Inter-slice Defender: An Anomaly Detection Solution for Distributed Slice Mobility Attacks

*Ricardo Misael Ayala Molina[1], *Nathalie Wehbe[1], Hyame Assem Alameddine[2], Makan Pourzandi[2], Chadi Assi[1]
[1]*Concordia University, Montreal, Canada* [2]*Ericsson, Montreal, Canada*
{ricardo.ayalamolina, nathalie.wehbe, chadi.assi}@concordia.ca,
{hyame.a.alameddine, makan.pourzandi}@ericsson.com

***Abstract*—With the evolution of Fifth Generation (5G) technology, network slicing has become a key enabler, providing flexibility and efficiency in segmenting the network to enhance service delivery. This technology enables User Equipment (UEs) to simultaneously connect or switch between Network Slices (NSs) to get access to multiple services with guaranteed quality of service. Nonetheless, switching between NSs, also known as Inter-Slice-Switching (ISS), can be maliciously exploited by attackers to cause a Distributed Slice Mobility (DSM) attack. DSM attack is a distributed denial of service attack that can disrupt both NSs and the 5G control plane. In this work, we develop Inter-slice defender, a novel Long Short-Term Memory (LSTM)-Autoencoder-based anomaly detection solution, tailored to detect DSM attacks. Inter-slice Defender leverages Third Generation Partnership (3GPP) Key Performance Indicators (KPIs) and Performance Measurement (PM) counters to detect two variations of the DSM attack that we devise. Our experimental results are based on DSM attacks simulations performed on a 5G testbed employing the open-source Free5GC testbed and UERANSIM simulator. They show that Inter-slice defender achieves an average F1-score of 98.75%, demonstrating its robustness in detecting these sophisticated attacks.**

***Index Terms*—Network slicing, 5G, security, inter-slice switching, anomaly detection, machine learning.**

I. Introduction

Network slicing stands as a pivotal technology and a key enabler, empowering 5G networks to provide intelligent services that necessitate a diverse range of requirements such as high data rates, ultra low latency, varying levels of security, dense connectivity, among others [1]. Its significance is underscored by its essential role in the establishment of virtual end-to-end networks known as Network Slices (NSs), each dedicated to specific vertical industry services [1]–[3].

The adaptability afforded by network slicing allows a User Equipment (UE) to seamlessly switch between NSs to meet its specific needs, such as Quality of Service (QoS), security level, service cost, and other factors [4]. Furthermore, contingent upon the policies set by Mobile Network Operators (MNOs), such as NS load management, UEs can be seamlessly transferred from one NS to another. In line with the 3rd Generation Partnership Project (3GPP) standards, a UE can be subscribed to up to sixteen NSs [5] and can connect concurrently to a maximum of eight NSs [4].

ISBN 978-3-903176-63-8 © 2024 IFIP
*These authors contributed equally to this work.

The inherent flexibility of NSs introduces a new security vulnerability susceptible to exploitation by attackers [6], [7]. In fact, whenever a UE transitions from one NS to another, also known as Inter-Slice Switching (ISS) [4], it must undergo a re-initiation of its authentication and registration procedures, among others [8]. These procedures generate a flood of signals within the Control Plane (CP) and between it and the UE. As such, attackers can trigger a high number of ISS events to generate an influx of signaling messages which can overwhelm Network Functions (NFs) such as those involved in the ISS procedures (e.g., Access Mobility Management Function (AMF), Session Management Function (SMF), etc.). Consequently, such attack, known as Distributed Slice Mobility (DSM) attack, can cause a Distributed Denial of Service (DDoS) which can disrupt the connection of legitimate UEs [9].

DSM attack along with its economical and performance damage was first discussed in [9]. The same authors then proposed a protocol to secure the network against DSM attacks [10]. The proposed protocol automatically selects an NS for a UE based on its subscription and services offered by external networks. Subsequently, [11] developed a DSM attack detection solution based on evaluating the average waiting time and the average switching rate. To the best of our knowledge, [11] is the only work which addressed DSM attack detection while other works [12]–[15] focused on DDoS attack detection in general using Machine Learning (ML) techniques without evaluating the effectiveness of their solutions in detecting the DSM attack.

In this work, we address the shortcoming in the literature and propose Inter-slice defender, a novel network slicing anomaly detection solution to detect DSM attack and two of its variations. We devise the Random Slice Attack (RSA) and Target Slice Attack (TSA). RSA consists of switching UEs to random NSs while TSA accounts for switching UE to pre-selected targeted NS. We simulate RSA and TSA in a 5G testbed and study their impact on the network. Our contributions are summarized as follows:

- We develop Inter-slice defender, a novel network slicing anomaly detection solution leveraging the Long Short-Term Memory (LSTM)-Autoencoder ML model to detect variations of the DSM attack. LSTM-Autoencoder combines the advantages of both; LSTM in capturing long-term dependencies in a sequence of data, and

Autoencoder in learning the most relevant features. This makes LSTM-Autoencoder a good candidate to profile ISS events that involve multiple intertwined and sequential 5G procedures such as UE authentication, registration, Packet Data Unit (PDU) session establishment and deregistration.
- We train and test our Inter-slice defender model using time-based 3GPP Key Performance Indicators (KPIs) and Performance Measurements (PM) counters to profile CP NFs and NSs behavior. To the best of our knowledge, we are the first to explore the efficiency of these features in detecting network slicing attacks.
- Using the open source Free5GC testbed [16] and the UERANSIM [17] simulator, we build a 5G network with four different NSs and we adapt it to simulate different variations of the DSM attack.
- We simulate normal and DSM attack network traffic on our 5G testbed and collect the related data. We leverage the collected data to calculate 3GPP KPIs and PM counters that are used by our Inter-slice defender solution to detect DSM attack. To the best of our knowledge, the generated dataset is the first of its kind for network slicing anomaly detection.
- We design two variations of the DSM attack, namely, Random Slice Attack (RSA) and Target Slice Attack (TSA) and evaluate their impact in causing a DDoS on the 5G network as a result of overloading the AMF.
- We perform an experimental analysis of our novel Inter-slice defender solution and show its robustness in detecting RSA and TSA with an average F1-score of 98.75%.

