How DoS attacks can be mounted on Network Slice Broker and can they be mitigated using blockchain?

Tharaka Hewa*, Anshuman Kalla†, Pawani Porambage‡, Madhusanka Liyanage§, Mika Ylianttila¶
*†‡§¶Centre for Wireless Communications, University of Oulu, Finland
§School of Computer Science, University College Dublin, Ireland
Email: *‡§¶[firstname.lastname]@oulu.fi,†anshuman.kalla@ieee.org,§madhusanka@ucd.ie

***Abstract*—Several recent works talk about the potential use of network slice brokering mechanism to facilitate the resource allocation of network slicing in next generation networks. This involves network tenants on the one hand and resource/infrastructure providers on the other hand. However, the potential downside of deploying Network Slice Broker (NSB) is that it can be victimized by DoS (Denial of Service) attack. Thus, the aim of this work is three fold. First, to present the possible ways in which DoS/DDoS attacks can be mounted on NSB and their adverse effects. Second, to propose and implement initial blockchain-based solution named as Security Service Blockchain (SSB) to prevent DoS attacks on NSB. Third, to enumerate the challenges and future research directions to effectively utilize blockchain for mitigating DoS/DDoS attacks on NSB. To evaluate the performance the proposed SSB framework is implemented using Hyperledger Fabric. The results manifest that the latency impact of the legitimate slice creation over scaled up malicious traffic remains minimal with the use of SSB framework. The integration of SSB with NSB results in gaining several fold reduction in latency under DoS attack scenario.**

***Index Terms*—Network Slicing, Network Slice Brokering, Blockchain, Smart Contracts, 5G, IoT**

I. Introduction

5G mobile networks are designed with the vision to accommodate the diverse service requirements for different stakeholders (e.g. Mobile Virtual Network Operators, Over The Top service providers, and industry verticals) in telecommunication ecosystem [1]. Thus, it is required to customize and share same 5G physical infrastructure between these stakeholders to satisfy their diverse service requirements. In this context, Network Slicing (NS) [2] has emerged as one of the building blocks of the 5G and B5G networks. Furthermore, the concept of NSB has been introduced to better orchestrate the overall process of NS that includes tasks like admission control of requests from legitimate network tenants, sending across such requests to the multiple infrastructure/resource providers, slice negotiation, slice selection, and ensuring SLA compliance [3]–[6]. Thus, NSB acts as a mediator to facilitate the process of NS that enables trustworthy resource trading between multiple vendors and stakeholders in an open market.

A. Motivation

Though NSB offers numerous advantages, nevertheless, it is vulnerable for new security limitations and issues that can hinder the deployment of network slices. In particular, the issues corresponding to DoS and DDoS attacks are envisioned to be of high importance. In DoS/DDoS attack, the compromised network tenants(s) and/or the compromised Mobile Network Operators (MNOs) maliciously overwhelm NSB entity with the intention to either slow down its working or sabotage it. As a result, there can be adverse effects like delaying the creation of genuine and legitimate network slices, inefficient utilization of computational and network resources, glitches in SLA compliance, and diminishing the trust in the network and associated services. Hence, DoS/DDoS attacks can be a significant impediment for the deployment of NSB service towards network automation so it is essential to design a mechanism to mitigate such attacks on NSB.

On one hand, there exists numerous research like [7] and [8] that aims to secure network slices against DoS attacks. On the other hand there are works such as [9]–[11] that intends to deal with DoS/DDoS attacks on 5G networks. However, none of them discuss about DoS/DDoS attacks on NSB entity.

Blockchain is a well-known Distributed Ledger Technology (DLT), which has a huge potential to be used as a supporting technology for 5G and 6G networks [12]–[14]. There have been significant number of recent works that leverage blockchain for decentralized NSB [15]–[20]. Nevertheless, to the best of our knowledge there exists no work that investigates DoS/DDoS attack on NSB and utilizes blockchain to mitigate such attacks. Table I summarizes the feature-wise position of our work with the related works. This work uses 'blockchain as a service' to mitigate DoS attacks. Some of the advantages of blockchain-based decentralized service (over centralize service) that motivated us to use blockchain technology are cryptographically secure distributed ledger, P2P network of nodes, consensus-based decision making, ledger comprising data and smart contracts distributed among the nodes, non-existence of single-point-of-failure, and transparency with pseudonymity.

B. Contribution

The contributions of this paper are as follows:

- To identify the possible ways in which DoS/DDoS attacks can vandalize the functioning of NSB.
- To propose an initial blockchain-empowered smart contract driven profiling mechanism — Security Service Blockchain (SSB) — to mitigate DoS attacks on NSB.

