generating block creates a new block and records to it some amount of transactions from the memory pool but no more than maximum amount for block (block size).

And now let's take a look at points which can have impact on the performance of a distributed blockchain-based network. The first one is the number of nodes and related to the number of connections and number of message propagation steps. As a step we mean a situation when a message is received by the node and then reseeded to other nodes.

Our simulation shows that with almost constant number of connection per node we have a linear dependency between the number of connections and the number of nodes (Figure 1(a)).



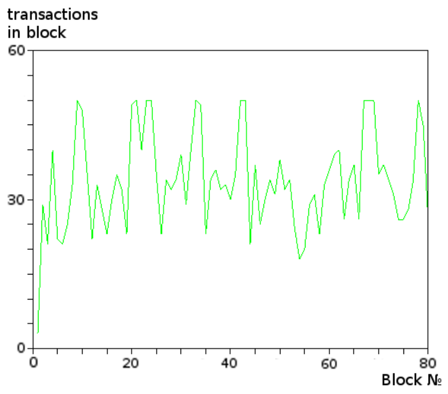
(a)



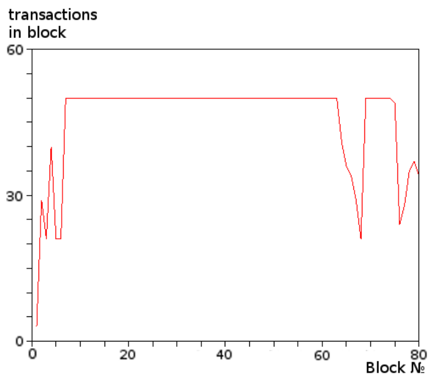
(b)

Figure 1 – Number of connections (a) and steps for full message propagation (b) depending of number of nodes

But the number of connections causes a huge performance impact only on decentralized system, because central nodes must process them all, but with the decentralized network each node has a limited number of connections. And instead of sending a message directly to all nodes in the network, the message is sent and received a few times by network nodes until it reaches the last node and will be fully propagated over the network. So in this case the number of steps will be more important, because the number of steps will be proportional to the message propagation time. So let's look how the number of message propagation steps depends on nodes count (Figure 1(b)).



(a)



(b)

Figure 2 – 20 nodes block filling in abnormal state (a) and under attack state (b)

As we see the number of steps is not increasing so significantly (for 1000 nodes it's only about 6 steps), so this type of a decentralized network is suitable for making a big decentralized network without powerful central nodes and message exchange will be pretty fast.

But the weakest area of a blockchain-based network is a writing transaction (in our case it's alerts) to blocks. And the main point here is to keep balance between keeping blocks filled and having reserved resources for a case of attack or some network anomalies. But this will cause a fast increasing of a blockchain size by huge blocks which are much harder to propagate and verify. And if a disk space is not a big problem in our days, when we have a really big hard drives with more than 10TB, block propagation and the verification can be a problem, because all nodes must accept a new block before the next block is generated. So most cryptocurrencies are choosing a smaller block size for their blockchain. For example at this time bitcoin has a theoretical block size of 4 megabytes [29] and blocks are generated every 10 minutes. So this kind of block can contain up to 8000 transactions with a size of 512 bytes, it's about 13 transactions per second. It's still suitable for cryptocurrency, but will be not so good solution for an intrusion detection system.

But here we have one significant difference with cryptocurrencies – cryptocurrency in average has a stable level of transactions. But in the case of intrusion detection most of time blocks will be almost empty, but when the system is under attack, blocks will be filled by a huge amount of transaction.