The remainder of the paper is organized as follows: Section II presents the literature review. Section III outlines two DSM attack variations, along with their assumptions and threat model. Section IV describes Inter-slice defender and its architecture. Section V introduces our simulation setup including our 5G testbed, simulation scenarios, and datasets. Section VI highlights the impact of RSA and TSA on the 5G network. Section VII discusses the experimental results. Finally, the work is concluded in Section VIII.

II. Background and literature review

A. DSM Attack

A limited number of research studies have been dedicated to analyzing, detecting and preventing DSM attacks. For instance, Sathi et al. [9] were the first to propose the DSM attack and theoretically study the economic and performance damage it can cause on the network. They note that such damage exceeds those resulting from other DDoS attacks. Sajjad et al. [4] detailed 3GPP-based procedures enabling UE mobility and its transition between NS. They highlighted critical UE mobility challenges and suggested potential avenues for future research. Bisht et al. [11] illustrated the consequences of ISS events that result in a DDoS attack. They employed an algorithm based on two metrics, average waiting time and switching rate, to identify and block compromised UEs. From a different perspective, Sathi et al. [10] introduced a preventive approach through a novel NS selection protocol. The protocol suggests that the network chooses the best NS for the UE based on its subscription and data network services which contradicts 3GPP 5G standards that allow the UE to request the services of a specific NS.

The work tackling DSM attack is limited to theoretical studies and does not consider the practical implementation of this attack in a real 5G network. Further, DSM attack detection solutions are limited to [11], which followed a simplistic algorithmic approach based on two metrics that were not tested on a real 5G network slicing-based dataset, thus overlooking the complexity of this attack.

Contrary to the aforementioned efforts, we bridge existing research gaps by providing a practical implementation of the DSM attack and its variations in a 5G testbed and analyze their impact on the network. We craft an LSTM-Autoencoder-based anomaly detection model to detect DSM attacks' complex patterns which we train and test on network-slicing-specific features (i.e., 3GPP KPIs and PM counters) extracted from a 5G dataset that we generate.

B. Traditional DDoS Attacks Targeting Network Slices

DDoS attacks are known to be the most disruptive attacks in the realm of cybersecurity [18]. While DSM has been identified as a DDoS attack on NSs, DDoS flooding attacks, such as User Data Protocol (UDP) lag and Transmission Control Protocol (TCP) SYN, among others, have also gained attention. DDoS flooding attacks can be caused by exploiting vulnerabilities in the high number of UEs in 5G networks and targeting its NSs. To detect these attacks, Deep Learning (DL) and mathematical models have been proposed. For example, Khan et al. [12] used a bidirectional-LSTM model to detect DDoS attacks (i.e., UDP flooding and TCP SYN attacks) in two NSs. They evaluated the impact of these attacks by measuring the incurred bandwidth and latency. Similarly, the authors in [13] devised a framework leveraging LSTM to identify DDoS attacks and detect if UEs' NS requests are either legitimate or indicate an attack.

Further, Thantharate et al. [14] developed Secure5G that employs a convolutional neural network for early DDoS attack detection by analyzing network traffic patterns collected from a 5G testbed. Secure5G detects anomalous UEs' requests targeting multiple NSs and redirects suspicious UEs to a quarantine NS. Sattar et al. [15] proposed an NS isolation mechanism as a mitigation measure against DDoS attacks using a mathematical model. They evaluated their model's performance by measuring its impact on bandwidth, response time, and round-trip time.

Most of the aforementioned work on NS attack detection focuses on general DDoS attacks that are not NS specific. Despite the efforts presented in some of these works to test these attacks on a 5G testbed and use the generated dataset for DDoS attack detection, their anomaly detection models remain limited to flow-based features that do not capture any NS-specific characteristics. Moreover, these studies do not examine nor evaluate the performance of their detection solutions in detecting the DSM attack and its variations.

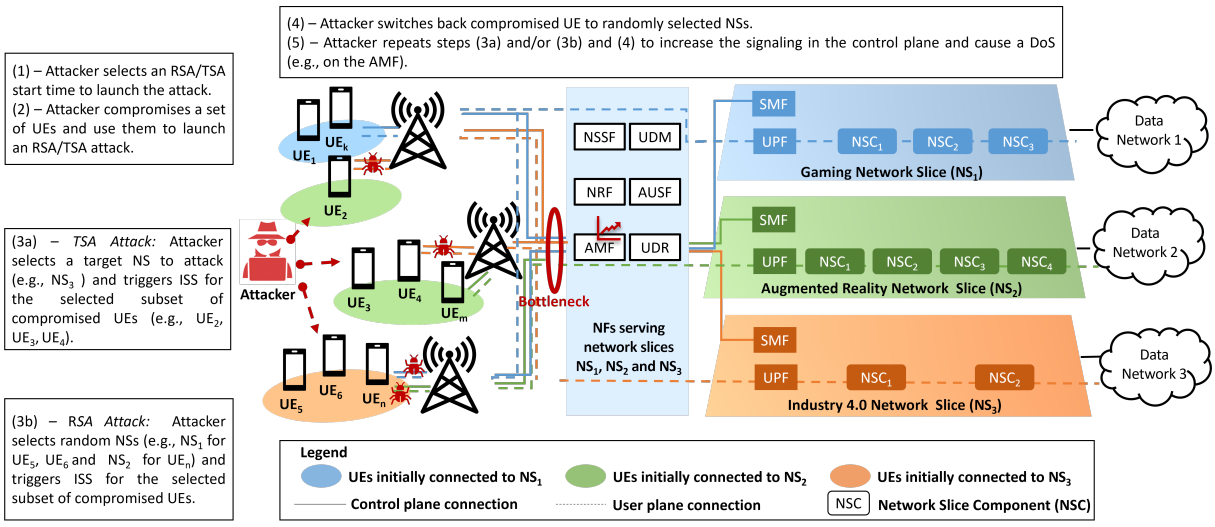


Figure 1: DSM attack and its variations (i.e., RSA, TSA).

