$\mathcal{B}$SIEM-IoT: A blockchain-based and distributed SIEM for the Internet of Things

Andrés Pardo Mesa[1][0000−0002−3410−8330], Fabián Ardila Rodríguez[1][0000−0002−2055−7901], Daniel Díaz López[1][0000−0001−7244−2631], and Félix Gómez Mármol[2][0000−0002−6424−3322]

[1] Colombian School of Engineering Julio Garavito, Bogota, Colombia
{andres.pardo-m, fabian.ardila}@mail.escuelaing.edu.co
daniel.diaz@escuelaing.edu.co
[2] Faculty of Computer Science, University of Murcia, Murcia, Spain
felixgm@um.es

Abstract. The paper at hand proposes $\mathcal{B}$SIEM-IoT, a Security Information and Event Management solution (SIEM) for the Internet of Things (IoT) relying on blockchain to store and access security events. The security events included in the blockchain are contributed by a number of IoT sentinels in charge of protecting a group of IoT devices. A key feature here is that the blockchain guarantees a secure registry of security events. Additionally, the proposal permits SIEM functional components to be assigned to different miners servers composing a resilient and distributed SIEM. Our proposal is implemented using Ethereum and validated through different use cases and experiments.

Keywords: IoT · Intrusion detection system · Blockchain · SIEM

1 Introduction

The Internet of Things (IoT) has brought uncountable benefits in a number of diverse and relevant environments. Yet, one of its current major drawbacks lies in the lack of security solutions to protect these systems against cyber attacks. One approach in this regard consists in processing the security events coming from such ecosystem and use them to prevent, detect and mitigate security incidents [2]. Security events, stemming either from IoT devices or from intermediate security components, are collected and sent toward a centralized Security Information and Event Management (SIEM) server to detect such incidents using one of its available modules (correlation rules, policies, statistic models).

In this regard, the integrity of the security events is critical, since an alteration of this data could induce false alarms. Likewise, availability is another security requirement for those security events: all the security events should be available to the SIEM modules in a timely manner, as well as resilient against denial attacks. Furthermore, traceability is also a key requirement here. A comprehensive registry of all event operations should be kept and maintained to support an effective audit in case of a potential security violation.

Finally, a centralized architecture to detect intrusions in IoT ecosystems constitutes a single-point of attack and a bottle-neck that in case of failure would impact adversely all related security functions, mainly containment and recovery. Thus, resiliency becomes another requirement for the security infrastructure, so the security functions can not be interrupted.

In this paper, we present $\mathcal{B}$SIEM-IoT, a blockchain-based and distributed SIEM to detect attacks against IoT devices. This proposal is built over a blockchain architecture, allowing interoperability between components of the IoT ecosystem that contribute information related to security events. Every security event is effectively protected in terms of integrity and non-repudiation due to the intrinsic features of the blockchain [7]. Further, smart contracts (SC) [8] in the blockchain guarantee a consistent behavior of the system, including the authorization of actions over the security events. $\mathcal{B}$SIEM-IoT is able to consume local threat intelligence, enabling the detection of distributed attacks which can only be discovered by correlating security events coming from different sources. Moreover, our proposal connects to different external sources to get updated threat intelligence and improve the analysis of the security events within the blockchain.

The main contributions of this paper are as follows:

- A distributed SIEM proposal for IoT scenarios leveraging the benefits of a blockchain (server-less operations, integrity, non-repudiation and resiliency).
- Development of methods in a smart contract to handle blocks of security events and detect attacks from the security events available in the blockchain.
- Integration of the *External* and the *Internal Threat Intelligence* of the $\mathcal{B}$SIEM-IoT to make local validations originated in smart contracts.
- The evaluation of the proposal and its features through exhaustive experiments, which in turn proved the feasibility of the solution for organizations.

2 Background

Blockchain is a decentralized P2P network where all transactions are validated by all the nodes and recorded in a distributed and immutable ledger. Consensus is the core of the blockchain technology as it guarantees the reliability of the network, and some of the existing types are presented next [11]:

- Proof of Work (PoW): A transaction is approved if at least half plus one of the nodes in the P2P network accept it.
- Proof of Stake (PoS): The node who has more wealth has greater probability to participate in the consensus and create a block.
- Proof of Importance (PoI): The nodes that can create a block are the ones with the greatest number of transactions into the network.
- Proof of Authority (PoA): Only some nodes are explicitly allowed to create new blocks and secure the blockchain.