- To evaluate the performance of the proposed SSB by implementing it on a hyperledger fabric based experimental setup and discuss the obtained results.
- To pin point the challenges which still needs attention to make NSB completely immune to DoS/DDoS attacks.

TABLE I: Features Comparison with Key Related Works

Features	[15]	[19]	[18]	[16]	[17]	[20]	SSB
DDoS prevention	No	No	No	No	No	No	Yes
Consensus oriented tenant/MNO registration	No	No	No	No	No	Yes	Yes
Consensus oriented request authorization	No	Yes	No	No	No	No	Yes
Dedicated blockchain for security	No	No	No	No	No	No	Yes

Rest of the paper is organized as follows. Section II identifies different ways of mounting DoS/DDoS attacks on NSB and their adverse effects. Section III presents a blockchain-based solution to mitigate DoS attack on NSB. Section IV provides the implementation details and discusses the results. Section V delineates the challenges paving way for future research directions. Finally, section VI concludes the work.

II. DoS and DDoS Attack on NSB

The role of 5G NSB is introduced as a novel business model to enable the dynamic interoperability and resource trading requirements of market players such as infrastructure providers, consumers, and MNOs in trading the network and computational resources [3]. NSB is acting as a mediator between the MNO and the network tenants. Based on the resource requests received from the tenants, NSB creates a network slice template and broadcasts it to the prospective MNOs. After receiving the offers (i.e., the price list for available resources) from the MNOs, NSB selects the best matching offer for the given request and facilitates to provide the network slice to the tenants. According to the architecture of NSB, mainly there are two contacting points (i.e., tenants and MNOs) from where DoS attacks can be mounted on the NSB. In the rest of the section, we discuss the most probable four scenarios where DoS/DDoS attacks may occur on NSB.

A. Malicious resource requests sent by compromised IoT tenants

Compromised IoT tenants can send extremely large number of resource requests (which is also referred as slice request) with malicious intention to NSB (i.e., DoS attack). This leads to generation of subsequent events within the brokering entity as per the sequential workflow of NSB. For instance, a malicious request may consists of large number of resource parameters that requires resource-intensive execution. In such a scenario, the relevant NSB's modules for each step will run extensively to perform different activities including creation of slice blueprints and disseminating them to the MNOs. The high volume of transactions utilize the computational resources and deplete the storage with malicious traffic. Thus, NSB will be overloaded which will soon make the slicing service unavailable. Furthermore, the effects of the attacks will be reflected on the MNOs since the MNOs will continuously respond to the requests received from NSB under attack. The overall effect is that such malicious requests will overshadow the legitimate resource requests in the NSB.

B. Malicious resource offers sent by compromised MNOs

NSB can be potentially affected by the malicious resource offers sent by MNOs (i.e., DoS attack). The severity depends on the computationally intensive nature of the selection algorithm powering the NSB. Moreover, the malicious resource offers sent by a compromised MNO may intentionally include numerical values which results in overflow of the memory heap of NSB thereby impacting it capabilities. Furthermore, the malicious MNOs may intentionally send messages with extensive length which overload the messaging protocols and data buffers of the API services of the NSB. Thus, NSB may fail to receive the legitimate resource offers under such attack.

C. Malicious resource requests sent by colluding IoT tenants

A DDoS attack can be launched by a malicious group of IoT tenants which under an attacker's control can operate unlawfully towards a common ill objective. Every member of the colluding group of IoT tenants send a permissible number of requests to NSB. However, collectively they might get success in bring down the services of NSB. Such an attack, within limited time period, can result in overloading of API related message streams, memory overflows and over-utilization of computational resources. Early detection of such colluding group of IoT tenant is relatively difficult when compared with the malicious requests sent by a single tenant.

D. Malicious resource offers sent by colluding MNOs

A subset of MNOs under an external malicious control can be made to collude and send bogus resource offers to NSB (i.e., DDoS attack). This kind of attack can trap the slice selection algorithm and hog the resources. Thus the group of organized malicious MNOs does not allow NSB to attend the legitimate resource offers and fullfil the service delivery.

III. Proposed Architecture to Mitigate DoS Attack on NSB

We proposed a novel Security Service Blockchain (SSB) for protecting the slice broker from DoS attacks launched by malicious tenants and MNOs. The main objective of SSB is to ensure the persistent operation of the NSB for genuine members even under the presence of compromised tenants and/or MNOs. In the proposed solution, we utilize SSB as the smart contract based security gateway to validate each request and control the access of IoT tenants as well as MNOs to the slice broker. The proposed SSB ensures that all resource requests and resource offers committed to the slice broker are

valid and approved by the consensus process of the dedicated secure blockchain network. Figure 1 illustrated the deployment of proposed SSB mechanism.