In contrast to these studies, we explore the efficiency of NS specific features in detecting DSM attacks. For that, we collect 5G network traffic from a network slicing based 5G testbed that we deploy. We use the generated traffic to calculate 3GPP-based features (i.e., KPI and PM counters) that can profile NSs and NFs behavior. To the best of our knowledge, we are the first to devise a 5G network slicing-based dataset and use related features to detect DSM attacks.

III. DSM Attack - Threat Model and Variations

The DSM attack is a DDoS attack that exploits UEs ISS events to disrupt the performance of both the 5G network CP and its NSs. Such disruption is caused by the signaling associated with the high number of malicious ISS events. In fact, ISS triggers the PDU session release procedure to release the UE PDU session in its current NS, registration, and PDU session establishment to a new desired NS, and other procedures, such as UE authentication to the network [4], [8]. These procedures involve a sheer volume of signaling messages within the 5G CP and the NSs. In the following, we detail the DSM attack assumptions, and threat model while highlighting two of its variations (i.e., RSA and TSA).

A. Assumptions

To perform RSA and TSA, the following assumptions are made:

1) **Access to a set of compromised UEs.** UEs and Internet of Things (IoT) devices are known to be highly vulnerable to attacks such as those noted in [19]. An attacker, can thus compromise a set of UEs and use them as a botnet to perform a DSM attack.
2) **UEs NS configurations and credential information.** We assume that the attacker has access to compromised UEs credentials (i.e., cryptographic keys, NSs configuration, etc.) and can use them to successfully connect to the MNO network and its NSs.
3) **Remote activation of ISS.** We assume that the attacker is able to remotely access and manipulate the compromised UEs in order to force them to trigger ISS to switch between their accessible NSs.

B. DSM Attack Variations - Threat Model

In this work, we exploit two variations of the DSM attack, illustrated in Fig. 1:

Random Slice Attack (RSA). To perform a RSA, the attacker selects different subsets of compromised UEs, and simultaneously switch them to randomly selected NSs. RSA creates varying loads on these random NSs as a result of the varying number of connected UEs.

Target Slice Attack (TSA). To perform a TSA, the attacker selects a target NS to attack and triggers ISS events to switch the compromised set of UEs to the selected NS.

RSA and TSA can be performed as followed:

1) **Select RSA/TSA start time.** The attacker identifies the network's peak times, during which a substantial number of UEs are connected to the network in order to perform the RSA/TSA. This will impose a more pronounced impact on the CP's NFs by augmenting the peak network load with a high frequency of ISS events that will be initiated by the attack.
2) **Compromise UEs.** The attacker compromises a set of UEs and use them as botnet to launch the RSA/TSA.
3) **Launch RSA/TSA** To launch a RSA, the attacker identifies multiple random NSs (i.e., one per each compromised UE) and switches the compromised UEs to these NSs by triggering ISS for each of them. In contrast, to launch a TSA, the attacker switches the compromised UEs to a target, pre-selected NS.
4) **Switch back compromised UEs to random NSs.** The attacker switches back some or all the compromised UEs to random NSs in order to introduce random ISS patterns for UEs and make the attack look stealthy.
5) **Repeat steps (3) and (4).** For increased and varying impact on the network and its NSs, the attacker can repeat steps (3) and (4) using the same or different subset of compromised UEs at different times, mak-

Table I: Features of Inter-slice defender model.

Type	3GPP KPI features	Definition
3GPP-NS	*Registration success rate of one single NS	Success ratio of registration procedures (i.e., ratio of number of successful registrations over total number of attempted registrations) within a single NS for a specific AMF set.
	*PDU session establishment success rate of one NS	Rate of successful PDU session establishment request over total number of attempted requests across all SMFs associated with a specific NS.
	*Mean number of PDU sessions of network and NS	Average number of successful PDU session within a specific NS.
	*Maximum number of PDU sessions of NS	Maximum number of successfully established PDU sessions within a single NS.
	3GPP PM Counter features	**Definition**
3GPP-AMF	Number of initial registration requests	Total number of initial registration requests that AMF receives.
	Number of successful initial registrations	Count of successful initial registrations processed by the AMF.
	Total number of attempted service requests	Number of attempted service requests including those initiated by the network and those initiated by UEs.
	Total number of successful service requests	Cumulative count of successful service requests accounting for those initiated by both the network and by UEs.
3GPP-SMF	+Number of PDU session creation requests	Number of PDU session creation requests received by the SMF.
	+Number of successful PDU session creations	Number of PDU sessions successfully established by the SMF.
	+Number of failed PDU session creations	Count of PDU sessions successfully created by the SMF.
	*Max time of PDU session establishment	Maximum time for PDU session establishment in each granularity period divided into sub-counters for each NS.
	Number of released PDU sessions (AMF initiated)	Number of PDU sessions released at SMF with initiation originating from the AMF.
3GPP-NSSF	Number of NS selection requests	Total number of NS selection requests that the NSSF receives.
	Number of successful NS selections	Total successful NS selections executed by the NSSF.
	Number of failed NS selections	Number NS selection attempts that failed at the NSSF.
	Non-3GPP PM Counter feature	**Definition**
AMF	Number of failed initial registrations	Number of unsuccessful registrations.

* Feature computed per each NS; + Feature computed per each SMF

ing the attack harder to detect while always causing disruptions to the network.

C. Illustrating Example

To better explain RSA and TSA, we represent in Fig. 1, a telecommunication network with multiple base stations and connected UEs. Note that the limited number of UEs in the figure is only for illustrative purposes and a larger number is assumed in real networks. The network is configured with three different NSs, each providing services for a specific vertical industry. For simplicity and without loss of generality, we account for a gaming NS (i.e., NS_1), an augmented reality NS (i.e., NS_2) and an industry 4.0 NS (i.e., NS_3). Each of these NSs has its own SMF, User Plane Function (UPF) and a set of Network Slice Components (NSCs). An NSC is a capability (i.e., abstraction of an NF) that fulfills the functional requirements of an end-to-end NS [20]. These NSs are served by a common set of CP NFs represented in the blue box in Fig. 1. Particularly, the sharing of AMF by-design (i.e., based on 3GPP) between NSs is intrinsic to allow the UEs to connect simultaneously many of them [21].