In general, blockchain proposes two key ways to build a network [9], namely, permissioned and permissionless blockchains, being the main difference the level

of governance implemented by each node. Permissionless blockchains (i.e public blockchains) allow anyone to become a node and belong to the network. Nodes on this blockchain can perform any task if they have the physical capability (e.g., mine blocks, validate transactions, etc.). In turn, permissioned blockchains (i.e. private blockchains) restrict the nodes belonging to the network and performing tasks. A relevant feature of this kind of blockchain is that it may choose the level of decentralization on the network, i.e., fully or partially decentralized.

With blockchain one can develop *Decentralized Applications* or *DApps*. To do so, a *Dapp* requires a back-end component, and in this regard, blockchains implement *smart contracts* (*SC*) to support any required operation by the application logic. Ethereum [1] is an open source platform to create *smart contracts*.

3 State of the art

A number of proposals have arisen in the last years to protect IoT ecosystems. Thus for instance, [2] proposes a security architecture employing security events. Such architecture relies on a multi-relation between: i) security events categories, providing information about the impact of an attack over a given IoT device, ii) vulnerabilities, to explain the causes of the attack, and iii) attack surfaces, yielding information on how the attack was conducted.

In turn, authors of [4] propose an IoT security framework for a smart home scenario. This framework applies a novel instance of blockchain by eliminating the concept of PoW and the need of coins. This work relies on a hierarchical structure that coordinates methods over the blockchain network to keep the security and privacy benefits offered by this technology. Such hierarchical structure is more suitable for the specific requirements of IoT since tasks on the network are performed in a different and adjusted manner than a common blockchain such as Bitcoin [3]. The framework proposes to manage the network and the belonging devices with the methods *store, access, monitor, genesis* and *remove*.

A blockchain-based framework to support access control in IoT is introduced in [10], implementing multiple smart contracts: i) Access Control Contract (ACCs) to manage the authorization of users over an IoT device, ii) Judge Contract (JC) to implement a misbehavior-judging method to facilitate the dynamic validation of the ACCs, and iii) Register Contract (RC) to register the information of the access control and misbehavior-judging methods plus their smart contracts. When an access request arrives to the framework, different validations are done with the smart contracts before resolving such request.

In addition, [5] investigates on the applicability of a blockchain to develop the next-generation SIEM 3.0 systems, designed to detect information security incidents in a modern and fully interconnected organization network environment. This work brings the next generation of SIEM to a qualitatively new and higher level by proposing a methodology for its evaluation based on the *B method*, the most popular formal method to be used in industry projects and safety-critical system applications to allow for highly accurate expressions of the properties required by specifications and models systems in their environment.

As observed, there are already works dealing with cyber security for IoT scenarios and blockchains to tackle different IoT challenges. In particular, we found that blockchain has been applied to support IoT operations like data synchronization, communication or access control. In the paper at hand, we propose $\mathcal{B}$SIEM-IoT which, in contrast to all previous proposals, is specifically focused on the management of IoT security events. Our proposal brings the principal security features of blockchain to a regular SIEM to finally compose a security solution which is specifically focused on IoT, resilient, trust-oriented, auditable and scalable. To our best knowledge, there is no security solution applicable to IoT ecosystems holding these attributes with verifiable functionality.

4 $\mathcal{B}$SIEM-IoT

Our proposed blockchain-based and distributed SIEM for IoT, $\mathcal{B}$SIEM-IoT, validates and analyzes the compilation of security events stored in a distributed ledger of a blockchain that keeps completely safe all the information against any kind of unexpected modification. Additionally, our solution uses both internal and external threat intelligence to identify suspicious behaviors and promptly warn about an in-progress attack. Thus, $\mathcal{B}$SIEM-IoT must satisfy these goals:

- **Resilient**: In order to offer a high availability of security services, the solution should provide a go on alive capability, ensuring protection of IoT devices and attack detection, even if the SIEM gets in a hostile situation.
- **Trust-oriented**: Only trusted nodes, i.e., IoT sentinels [6], must be allowed to create transactions containing security events, avoiding data pollution.
- **Auditable**: The solution must be able to audit the block of events to identify key elements in an incident response procedure, such as identifying node(s) that issued an event or discovering causality relation between events.
- **Scalable**: The solution should be able to integrate new IoT Sentinels into the blockchain network without impacting adversely other existing nodes.