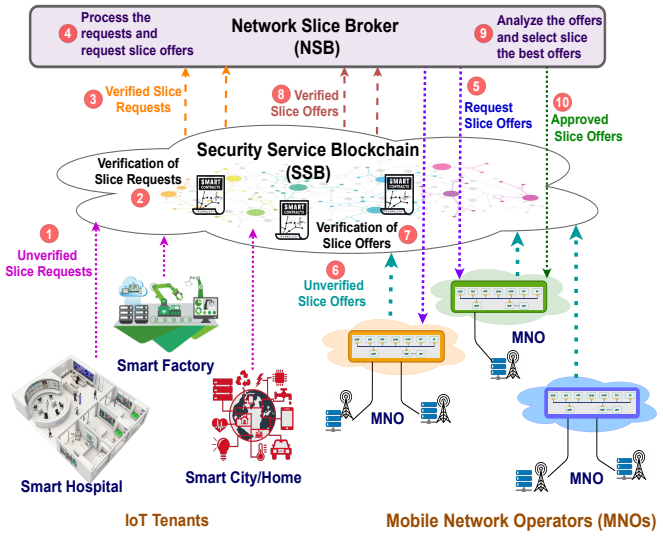

Fig. 1: Proposed Security Service Blockchain (SSB)

A. Assumptions

We make few assumptions while developing the proposed SSB mechanism. These assumptions are made to match the proposed system with real world deployments.

1) At any given time, the underlying infrastructure, comprising computational and storage resources, where the NSB is deployed is finite. Once the resources are fully utilized, transactions will not be processed.
2) The capacity of communication channels, including message buffers to communicate between IoT tenants, NSB, and MNOs are limited. Once the message buffer capacity has been exceeded, NSB will not receive neither resource requests from tenants nor resource offers from MNOs.
3) The concurrent transaction processing capacity of the NSB for IoT tenants and MNOs are limited. Once the processing capacity has been overloaded, the transactions will not be processed as anticipated which is regarded as DoS effect.
4) The minor impact of latency for a legitimate request over scaled up malicious requests denotes that the effect of a malicious requests is minimal to the legitimate request.

B. Prerequisites

Each IoT tenant and MNO requires to register with slice brokering ecosystem to acquire the transaction profile. The transaction profile defines the scope of services eligible to the individuals by NSB. Table II reflects an example transaction profile of an IoT tenant and MNO. The transaction profile registration completes upon the consensus procedure including storage of the distributed ledger.

C. Transaction profile based verification

The malicious resource requests launched from compromised tenants as well as compromised MNOs need to be distinguished from the legitimate requests. The tenants and MNOs require to agree on the limits for the parameters as defined in Table II prior to consume the slice brokering service. These profiles will be stored in the immutable ledger in blockchain upon the consensus process. We propose to utilize the blockchain based transaction profiles to verify each request launched by the tenant and MNO to the SSB. After the transaction profile verification, if the requests are within the conditions defined in profile, the SSB smart contract invokes NSB as an off-chain request to submit legitimate request to NSB. Algorithm 1 denotes the corresponding steps encoded on the smart contract.

TABLE II: Transaction profile for network tenant and MNO

Parameter	Value for tenant	Value for MNO
Messages per day	1000	100
Max message length	200 bytes	400 bytes
Type of message	Resource request	Resource offers
Maximum parameter count	100	200
Maximum parameter value	1000	1000

Algorithm 1

Require: $M_1 = \{ID, request, currentTime\}$
$TP \leftarrow$retrieveProfileFromLedger(ID)
if $isValidMessage(request, TP)$ **then**
  if $messagesToday \leq TP.messagesPerDay$ **then**
    if $messageLength \leq TP.maxMessageLength$ **then**
      if $parameterCount \leq TP.maxParameterCount$ **then**
        if $validateParameterValues(request)$ **then**
          $invokeNSB(request)$
          $updateSSBLedgerLegitimateRequest(request)$
        end if
      else
        $updateSSBLedgerInvalidParamValue(request)$
      end if
    else
      $updateSSBLedgerInvalidParamCount(request)$
    end if
  else
    $updateSSBLedgerInvalidLength(request)$
  else
    $updateSSBLedgerInvalidMessageCountrequest)$
  end if
end if

D. Advantages of SSB

Computational offloading: NSB is envisioned to be extremely busy with (i) the processing of large number of resource requests from network tenants and resource offers from MNOs, (ii) selecting best offer from the pool of resource offers, (iii) monitoring SLA compliance, and (iv) communicating with slice manager of each MNO. In this scenario, handling DoS attacks with separate blockchain (SSB) offloads the DoS prevention overhead from NSB and ensures persistent service to the legitimate tenants over attacks.