We consider UEs initially connected to a specific NS identified by the circle they belong to (i.e., $\{UE_1, UE_k\}$ connected to NS_1, $\{UE_2, UE_3, UE_4, UE_m\}$ connected to NS_2, and $\{UE_5, UE_6, UE_n\}$ connected to NS_3). We assume that the attacker compromises some of these UEs and uses them to perform RSA and TSA. More precisely, UE_2, UE_3, and UE_4 are compromised and used to perform a TSA towards NS_3 (i.e., the targeted NS). UE_5, UE_6 and UE_n are used to perform an RSA. The random NS selected for ISS of UE_5, UE_6 is NS_1 while NS_2 is selected for UE_n. The user and CP connections shown in Fig. 1 reflect the UEs connections after the attacks. RSA and TSA lead to a DDoS on the network, particularly impacting the AMF. The AMF becomes a bottleneck as it is the first point of contact between the UEs and the network and is involved in the ISS related procedures.

IV. Inter-slice defender: Network slice anomaly detection solution

We present Inter-slice defender (Fig. 2), our novel NS anomaly detection solution that leverages LSTM-Autoencoder, 3GPP KPIs [22] and PM counters [23] (Table I) to detect DSM attack and its variations. The Inter-slice defender is composed of three modules detailed below.

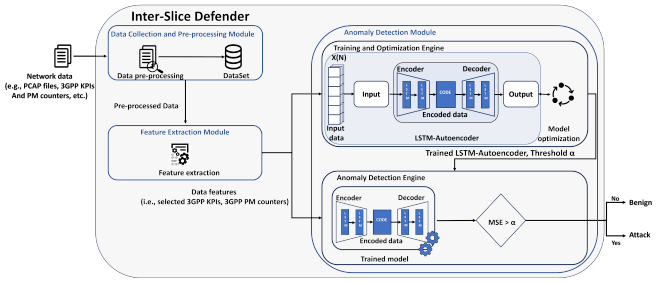


Figure 2: Inter-slice defender architecture.

A. Data Collection and Pre-processing Module

The data collection and pre-processing module (Fig. 2) collects data (e.g., PCAP files, 3GPP KPI, PM counters, etc.) from the 5G network and pre-processes it to be ready for use by the feature extraction module. The collected data can include PCAP files depicting signaling messages between the 5G network components (i.e., Radio Access Network (RAN) and core NFs) and which pertain to the different 5G procedures such as ISS, registration, etc. It can also account for 3GPP KPIs and PM counters (Table I) that are usually calculated at the different 5G NFs, and can be available at the Network Data and Analytics function (NWDAF) [24]. NWDAF is a 5G NF that facilitates data collection and analysis. It can collect KPIs from the 5G NFs and calculate others such as those KPIs related to NSs [24].

The data collected by the data collection and pre-processing module then undergoes a transformation that simplifies the extraction of features. For instance, in this work, we use TShark [25] to collect PCAP files from the network. We merge these files into a single one while maintaining their chronological order. This will serve the accurate calculation of time series based features by the feature extraction module. Then, we pre-process the unified PCAP file by extracting the most relevant information (e.g., IP source, IP destination, port destination, HTTP/2 header, etc.) for feature extraction. We save them in a CSV file that is then shared with the feature extraction module.

B. Feature Extraction Module

The feature engineering module acquires the pre-processed data from the data collection and pre-processing module and uses it for feature extraction. This data is used for feature extraction, selection, and normalization to extract 3GPP KPI [22] and PM counters [23] such as those listed in Table I. These features capture the 5G network and its NSs normal behavior, along with the abnormalities that can be caused by a DSM attack. The calculated features are time-series based, suitable for LSTM-Autoencoder required input format. To select the most pertinent features, a variance threshold process is employed which eliminates features characterized by minimal variations or perceived as noise.

Consequently, a total of 17 distinctive features (Table I) are selected to comprehensively scrutinize, analyze, and establish the normal behavioral patterns exhibited by 5G networks. They can be classified per type that reflects their calculation. For instance, some features are calculated per NF (i.e., AMF, SMF, NSSF), mainly those NF involved in ISS related procedures (e.g., "Number of PDU session creation requests" feature is calculated at the SMF as it is involved in PDU session establishment procedure) while others are calculated per NS. With the exception of "Number of failed initial registrations" feature that is not standardized by 3GPP, the computation of the remaining features follows 3GPP specifications. These features can be collected from the NWDAF or the NFs if available or can be calculated from the PCAPs as in this work. Finally, it is worth noting that in the case where any of the aforementioned NFs is dedicated to an NS, its related features will then reflect the NS it serves.

C. Anomaly Detection Module

The anomaly detection module is composed of two engines, mainly the training and optimization engine and the anomaly detection engine, detailed hereafter.

Training and optimization engine. The training and optimization engine is dedicated to training the anomaly detection model based on LSTM-Autoencoder [26], optimizing its architecture, hyper-parameters, and selecting a threshold that will lead to good detection performance. The choice of LSTM-Autoencoder lies in its capability of capturing temporal dependencies in the data through an LSTM and combining it with an Autoencoder. Autoencoder creates an effective representation of the sequential patterns in the data by encoding and decoding them. This yields very efficient for DSM attack detection where dependencies and precedence constraints of the different 5G procedures representing ISS are highly descriptive of the attack. For instance, an ISS event to a new NS cannot occur before de-registering from the current NS and registering to the new one (Section V-B). Further, the frequency and attack patterns over time can be captured by LSTM through its cells capabilities that can remember values over different time intervals [26]. With Autoencoder coming into play, the most relevant features representing the data are learned.

An Autoencoder is an unsupervised model that learns to reconstruct the input data [26]. When used for anomaly detection, it is trained on mostly benign data. Hence, it will fail to reconstruct any anomalous input, which will result in a relatively big difference between the input and its reconstructed version. This difference is known as the reconstruction error and is used to depict anomalies by comparing against a threshold as we explain next. The training and optimization engine determines this threshold such that the F1-score is maximized (see Section VII-B).