It is important to understand that a blockchain network is composed of *nodes*. While the IoT sentinels are the only ones who may create transactions in the blockchain, solely some special nodes, called miners, can receive transactions and mine (create) new blocks to be added to the blockchain. Moreover, both IoT sentinels and miners participate in the consensus algorithm.

The architecture of our proposal $\mathcal{B}$SIEM-IoT$= (\mathcal{D}, \mathcal{S}, \mathcal{M}, \mathcal{T})$ is shown in Figure 1, encompassing the following elements: IoT devices ($\mathcal{D}$), IoT sentinels ($\mathcal{S}$), distributed SIEMs (miners $\mathcal{M}$) and external Threat Intelligence providers ($\mathcal{T}$).

4.1 IoT devices

IoT devices ($\mathcal{D} = \{D_1, \ldots, D_{n_D}\}$) are widely deployed nowadays, including scenarios like smart homes and smart offices, amongst others. Wherever they operate, they communicate with each other and/or with other entities in the overall Internet. Due to the negative impact that a successful cyber attack would have on these (usually unprotected) devices, their communications must be secured.

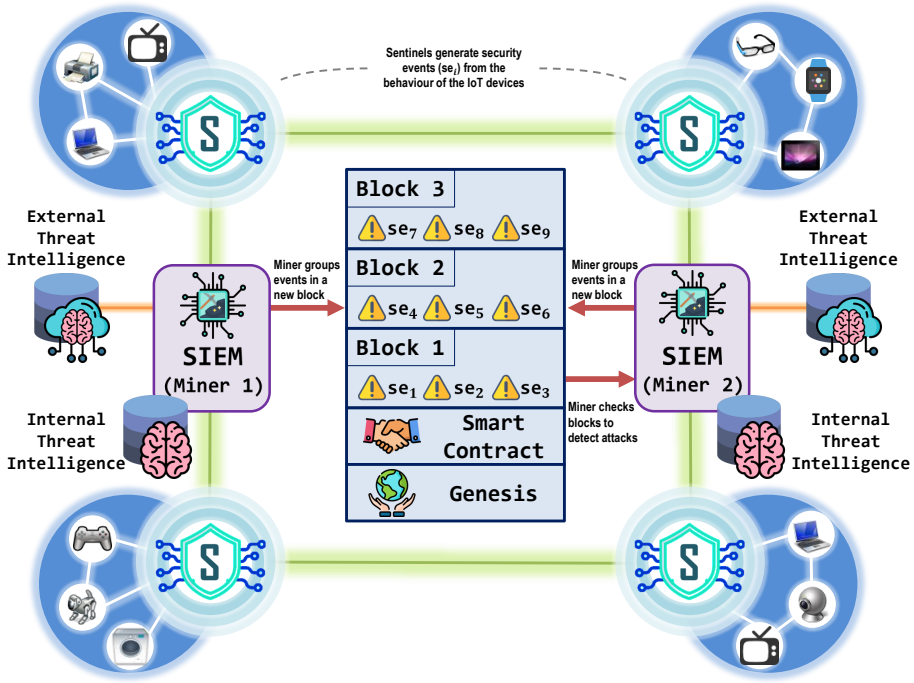

Fig. 1. Architecture of the blockchain-based and distributed SIEM, $\mathcal{B}$SIEM-IoT

4.2 IoT sentinels

IoT sentinels ($\mathcal{S} = \{S_1, \ldots, S_{n_S}\}$) are in charge of shielding all the IoT devices in their nearby against cyber attacks. In this regard, whenever an intrusion attempt happens, the IoT sentinels generate the corresponding security event se_i and integrate it into a transaction that will subsequently be sent to the distributed SIEM (miner), who will evolve it into a block and add it to the blockchain.

Thanks to the benefits offered by the blockchain network, the sentinel here is only required to gather and keep a small portion of security events before creating an actual transaction. Thus, the sentinel just needs to run a lightweight blockchain client, turning such sentinel into a new node of the blockchain with the capacity to create transactions and to participate in a consensus validation.