Blockchain-based transaction profiling using smart contracts: The transaction profiles stored in the blockchain are

cryptographically verified and immutable. Furthermore, smart contract based operations ensure that the malicious groups cannot tamper the profile information in the blockchain.

Efficient data logging: Most of logging activity related to resource requests and resource offers are of no use once the slice has been released and payment settlement has been done. Thus, such data logs need not to be maintained on NSB otherwise it will quickly deplete the storage capacity. Thus, storing such data in separate blockchain SSB allows to deal efficiently with such data of local and temporal importance.

IV. Implementation and Results

In this section, we explain the experimental setup and elaborate the experiments performed to to investigate the behavior of proposed architecture in a near realistic environment.

A. Infrastructure placement of the implementation setup

As illustrated in Figure 2, the Fog tenants simulated by the Raspberry Pi. The MNOs are simulated by Virtual Machines (VM). The malicious tenants and MNOs also simulated by the selected fog nodes and VMs. A cloud instance with Ubuntu 19.04 operating system used to deploy the SSB with public IP access. The processor is Intel(R) Xeon(R) CPU 2.33GHz with 16GB RAM. A router with 5G and LAN connects the components to internet. Lenovo Thinkpad T14 laptop used as the host machine for NSB, which is a blockchain-based slice broker deployment. The NSB deployed on a Ubuntu 18.04 VM. We used two instances of Hyperledger Fabric blockchain platform with 5 nodes and RAFT consensus based mining service deployed for each SSB and NSB. Communication between each component facilitated using MQ Telemetry Transport (MQTT) broker deployed on cloud server.

B. DoS attacks and legitimate requests simulation for latency evaluation experiments

In the different experiments, the scaled up malicious resource requests and MNO offers simulated using multithreaded software codes run on Raspberry Pi and VMs. Different messages(slice requests/resource offers) with a specific number of concurrent transactions launched by the program on each trial. Each trial performed 100 times and measured the mean end to end latency to complete entire set of transactions in the trial.

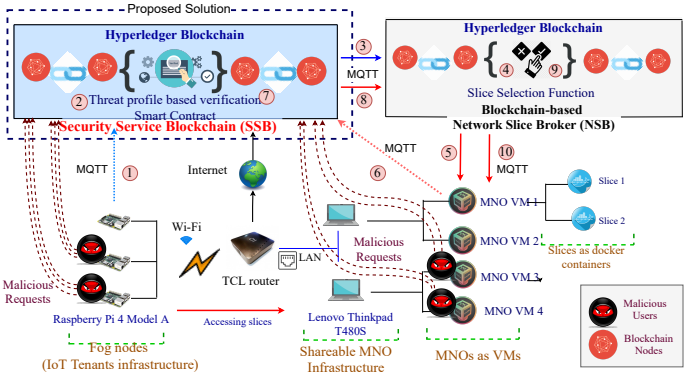

Fig. 2: The implementation setup

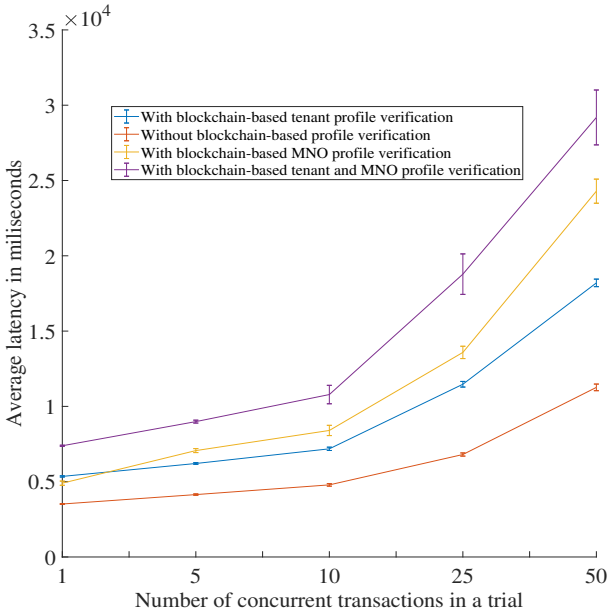

Fig. 3: Slice creation latency with and without SSB tenant profile verification for legitimate resource requests

Furthermore, the mean and standard deviation for each test presented graphically as the results.

