Detecting Workload-based and Instantiation-based Economic Denial of Sustainability on 5G environments

Jorge Maestre Vidal, Marco Antonio Sotelo Monge, Luis Javier García Villalba
Department of Software Engineering and Artificial Intelligence
School of Computer Science, Complutense University of Madrid
C/ Prof. José García Santesmases, 9, Ciudad Universitaria, 28040, Madrid, Spain
{jmaestre, masotelo}@ucm.es, javiergv@fdi.ucm.es

ABSTRACT
This paper reviews the Economic Denial of Sustainability (EDoS) problem in emerging network scenarios. The performed research studied them in context of adaptive approaches grounded on self-organizing networks (SON) and Network Function Virtualization (NFV). In particular, two novel threats were reviewed in depth: Workload-based EDoS (W-EDoS) and Instantiation-based EDoS (I-EDoS). With the aim to contribute to their mitigation a security architecture with network-based intrusion detection capabilities is proposed. This architecture implements machine learning techniques, network behaviour prediction, adaptive thresholding methods, and productivity-based clustering for detecting entropy-based anomalies based on the observed workload (W-EDoS) or suspicious variations of the productivity observed at the virtual instances (I-EDoS). A detailed experimentation has been conducted considering different calibration parameters under different network scenarios, on which the security architecture has been assessed. The results have proven good accuracy levels, hence demonstrating the proposal effectiveness.

CCS CONCEPTS
• Networks → Network management; Cloud Computing; • Security and privacy → Network Security;

KEYWORDS
Economical denial of sustainability, information security, intrusion detection systems, network function virtualization, self-organizing networks

1 INTRODUCTION
The complexity and sophistication of emerging network architectures has noticeably increased and nowadays, they demand more agile, robust and effective network management paradigms, were their scalability is mandatory. In the last years, 5G networks have emerged as a promising technology towards the fulfillment of the challenging requirements posed by the current and future communication scenarios [26]. They have motivated a smart integration of innovative communication network solutions, such as Network Function Virtualization (NFV), cloud computing, Software Defined Networking (SDN), artificial intelligence, Self-Organizing Networks (SON), among others. In particular, the suitable combination of SDN and SON is considered one of the most relevant to accomplish the 5G Key Performance Indicators (KPI) [17]. Because of this, recent 5G projects have been integrating such technologies to incorporate cognitive capabilities for the inference of the network status, thus enhancing the autonomic management capacity [23] when dealing with heterogeneous network environments [2]. A clear example of this is observed in the SELFNET project [32], where a 5G-oriented framework for self-organizing management is proposed.

The research introduced in this paper is thereby focused on SON-networks as promising solutions for fulfilling the aforementioned challenges. Originally, SON networks were proposed as a response to address the problem of LTE mobile network efficiency [4], being consequently standardized by the Third Generation Partnership Project (3GPP) on which their capability to reduce operational costs by automation is remarked [1]. In this way, SON poses a transition from traditional management paradigms where human intervention is mandatory (open-loop) towards a fully automated model (closed-loop). Another important topic of this research is the role of cloud computing in the SON context, which has allowed the virtualization of network functions aimed to address scalability issues of network infrastructures [45], which in the meantime yields the reduction of costs in the deployment of sensors and actuators involved at SON. That network elasticity is orchestrated through auto-scaling policies, which expose vulnerabilities that can be exploited by an attacker with the aim to produce an economical overspending of the target victims, hence making a cloud service unsustainable [6]. This effect is known as Economical Denial of Sustainability (EDoS), and it poses security threats which have not been reviewed in depth by the research community, being frequently confused with flooding-based or complexity-based Denial of Service (DoS) attacks. EDoS threats have gained sophistication with the expansion of the next generation technologies, hence demanding the deployment of detection strategies toward their mitigation [40]. The research presented throughout this paper contributes with an in-depth review of the EDoS problem in conventional cloud infrastructures and their adaptation to self-organizing scenarios. It has entailed the distinction of two main threats: EDoS based on the exploitation of the network elements workload (W-EDoS), and EDoS based on fraudulent instantiation of virtualized network functions (I-EDoS). It is also proposed a multilayered architecture compatible with the
ETSI-NFV [16] model for their detection, which combines machine learning techniques, prediction methods and clustering algorithms. The effectiveness of the detection strategy has been assessed in a real SON environment, which has exposed promising preliminary results.