Anomaly detection engine. The anomaly detection engine uses the trained LSTM-Autoencoder model provided by the training and optimization engine along with the selected threshold α to perform real-time anomaly detection (Fig. 2). Such detection is based on comparing the reconstruction error provided by the LSTM-Autoencoder against the selected threshold α. In this work, we use the Mean Square Error (MSE) [26] as the reconstruction error metric given its sensitivity to outliers, however, other metrics such as the root mean squared error and mean absolute error can also be used. The MSE represents the degree of alignment between the model predicted outcomes and the actual ground truth. Given that LSTM-Autoencoder is trained to learn benign behavior, it is expected to succeed in reconstructing benign data which will lead to a low reconstruction error (i.e., MSE). In contrast, a high reconstruction error is expected in case of an anomaly. Thus, if the $MSE \leq \alpha$, the data is classified as benign, while an $MSE > \alpha$ depicts malicious data.

V. Environmental Setup And Data Simulation

Given the lack of a 5G dataset suitable for training and testing our Inter-slice defender solution, we present, in this section, our environmental setup and data simulation strategy. We highlight our 5G testbed that we deploy to simulate normal network traffic in addition to RSA and TSA.

A. 5G Testbed

We employ the open source free5GC-compose and free5GC all-in-one implementation [16], adhering to the 3GPP standard [27] to build our testbed, depicted in Fig. 3. The testbed runs free5GC-compose version 3.3.0 as the CP on a Virtual Machine (VM) operating Ubuntu 20.04 – Focal. The VM features 8 virtual CPUs, 8 GB of RAM, and a 60 GB of hard drive, with each NF encapsulated within this virtualized environment. To enhance the realism of our simulation environment, we opt to segregate the RAN from

the CP. This segregation involves installing UERAMSIN 3.2.6 [17], a UE and RAN simulator, on a distinct VM. Our devised testbed comprises four NSs, each featuring a dedicated UPF installed in a separate VM, and a dedicated SMF. For the installation and configuration of all UPFs, we use the free5GC all-in-one version 3.3.0 [16] while the SMFs are deployed on containers in the VM hosting the CP. Furthermore, our testbed hosts 92 UEs configured to be able to connect to the existing four different NSs. The creation and management of the VMs within our framework are orchestrated through OpenStack [28].

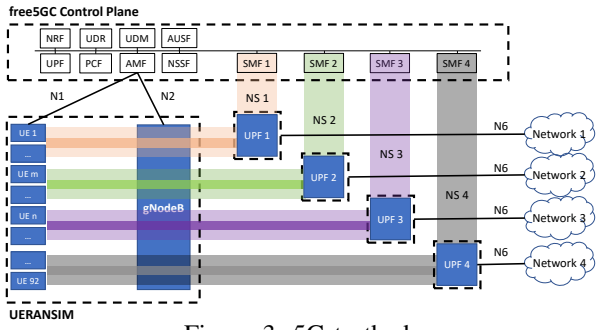


Figure 3: 5G testbed.

B. Simulation of Normal and Attack Network Traffic

To collect benign and attack data, we perform three different simulations. The initial simulation replicates a 5G network operating under normal conditions without any external attacks. In contrast, the subsequent two simulations were conducted to assess the impact of RSA and TSA.

Normal network traffic simulation. Throughout the 120-minute duration dedicated to this simulation, 92 UEs are used to simulate varying loads of normal network traffic. Each UE randomly triggers many 5G procedures from those detailed in Table II. Notably, the 5G procedures listed under the "Possible subsequent procedures" column are executable solely and exclusively if the 5G procedure indicated under the "Triggered procedure" column has been successfully completed. This logical interdependence underscores the sequential nature of these procedures. For instance, as outlined in Table II, following the successful completion of a registration procedure, several events can be triggered: ISS, Uplink, Downlink, UE release PDU session, and gNodeB release PDU session. Finally, the benign activity of the UEs is recorded in PCAP files. The latter constitutes the benign dataset that will be used for our LSTM-Autoencoder model training and testing.

RSA and TSA Simulations. To conduct the RSA and TSA simulations, we adhere to the same approach as in the benign simulation. We perform the simulation of each attack using a total of 92 connected UEs out of which 28 are compromised and used for the attack. Following our threat model (Section III), the attacker strategically decides on the time to launch the RSA or TSA such that it coincides with the network's peak activity. As a result, during the attack simulations, the network operates normally for the first 60 minutes, after which the attack is initiated when the load on the network is designed to be at its peak. When the attacks start, the 28 compromised UEs are connected to any of the four NSs and are used to perform ISS events in the quest of overloading the CP.

Table II: Logical dependency between 5G procedures.

Triggered procedure	Possible subsequent procedures
ISS	ISS, Uplink, Downlink, UE release PDU session, gNodeB release PDU session
Registration	ISS, Uplink, Downlink, UE release PDU session, gNodeB release PDU session
Uplink	ISS, Downlink, UE release PDU session, gNodeB release PDU session
Downlink	ISS, Uplink, UE release PDU session, gNodeB release PDU session
UE release PDU session	ISS, Downlink, Uplink, gNodeB release PDU session
gNodeB release PDU session	ISS, Uplink, Downlink

C. Datasets for Anomaly Detection Model

The data generated from the aforementioned simulations is used to create different datasets to facilitate the training and evaluation of our Inter-slice defender model. These datasets include benign and/or attack records (Table III).

Table III: Datasets statistics.

Dataset type	Total number of records	Benign records	Attack records	
			RSA	TSA
Training + validation	40000	40000	0	0
Optimization	10000	5000	2500	2500
RSA Test	20000	10000	10000	0
TSA Test	20000	10000	0	10000

1) Training and validation dataset: This dataset exclusively comprises benign data obtained from the normal network traffic simulation. It serves as the foundation for training and validating our model.

2) Optimization dataset: The optimization dataset includes both benign and malicious data extracted from the data generated from the RSA and TSA simulations. An equal number of benign and attack records were extracted from each of the attack simulations to generate this dataset. The optimization dataset is used to determine the threshold that maximizes the F1-score for distinguishing between benign patterns and potential attacks, as we explain in Section VII-A.

3) Test datasets: To rigorously evaluate the performance of our anomaly detection model, three different test datasets incorporating both benign and malicious data are considered. A dataset per each DSM attack variant (i.e., RSA test dataset, TSA test dataset) is generated as a result of the RSA and TSA simulations respectively. This approach allows for a thorough assessment of the model's ability to discern between normal and anomalous patterns.