C. Experimental configuration on blockchain network

The $BatchTimeOut$ parmeter of Hyperledger Fabric, which is the duration to wait before processing a batch of transactions by the mining nodes configured as 500ms, 1000ms, and 2000ms on different experiments for SSB. The $BatchTimeOut$ configured fixed on NSB for all experiments.

D. Performance impact incurred by the SSB integration for the scaled up legitimate resource requests

The objective of this experiment is to evaluate the latency impact incurred on resource requests by each verification on SSB and compare with the latency of SSB-less NSB. In the proposed architecture, SSB first receives resource request by tenants or resource offers by MNOs before sending to NSB. Each message is verified with the profile information in the smart contract of SSB. The smart contract ensures whether the message is within the limits defined in the profile and approves the transaction. Once the approved transaction committed to the ledger, it will included to the block and disseminated. This procedure incorporates additional latency of block generation to the existing slice brokering process. We evaluated the impact of latency incurred by the integration of SSB to the NSB. Figure 3 presents the latency impact of SSB integration for a legitimate transactions when SSB and NSB block generation time set to 1000ms. The number of concurrent transactions have been increased upto 50 transactions per batch in order to evaluate the system performance over scaling up requests.

*1) **Tenant profile verification latency**:* In this evaluation, resource requests sent to the SSB simulating legitimate tenant requests. Upon tenant profile verification and approval by the SSB, the request forwarded to the NSB. No MNO verification

performed. Obviously, the block generation time affected as an additional latency with the integration of SSB when comparing with SSB not integrated transaction latency.

*2) **MNO profile verification latency**:* In this evaluation, resource offers sent to the SSB to simulate legitimate MNO resource offers. Upon MNO profile verification and approval by the SSB, the request forwarded to the NSB for selection of resource offer. No tenant verification performed. The effect of additional block mining for the transaction verification in SSB is reflected from the results. However, the MNO verification latency is slightly higher than the tenant profile verification since each resource request consists of multiple MNO offers and verification of all MNOs takes time.

*3) **Tenant and MNO profile verification latency**:* In this evaluation, resource requests sent to the SSB simulating legitimate tenants. The tenant verification performed for the resource request and MNO profile verification performed for resource offers which incur more latency.

However, the additional latency is required to be accepted when additional layer of security has been integrated. This latency is dependent on the block generation time which is currently configured as 1000ms. The block time can be reduced upto 200ms as well as increased. Hence, the latency is configurable according to the transaction traffic. Furthermore, when the throughput is getting increased, the standard deviation is slightly increased as per the Figure 3. The key reason we observed was the dissemination of requests within multiple blocks. The block generation time is 1000ms in the implementation setup and when the entire batch of transactions not included into the same block, the latency is increased since all transactions in multiple blocks required to be completed. However, in the next experiments, obviously SSB is beneficial to defend NSB on higher number of malicious requests.

E. Behavior of slice creation latency with malicious traffic

The objective of this experiment is to evaluate the latency impact of legitimate resource requests while SSB is defending the malicious requests launched. It is assumed that the compromised tenants launch the DoS attack as slice requests and compromised MNOs launch the DoS attacks as malicious resource offers. When the SSB is integrated to the NSB, all messages are verified by the SSB along with the profile information. The NSB receive approved resource requests and MNO offers as off-chain invocation committed by the SSB. The NSB receives slice request or MNO offers as an off-chain invocation of smart contract if and only if the SSB request is within the security verification criteria. We assume that the uninterrupted service delivery with minimal impact on latency for the legitimate resource requests over malicious requests which simulate DoS attack attempt reflects that the proposed system is capable to defend NSB from DoS attacks.

In this experiment, we compared the latency of a single legitimate transaction over NSB in an attack scenario in SSB integrated and non-integrated cases. First, we evaluated the end to end latency of a legitimate transaction over different number of malicious resource requests and resource offers. The malicious resource requests leading to a DoS attack simulated using multi-threaded program running on Raspberry Pi. We assume the lower impact to the latency of a legitimate transaction over malicious requests reflect that there is no interference to the legitimate request by the malicious transactions.

