Combined NFV and SDN Applications for Mitigation of Cyber-Attacks Conducted by Botnets in 5G Mobile Networks

Giacomo Bernini*, Pietro G. Giardina*, Gino Carrozzo*, Alberto Huertas Celdrán†, Manuel Gil Pérez†, Jose M. Alcaraz Calero, Qi Wang‡, Konstantinos Koutsopoulos§, and Pedro Neves¶

* Nextworks, 56122 Pisa, Italy
Email: g.bernini@nextworks.it; p.giardina@nextworks.it; g.carrozzo@nextworks.it
† Dept. Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071 Murcia, Spain
Email: alberto.huertas@um.es; mgilperez@um.es
‡ University of the West of Scotland, Paisley PA1 2BE, UK
Email: jose.alcaraz-calero@uws.ac.uk; qi.wang@uws.ac.uk
§ Creative Systems Engineering, 28is Oktovriou (Patision) 119, 11251 Athens, Greece
Email: k.koutsopoulos@creativese.eu
¶ Altice Labs, 3810-106 Aveiro, Portugal
Email: pedro-m-neves@alticelabs.com

***Abstract*—5G networks are envisioned to support substantially more users than the current 4G does as a direct consequence of the anticipated large diffusion of Machine-2-Machine (M2M) and Internet of Things (IoT) interconnected devices, often with significantly higher committed data rates than general bandwidth currently available into Long Term Evolution (LTE) and broadband networks. The expected large number of 5G subscribers will offer new opportunities to compromise devices and user services, which will allow attackers to trigger much larger and effective cyber-attacks. Significant advances in network management automation are therefore needed to manage 5G networks and services in an efficient, scalable, and effective way while protecting users and infrastructures from a wide plethora of advanced security threats. This paper presents a novel self-organized network management approach for 5G mobile networks where autonomic capabilities are tightly combined with Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) technologies so as to provide an effective detection and mitigation of cyber-attacks.**

Keywords–SDN; NFV; 5G; cyber-protection.

I. INTRODUCTION

5G aims to provide a scalable network infrastructure to meet the exponentially-increasing demands on mobile broadband access, both in terms of the number of connected users and required bandwidth. In this context, cyber-security widely recognized as a well-known challenge is already targeting all the layers of any ICT system, and it is becoming even more crucial to protect infrastructures due to the potential size and effects of cyber-attacks. Increased availability, service continuity, resilience, and delivery assurance for a broad spectrum of 5G services and applications are some of the security keywords that when combined with the anticipated levels of 5G mobility are required for the evolution of current security infrastructures towards more flexible, dynamic, and adaptive solutions.

Deep and extensive cloudification of services, with integration of edge and centralized clouds is another 5G key aspect that is accelerating the adoption of Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) technologies as key enablers of truly dynamic and automated service management in 5G networks, including multiple recurrent provisioning, maintenance, and service resilience [1].

The combination of NFV and SDN capabilities is the base for a required paradigm shift in the way 5G networks will be planned, deployed, and operated, i.e., to truly achieve autonomic management of 5G networks. The main target is to achieve a highly intelligent management platform for smart self-management of complex networking scenarios where proactive and reactive actions are automated in order to resolve and mitigate a wide plethora of networking problems (from performance issues to network failures and cyber-attacks), thus minimizing the intensive manual maintenance and troubleshooting tasks for network operators, leading to significant decrease in operational costs.

Such a novel paradigm for fully-automated and highly intelligent self-organized network (SON) management must provide four key functionalities in a continuous loop for automated detection and reaction to cyber-attacks, following the well-known MAPE approach: Monitor, Analyze, Plan, and Execute, as depicted in Figure 1. In *Monitor* phase, dedicated SDN-enabled sensors are deployed in the network infrastructure to facilitate system-wide distributed monitoring. These sensors are basically not only traditional monitoring sensors deployed in physical infrastructures, but also NFV and SDN applications spread across edge and core ETSI-compliant NFV Infrastructures (NFVI) Points of Presence (PoPs) that enable end-to-end user, network, and service awareness by means of specialized security-related metrics. At *Analyze* phase, the heterogeneous and specialized metrics collected during monitor feed data analysis processes encompassing scalable data analytics and machine learning techniques to produce key indicators and symptoms in the form of high-level metrics for particular 5G service affecting conditions, such as security threats, intrusions, denials of service, etc. At *Plan* phase indicators and symptoms generated by the data analysis processes are then combined and further analyzed to produce high-level tactical actions with the aim of reacting (possibly in a proactive mode) to the diagnosed conditions. The goal of the Plan task is to take decisions upon heterogeneous network security issues, which translate into actions to be enforced for network and service re-configuration. Finally, at *Execute* phase, the action plan is enforced over the heterogeneous physical and virtualized 5G network infrastructure. Either deployment of new SDN-enabled virtualized actuators, or re-configuration of