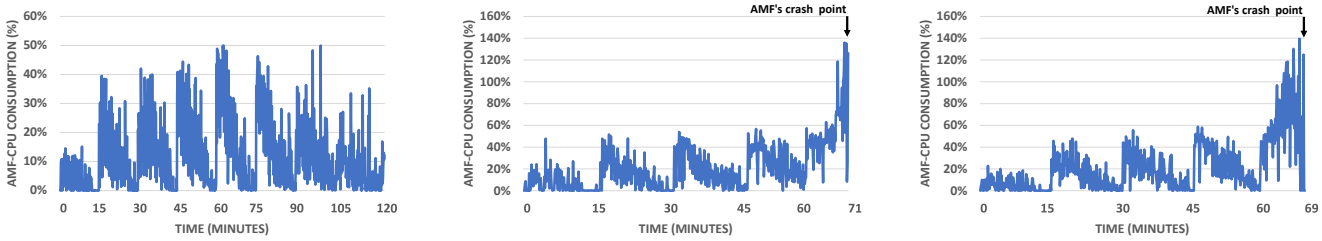
It is crucial to highlight that these datasets are meticulously designed to be mutually exclusive, ensuring that there are no redundant records between them.

VI. DSM Attack Impact on 5G Control Plane

To assess the impact of the RSA and TSA on the 5G network, we observe the CPU utilization of the different 5G CP NFs during RSA and TSA simulations and compare it with their CPU utilization during normal network traffic.

A. Impact on AMF

We first focus on the impact of the attacks on the AMF given that it is the CP NF that is involved the most in UE to 5G network communication and is the first CP point of contact. In fact, despite the role AMF plays in UE registration, authentication, and NS selection and allocation,



(a) AMF - Benign simulation (b) AMF - RSA (c) AMF - TSA

Figure 4: AMF CPU consumption during benign and attack simulations.

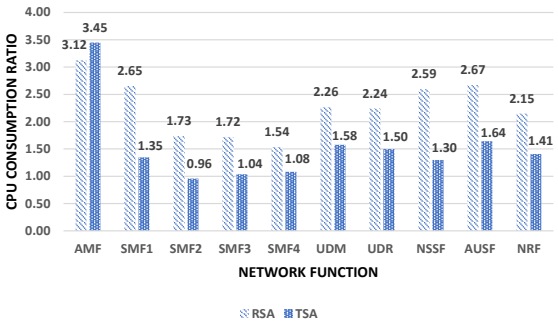


Figure 5: CPU consumption ratio during RSA and TSA.

it is usually shared among different NSs (as explained in Section III-C). Thus, when a significant number of UEs simultaneously perform ISS, such as in the case of RSA and TSA, we observe a significant increase in the AMF CPU utilization which leads to a DDoS (Fig. 4).

As explained in Section V-B, we simulate varying loads of normal network traffic and trigger the RSA and TSA at the peak load depicted at $time = 60\ minutes$. Fig. 4a shows the AMF CPU utilization during normal network traffic simulation and depicts the highest utilization of 49.95% at $time = 60\ minutes$. In contrast, Fig. 4b and 4c show that the AMF CPU consumption reaches 135.98% and 139.55% during the RSA and TSA respectively which are launched at the peak network load (i.e., $time = 60\ minutes$). The figures show that RSA and TSA last for 11 *minutes* and 9 *minutes* respectively before the crash of the AMF, and hence the whole network. This shows the DDoS impact the DSM attack can have on the 5G network and further emphasizes the need to secure 5G networks against it. Finally, it is worth noting that the AMF CPU utilization could exceed 100% in our testbed because our CP NFs are containerized and can borrow from each other the CPU unused resources. Nonetheless, if the network does not allow sharing of unused CPU resources, the DDoS impact would have been observed earlier.

B. Overall Impact on 5G CP NFs

Despite the significant impact RSA and TSA have on the AMF, we observe a similar increase in the CPU consumption on other CP NFs as shown in Fig. 5. In fact, this figure shows the average CPU ratio for all the CP NFs during both RSA and TSA simulations. This ratio is calculated by dividing the average CPU consumption for each NF during the attack period (i.e., $[60\ minutes - 71\ minutes]$ for RSA, $[60\ minutes - 69\ minutes]$ for TSA) over that during the benign simulation for that same period.

Note that except for the AMF, the increase in the CPU utilization for all the NFs is greater during the RSA than during the TSA. Fig. 5 shows that RSA has a bigger impact on the 5G CP NFs. However, TSA has a higher impact on the AMF than the RSA, which results in degrading its performance in handling the requests, many of which ended up being dropped. Thus, fewer requests were forwarded to other CP NFs during TSA than during RSA which explains a smaller increase in their CPU utilization. Finally, unlike the AMF which CPU consumption exceeds 100% during the attacks, we observe that CPU consumption of other CP NFs remains under 50%.

VII. Experimental Results

In this section, we evaluate Inter-slice defender and assess its effectiveness in detecting RSA and TSA while examining its performance against contaminated data.

A. LSTM-Autoencoder Architecture Selection

To determine the best architecture for Inter-slice defender LSTM-Autoencoder that is able to efficiently detect DSM attack and its variations, we train and validate the performance of multiple architectures and examine their performance. Thus, we allocate 20% of the training dataset as a validation dataset, and we train the model using the remaining portion of the training dataset (Table III). To select the model hyperparameters, we apply the K-fold cross-validation technique [29] for both training and validation. K-fold cross-validation is an effective method for validating and refining DL models, as it helps prevent overfitting and offers a more thorough approach to evaluate model performance. Additionally, to further mitigate the risk of overfitting, our model incorporates L1 regularization and dropout techniques within the encoder.

We test and evaluate various LSTM-Autoencoder architectures and select {64, 32, 32, 32, 64} as Inter-slice defender model architecture given that it provides the best detection performance. This architecture is composed of two LSTM layers of 64 and 32 neurons respectively, forming the encoder, and a decoder having the encoder's mirrored architecture. The code of the LSTM-Autoencoder is 32 neurons. We train our model with the selected architecture after fine-tuning its hyperparameters (Table IV). Its training time is equal to 21.92 seconds.