The lightweight blockchain client allows the sentinel to handle smart contracts, and such smart contracts, in turn, are employed to execute several useful operations. For instance, the sentinel is able to format new security events and add them to a transaction when a given threshold of collected security events is reached. Likewise, the sentinel can also delete a specific security event from an already created transaction, so to avoid storing trash data.

4.3 Distributed SIEM (*miner*)

In contrast to the IoT sentinels, the distributed SIEM acts as the miner node in the blockchain ($\mathcal{M} = \{M_1, \ldots, M_{n_M}\}$) and must have the highest hardware features. Hence, the distributed SIEM is in charge of creating new blocks containing the transactions got from the IoT sentinels. To this end, the miner employs its computational power to solve a challenge in the blockchain network.

Moreover, the distributed SIEM will decode the information received from the lightweight blockchain client (running within the IoT sentinel) and transfer it to the external Threat Intelligence under specific formats, depending on the IoT source where the data were collected.

4.4 External Threat Intelligence

The external Threat Intelligence ($\mathcal{T}$) is provided by a third-party service analyzing malware campaigns addressed to the most prominent industries and identifying Indicators of Compromise (IoC) and Indicators of Attack (IoA) that can help another organization to detect an ongoing attack or to investigate a past attack sharing some common features with a known attack.

Intelligence information delivered by an external Threat Intelligence provider is definitely useful for $\mathcal{B}$SIEM-IoT, as it may use it to analyze security events that exist in the blockchain and consequently detect IoT attacks. $\mathcal{B}$SIEM-IoT is also able to incorporate this info from a third-party into its internal Threat Intelligence database, so it can be usable in the attacks detection. It is important to note that $\mathcal{B}$SIEM-IoT is a distributed solution composed by a set of SIEMs, each one having different security functions and even connected to different external Threat Intelligence providers

5 Use cases

5.1 Adding blocks of security events to the blockchain

As stated before, IoT sentinels are the only nodes in the blockchain network able to generate transactions containing security events. Yet, this action should only be granted when such devices are trustworthy enough. The novel implementation of $\mathcal{B}$SIEM-IoT includes a strategic permissioned operation mode to guarantee the control and reliability of the information to be added to the blockchain.

Further, for the sake of efficiency, IoT sentinels may also group security events and include them all within the same transaction. This feature avoids creating one block for each security event, which could impact the performance of the blockchain. Thus, the **Threshold of Security Events** ($\lambda_{se} > 0$) is defined as the minimum number of events that must be grouped to create a transaction and is set previously in the configuration of the sentinels.

Finally, whenever a transaction is created by an IoT sentinel, the latter sends it to a distributed SIEM, who will in turn mine a new block with such transaction and add it to the blockchain.

5.2 Consuming the blockchain to detect distributed attacks

When $\mathcal{B}$SIEM-IoT is launched, IoT sentinels start building security events for every incident they detect. Hence, when a distributed cyber attack arises in the protected network, aiming at different IoT devices, the IoT sentinels shielding each of those victim IoT devices generate the corresponding security events.

While the IoT sentinels keep accumulating security events, they send transactions (once the threshold λ_{se} is reached) to be validated and processed by the distributed SIEM (miners). The miner processes the transaction, evolves it into a new block with all the security events and adds it to the blockchain.

In case of a distributed attack, security events related to at least two victim IoT devices are reported and added to the blockchain. If the security events are reported by two different IoT sentinels, then each of them sends its corresponding transaction to a miner. After the respective blocks are added to the blockchain, the miner consumes the security events and analyzes them using its local threat intelligence. This analysis includes the validation of security rules and policies employed to correlate security events and consequently identify distributed cyber attacks. To this end, miners can retrieve information from previous blocks stored in the blockchain. In the course of the validation process, the relevant security events are spotted and correlated to raise an alarm about the suspicious behavior.

5.3 Detecting attacks under hostile scenarios

$\mathcal{B}$SIEM-IoT is resilient against unexpected situations or even attacks aimed at the SIEM itself, without affecting its overall performance. Thus, if a miner becomes the target of a cyber attack, leading to its operational disruption, IoT sentinels would still keep generating transactions of security events. Further, the redundant and distributed additional miners, would in turn keep supporting the validation tasks needed to maintain the expected operational mode of the SIEM.