 ISBN: 978-1-61208-546-3

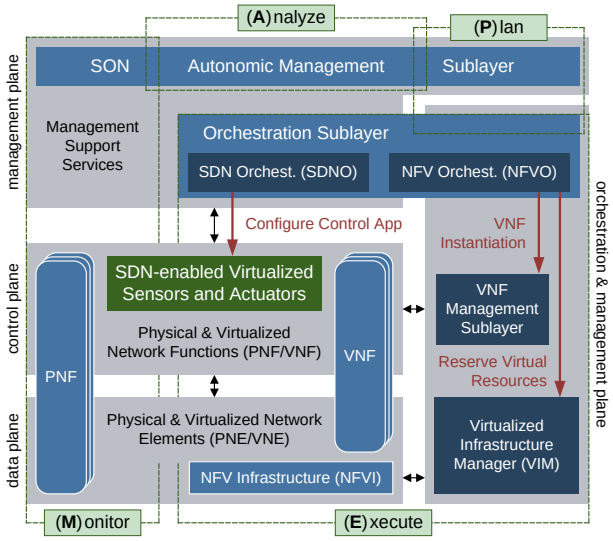

Figure 1. Autonomic network management architecture with NFV & SDN.

existing NFV and SDN applications may be included in action plans. End-to-end service orchestration is needed to coordinate lifecycle management of end-to-end 5G services composed by NFV and SDN applications.

The above concepts are the building blocks in the reference architecture depicted in Figure 1 where the self-organized management platform suitable for 5G networks and services is addressed under the SELFNET research project, funded by the EC under the Phase 1 of the 5G Public Private Partnership (5G-PPP) within the H2020 Framework Programme. SELFNET specifically addresses these network management challenges of 5G networks by closing the control loop while leveraging and enhancing standard NFV management and orchestration approaches currently targeted by ETSI NFV [2].

This paper presents a self-organized management approach, compliant with the MAPE approach, which aims at detecting and mitigating cyber-attacks in 5G mobile networks. Dedicated NFV and SDN applications for the purpose of cyber-attacks conducted by botnets are described (Section II), as well as orchestration (Section III) and lifecycle (Section IV) management principles and workflows for their effective operation in 5G scenarios. Conclusions are finally drawn in Section V.

II. Detection and Mitigation of Botnets in 5G Mobile Networks

5G networks, like any other radio communication systems, are prone to being compromised by attackers who use elements connected to the 5G antennas (the users equipments –UEs), to serve as a stepping stone for launching cyber-attacks, such as Distributed Denial of Service (DDoS) attacks, among others. In 5G, the detection and mitigation of botnets will present an even greater challenge due to the massive number of connected devices as well as a higher data rate. To this end, we propose decoupling traditional botnet detection procedures into two phases at two complementary levels of abstraction; namely:

1) Detection at *high-level* to proactively identify suspect Command & Control (C&C) channels, by monitoring and analyzing network traffic flows exclusively.
2) Fine-granularity detection at *low-level* through Deep Packet Inspection (DPI) to confirm the real existence of the C&C channels detected in the previous phase.

Monitoring network flows during the first high-level detection phase allows us to analyze big volumes of data quickly and in near real-time. A deep analysis of network packets is not feasible in a first step due to the massive amount of traffic in the network. For this reason, the deep analysis is conducted in a second step between the peers identified as suspects during the first step, but carrying it out for a much smaller number of peers. The *sensor* in charge of gathering the traffic network flows to be subsequently analyzed, from a high-level detection perspective, is called *Flow-Based Monitoring* (FBM), while the deep analysis is performed by a DPI tool, such as Snort [3]. The latter also acts as a sensor, although at lower granularity than the other. Both phases make reference to the two detection control loops defined in the Self-Protection use case within the 5G-PPP SELFNET project [4], which is augmented in this paper by the full integration of end-to-end orchestration and application management components so as to proactively detect potential botnets in 5G mobile networks.