Table IV: LSTM-Autoencoder hyperparameters.

Hyperparameter	Value
Epochs	20
Dropout	0.1
L1 regularization	0.15
Batch size	16
Loss Function	MSE
Learning rate	0.01
Optimizer	Adam
Hidden activation function	Relu

B. Reconstruction Error Threshold Selection

To detect anomalies, there is a need to compare the reconstruction error (i.e., MSE) provided by Inter-slice defender model against a selected threshold α (Section IV-C). In this work, we resort to the F1-score, that represents the harmonic mean between precision and recall, to determine α. We evaluate the performance of our model against different threshold values, as shown in Fig. 6, using our optimization dataset (Table III). Fig. 6 depicts that with the increase of the threshold value, the precision increases and stabilizes at 100% while the recall decreases as attack data starts getting misclassified as benign. After calculating and comparing the F1-score for the different threshold values presented in Fig. 6, we select $\alpha = 0.1408$ that maximizes the F1-score. We use this value in the remainder of the experiments to evaluate the model performance.

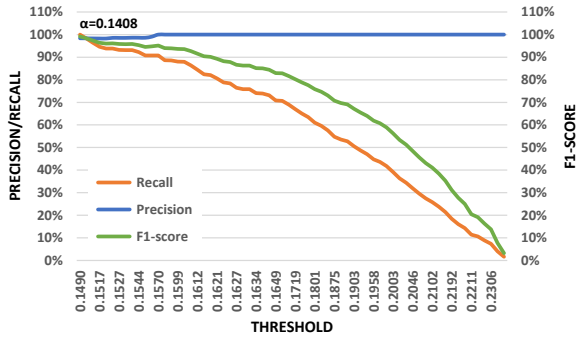


Figure 6: Threshold selection.

C. Inter-slice Defender Performance

1) Impact of Timesteps: Recall that LSTM is a recurrent neural network that is known for its ability to detect dependencies in a sequence of data. It uses a lookback hyperparameter that determines the number of previous timesteps within a sequence that will be used to predict the next timestep. Thus, we evaluate in Fig. 7, the performance of Inter-slice defender over multiple values of the lookback hyperparameter (i.e., number of timesteps) when tested on RSA and TSA test datasets (Table III). Our results show that Inter-slice defender detection performance degrades for these test datasets when the number of timesteps increases. This is because UEs are highly dynamics and their long term communication patterns and resulting load on the network may not necessarily be indicative of their most recent behavior. Thus, the most updated KPIs and counters are better suited for predicting the next benign or attack behavior. Nonetheless, the model performance remains satisfactory with an F1-score above 87% even when $timesteps = 8$. In the remainder of the experiments, we evaluate the model performance with $timesteps = 1$ as it provides the best detection performance with an F1-score of 99.1% and 98.4% when tested with RSA and TSA datasets, respectively.

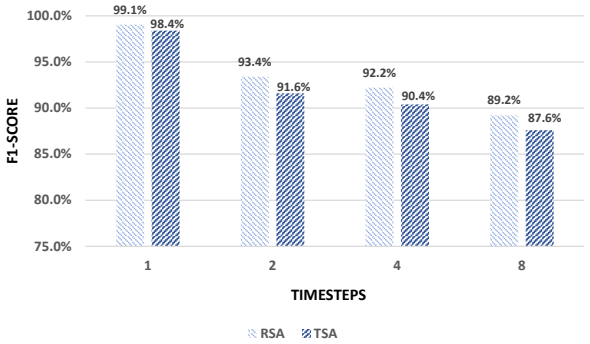


Figure 7: Inter-slice defender performance over different lookback values (i.e., timesteps).

2) Detailed performance with timesteps = 1: Given that Inter-slice defender model has shown the best performance with $timesteps = 1$ and a threshold $\alpha = 0.1408$, we take a closer look at this performance in Fig. 8. Fig. 8 depicts F1-scores of 99.1% and 98.4% recorded for RSA and TSA datasets, respectively, leading to an average F1-score of 98.75%. Further, our tests show that the model exhibits a perfect recall for RSA and TSA test datasets revealing that the model is able to correctly discern RSA and TSA. However, the degraded precision depicts that the model misclassifies benign data as anomalous.

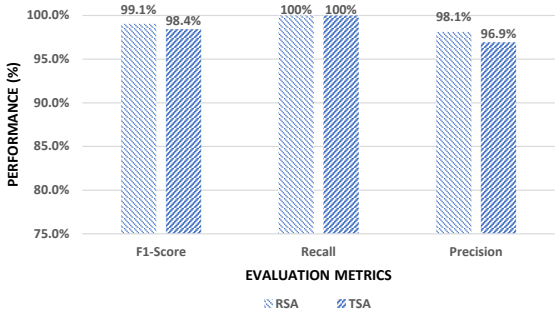


Figure 8: Detailed Inter-slice defender performance with a timesteps = 1 and threshold $\alpha = 0.1408$.

3) Impact of contamination: Given that it is hard to obtain purely benign data in real-world deployment, we assess the robustness of Inter-slice defender model in the presence of contaminated data. Thus, we retrain our model with the training dataset (Section V-C) after contaminating it with different percentages of RSA and TSA records. Then, we test the model with the RSA and TSA test dataset (i.e., after excluding RSA and TSA records used for training) while fixing the $timesteps = 1$ and the threshold $\alpha = 0.1408$. Fig. 9 shows that the performance of our model significantly degrades, especially with 1% and 2% of contamination, with the increase of the contamination percentage. However, starting with 3% of contamination, only a slight degradation in the performance is observed while maintaining an F1-score

above 84%. Nonetheless, this test shows that further fine-tuning of the model needs to be performed, upon the need, in real network deployment.

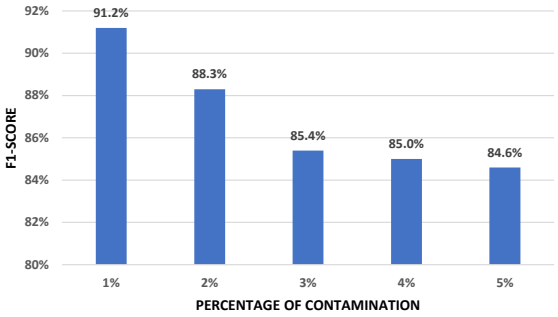


Figure 9: Inter-slice defender performance in the presence of contaminated data.