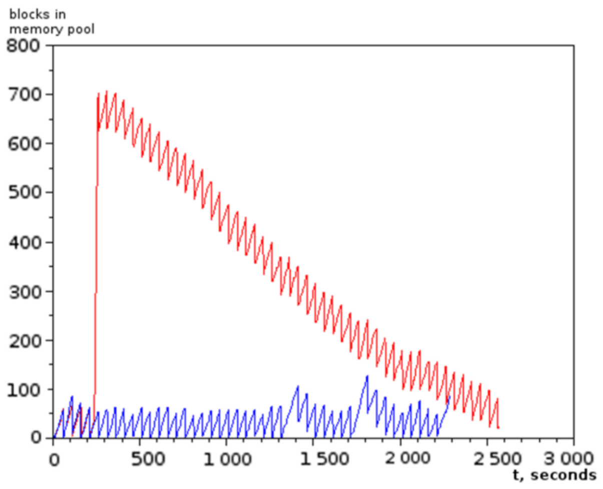
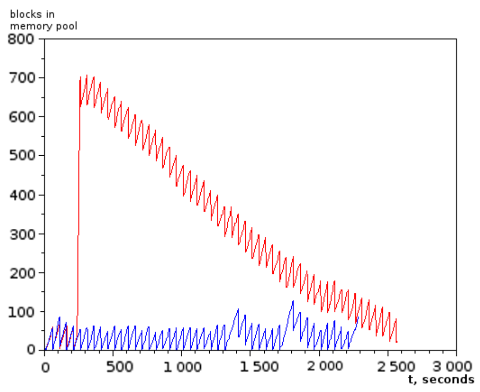


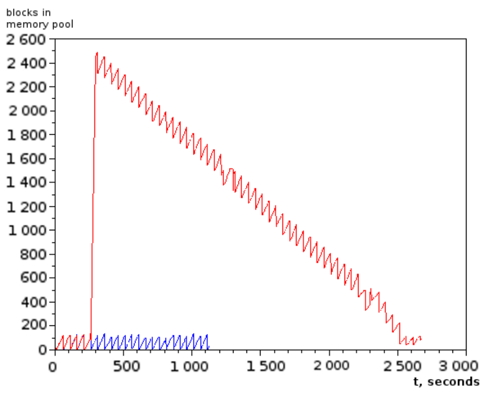
Figure 3 – Memory pool usage on 20 nodes blockchain (50 transactions per block)

For our model we defined blocks with a constant number of transaction not an amount of information because we simplified a model by making all transaction the same size. And now we are going to evaluate block filling and the memory pool usage for our model with different number of nodes. So let's start with 20 nodes (Figure 2).

Of course a block size which we choose (for 20 nodes we choose 50 transactions per block and one block per minute) is a little small for under attack state and close to limit for a normal state with some random anomalies. But from this plots we can just see that under attack the blocks are fully used so let's take a look at memory pool usage (Figure 3).



(a)



(b)

Figure 4 – Memory pool usage on (a): 40 nodes (80 transaction per block), (b): 80 nodes (160 transaction per block)

Here we can see that the attack causes a huge increasing of a memory pool, and recovering pool size to normal state takes about a 30 times more time than attack duration and about a 64 blocks. So let's take a look at the similar plots for 40 nodes (Figure 4 (a)) and 80 nodes (Figure 4 (b)).

Here we can see the same situation, even for the short time attacks it takes a lot of time to utilize all transactions from the memory pool (for these two cases it takes about 45 blocks). And it's really bad situation for a blockchain-based system, because while a memory pool contains a lot of still unused transactions system is unstable. And such long transaction utilizing time makes the system vulnerable for long attacks because the memory pool can be overfilled and the node can crash.

4.3. Ways to improve

So a blockchain optimized for a normal state of IDS is not suitable because of unstable behaviour under the attack. But there are a few ways to improve this situation. The first way is to search optimal parameters for some IDS network by increasing a maximum block transaction count and generating blocks more often. But this choice is suitable only for a small network with limited amount of nodes, because if a block size is too big or blocks are generated too often then we are returning to performance problems which were described earlier.

Another way to improve under attack performance is to decrease the number of transactions by using advanced transaction packaging and compression, so that a set of transactions becomes packed into a single one (main transaction data can be saved outside the blockchain and blockchain will contain just minimal information for transaction verification). But if we are using this optimization we are increasing a transaction delay. On the other hand processing a bigger amount of data can help to compress the transaction more effective than a set of small transactions.

One more way of improvement is a blockchain modification which adds a possibility of a dynamic block size and dynamic bock timeout. Of course there must be some limitations, but making a block for example n times bigger and making timeout m times shorter can increase the memory pool utilization speed up $m \times n$ times. But the main problem here is that this solution needs not standard blockchain with modified consensus protocol which will allow to control block generation time.