This paper is divided into seven sections, being the present introduction the first of them. Section II reviews the state of the art about EDoS attacks related with SON environments and the proposals for their mitigation. Section III defines the W-EDoS and I-EDoS attacks and their characterization. In section IV, the proposed approach for detecting EDoS threats is introduced. Section V describes the evaluation methodology conducted throughout the experimentation. In section VI the experimental results are discussed. Finally, Section VII presents the conclusions and highlights the future research lines.

2 BACKGROUND

This section describes the main characteristics of EDoS attacks, and the efforts proposed by the research community towards their mitigation.

2.1 Economical Denial of Sustainability

The expression Economical Denial of Sustainability was coined by C. Hoff in 2008 [10] [11] to describe attacks originally targeted against cloud computing platforms, in which the intruder has the goal to fraudulently increase the economic expenditures derived from the maintenance of the hosted cloud services. Therefore, their main consequence is to affect the economic viability in the wake of higher expenses, which can motivate either the migration to other cloud provider or, even worse, the service unsustainability. Interested in this new threat, R. Cohen [31] extended its definition pointing out the exploitation of vulnerabilities of self-scaling processes as the most implemented procedures to achieve the aforementioned fraud, an approach that nowadays is mainly supported by the research community. Although EDoS introduces a new paradigm of intrusion inherent in emerging network technologies, it has drawn the attention of different organizations for information security, which usually refer to EDoS as Reduction of Quality (RoQ) [9] attacks or Fraudulent Resource Consumption (FRC) [36] threats that typically take advantage of the payment-for-service solutions offered by the cloud computing suppliers [30]. These threats usually try to go unnoticed by monitoring elements via registering consumption distributions and requests that resemble those of normal and legitimate clients [10] [11]. Therefore, it is common to undertake the intrusion by requesting computationally expensive requests [36]. This also poses a representable difference with events of legitimate nature capable of jeopardizing the availability of the protected system, such as the massive access of legitimate users to the hired services, commonly referred as flash crowds [44]. At the present time, there are different techniques to perpetrate EDoS threats, for example, by requesting large files or costly queries to databases [7]. HTTP requests linked from XML content [41], or by exploiting specific vulnerabilities of the web service platforms [46][35][34]. In addition to causing an economic impact, EDoS attacks potentially lead to other secondary risks. G. Sonami et al. [36] reviewed this problem by pointing out different collateral damages, which vary depending on the role of each actor in a cloud computing deployment. For example, the provider tends to lose reputation while customers decide to contract cheaper services to rival enterprises. Clients also may pay an excessive amount of money for services that they were not using. These threats also may affect the operational capacity of the services at the different information processing layers that support them, this being the case of infrastructure, network function virtualization or multitenacy [9][35].

2.2 Countermeasures

Despite the growing relevance of the EDoS threat at the emerging networking landscape, the bibliography does not provide an extensive number of publications that address the challenges it poses. They usually describe solutions based on analyzing network-level metrics typical on flooding-based denial of service recognition. In order to facilitate their understanding, the contributions are classified as they are classical organized at the research related to conventional DDoS defense [35]: detection, mitigation/prevention, and source identification.

Detection. The publications at this field aim on identifying the EDoS attacks. A significant portion of them analyzed local-level metrics for modeling the resource consumption and self-scaling processes of the monitored environment [35]. Other publications lie on studying network-level data [20] and the browsing habits of the clients [34]. Note that although the research focused on local metrics has proven to be effective by best fitting the definition of EDoS attacks proposed by Hoff [10] [11], the network-based solutions are able to take advantage of the state-of-the-art about flooding-based DDoS and the emerging communication paradigms.

Mitigation and Prevention. The contributions towards EDoS mitigation trend to focus on increasing the restriction level of the protected system through access control techniques. Tuning tests based on image recognition [22] or resolution of cryptographic puzzles [25] are usually the most commonly applied methods. In contrast to the detection techniques, they do not require the previous identification of the threat, but their deployment usually penalizes the user Quality of Service or the operational expenditure. It worth emphasizing that most of the proposals categorized as mitigation solutions can be implemented as prevention measures, hence ignoring previous threat identification stages.

Source Identification. Finally, the research that aims on discovering the origin of EDoS situations attempts to track the attacker. Because of the complexity that this challenge implies, the scope of identifying the threat source is often reduced to get as close as possible to the attacker. The bibliography related with the defense against DDoS serves to this purpose [21], being worth to highlight among the previous publications those based on analyzing error messages [3], honeypot deployment [42] and packet marking [43].