5.4 Auditing a security incident

Thanks to the traceability provided by the blockchain, along with the immutability of its blocks, all the information recorded in the blockchain is permanently available to be consumed in the future. Besides the security events, each block also contains data such as the address and ID of the sentinel who created the events, creation date and any information that can be useful for further analysis. Such approach allows $\mathcal{B}$SIEM-IoT to guarantee a completely auditable system.

5.5 Scaling an IoT Security Infrastructure

By leveraging the scalability properties of blockchain, $\mathcal{B}$SIEM-IoT permits integrating further IoT sentinels as well as distributed SIEMs (miners) effortlessly. It is worth noting that every new node in the network (either sentinel or distributed SIEM) must be granted beforehand, prior to their actual functioning.

6 Experiments

Several preliminary experiments were conducted on the proposed solution to prove its suitability in an IoT ecosystem. Since $\mathcal{B}$SIEM-IoT is composed of different elements, as shown in Figure 1, the experiments developed in this paper have used the following infrastructure:

- IoT sentinels: Each sentinel has been deployed on a Raspberry Pi 3 model B, equipped with a quad core 1.2GHz CPU, 1GB RAM, 16GB Hard Disk and OS Ubuntu Mate 16.1.
- Distributed SIEMs (miners): One SIEM (A) has been deployed in a desktop computer, equipped with a core i3 3.4GHz x4 CPU, 5.71GB RAM, 1.82TB Hard Disk and OS Debian. The other SIEM (B) was deployed on a laptop Lenovo L470 equipped with Intel Core i7 7500U (2.7 GHz), 16 GB RAM, 512 GB Hard Disk and OS Debian. All SIEMs have been tested using Alienvault OSSIM[3] (Open Source SIEM) version 5.5.1.

For the ease of reading, the experiments settings are reported in Subsection 6.1, while a significant analysis of the results is carried out in Subsection 6.2.

6.1 Settings

The experiments were conducted by running one Ethereum [1] node on each physical component, i.e., the IoT sentinels and the SIEMs (miners). The SIEMs (miners) were able to create mined blocks thanks to their computational capabilities, whereas the IoT sentinels were only able to create transactions.

Each mined block in $\mathcal{B}$SIEM-IoT is composed of a block header and a transaction. The header contains regular Ethereum header data (time stamp, difficulty, gas limit, uncles hash, gas used, among others) and the transaction includes in the **data** field the security events that were generated by IoT sentinels.

As mentioned in Section 5.1, $\mathcal{B}$SIEM-IoT is based on a permissioned blockchain that allows only known nodes (IoT sentinels and SIEMs) to be part of the network. The consensus mechanism was the one supported currently by Ethereum, i.e. PoW; however, as Ethereum evolves, a more efficient consensus mechanism, e.g. PoS, could be used instead of PoW. PoS would reduce the time and effort that are currently required for the mining process.

The reward system for $\mathcal{B}$SIEM-IoT defines its own token, which is similar to Ether, but only valid internally. In a real scenario, users interested in protecting his own IoT devices could host an IoT sentinel connected to $\mathcal{B}$SIEM-IoT to share security events. Additionally, distributed SIEMs could be hosted by different security providers at different levels like i) Internet Service Providers (ISP), which can be interested in providing security for residential customers, ii) National Computer Emergency Response Teams (CERTs), monitoring security incidents with a possible massive impact, or iii) Security vendors, which can offer IoT

[3] https://www.alienvault.com/products/ossim

security protection under a subscription. In this context, even if all blockchain nodes are identified, not all nodes are necessarily trusted for sharing security events. Security events are fundamental to detect and prevent attacks through the use of Threat Intelligence.

The experiments were carried out using several clients that ease the implementation of $\mathcal{B}$SIEM-IoT, namely: i) A Remix [4] client for the IoT sentinel, which is in charge of grouping and encoding security events to be added to a new transaction, ii) a JavaScript client for the SIEM (miner), running in the desktop computer and responsible for listening and capturing new transactions of the blockchain, in order to decode security events and make them understandable for the OSSIM server, and iii) a JavaScript client, running in the Raspberry Pi and emulating the monitoring action that an IoT sentinel performs to generate a set of security events.

6.2 Analysis of results

This Section offers an in-depth analysis of the outcomes from the experiments conducted over the $\mathcal{B}$SIEM-IoT. The obtained results will be organized around two kind of metrics (performance, blockchain) as shown in Table 1.