*1) **Transaction latency comparison with and without SSB integration on malicious resource requests** :* Figure 4 reflects the interference to the legitimate slice requests with and without SSB integration on different $BatchTimeOut$ configurations on SSB. The experiment performed in 500ms, 1000ms, and 2000ms block $BatchTimeOut$ configurations of SSB. The impact to the legitimate request over the malicious resource requests have been considered in when the SSB is not integrated. Apparently, the end to end latency is increasing when the number of malicious resource requests increased. This latency increase indicates the service failures over the flooded requests. When the NSB is being flooded with the requests, the mining nodes are overloaded and functionality of NSB is hindered. In contrast, the SSB integrated experiment proves that the latency does not being significantly affected by the flooded resource requests. We observed that when 50 malicious transactions are launched, the mean latency of the legitimate transaction is 115 seconds on SSB-less NSB. In contrast, the mean latency is 5 seconds when the SSB has been integrated to the NSB. When SSB is integrated, the NSB does not reach the malicious traffic and preserves NSB resources for legitimate requests.

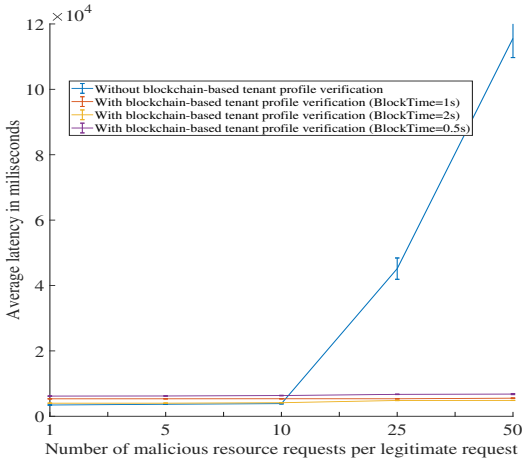

Fig. 4: Latency of single legitimate transaction with and without SSB over malicious resource requests

*2) **Transaction latency comparison with and without SSB integration on malicious resource offers**:* Figure 5 reflects the interference to the legitimate resource offers with and without SSB integration on different $BatchTimeOut$ configurations on SSB. The experiment performed in 500ms, 1000ms, and 2000ms block $BatchTimeOut$ configurations of SSB. The malicious resource offers leading to a DoS attack simulated using the multithreaded program running on VM. The end to end latency increase when the number of resource offers increased in the system without SSB. This latency increase indicates the service failures over the flooded malicious resource offers.

When the NSB is being flooded with the malicious offers, the mining nodes are overloaded and the functionality of NSB is hindered. In contrast, the SSB integrated experiment proves that the latency does not being affected by the flooded resource offers when integrated with SSB. The experiment performed in 500ms, 1000ms, and 2000ms $BatchTimeOut$ configurations of SSB. We observed that when 50 malicious transactions are launched, the mean latency of the legitimate transaction is 119 seconds on SSB-less NSB. In contrast, the mean latency is 5 seconds when the SSB has been integrated to the NSB. When SSB is integrated, the NSB does not reach the malicious traffic and preserves NSB resources for legitimate requests.

*3) **Transaction latency comparison with and without SSB integration on malicious slice request with malicious resource offers**:* Figure 6 reflects the interference to the legitimate resource offers with and without SSB integration on different $BatchTimeOut$ configurations on SSB. The experiment performed in 500ms, 1000ms, and 2000ms $BatchTimeOut$ configurations of SSB. The malicious resource requests and offers leading to a DoS attack simulated using the multithreaded program running on Raspberry Pi and VM. The end to end latency increase when the total number of resource requests and resource offers increased. This latency increase indicates the service failures over the flooded malicious traffic. When the NSB is being flooded with the malicious resource requests and offers, the mining nodes are overloaded and the functionality of NSB is hindered. In contrast, the SSB integrated experiment proves that the latency does not being affected by the flooded resource requests and offers when integrated with SSB. We observed that when 50 malicious transactions are launched, the mean latency of the legitimate transaction is 35 seconds on SSB-less NSB. In contrast, the mean latency is 6.9 seconds when the SSB has been integrated to the NSB. When SSB is integrated, the NSB does not reach the malicious traffic and preserves NSB resources for legitimate requests.

When the SSB smart contract as illustrated on Algorithm 1 has been focused, the transaction profile verification smart contract performs a querying operation initially. The ledger

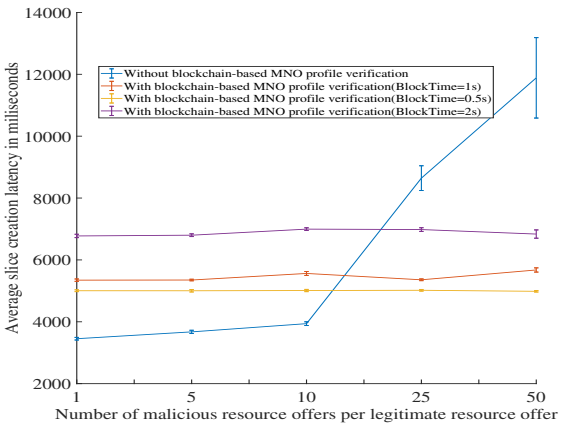

Fig. 5: Latency of single legitimate transaction with and without SSB over malicious offers