3 EDoS IN THE SON ENVIRONMENT

Hoff [10][11] pointed out the great similarity that EDoS activities present with respect to the legitimate traffic. It is then assumable that, in the context of a client-server architecture, that similarity is expressed in terms of the set of clients and the requests they generated, thus taking into account their number, distribution over time and computational complexity. These traits characterize both
normal and EDoS situations, consequently being assumed throughout the performed research. The following subsections define each type of attack, its characterization and impact on the cost model.

### 3.1 W-EDoS: Workload-based EDoS

An attack of Economic Denial of Sustainability based on Workload (W-EDoS) is characterized by the execution of operations of high computational cost in the virtual instances hosted on a cloud computing provider. They are executed at server-side, thus generating a high workload in response to seemingly legitimate client requests. Under this premise, the existence of a W-EDoS attack is assumed when a monitored network environment presents conditions of similarity with legitimate network traffic, but where the average workload per request is significantly greater in terms of quantity and distribution. Fig. 1 shows a representation of a W-EDoS attempt launched on an instantiated VNF. The effect of the W-EDoS attack is to force the SON management layer to scale the instantiated VNFs vertically or horizontally, hence implying to waste additional computational resources (computation, storage, etc.) hired by payment per use policies, which causes negative effects in the economic sustainability of the offered services they support.

### 3.2 I-EDoS: Instantiation-based EDoS

An attack of Economic Denial of Sustainability based on Instantiation (I-EDoS) is characterized by the exploitation of some existing vulnerability either in the cloud service platform or in virtual functions, that leads to the automatic creation of additional VNF instances in one or several points of the network. In this way, an increase in the number of deployed instances is observed. Note that their average productivity is typically considerably lower, since their deployment would not have been necessary under legitimate circumstances. Therefore, the existence of an I-EDoS attack is assumed when a monitored network environment displays conditions of similarity with legitimate network traffic; but with a significant increase in the number and distribution of virtual instances, as well as a decrease in their average productivity. Fig. 2 shows a graphic representation of an I-EDoS attack in which the cloud service platform exposes a vulnerability that triggers the creation of additional virtual instances with different degree of productivity. The group of unproductive instances was fraudulent instantiated by the attacker, which causes extra costs derived by the time they remain in execution and their resource consumption, in this way jeopardizing the economic sustainability of the offered services.

### 4 DESIGN PRINCIPLES AND ARCHITECTURE

The performed research aimed on distinguishing legitimate situations from those related to EDoS attacks in self-organized scenarios. The following describes its design principles, architecture, and the EDoS threat discovery approach.

#### 4.1 Design Principles

Thorough this section the requirements, assumptions and limitations (scope) of the performed research are detailed, which are enumerated as follows:

- The architecture must be capable of detecting W-EDoS and I-EDoS attacks assuming the characteristics described in the previous section, in this way distinguishing them from legitimate activities (typified as normal traffic and flash crowds).
- The detection of conventional flooding-based DoS attacks is beyond the scope of the performed research.
- The self-organized networks pose complex monitoring scenarios in which a large number of sensors collects information about the state of the network in real time. This information should be aggregated into observations that can be treated by high-level analytical tools. Although in the experimentation the impact of the data granularity is briefly discussed, the introduction of methods for data granularity calibration is postponed for future investigation.
- The correlation and management of the discovered incidents are beyond the scope of this publication. However, it is assumed that the acquired knowledge must be notified to the security management layers.
4.2 Architecture

Fig. 3 illustrates the proposed architecture, which was designed in accordance with the most widely accepted framework for Network Function Virtualization (ETSI-NFV) and next generation networks (5G) [16]. Accordingly, the data decoupling and data plane management make possible the distinction of the different functional layers. The Virtualization Layer is executed on the Physical Layer commonly implemented with Commercial-Off-The-Shelf (COTS) hardware. At a higher level, the Cloud Layer manages the automatic instantiation of Virtual Network Functions (VNFs) through interaction with the Virtualization Layer, which is responsible for providing the requested resources. The deployed Cloud environment interconnects VNFs through the underlying virtual network composing one or more Network Services (NS) accessible to users. It is also assumed that the Cloud Layer has the ability to extract monitoring metrics, which are subsequently analyzed in the SON Autonomic layer in the following steps:

Data collection. In SON environments the sensors (S) play an important role by monitoring custom metrics at the application-level, such as response times, memory consumption per process, NFV instances productivity, etc. Likewise, cloud computing platforms dispose of monitoring tools (e.g. Ceilometer [27]) capable of offering a significant number of metrics related to the usage mode of the network and the performance of the instantiated resources; e.g. CPU or memory consumption, latency, etc. In this way, the architecture collects information from both sensors (ALM) and cloud platform (VIFM).