Category	Name	Description
Performance	CPU	SIEM (miner) CPU usage along an experiment time lapse
	RAM	SIEM (miner) RAM usage along an experiment time lapse
Blockchain	Number of blocks	Blocks added to the blockchain
	Gas used	Cost of carrying out an operation(s) in the Ethereum network
	Difficulty	Measure of how difficult is to generate a new block

Table 1. Performance and blockchain metrics for $\mathcal{B}$SIEM-IoT

To validate the capabilities of $\mathcal{B}$SIEM-IoT, two scenarios have been considered and tested:

i. Scenario 1: No critical security events (e.g. informational syslog message) are communicated from the IoT sentinel to the distributed SIEMs, which can be retained in the sentinel until reaching a Threshold of Security Events (λ_{se}=5), and then be grouped in one transaction, until a total of 10,000 transactions is reached.
ii. Scenario 2: Critical security events (e.g. emergency syslog message) need to be communicated in a short time from the IoT sentinel to the distributed SIEMs, incorporating 1 security event per transaction, until reaching a total of 1,000 transactions.

[4] https://remix.ethereum.org/

In both cases, all the metrics have been measured over the SIEM (miner). Figures 2, 3, 4 and 5 plot the measures for each metric for both cases.

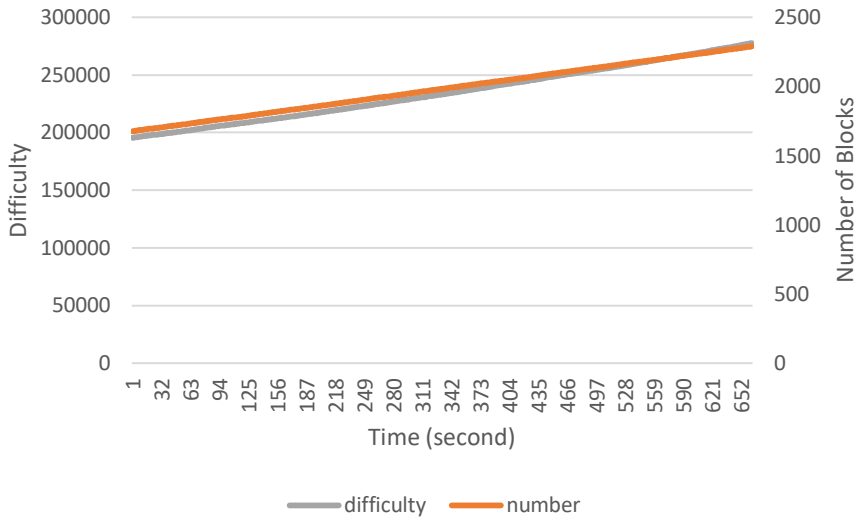

Fig. 2. Blockchain metrics for scenario 1 with 4085 transactions with 5 events per transaction

The outcomes of scenario 1 observed in Figure 2 show how each function in a smart contract generates a gas value that defines how complex it is to execute the method in the corresponding Ethereum node. Since the first transaction was mined, 663 seconds elapsed until the miner created the last transaction, so, on average, each block took approximately one second to be mined.

On the other hand, the difficulty and number of blocks are directly proportional, given that every new block increases the complexity to calculate a new hash, and the difficulty considers this hash rate to be calculated.The number of blocks increases in a rate of 0.92 blocks per second, while the difficulty raises in a rate of 123.73 points of difficulty per second.

Finally, the performance metrics for the SIEM 1 in the scenario 1 (see Figure 3) show a maximum percentage of 28.5 of used memory with some gaps where the usage of CPU is zero. When the miner is in mining process, it used practically all the CPU capability (i.e. four cores). On the other hand, the performance metrics for the SIEM B in the scenario 1 (see Figure 3) show a constant percentage of 28.5 of used memory with some gaps where the usage of CPU is zero.