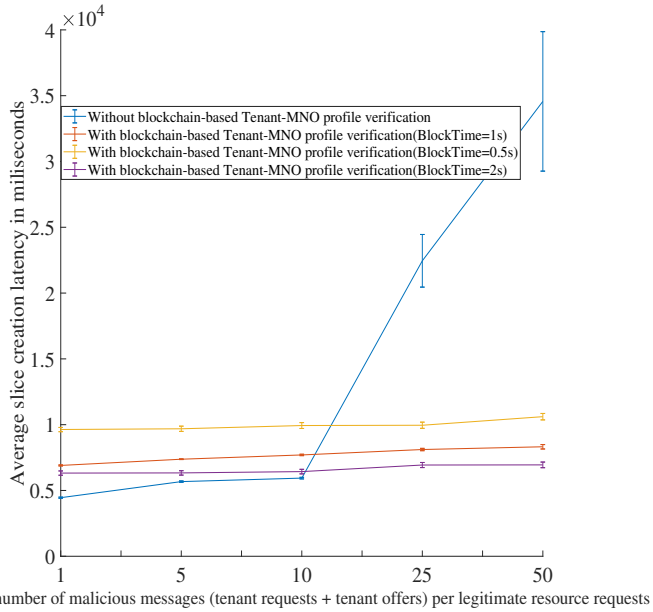

Fig. 6: Latency of single legitimate transaction with and without SSB over malicious resource requests+resource offers

updates on failure conditions are optional and no significant affect the sequential flow of a transaction. The transaction approval invokes mining nodes for the verified transactions only. Hence, in the malicious attempts, the utilization of SSB is limited to the querying and verification operation. The NSB utilized if and only if either the resource request or resource offer is within the specific transaction profile limits. Therefore, no overhead incurred on the NSB by flooding type requests. The end to end latency of legitimate requests not being significantly affected with the proposed method. However, significant latency advantage is obvious when NSB is running without SSB on lower malicious message rates such as 1, 5, and 10. This is a common observation on Figure 4, Figure 5, and Figure 6. The key reason is the additional block mining latency incurred by SSB. However, in higher malicious message rates such as 25 and 50, SSB outperforms NSB standalone deployment in terms of latency. Obviously, SSB integration is safe and advantageous in terms of DoS prevention in the scalable environments. However, the evaluation was limited upto 50 malicious requests per batch due to infrastructure limitations. We observed that the RAFT consensus nodes of Hyperledger take more than 45 seconds to process the transaction. The transactions which have been not approved within 45 seconds will not be added to the blocks.

V. Challenges and Limitations

A. Challenges

1) Identification of valid requests launched by malicious groups to attack SSB: The SSB performs an individual validation on the resource requests and offers to evaluate whether each request is within the limits. However, it is possible to an organized group of registered tenants can explicitly send individual requests to attack the NSB. In such case, SSB

verification will be passed since the request is within the profile verification criteria. The resultant will affect the NSB.

2) Latency incurred by the SSB block verification: When the Figure 3 has been focused, the impact of additional latency incurred by the verification steps is obvious. However, this latency depends on the block mining interval defined on the SSB and can be adjusted by configuration.

3) Additional overhead to operate blockchain network: SSB is a dedicated blockchain network which has been employed in parallel to the NSB. Hence, the NSB includes the ordinary operational overheads applicable to a generic blockchain network, including computational overheads.

B. Limitations

1) Authentication of messages exchanged in different channels are not present: In the proposed architecture, authentication of messages exchanged are not present. The messages exchanged can be forged even after the verification by SSB.

2) Privacy over the SSB records and messages are not enforced: The transaction records as well as profile information have not been enforced with privacy preservation technique.

3) Access control to the network services along with service level agreements are not specifically focused: Although we address, specifically, the mitigation of DoS attacks with the so called SSB NSB, there are different other security aspects that can be considered in blockchain-based NSBs. For instance, there are possibilities to use dynamic profiling smart contracts and consensus algorithms with blockchain-based NSBs for managing security-oriented service level agreements (SSLAs) for local network operators and infrastructure providers running on a common platform. Through the dynamic profile status, the stakeholders are authorized to access the functions of NSB. Moreover, it can be upgraded with lightweight security solutions in terms of Authentication, Authorization and Accounting (AAA).