Once the second detection phase confirms the actual existence of the botnet, our approach is based on the deployment of a reaction based on the so-called deception approach to counter the botnet and, consequently, the potential cyber-attacks that it could produce. This reaction consists of deploying and enforcing a virtualized and personalized honeynet as an *actuator* (HNet) by using honeytoken techniques [5] to isolate the UEs shaping the botnet. It aims at cloning the botnet zombies –UEs known as *bots*– to emulate their behaviors. As a result, the real attacker (i.e., the botnet's owner) will not be aware that part of the attack actions have been disabled by the HNet. Sensors and actuators, as detailed earlier and summarized in Table I, are dynamically deployed and operated as NFV applications (see Section IV).

For their operation, i) the DPI sensor needs access to the raw network packets exchanged between the suspicious peers to be inspected and ii) the HNet actuator requires the network flows of the C&C channels detected by the DPI to be redirected to the emulated bots while blocking the real ones. To this end, an actuator listed in Table I as FlowT has been implemented as an SDN application to reconfigure the flow tables of the virtual switches providing the following two features:

- *Network Flow Mirroring* to send copy of the network packets of given peers to the DPI for their inspection, thereby starting the second detection control loop.
- *Network Flow Diversion* to redirect the network flows of given peers to the HNet, where they are isolated to avoid cyber-attacks (*mitigation phase*) and learn new knowledge when changing botnet behavioral patterns.

The whole set of detection and mitigation actions for these two control loops mostly matches the bottom layers in Figure 1. The *SON Autonomic Management Layer* sitting on top provides those autonomic features needed to have an intelligence-driven 5G management platform. It is in charge of analyzing the metrics and events gathered from the sensors and deciding *when*, *where*, and *how* to deploy and configure sensors and actuators in a coordinated way.

Figure 2 shows the overall workflow aimed at detecting and mitigating botnets in 5G networks, expanding the previous SON Autonomic Management Layer in a pool of modules addressing the MAPE capabilities, as detailed in Section III.

 ISBN: 978-1-61208-546-3

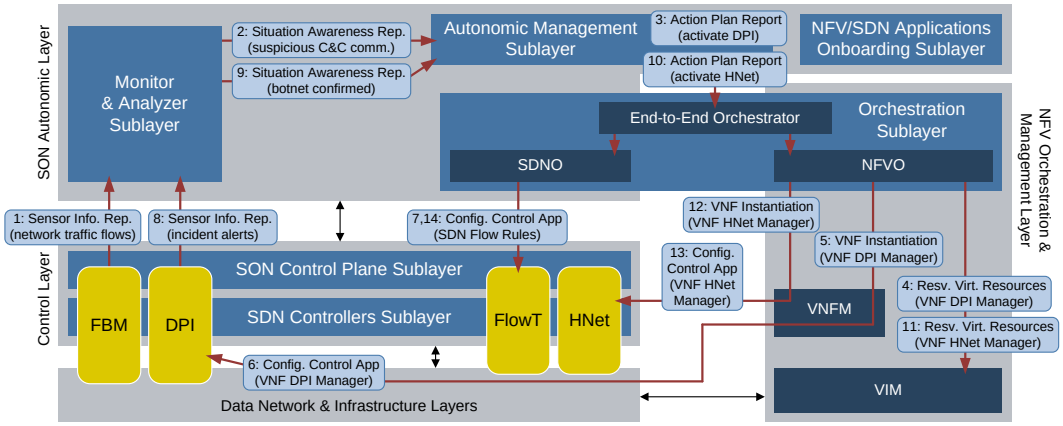

Figure 2. Overall workflow for detecting and mitigating botnets in 5G mobile networks.

TABLE I. SET OF NFV- AND SDN-APPS DEFINED FOR THE SELF-PROTECTION USE CASE.