5. Conclusion

A blockchain technology is showing good results for providing a verifiable manner for information exchange in decentralized intrusion detection systems. So it is possible to use this technology to decrease impact of malicious nodes in collaborative IDS which can generate untrustful signatures. Motivated by recent researches related to using a blockchain in IDS, we focused on evaluating a blockchain performance for a blockchain when it's used with IDS.

The result of our modeling shows that a blockchain, as it is used now in cryptocurrencies, is not an optimal solution for intrusion detection systems, because transactions rates are significantly different. But there are two ways to optimize a blockchain performance. First of all its decreasing the number of transactions per second, which can be done without blockchain protocols modification. Another option is to modify a blockchain consensus protocol to make it possible to create blocks with a variable size and generating timeout.

This work is an early research study in this area, showing blockchain effectiveness for using in collaborative IDSs. The main purpose is to complement the literature and stimulate more research on this topic.

REFERENCES

1. Shelke M.P.K., Sontakke M.S., Gawande D.A.D. Intrusion detection system for cloud computing. *International Journal of Scientific & Technology Research*. 2012. Vol. 1, N 4. P. 67–71. URL: http://www.ijstr.org/paper-references.php?ref=IJSTR-0512-5114.
2. Aljawarneh S., Aldwairi M., Yassein M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*. 2018, Vol. 25. P. 152–160. URL: http://www.sciencedirect.com/science/article/pii/S1877750316305099.
3. Li W., Tug S., Meng W., Wang,Y. Designing collaborative blockchained signature-based intrusion detection in IoT environments. *Future Generation Computer Systems 96*. 2019. Jul. P. 481–489. URL: http://www.sciencedirect.com/science/article/pii/S0167739X18327237.
4. Meng W., Tischhauser E.W., Wang Q., Wang Y., Han J. *When intrusion detection meets blockchain technology: a review*. 2018. Vol. 6. P. 10179–10188.
5. Wu Y.S., Foo B., Mei Y., Bagchi S. Collaborative intrusion detection system (CIDS): a framework for accurate and efficient IDS. *19th Annual Computer Security Applications Conference*. *Proceedings*. 2003. Dec. P. 234–244.
6. Snapp S., Brentano J., Dias G., Goan T., Heberlein L., Ho C., Levitt K., Mukherjee B., Smaha S., Grance T.D. Distributed intrusion detection system – motivation, architecture, and an early prototype'proc. *The 14th national computer security conference*. Washington, 1991. P. 167–176.
7. Barford V.Y.P., Jha S., Li Z., ChenY., Beach A. Global intrusion detection in the domino overlay system. *Proceeding of NDSS*. 2004. Vol. 4, N 8. Towards Scalable and Robust Distributed Intrusion Alert Fusion with Good Load Balancing. *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense*. LSAD. New York, NY, USA. 2006. P. 115–122. URL: http://doi.acm.org/10.1145/1162666.1162669, event-place: Pisa, Italy.
9. Afzali Seresht N., Azmi R. MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach. *Engineering Applications of Artificial Intelligence*. 2014. Vol. 35. P. 286–298. http://www.sciencedirect.com/science/article/pii/S0952197614001444.
10. Li W., Meng Y., Kwok L.F. Enhancing Trust Evaluation Using Intrusion Sensitivity in Collaborative Intrusion Detection Networks: Feasibility and Challenges. *Ninth International Conference on Computational Intelligence and Security*. 2013. Dec. P. 518–522.
11. Li W., Meng W., Kwok L.F. Design of Intrusion Sensitivity-Based Trust Management Model for Collaborative Intrusion Detection Networks. *Trust Management VIII*. IFIP Advances in Information and Communication Technology / eds. J. Zhou, N. Galoz, J. Zhang, E. Gudes. Springer, Berlin, Heidelberg, 2014. P. 61–76.
12. Meng Y., Kwok L.F. Enhancing False Alarm Reduction Using Voted Ensemble Selection in Intrusion Detection. *International Journal of Computational Intelligence Systems*. 2013. Vol. 6, N 4. P. 626–638. URL: https://doi.org/10.1080/18756891.2013. 802114.
13. Li W., Meng W., Kwok L.F. Ip, H.H.S.: PMFA: Toward Passive Message Fingerprint Attacks on Challenge-Based Collaborative Intrusion Detection Networks. *Network and System Security*: Lecture Notes in Computer Science / eds. J. Chen, V. Piuri, C. Su, M. Yung. Springer International Publishing, Cham. 2016. P. 433–449.
14. Li W., Meng W., Kwok L.F. Ip H.H.S. Developing advanced fingerprint attacks on challenge-based collaborative intrusion detection networks. *Cluster Computing*. 2018. Vol. 21, N 1. P. 299–310. URL: https://doi.org/10.1007/s10586-017-0955-8.
15. Li W., Meng W., Kwok L.F. Investigating the Influence of Special on Attacks on Challenge-Based Collaborative Intrusion Detection Networks. *Future Internet*. 2018. Vol. 10, N 1. 6 p. URL: https://www.mdpi.com/1999-5903/10/1/6.
16. Wust K., Gervais A. Do you Need a Blockchain? *Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2018. Jun. P. 45–54.
17. Alexopoulos N., Vasilomanolakis E., Ivanko N.R., Muhlhauser M. Towards Blockchain-Based Collaborative Intrusion Detection Systems. *Critical Information Infrastructures Security*: Lecture Notes in Computer Science / eds. G. D'Agostino, A. Scala. Springer International Publishing, Cham, 2018. P. 107–118.
18. Sharma P.K., Singh S., Jeong Y.S., Park J.H. DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks. *IEEE Communications Magazine*. 2017. Vol. 55, N 9. P. 78–85.