Data Aggregation. The high volume of data generated by the monitoring tasks requires to run periodic aggregation procedures while generate time series able to be handled by the analytic components, by this approach being empowered their projection to future observations. At application-level, this is achieved through the Feature Extraction (FE), which implements at least the methods involved in EDoS detection described in the forthcoming sections, for example, the measurement of the data disorder by entropy analysis. On the other hand, the metrics directly gathered from the cloud computing platform are extracted and added (VRA) through queries to the API of the monitoring tool. In both cases, the granularity of the time series is determined by the periodicity with which the aggregation operations are executed.

EDoS Detection. The discovery of EDoS situations is addressed by the analytics and decision-making stages. In this framework, the first of them allows the inference of predictive models (MD) applied to time series of aggregated metrics, which results are considered for building prediction intervals (AI) based on the estimated error per observation. Consequently, unexpected behaviors are deduced when the observations are outside the prediction interval. Besides that, groups of instances based on the similarity (SM) observed at their productivity indicators are clustered, thus giving rise to the identification of groups with low productivity potentially related with I-EDoS situations. At decision-making stage, the analyzed data is taken into account to create inference rules designed to detect anomalies (AD) that reflect the presence of an EDoS threat, hence assuming as factual knowledge the information directly gathered from the monitored environment or acquired by the previous analytical steps.

Notification. The inferred conclusions are notified as possible EDoS situations. They serve the purpose of avoiding the creation of instances whose fraudulent origin generates surcharges derived from their usage.

4.3 W-EDoS detection

The following details the W-EDoS detection metrics and the analytical processes this task involves:

W-EDoS metrics. According to the W-EDoS definition, this type of attacks maintains a condition of network similarity with the normal and legitimate usage model but displaying significant variations in terms of VNF workload. Because if this, the detection strategy considers the CPU consumption ($X_{cpu}$) and the response time at application level ($X_{app}$) as W-EDoS indicators. It is important to clarify that the first of them measures the CPU consumption at operating system level, while the second measures the total time required to process each request at server-side. With the motivation of discovering unexpected behaviors, the first performed step is to analyze the variations in $X_{app}$, which is achieved by studying their disorder degree in fixed time intervals. The reviewed literature suggests the correlation of these observations in terms of entropy [20, 29, 37], as commonly accepted for classical DDoS recognition. As indicated by Bhuyan et al. [8], the entropy defined by Rényi provides a general-purpose solution particularly effective at this type of problems. It is defined by $H_\alpha(X_{app})$ in the following equation, being $\alpha$ the entropy order, $\alpha \geq 0$ and $\alpha \neq 1$.

$$H_\alpha(X_{app}) = \frac{1}{1 - \alpha} \log \sum_{i=1}^{n} p_i^\alpha$$

where $X$ is the random variable with $n$ possible outcomes and corresponding $P_i$ with ($i = 1,2, ..., n$) probabilities. For experimental purposes, the normalized solution $H_\alpha(X_{app})/\log n$ is considered. Note that if $\alpha = 1$, the particular case is observed in which the Rényi entropy coincides with that of Shannon. The successive measurements of entropy give rise to the creation of the time series:

$$H_{\alpha}(X_{app})_{t=0}, H_{\alpha}(X_{app})_{t=1}, \cdots, H_{\alpha}(X_{app})_{t=n}$$

and the CPU consumption indicators expressed as the time series:

$$(x_{cpu})_{t=0}, (x_{cpu})_{t=1}, \cdots, (x_{cpu})_{t=n}$$

The rest of analytical steps to detect W-EDoS are the same for $X_{cpu}$ and $X_{app}$. Henceforth, $X$ is used to refer indistinctly to any of them.