Name	Type	Short description	Input data	Output data
FBM	NFV-App (sensor)	Flow-based monitoring for high-level detection of suspect C&C channels	Raw network packets	Metrics on the network status: Number of packets, size, frequency between peers, etc.
DPI	NFV-App (sensor)	Deep packet inspection for detecting distributed threats (e.g., bots) at low level	Raw network packets	Alerts or events that report incidents on a cyber-attack or the confirmation of existence of bots
HNet	NFV-App (actuator)	Build a virtualized and personalized honeynet by creating fake bots	List of new threat targets (incl. the attackers' IP addresses to be diverted) and configuration	Network traffic sent to the C&C server emulating the real bots' behavior
FlowT	SDN-App (actuator)	Reconfigure the flow tables of the vSwitches to enable DPI and HNet	New flow tables for the affected/under attack virtual network	New traffic steering policies to be routed to the OpenDaylight controller(s)

Specifically, pattern recognition techniques are being used in SELFNET for the first high-level detection phase, so as to identify similar patterns in behavior between peers, while DPI tools with 5G support, such as Snort, provide the expected deep analysis in raw network packets depending on the botnet type (identified as suspected by the first autonomic loop). These detection phases are encompassed by the *Monitor & Analyzer Sublayer*, which feeds the *Autonomic Management Sublayer* with potential *symptoms* about a security-related problem. This latter examines the detected symptom, searching the *causes* that produced it and deciding certain actions (*tactics*) through machine learning techniques to be enforced. Reinforcement learning techniques are also used to further refine the previous decision-making process, by making use of reward (and punishment) mechanisms in case the tactical actions help to detect other bots.

With a pool of sensors and actuators already deployed in the network, 5G mobile users can move from one location to another, implying that deployed detection or mitigation functions will need to be adapted accordingly. They should be moved following users' mobility, in order to continue monitoring and analyzing those UEs under inspection (detection phase) or even emulating their behaviors (mitigation phase) as was previously done. As an example, consider the 5G mobile scenario shown in Figure 3.

In this figure, one or more UEs served from RAN1 move and are served by RAN2. In the case that they were being analyzed (by a DPI) or emulated (by a HNet) in RAN1, the corresponding sensors and/or actuators capabilities will be dynamically migrated to RAN2 to keep their detection and mitigation processes up and running (either with new deployments or re-using existing applications in RAN2).

For scalability purposes, multiple distributed SDN controllers may be deployed to control specific segments of the network, all managed by the *Orchestration Sublayer* acting as a centralized coordination point.

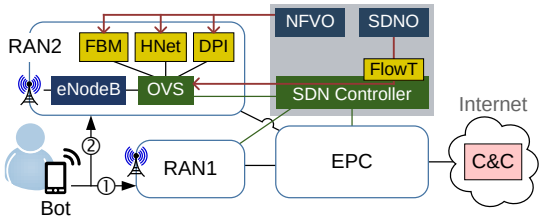

Figure 3. 5G mobile scenario where combining NFV- and SDN-Apps.

III. END-TO-END ORCHESTRATION OF COMBINED NFV AND SDN APPLICATIONS

The evolution towards a 5G enabled service architecture will lead to the creation of a new service environment, built over a mesh of micro data centers (also known as PoPs) coordinated by advanced service management platforms. These platforms, including advanced virtual infrastructure management platforms, will provide enhanced agility for new services creation and operation, being also a contributor for costs reduction due to use of common purpose hardware.

At present time, the creation of a new service requires the setup of an engineering project to coordinate and govern the configuration of several distinct network elements and the creation of specific service logic in proprietary delivery and control platforms. Additionally, the management of this distributed service intelligence over several network appliances (also known as Physical Network Functions –PNFs) requires the setup of complex management processes. All this together compromises the agility to launch new services. The migration

 ISBN: 978-1-61208-546-3

of service logic to virtualized (VNFs) and software-defined functions (SDN-Apps), allows the service provider to reduce significantly the operational impact of launching new services.