VIII. Conclusion

In this work, we introduced, tested, and evaluated two variants of the known DSM attack, namely RSA and TSA, that exploit ISS procedures. We analyzed the impact that these attacks have on the 5G network using our free5GC testbed and UERANSIM simulator. We showed that they cause a DDoS on the network due to the resulting overload on the AMF. Further, we developed Inter-slice defender, a novel anomaly detection solution to detect RSA and TSA inter-slice attacks using an LSTM-Autoencoder model trained on 3GPP KPIs and PM counters. The use of these 3GPP features makes our Inter-slice defender solution easily deployable as part of the NWDAF, as those features are usually/can be made available in this 5G NF. Inter-slice defender was evaluated under different conditions (i.e., architectures, timesteps, contaminated data, test datasets, etc.) and has shown its ability to detect RSA and TSA attacks with an average F1-score of 98.75% and a perfect recall.

Finally, as a future work, we aim at studying the performance and economic impact that the DSM attack and its variations can have on NSs when the AMF and CP NFs resources are large enough to contain the attack.

Acknowledgments

This research is made possible through the financial support of Concordia University, Ericsson research, Montreal, and a grant from the National Cyber security Consortium Canada under the Cyber Security Innovation Network.

References

[1] M. Chahbar, G. Diaz, A. Dandoush, C. Cérin, and K. Ghoumid, "A comprehensive survey on the e2e 5g network slicing model," IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 49–62, 2020.

[2] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network slice lifecycle management for 5g mobile networks: An intent-based networking approach," IEEE Access, vol. 9, pp. 80 128–80 146, 2021.

[3] S. Zhang, "An overview of network slicing for 5g," IEEE Wireless Communications, vol. 26, no. 3, pp. 111–117, 2019.

[4] M. M. Sajjad, C. J. Bernardos, D. Jayalath, and Y.-C. Tian, "Inter-slice mobility management in 5g: motivations, standard principles, challenges, and research directions," IEEE Communications Standards Magazine, vol. 6, no. 1, pp. 93–100, 2022.

[5] 3GPP, "System architecture for the 5G System(5GS): TS 23.501 V18.4.0," 2023.

[6] R. F. Olimid and G. Nencioni, "5g network slicing: A security overview," IEEE Access, vol. 8, pp. 99 999–100 009, 2020.

[7] C. De Alwis, P. Porambage, K. Dev, T. R. Gadekallu, and M. Liyanage, "A survey on network slicing security: Attacks, challenges, solutions and research directions," IEEE Communications Surveys & Tutorials, 2023.

[8] 3GPP, "Procedures for the 5G System (5GS): TS 23.502 V18.4.0," 2023.

[9] V. N. Sathi and C. S. R. Murthy, "Distributed slice mobility attack: A novel targeted attack against network slices of 5g networks," IEEE Networking Letters, vol. 3, no. 1, pp. 5–9, 2020.

[10] V. N. Sathi and C. S. R. Murthy, "Dsm attack resistant slice selection in 5g," IEEE Wireless Communications Letters, vol. 10, no. 7, pp. 1469–1473, 2021.

[11] H. Bisht, M. Patra, and S. Kumar, "Detection and localization of ddos attack during inter-slice handover in 5g network slicing," in 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023, pp. 798–803.

[12] M. S. Khan, B. Farzaneh, N. Shahriar, N. Saha, and R. Boutaba, "Slicesecure: Impact and detection of dos/ddos attacks on 5g network slices," in 2022 IEEE Future Networks World Forum (FNWF). IEEE, 2022, pp. 639–642.

[13] N. A. E. Kuadey, G. T. Maale, T. Kwantwi, G. Sun, and G. Liu, "Deepsecure: Detection of distributed denial of service attacks on 5g network slicing—deep learning approach," IEEE Wireless Communications Letters, vol. 11, no. 3, pp. 488–492, 2021.

[14] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, and P. Kankariya, "Secure5g: A deep learning framework towards a secure network slicing in 5g and beyond," in 2020 10th annual computing and communication workshop and conference (CCWC). IEEE, 2020, pp. 0852–0857.

[15] 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.

[16] Free5GC, "Free5gc," https://free5gc.org/, 2023, [Online; accessed Jul-2023].

[17] Aligungr, "Ueransim," https://github.com/aligungr/UERANSIM, 2023, [Online; accessed May-2023].

[18] M. A. Al-Shareeda, S. Manickam, and M. Ali, "Ddos attacks detection using machine learning and deep learning techniques: Analysis and comparison," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 2, pp. 930–939, 2023.

[19] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2702–2733, 2019.

[20] V. N. Sathi, M. Srinivasan, P. K. Thiruvasagam, and S. R. M. Chebiyyam, "A novel protocol for securing network slice component association and slice isolation in 5g networks," in Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2018, pp. 249–253.

[21] 3GPP, "System Architecture for the 5G System (5GS): TS 23.501 V18.3.0," 2023.

[22] 3GPP, "Management and orchestration; 5G end to end Key Performance Indicators (KPI): TS 28.554 V18.2.0," 2023.

[23] 3GPP, "Management and orchestration; 5G performance measurements: TS 28.552 V18.3.0," 2023.

[24] M. A. Garcia-Martin, M. Gramaglia, and P. Serrano, "Network automation and data analytics in 3gpp 5g systems," IEEE Network, pp. 1–1, 2023.

[25] TSHARK.DEV, "Tshark.dev," https://tshark.dev/, 2023, [Online; accessed Jul-2023].

[26] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using lstm based autoencoder," in Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 37–45.

[27] 3GPP, "The 5g standard," https://www.3gpp.org/, 2023, [Online; accessed Aug-2023].

[28] OpenStack, "An openinfra foundation project," https://www.openstack.org/, 2023, [Online; accessed Apr-2023].

[29] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 8, pp. 1586–1594, 2020.