Unexpected behaviors derived from W-EDoS. The proposed detection method lies on deciding whether the estimation $X_{t=m}$ at time horizon $m$ differs significantly from $X_{t=m}$. This requires predicting time series of variable $X$ in a predetermined horizon, which allows comparing the forecasted values with the actual observations. The Double Exponential Smoothing (DES) predictive algorithm has been implemented, because it reduces the adaptation time by requiring shorter time series for data modeling, in this way outperforming autoregressive solutions as ARIMA [34]. Its adjustment parameters are auto-calibrated as described in [24] but instead of inferring variations with respect to the estimated points, prediction intervals are constructed as suggested in [19]. They are expressed considering the prediction error $e_t$ based on the Mahalanobis distance at $t$, particularly when $t = m$, according to the following equation:
The Prediction Interval (PI) is expressed as follows:

\[ PI = x_{t=n} \pm \eta \sqrt{\sigma^2(\varepsilon_t)} \]  

where \( \sigma^2 \) is the variance of the prediction error \( \varepsilon_t \). Consequently, let \( X'_{t=0} \) and its prediction \( \hat{X}_{t+m} \) at horizon \( m \), the observation \( X_{t=m+n} \) is considered a workload-based unexpected behavior if \( \varepsilon_t \not\in PI \), i.e. when \( \hat{X}_{t=m+n} \) and \( X_{t=m+n} \) differ significantly. Since \( X_{cpu} \) is a variable independent from \( X_{app} \), the proposal assumes that each \( X_{t=m} \) unexpected observation at both \( X_{cpu} \) and \( X_{app} \) unmask a potential W-EDoS threat if \( X_{cpu} \) displays increasing trend, in this case reporting a W-EDoS incident.

### 4.4 I-EDoS detection

The I-EDoS detection metrics and the adopted analytical procedure are described below:

**I-EDoS metrics.** The I-EDoS threat preserves a condition of network similarity with the normal and legitimate usage model. However, and as previously indicated, these attacks are characterized by the appearance of new instances, which causes a direct relationship between the new NFVs deployment and their low productivity. Consequently, two metrics are mainly taken into account for I-EDoS detection: the number of VNFs instantiated per observation \( (Y_t) \), and their productivity \( (Z_t) \), where \( Z \) is the set \( Z = \{z_1, \ldots, z_Y, Y \geq 0\} \) that defines the productivity of the different virtual instances of the observation at \( t \). In analogy to the proposed solution for W-EDoS detection, they are monitored over time, hence leading to the generation of the following time series:

\[
Y_{t=0}, Y_{t=1}, \ldots, Y_{t=n}, (Y^n_{t=0})
\]

\[
Z_{t=0}, Z_{t=1}, \ldots, Z_{t=n}, (Z^n_{t=0})
\]

where an observation at \( t, 0 \leq t \leq n \) is suspicious when \( Y_t \) displays a significant increase and \( Z_t = \{z_1, \ldots, z_Y\} \) contains a group of VNFs instances with clear low productivity, which is referred as lazy group. They are suspicious of deriving in an additional resource consumption and empowering the anomalous raising of \( Y_t \).

**Unexpected behaviors derived from I-EDoS.** As in W-EDoS attack detection, at I-EDoS situations there is a significant increase in the number of instances \( Y \) when for a time horizon \( m \) the calculated error between its forecasted value \( \hat{Y}_{t=n+m} \) and its observation \( Y_{t=n+m} \) falls outside the previously defined prediction interval \( (PI) \). When an auto-scaling action has triggered the creation of new VNFs instances with productivity \( Z_t = \{z_1, \ldots, z_Y\} \) it is possible to assess if part of them are involved in an I-EDoS attack by applying a density-based clustering; in the solution implemented at the performed experimentation, this method is particularized through a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [15]. This approach considers the existence of groups of observations based on the density of its closest K-neighbors. The observations that are not reachable within the same group are considered outliers [12]. DBSCAN has been chosen because it is tolerant to noise and does not require previous estimation of the number of groups, being configured at the experimentation by an heuristic approach recommended in [33]. DBSCAN is executed per set of productivity values \( Z_t = \{z_1, \ldots, z_Y\} \), and the result is a set of \( K \) clusters represented by \( C_t = \{c_1, \ldots, c_k\} \). Let \( Z_t = \{z_1, \ldots, z_Y\} \) the set of productivity measures at the instances in \( t \) classified as \( C_t = \{c_1, \ldots, c_k\} \) with \( K \geq 0 \) and ordered as \( s(C_t) = \{c_1, \ldots, c_k\} \), there is an I-EDoS based unexpected behavior (labeled as possible I-EDoS at \( t \)) when a significant growth at the time of creation of the VNFs instances belonging to \( c_1 \) is observed, where \( c_1 \) is the least productivity (lazy) group of VNFs.