A complete autonomic management loop opens the possibility to explore a wide set of new use-cases for service providers, usually known as self-* (e.g., self-healing, self-protection, self-optimization). Nevertheless, the challenges to orchestrate NFV and SDN applications are much more complex when compared with physical functions. Virtual and software-defined functions can be dynamically on-boarded, provisioned, started, paused, and stopped, whereas the management procedures over physical functions are much more limited and static. The instantiation of a virtual and/or software-defined network service is expected to be a fully automated procedure, without human intervention, while the instantiation of a physical service sometimes requires the explicit intervention of human resources on the field.

In this context, Operational Support Systems (OSS) are naturally evolving to deal with such a wide and differentiated pool of resource types (VNFs, PNFs, and SDN-Apps) in order to provide end-to-end services composed by any combination of virtual, legacy, or SDN-based function. In particular, the need for a holistic orchestration component, which combines all the required logical resources management to deliver a service, must be provided. This component, known as End-to-End Service Orchestrator (E2E-SO) and its main interactions is briefly depicted in Figure 4.

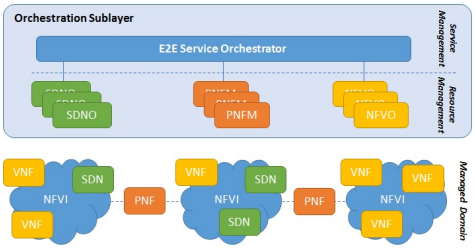

Figure 4. 5G End-to-End Service Orchestrator high-level concept.

According to the E2E-SO concept, software components (SDN-Apps and VNFs) are dynamically managed and configured for provisioning E2E services through the utilization of Resource and Application Management modules. The displayed PNFM concept regards the need for on-the-fly reconfiguration of PNFs and delivery of an E2E service that spans beyond the borders of a single PoP. This need is addressed in the context of the Application Management abstraction that is realized by the instantiation of a pool of adaptation objects per resource instance (PNF, SDN-App, VNF), which enable a scalable orchestration approach spanning multiple domains and PoPs. This pool of objects provides a common interface to be utilized by the E2E-SO during instantiation, configuration, and reconfiguration of the three resources types. In the case of PNFs, the adaptation objects are generated by manual registration of the PNFs available in each PoP, whereas for the SDN-Apps and VNFs the lifecycle management triggers the automated provisioning of the related configuration objects.

Applying the E2E orchestration concept to the concrete case of cyber-attacks, the E2E-SO will have a key role in both autonomic loops of the scenario, being the entity responsible for orchestrating all the required elements to deliver the action required by the autonomic manager, as depicted in Figure 2.

Before the first control loop, the E2E-SO is key to deploy and configure the FBM VNF sensor, which will be responsible to provide the flows monitoring and therefore trigger the first control loop. Thereafter, during the first autonomic loop, based on the monitored flows (step 1) through the FBM VNF, a suspicious botnet symptom is detected (step 2). As a result of this symptom, in order to ensure that a cyber-attack is happening, the autonomic manager requests the E2E-SO to activate a DPI to analyze the suspicious botnet network traffic (step 3). The E2E-SO instantiates (if not yet instantiated for other tenants) and configures a DPI VNF sensor (in this case Snort) according to the information (e.g., location, botnet signature, etc.) provided by the autonomic manager (steps 4, 5, and 6). The E2E-SO interacts with the NFVO to deploy and configure the DPI VNF. Additionally, still as a result of the first autonomic loop, the E2E-SO is also responsible for requesting the SDNO to activate and configure the FlowT SDN-App to apply, through the SDN Controller, the mirroring of the potential zombie network traffic towards the deployed DPI VNF (step 7).

At this stage the second autonomic loop is started and, as a result, when a potential zombie is confirmed as an attack, an event is triggered by Snort (step 8). This event is processed, filtered, enriched, and provided to the autonomic manager (step 9) confirming that a cyber-attack is taking place. Consequently, the autonomic manager requests the E2E-SO the isolation of the attack (step 10). As a result, the E2E-SO requests the NFVO the deployment and configuration of a HNet VNF in order to create emulated zombies (steps 11-13). Finally, the E2E-SO has to coordinate the diversion of the attacker network traffic towards the HNet by requesting the SDNO to configure (step 14) the FlowT SDN-App for this action. In the end, zombies being attacked are isolated.