With regards to scenario 2, in Figure 4 we observe that the time elapsed between the mining of the first and last block for this test was 43 seconds. In

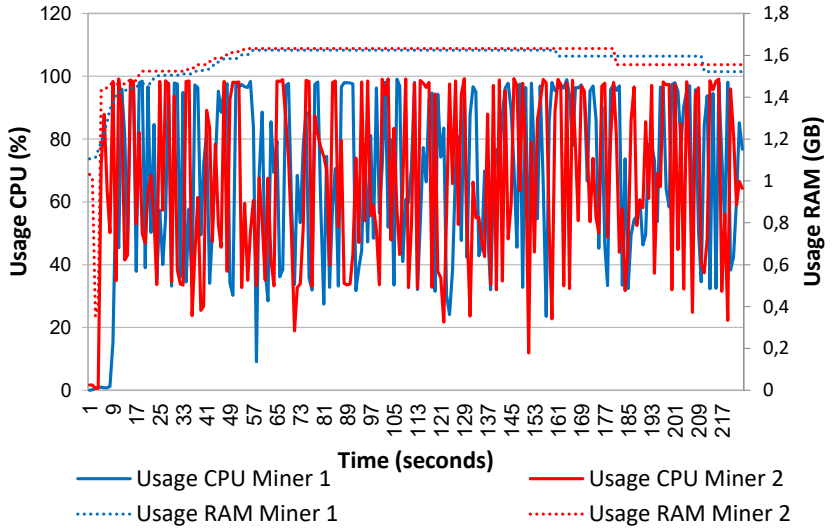

Fig. 3. Performance metrics for scenario 1 with 4085 transactions with 5 events per transaction

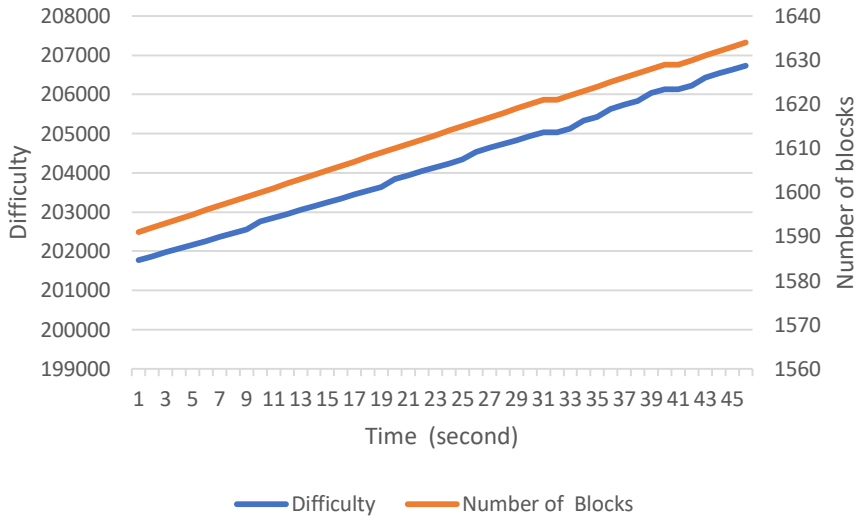

Fig. 4. Blockchain metrics for scenario 2 with 1000 transactions with 1 event per transaction

this case, where we have a greater number of transactions but lower quantity of events per transactions, every block mined took approximately 0.043 seconds.

After both analysis and having in mind that difficulty is adjusted periodically as function of how much hashing power has to be deployed by the network of miners, it is possible to observe that it increases with the time at different rates for each case . The above understanding let us realize that the difficulty rate is related to the block production rate which should change when more miners join the network.

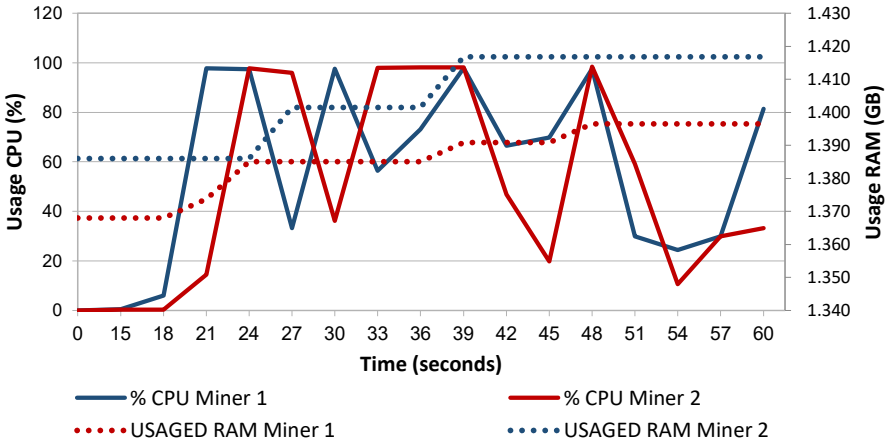

Fig. 5. Performance metrics for scenario 2 with 1000 transactions with 1 event per transaction

As for performance capabilities (see Figure 5), in this scenario we found a similar behavior compared to the first scenario. That is to say, the miners used almost all its resources for both RAM memory and CPU usage. In this test, however, the CPU does not have gaps of zero usage, but it is rather continually in use.