![Figure 3: SON Architecture for EDoS detection](image-url)
5 EXPERIMENTATION

This section presents the network environment where the EDoS detection approach has been evaluated. The Cloud Layer and related SON components are described below.

5.1 Testbed

Fig 3. illustrates the experimental testbed where the Cloud Layer has been implemented with Openstack [28]. It has been deployed in two servers: Controller and Compute. The Controller server hosts the network service (Neutron), and the Compute node provides orchestration (Heat), clustering (Senlin) and telemetry (Ceilometer) services; on which the auto-scaling policies are supported. All Openstack services are communicated via RabbitMQ message exchange buses. On the other hand, the processing stages of the SON autonomic layer combine custom implementations and open source tools. Thus, the Collection node periodically fetches the response times calculated per instance; whereas the metrics related with the instantiated VNFs are gathered by Ceilometer. Then, data aggregation functions are applied, firstly aiming on calculating the entropy from data of the central node; and secondly, by querying the Ceilometer API for obtaining the average CPU consumption of the instantiated VNFs per observation. The time series feed the algorithms implemented for the detection stage. The acquired factual knowledge is analyzed by production rules configured in Drools with the aim of inferring unexpected behaviors labeled as potential EDoS situations [38].

5.2 W-EDoS characterization

An HTTP REST web service that supports GET requests to seven URIs (numbered 1 to 7) has been implemented in a virtual Openstack instance, each URI with a different response time, from the simplest (18.56ms) to the most complex queries (36.73ms). An eighth URI with 226.04 ms of average response time is also implemented, which represents the point of greatest computational cost that can be exploited as vulnerability. The metrics required for EDoS detection are collected per second, which serve for building time series and calculating the Rényi entropy degree of the monitored observations. On the other hand, the CPU based indicators are obtained per instance from the Ceilometer API, thus creating additional time series. In the experimental test, the requests have been launched from 500 clients implemented as Python threads, that in normal traffic situations randomly communicate with URIs 1 to 8, while in attack scenarios only URI 8 is requested. In both situations, a self-scaling policy that creates a new instance of the web service has been configured, which occurs when the average CPU consumption reported is greater than 60% in a one-minute time interval. Two adjustment factors allowed to configure the attack intensity: the number of compromised nodes, and the variation of the connection rate per second. From them, the rules for discovering unexpected behaviors derived from W-EDoS were configured.

5.3 I-EDoS characterization

At the I-EDoS scenario, the implemented REST application has been modified to expose a single URI that performed request with an average execution time of 27.89ms. For hosting the virtual image instances, an Openstack cluster was created with minimum length of 2 VNFs and maximum length of 12. The implemented auto-scaling policy orchestrated the creation of a new NFV instance when the average CPU consumption was higher than 80%; and the removal of an instance of the lower productivity cluster when this value was less than 40%. A stress-test was launched on the server for establishing the default productivity level. This has been evaluated with Httperf [18], and the obtained results reflected the lowest achieved productivity when the connection rate per second was less than 10, in this way causing a maximum CPU consumption of 39.1% that approached the lowest threshold of the configured auto-scaling policy. The optimal performance levels were recorded with a connection rate that varied from 10 to 40 per second, resulting in an average CPU consumption from 41.2% to 81.6%. In the aforementioned use case the percentage of connection errors was 0%. However, when the traffic injected above 40 connections per second, the CPU consumption reached its highest levels, thus registering values between 82.7 and 99.6% that exceeded the auto-scaling threshold and that posed connection errors higher than 10%.

The network parameters and the resulting productivity served for DBSCAN to identify the groups of VNFs that due to their behavior may be compromised by an I-EDoS situation. Their workload resembled a random Poisson distribution [5] where the expected value λ was the number of connections of the cluster at certain observation, for which has been tested by rates from 53 to 286 connections per second in a time period of three hours. The same default workload has been applied at both normal and attack scenarios. In the malicious situation, the VNF self-scaling was triggered through manipulating metrics gathered by Ceilometer, where it is assumed the ability of the attacker for exploiting vulnerabilities like CVE- 2016-9877 [13] to poison the information collected via RabbitMQ data buses. They enabled turning the original CPU readings (JSON messages) into fake values randomly ranging from 90% to 100%. The manipulated metrics were finally registered at the Ceilometer database, which led to fraudulently deploy additional VNFs instances due to auto-scaling policies.