IV. Automated Management of Sensors and Actuators Lifecycle

5G networks pose challenging requirements in terms of automation and performance for deployment of new services. The aspect of network management is being critical for the efficient provisioning and maintenance of new services in 5G networks. In accordance to the 5G KPIs [6], the reduction of services provisioning time from 90 days to 90 minutes is possible if the whole lifecycle of network functions and applications to be deployed and combined as services in the 5G network infrastructure is managed by means of coordinated and automated procedures.

Common and homogeneous mechanisms and procedures are needed to manage the whole lifecycle of individual NFV- and SDN-Apps (i.e., instantiation, configuration, start, stop, scale, termination, etc.) irrespectively of their specific logic or function. This means that sensors and actuators listed in Section II need to be properly encapsulated to be coordinated by the E2E-SO and achieve a high degree of automation in the detection and mitigation of cyber-attacks.

A. NFV and SDN Applications Onboarding

The very first aspect of lifecycle management of NFV- and SDN-Apps refers to their onboarding in the management platform. With reference to Figure 2, this is provided by the *NFV/SDN Application Onboarding Sublayer*. When an

 ISBN: 978-1-61208-546-3

application is onboarded it becomes available to be instantiated, configured, and composed with other applications and network functions in support of specific E2E 5G services. However, the onboarding sublayer is more than a simple catalogue listing applications and network functions. It is a full onboarding service, managing a set of operations including applications onboard, enable, disable, update, and offboard, with management of software images for VNFs and SDN bundles upload for SDN-Apps. Moreover, the onboarding service is responsible for notifying all these operations within the management platform to all those components involved in the applications lifecycle management and operation (mostly E2E-SO, SDNO, and NFVO).

Starting from the VNF Package definitions in the ETSI MANO specifications [2], we defined the concept of App Package to support the one-click automated onboarding of both NFV- and SDN-Apps with a common approach valid for all kinds of sensors and actuators. An App Package is a single entity (in the form of a software archive), which encapsulates a given sensor or actuator. It is the container of all the information needed to operate a given application, including a set of information organized in folders and JSON files. In particular, the full structure of an App Package is shown in Figure 5.

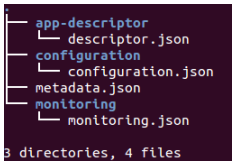

Figure 5. App Package structure.

In this context, the structure of an App Package includes a set of JSON files organized as follows:

- *metadata.json*: includes generic onboarding information for the application, like application family (sensor or actuator), class (VNF or SDN-App), and type, augmented with specific software image and bundles registration data. For VNFs, it also includes information of specific lifecycle scripts to be used to apply actions on the VNF.
- *app-descriptor/*: it is the folder containing the app descriptor file *app-d.json*, which provides information related to requirements for instantiation and management, in terms of resources to be allocated and configured for proper app operation. For VNFs, it follows the standard VNF Descriptor (VNFD) format defined by ETSI [2]
- *configuration/*: it is the folder containing the app configuration information file *configuration.json*, which provides information related to the configuration primitives actions exposed by the App. It models parameters and actions to be applied for proper management and control of the specific app operational behaviors
- *monitoring/*: it is the folder containing the app monitoring information file *monitoring.json*, which provides information related to the monitoring metrics exposed by the app when it is a sensor. It models both metric parameters and operations to collect them.

B. Lifecycle Management of NFV Applications: VNFM

The reference baseline for the NFV applications encapsulation is the ETSI MANO framework [2], with the VNFM as key component responsible for the lifecycle management of all the sensor and actuator VNFs listed in Section II. The VNFM aims to provide a unified and common approach for the lifecycle management of sensor and actuator VNFs, thus exposing primitives towards Orchestration Sublayer components (e.g., the NFVO) to instantiate and control the VNFs in the NFV Infrastructure. The VNFM functions as defined by ETSI can be considered as generic and common functions applicable to any type of VNF. The VNFM depicted in Figure 2 implements the following VNF lifecycle management operations specified by ETSI: 1) VNF instantiation, including VNF configuration according to the VNFD included in the correspondent App Package, which describes attributes and requirements to realize such VNF and provision it; 2) VNF instance modification, that basically consists into an update of the VNF configuration; 3) VNF instance scale out/in (i.e., allocate or terminate Virtual Machines in support of a given VNF) and up/down (i.e., increase or decrease virtual resources for a given VNF); and 4) VNF instance termination.