19. Vasilomanolakis E., Karuppayah S., Muhlhauser M., Fischer M. Taxonomy and Survey of Collaborative Intrusion Detection. *ACM Comput. Surv.* May 2015. Vol. 47, N 4. P. 55:1–55:33. URL: http://doi.acm.org/10.1145/2716260.
20. Vasilomanolakis E., Habib S.M., Milaszewicz P., Malik R.S., Muhlhauser M.: Towards Trust-Aware Collaborative Intrusion Detection: Challenges and Solutions. *Trust Management XI.* IFIP Advances in Information and Communication Technology / eds. J.P. Steghofer, B. Esfandiari. Springer International Publishing, Cham. 2017. P. 94–109.
21. Vukolic M. Rethinking permissioned blockchains. *Proc. of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts.* ACM. 2017. P. 3–7.
22. Gervais A., Karame G.O., Wust K., Glykantzis V., Ritzdorf H., Capkun S. On the security and performance of proof of work blockchains. *Proc. of the 2016 ACM SIGSAC conference on computer and communications security*. ACM. 2016. P. 3–16.
23. Tosh D.K., Shetty S., Liang X., Kamhoua C., Njilla L. Consensus protocols for blockchain-based data provenance: Challenges and opportunities. *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON).* 2017. Oct. P. 469–474.
24. Chen L., Xu L., Shah N., Gao Z., Lu Y., Shi W. On security analysis of proof of-elapsed-time (poet). *International Symposium on Stabilization, Safety and Security of Distributed Systems.* Springer, 2017. P. 282–297.
25. Gupta Y., Shorey R., Kulkarni D., Tew J. The applicability of blockchain in the Internet of Things. *2018 10th International Conference on Communication Systems Networks (COMSNETS).* 2018. Jan. P. 561–564.
26. Antonopoulos A.M. Mastering Bitcoin: Programming the open blockchain. O'Reilly Media, Inc. 2017. 113 p.
27. Bitcoin Developer Reference – Bitcoin. URL: https://bitcoin.org/en/developer-referencen #remote-procedure-calls-rpcs.
28. Help – AnyLogic Simulation Software. URL: https://help.anylogic.com/index.jsp?topic=/com.anylogic.help/html/agentbased/Connections+and+Networks.html.
29. What Is the Bitcoin Block Size Limit? *Bitcoin Magazine*. URL: https://bitcoinmagazine.com/guides/what-is-the-bitcoin-block-size-limit.

Стаття надійшла до редакції 20.01.2020