6 RESULTS

The following discusses the effectiveness of the proposal when assessed at the evaluation testbed. This section separates the results obtained when dealing with W-EDoS and I-EDoS situations.

6.1 Effectiveness at W-EDoS attacks

In Fig. 4 the effectiveness of the proposal when varying the Rényi entropy degree is illustrated. The lower λ values minimize the impact of the inferred noise, this being the main reason that led them to yield more accurate results. Consequently, during the rest of the experimentation the best observed adjustment achieved (i.e. λ = 1) was assumed. The W-EDoS attacks have been injected in intervals of 1%, 5% and 10%, where the percentage represents the proportion of malicious requests that characterize the attack intensity. Additionally, four scenarios have been studied based on the average of requests per second (px) performed by clients: 50; 60; 70; 80, where K is the adjustment value for the creation of the prediction intervals. It has been experimented with different values of K (from 0.1 to 6), this being the parameter that varies the degree of sensitivity of the detection. The best results were obtained when
6.2 Effectiveness at I-EDoS attacks

The I-EDoS situation recognition capabilities of the proposal have also been evaluated according to the attack intensity, which impact translates into a growth of 10%, 20%, 30%, 40% and 50% of the number of instantiated VNFs. As was easy to deduce, this adjustment parameter directly influenced the effectiveness of the proposal. This fact is illustrated in Fig. 6, where the ROC curve obtained at the different experimental conditions is displayed. In general terms, the hit rate experienced small and inconspicuous variations. At the first group of attacks (10%, 20%, 30%, 40%), a distance of 0.022 (0.025%) was observed between the minimum hit rate (TPR = 0.89 when 10% intensity) and the best hit rate (TPR = 0.91 when 40% intensity); note that as in the previous tests, the best adjustments were estimated according to the Yauden criteria. Likewise, when the attack gained intensity (50%) the hit rate slightly increased (TPR = 0.94). However, by taking into account the percentage of false positives the observed variations were more significant; in particular, the detection method registered FPR = 0.12 when 10% intensity; but when gaining intensity, the best configuration (at 40% and 50% intensities) resulted in FPR = 0.07, which represents an improvement of 58.3% over the worst result. This pattern can be observed in Fig. 6 where the AUC varies according to the attack intensity, being AUC = 0.9811 in the best adjustment and AUC = 0.9483 in the worst scenario. The variations in effectiveness is caused at the clustering stage based on the VNFs productivity. Thus, the more visible the attack, the greater the number of instances that belong to the group of unproductive instances. In view of the obtained results, it can be concluded that the proposed strategy is able to successfully identify I-EDoS situations at scenarios similar to those considered for evaluation.

7 CONCLUSIONS

The problem of Economic Denial of Sustainability (EDoS) in the SON landscape has been studied and defined from two paradigms: workload (W-EDoS) and instantiation (I-EDoS) exploitation. In this context, two novel detection strategies have been proposed, which were able to recognize each of them. Both were based on modeling the normal behavior of the protected system and the discovery of discordant activities at the monitoring environment. In particular, for W-EDoS recognition the study of significant prediction errors was adopted, which lies in analyzing the evolution of the CPU consumption and the entropy estimated on the response times at the application level calculated in VNFs instances. On the other hand, for I-EDoS detection purposes, the relationships between the growing of the number of instantiated VNFs belonging to low productive clusters was studied. The effectiveness of the proposal was proven through the performed experimentation, in which the impact of varying different adjustment parameters was studied (intensity of the attacks, confidence of prediction intervals or entropy degree). Consequently, it was possible to demonstrate that the proposal meets its main objective on the deployed testbed. However, it should be noted that aiming on enhancing the understanding of our contribution, some aspects also necessary for its application to real
ACKNOWLEDGMENTS

This work was partially funded by the JSAN Travel Award 2018 bestowed by the MDPI Journal of Sensors and Actuator Networks (JSAN). In addition, the authors sincerely appreciate the support of the European Commission Horizon 2020 Programme under the Grant Agreements number H2020-ICT-2014-2/671672 (SELFNET: Framework for Self-Organized Network Management in Virtualized and Software Defined Networks) and H2020-FCT-04-2015/700326 (RAMSES: Internet Forensic platform for tracking the money flow of financially-motivated malware).