As a consequence of the previously analyzed experiments, we can conclude that $\mathcal{B}$SIEM-IoT yields a performance represented by a high CPU consumption (98% approx) for the CPU and a medium RAM consumption (1.4GB approx) for the SIEMs (miners). Additionally, in both scenarios, $\mathcal{B}$SIEM-IoT showcased a stable behavior with an increasing difficulty as the number of blocks grew. Last but not least, it is important to note that a block containing more events, due to the grouping made by the IoT sentinel, could require more gas since the block size is bigger in this scenario.

7 Conclusions and future work

By leveraging the benefits of blockchains, this paper presented $\mathcal{B}$SIEM-IoT, contributing directly to the safety of IoT ecosystems managing the security events in

a strict way preserving integrity and non-repudiation. Additionally, $\mathcal{B}$SIEM-IoT offers desirable features for a sturdy security system such as resilience, trust-orientation, auditability and scalability. Experiments show that $\mathcal{B}$SIEM-IoT is able to get a desirable performance with low transaction times, which depends on the settings, being affected mainly by the Threshold of Security Events (λ_{se}) and the consensus method.

As for future works, we plan to allow new types of transactions in our solution according to the type of security event detected by the IoT sentinel, e.g. more critical security events could be added to the blockchain with a higher priority, whereas medium or low priority could be hold to be grouped. Finally, we will study the feasibility of building a new generation of IoT devices that can be blockchain-capable, qualified to report internal security events to the blockchain.

Acknowledgment

This work has been partially supported by the Escuela Colombiana de Ingeniería Julio Garavito (Colombia) through the project "Developing secure and resilient architectures for Smart Sustainable Cities" approved by the Internal Research Opening 2018 and by the project "Strengthening Governance Capacity for Smart Sustainable Cities" (grant number 2018-3538/001-001) co-funded by the Erasmus+ Programme of the European Union, as well as by a Leonardo Grant 2017 for Researchers and Cultural Creators awarded by the BBVA Foundation and by a Ramón y Cajal research contract (RYC-2015-18210) granted by the MINECO (Spain) and co-funded by the European Social Fund.

References

1. Antonopoulos, A., Wood, G.: Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly Media (2018)
2. Díaz López, D., Blanco Uribe, M., Santiago Cely, C., Vega Torres, A., Moreno Guataquira, N., Morón Castro, S., Nespoli, P., Gómez Mármol, F.: Shielding IoT against Cyber-Attacks: An Event-Based Approach Using SIEM. Wireless Communications and Mobile Computing **2018** (2018)
3. Dorri, Kanhere, S., Jurdak: Blockchain in Internet of Things: Challenges and Solutions. CoRR **abs/1608.05187** (2016)
4. Dorri, A., Kanhere, S., Jurdak, R., Gauravaram, P.: Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (2017)
5. Miloslavskaya, N.: Designing blockchain-based SIEM 3.0 system. Information and Computer Security **26**(4), 491–512 (2018)
6. Nespoli, P., Useche Peláez, D., Díaz López, D., Gómez Mármol, F.: COSMOS: Collaborative, Seamless and Adaptive Sentinel for the Internet of Things. Sensors **19**(7) (2019)
7. Tasca, P., J. Tessone, C.: A taxonomy of blockchain technologies: Principles of identification and classification. Ledger **4** (02 2019)

8. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., Wang, F.: Blockchain-enabled smart contracts: Architecture, applications, and future trends. IEEE Transactions on Systems, Man, and Cybernetics: Systems pp. 1–12 (2019)
9. Wust, k., Gervais, A.: Do you need a blockchain? In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). pp. 45–54 (June 2018)
10. Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., Wan, J.: Smart contract-based access control for the internet of things. arXiv preprint arXiv:1802.04410 (2018)
11. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: Architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress). pp. 557–564 (June 2017)