We implemented the VNFM as a stand-alone prototype [7] on top of the OpenBaton open source project, and enhanced it for this work to integrate the DPI and HNet VNFs. OpenBaton [8] is an ETSI MANO compliant tool, which can be easily integrated with existing cloud platforms like OpenStack [9] and adapted to different types of VNFs. For each VNF, a lifecycle management agent is embedded in the correspondent Virtual Machine to enable the communication with the VNFM and implement specific actions on the VNF according to the lifecycle scripts included in the App Package. The agent enables the containerization of VNFs into encapsulated VNFs and provide a common lifecycle management message bus interface based on RabbitMQ [10] towards the VNFM, i.e., in support of the four operations described above. In particular, with reference to the VNFs listed in Section II (i.e., the DPI and the HNet), and the workflow in Figure 2, this message bus interface is used in steps 6 and 13 for taking care of VNFs configurations during the two control loops.

For the DPI VNF, the configuration operation allows the Snort application to start inspecting those network flows identified by the Monitor and Analyzer components as suspected to belong to a potential botnet. The Snort VNF is indeed configured with the following parameters: i) IP addresses and ports identifying the suspected network flow involving the potential bot and C&C server and ii) the detection rule, specifying the pattern or payload content to be identified by the DPI engine. For the HNet VNF, the configuration operation enables the emulation of a bot identified by the second loop of Monitor and Analyzer detections. In practice, the HNet is configured by the VNFM with: i) type of botnet to be emulated; ii) frequency of the requests to be sent to C&C server; iii) identifier of the bot to be emulated; and iv) IP address of the C&C server.

C. Lifecycle Management of SDN Applications: SDNO

The concept of the SDN Application Management, as realized by SDNO foresees two types of applications: i) SDN-Apps, which regard software components that are deployed and executed in a runtime environment outside the SDN Controller and utilize its NorthBound Interface (NBI), this being, in practical terms, an SDN-App implementing a network application logic that is carried out through a number of transactions with the Northbound interfaces of the SDN Controller focusing on

 ISBN: 978-1-61208-546-3

specific high level tasks, such as extracting topology information or network metrics, applying any forwarding rules, etc.; and ii) SDN-Controller-Apps, which are software packages deployed directly in the SDN Controller runtime environment utilizing directly the services provided by other components of the controller or by southbound protocol plug-ins.

In essence, SDN-Apps intent to abstract the details of the Northbound Interface of the SDN Controller, as offered by the various features and bundles activated in the Controller, and while they are handling internally the complexity required to apply a particular forwarding rule or isolate and expose a view of the information that is available in the Controller, at the same time they are providing a uniform interface to be invoked in the context of an end to end service orchestration. This interface streamlines the way information has to be structured and contextualized so that the applications can be catalogued in terms of what is offering and what high level (abstracting SDN Controller NBI model) parameters are required in order to offer it. On the other hand, SDN-Controller-Apps are structured and developed according to the SDN Controller's principles and they are exporting their application model via the controller's NBI. Typically, the SDNO based approach focuses on the development of a number of SDN-Apps that enable a more effective and targeted use of what is offered by SDN-Controller-Apps.

An SDN-App may be included in various service compositions, which in turn may require that a separate instance of the app implementation is launched per service. Thus, contrary to the shared nature of SDN-Controller-Apps, SDN-Apps may be instantiated multiple times with each instance being associated with a particular tenant. Instantiation of an SDN-App can occur only after proper insertion of the app in the onboarding catalogue, which is reflected on the SDNO in terms of registration of the app implementation under the particular app type REST endpoint. The SDNO assigns to the newly registered app an implementation order identifier that is thereafter used by the SDNO to export the implementation for management purposes towards the E2E-SO. The supported management functions include app removal, which is triggered by the onboarding catalogue and instance management requested by the E2E-SO. For every app registered with the SDNO, the configuration extract from its descriptor (as presented in Section IV-A above) is collected from the notification generated by the onboarding catalogue through the message bus. This piece of information is processed by an adaptation object, which the SDNO is generating when an instance of the SDN-App is requested by the E2E-SO. The object analyses the configuration descriptor and provides a unique REST endpoint for every action the SDN-App is offering, to be invoked by the E2E-SP whenever the configuration of the SDN-App has to be updated. The configuration for all the apps is based on a key-value format. The adaptation object is also launching a Docker container [11] from a Docker image where the SDN-App binaries are installed. Those binaries are thereafter configured via execution commands through the Docker API according to the communication (remote execution, REST, environment variable settings) protocol indicated in the onboarding descriptor.