REFERENCES

[1] 3GPP TS 32.500 2008. Self-Organising Networks (SON): Concepts and requirements. (2008). http://www.3gpp.org/ftp/Specs/archive/32_series/32_500
[2] 5G PPP Security Working Group 2017. 5G PPP Phase I Security Landscape. (2017). https://5g-ppp.eu/white-papers/
[3] N.M. Alenezi and M.J. Reed. 2014. Uniform DoS traceback. Computers & Security 45, 1 (2014), 17–26.
[4] A. A. Atayero, O. I. Adu, and A. A. Alatise. 2014. Self organizing networks for 3GPP LTE. In Proceedings of the International Conference on Computational Science and Its Applications. Springer, Cham, 242–254.
[5] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski. 2003. Modeling Internet backbone traffic at the flow level. IEEE Transactions on Signal and Processing 51 (2003), 2111–2124.
[6] P.S. Barwa and S. Manickam. 2015. Critical Review of Economic Denial of Sustainability (EDoS) Mitigation Techniques. Journal of Computer Science 11 (2015), 855–862. Issue 7.
[7] A.S. Bhingarkar and B.D. Shah. 2015. A survey: Securing cloud infrastructure against edos attack. In Proceedings of the 15th International Conference on Grid Computing and Applications (GCA). Athens, Greece, 16–22.
[8] M. Bhuyan, D.Bhattacharyya, and J. Kalita. 2015. An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recognition Letters 51 (2015), 1–7. Issue 1.
[9] A. Bremler-Barr, E.Bros, and M. Sides. 2017. DDoS attack on cloud auto-scaling mechanisms. In Proceedings of 2017 IEEE Conference on Computer Communications (INFOCOM 2017). Atlanta, GA, US, 1–9.
[10] C. Hoff 2008. Cloud Computing: Security: From DDoS (Distributed Denial Of Service) to EDoS (Economic Denial of Sustainability). (2008). http://rationalsecurity.typepad.com/blog/2008/11/cloud-computing-security-from-ddos-distributed-denial-of-service-to-edos-economic-denial-of-sustainability.html
[11] C. Hoff. 2009. A Couple of Follow-Ups On The EDoS (Economic Denial Of Sustainability) Concept. . (2009). http://rationalsecurity.typepad.com/blog/edos/
[12] V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly Detection: A Survey. Comput. Surveys 41, 15 (2009). Issue 3.
[13] Common Vulnerabilities and Exposures 2016. CVE-2016-9877. (2016). https://www.cvedetails.com/cve/CVE-2016-9877/
[14] D. Gitzler, M. Roveri, C. Alippi, and R. Polikar. 2015. Learning in Nonstationary and Software Defined Networks) and H2020-FCT-04-2015/700326 (RAMSES: Internet Forensic platform for tracking the money flow of financially-motivated malware).
[15] J. Idziorek, M. Tannian, and D. Jacobson. 2012. Attribution of fraudulent resource consumption in the cloud. In Proceedings of the 5th IEEE International Conference on Cloud Computing. Honolulu, HI, USA, 99–106.
[16] J. Idziorek, M. Tannian, and D. Jacobson. 2012. Attribution of fraudulent resource consumption in the cloud. In Proceedings of the 5th IEEE International Conference on Cloud Computing. Honolulu, HI, USA, 99–106.
[17] G. Yao, J. Bi, and A. V. Vasilakos. 2015. Passive IP Traceback: Disclosing the Internet backbone traffic at the flow level. IEEE Transactions on Signal and Processing 51 (2003), 2111–2124.
[18] K. Wang, M. Du, S. Maharjan, and Y. Sun. 2017. Strategic Honeypot Game Model for Distributed Denial of Service Attacks in the Smart Grid. IEEE Transactions on Smart Grid 8 (2017), 2474–2482. Issue 5.
[19] G. Yao, J. Bi, and A. V. Vasilakos. 2015. Passive IP Traceback: Disclosing the Internet backbone traffic at the flow level. IEEE Transactions on Signal and Processing 51 (2003), 2111–2124.
[20] S. Yu, W. Zhou, J. Jia, S. Gao, Y. Xiang, and F. Tang. 2012. Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient. IEEE Transactions on Parallel and Distributed Systems 23 (2012), 1073–1080. Issue 6.
[21] Q. Zhang, L. Cheng, and R. Boutaba. 2010. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications 1 (2010), 7–18. Issue 1.
[22] W. Zhou, W. Jia, S. Wen, Y. Xiang, and W. Zhou. 2014. Detection and defense of application-layer DDoS attacks in backbone web traffic. Future Generation Computer Systems 38 (2014), 36–46.