VI. Conclusion

In this paper, we identified the potential DoS/DDoS attacks which can be mounted on NSB by malicious tenants and malicious MNOs. We discussed the impact of the DoS/DDoS attacks on NSB and highlighted the significance of handling and defending NSB for persistent service delivery. We proposed a threat profiling based security mechanism developed using blockchain and smart contracts. The experimental evaluation reflects that the proposed solution ensures smooth NSB service delivery to the legitimate tenants even under malicious traffic. In future, we plan to utilize the blockchain data as a training dataset to detect more complicated DoS/DDoS attacks.

Acknowledgment

This work has been performed under 6Genesis Flagship (grant 318927) and 5GEAR projects. The research leading to these results partly received funding from European Union's Horizon 2020 research and innovation programme under grant agreement no 871808 (5G PPP project INSPIRE-5Gplus). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains.

References

[1] "An introduction to network slicing," GSMA Network Tech. Report, 2017, accessed on 15.01.2021. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf

[2] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5g Networks," *IEEE Communications Surveys & Tutorials*, 2021.

[3] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5g network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[4] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5g network slicing resource utilization," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[5] H. Xu, P. V. Klainea, O. Oniretia, B. Caob, M. Imrana, and L. Zhang, "Blockchain-enabled resource management and sharing for 6g communications," *arXiv preprint arXiv:2003.13083*, 2020.

[6] M. A. Togou, T. Bi, K. Dev, K. McDonnell, A. Milenovic, H. Tewari, and G.-M. Muntean, "A distributed blockchain-based broker for efficient resource provisioning in 5g networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 1485–1490.

[7] P. Porambage, Y. Miche, A. Kalliola, M. Liyanage, and M. Ylianttila, "Secure Keying Scheme for Network Slicing in 5G Architecture," in *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2019, pp. 1–6.

[8] D. Sattar and A. Matrawy, "Towards Secure Slicing: Using Slice Isolation to Mitigate DDoS Attacks on 5G Core Network Slices," in *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2019, pp. 82–90.

[9] A. S. Mamolar, Z. Pervez, J. M. A. Calero, and A. M. Khattak, "Towards the Transversal Detection of DDoS Network Attacks in 5G Multi-tenant Overlay Networks," *Computers & Security*, vol. 79, pp. 132–147, 2018.

[10] H. Moudoud, L. Khoukhi, and S. Cherkaoui, "Prediction and detection of fdia and ddos attacks in 5g enabled iot," *IEEE Network*, 2020.

[11] K. Lalropuia and V. Gupta, "A Bayesian Game Model and Network Availability Model for Small Cells under Denial of Service (DoS) Attack in 5G Wireless Communication Network," *Wireless Networks*, vol. 26, no. 1, pp. 557–572, 2020.

[12] M. Latva-aho, K. Leppänen, F. Clazzer, and A. Munari, "Key drivers and research challenges for 6g ubiquitous wireless intelligence," 2020.

[13] T. Hewa, G. Gür, A. Kalla, M. Ylianttila, A. Bracken, and M. Liyanage, "The role of blockchain in 6g: Challenges, opportunities and research directions," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.

[14] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.

[15] J. Backman, S. Yrjölä, K. Valtanen, and O. Mämmelä, "Blockchain network slice broker in 5g: Slice leasing in factory of the future use case," in *2017 Internet of Things Business Models, Users, and Networks*. IEEE, 2017, pp. 1–8.

[16] K. Valtanen, J. Backman, and S. Yrjölä, "Creating Value through Blockchain Powered Resource Configurations: Analysis of 5G Network Slice Brokering Case," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2018, pp. 185–190.

[17] B. Nour, A. Ksentini, N. Herbaut, P. A. Frangoudis, and H. Moungla, "A Blockchain-based Network Slice Broker for 5G Services," *IEEE Networking Letters*, vol. 1, no. 3, pp. 99–102, 2019.

[18] N. Afraz and M. Ruffini, "5G Network Slice Brokering: A Distributed Blockchain-based Market," in *2020 European Conference on Networks and Communications (EuCNC)*. IEEE, 2020, pp. 23–27.

[19] L. Zanzi, A. Albanese, V. Sciancalepore, and X. Costa-Pérez, "NSBchain: A Secure Blockchain Framework for Network Slicing Brokerage," *arXiv preprint arXiv:2003.07748*, 2020.

[20] K. Antevski and C. J. Bernardos, "Federation of 5G services using Distributed Ledger Technologies," *Internet Technology Letters*, p. e193, 2016.