For the FlowT SDN-App, the SDNO prepares a Docker image with Python support and installs the FlowT implementation therein. For any instance of the app that is requested by the Orchestrator a separate Docker container based on the previous image is instantiated and an adaptor object is started. The adaptor object exports a number of action endpoints relating to the FlowT configuration actions (*"mirror", "divert", "mirror-del", "divert-del", "remove-all"*). The E2E-SO may post thereafter, to the proper REST endpoint identified by the FlowT instance ID, the required JSON formatted configuration sets as {*"key": "key name","value": "requested value for key"*} arrays, that communicate the specific parameters required for activating the related action. In the case of *"mirror"*, action the following parameters have to be defined: source and destination IP addresses of the flow, the IP address of Snort, and the identifier of the OVS instance to be configured. Similarly, for *"divert"* action the parameters are the same but instead of the Snort IP the HNet IP address has to be provided.

V. Conclusions

5G networks will need to be managed in a very flexible, dynamic, and scalable way, targeting a high degree of automation in the deployment of new services on top of virtualized and distributed infrastructures. This paper presented a self-organized network management approach where autonomic functions are combined with NFV- and SDN-Apps for detection and mitigation of cyber-attacks conducted by botnets. While NFV- and SDN-Apps, together with VNFM and SDNO lifecycle management functions, have been developed and integrated into virtualized testbed infrastructures, future work will involve implementation of E2E-SO and autonomic components.

Acknowledgment

This work was partially funded by the EC H2020 5G-PPP Programme under Grant Agreement number 671672 - SELFNET (*Framework for Self-Organized Network Management in Virtualized and Software Defined Networks*), and by a Séneca Foundation grant within the Human Resources Researching Training Program 2014 (FEDER/ERDF).

References

[1] C. Cleder Machado, L. Zambenedetti Granville, and A. Schaeffer-Filho, "ANSwer: Combining NFV and SDN features for network resilience strategies," Proceedings of the 2016 IEEE Symposium on Computers and Communication, pp. 391-396, June 2016.

[2] ETSI NFV ISG, "Network Functions Virtualisation (NFV); Management and Orchestration," ETSI GS NFV-MAN 001 V1.1.1, Dec. 2014.

[3] Sourcefire, Inc., "Snort: An open source network intrusion detection and prevention system," https://www.snort.org [retrieved: March, 2017]

[4] M. Gil Pérez and G. Bernini, "Self-protection against botnet attacks - Solutions by 5G PPP project SELFNET," Eurescom Message, pp. 13-14, Winter 2016.

[5] W. Fan, D. Fernández, and Z. Du, "Versatile virtual honeynet management framework," IET Information Security, vol. 11, no. 1, pp. 38-45, Jan. 2017.

[6] The 5G Infraestructure Public Private Partnership, "Key Performance Indicators (KPI)," https://5g-ppp.eu/kpis [retrieved: March, 2017]

[7] G. Bernini, E. Kraja, G. Carrozzo, G. Landi, and N. Ciulli, "SELFNET Virtual Network Functions Manager: A common approach for lifecycle management of NFV applications," Proceedings of 2016 5th IEEE International Conference on Cloud Networking, pp. 150-153, Oct. 2016.

[8] The OpenBaton project, "A ETSI NFV compliant MANO framework," http://openbaton.github.io [retrieved: March, 2017]

[9] The OpenStack project, http://openstack.org [retrieved: March, 2017]

[10] Pivotal Software, Inc., "RabbitMQ message queue service," https://www.rabbitmq.com [retrieved: March, 2017]

[11] Docker, "The world's leading software container platform," https://www.docker.com [retrieved: March, 2017]

 ISBN: 978-1-61208-546-3