Computer Networks 109 (2016) 39–49

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Patterns for the design of secure and dependable software defined networks

Nikolaos E. Petroulakis [a,b,*], George Spanoudakis [b], Ioannis G. Askoxylakis [a]

[a] Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion, Greece
[b] Department of Computer Science, City University London, London, UK

ARTICLE INFO

Article history:
Received 21 December 2015
Revised 5 June 2016
Accepted 20 June 2016
Available online 24 June 2016

Keywords:
Design patterns
Software Defined Networks (SDN)
Wireless networks
Security
Dependability
Drools

ABSTRACT

In an interconnected world, cyber and physical networks face a number of challenges that need to be resolved. These challenges are mainly due to the nature and complexity of interconnected systems and networks and their ability to support heterogeneous physical and cyber components simultaneously. The construction of complex networks preserving Security and Dependability (S&D) properties is necessary to avoid system vulnerabilities, which may occur in all the different layers of Software Defined Networking (SDN) architectures. In this paper, we present a model based approach to support the design of secure and dependable SDN. This approach is based on executable patterns for designing networks able to guarantee S&D properties and can be used in SDN networks. The design patterns express conditions that can guarantee specific S&D properties and can be used to design networks that have these properties and manage them during their deployment. To evaluate our pattern approach, we have implemented executable pattern instances, in a rule-based reasoning system, and used them to design and verify wireless SDN networks with respect to availability and confidentiality. To complete this work, we propose and evaluate an implementation framework in which S&D patterns can be applied for the design and verification of SDN networks.

1. Introduction

The design of complex system networks is of paramount importance due to their increasing role in the implementation of Cyber–Physical Systems (CPS) and Software Defined Networking (SDN) involving integrated ICT and physical components and devices. However, the design of such networks effectively encounters difficulties which need to be resolved. These difficulties stem from the highly distributed and heterogeneous nature of SDN and the extent of intelligence, dependability and security that they need to demonstrate during their operation. The design and verification methods for developing secure and dependable system networks is necessary and should be considered at the design level to guarantee security and mitigate safety threats, on remote monitored and managed networks. Especially, with the fast growth of SDN and the integration with 5G network architectures [1], the design of networks enters in a new era and makes necessary a careful investigation of the new security and dependability risks, which have not been relevant in legacy systems. One of the challenges of future networks is to develop SDN capabilities tailored to CPS and drive the reconfiguration of these capabilities through network configuration specifications embedded in critical infrastructures.

SDN allows network programmability and control to be decoupled from the forwarding plane and the forwarding plane to be directly programmable by the control plane. In this paper, we present a model driven approach to the design and verification of secure and dependable SDN networks that is based on S&D network design patterns (referred to as *S&D patterns* in the rest of this paper). These patterns can be used to design and/or verify SDN network infrastructures and identify suitable paths and nodes that can guarantee S&D properties. S&D patterns can be used to design SDN infrastructures, and determine also the type, location and connectivity of end nodes with forwarding devices. At the control layer, S&D patterns can ensure secure connectivity between the controllers and the programmable switches. In this paper, we give a detailed description of the scheme for specifying S&D patterns and their use for the design of S&D preserving SDN networks. The main contribution of the approach is that our approach encodes designs of network topologies, which are proven to satisfy S&D properties, as design patterns. In addition, S&D patterns can be used for the definition of optimal paths which are able to guarantee S&D properties in deployed networks. A first definition of our pattern-based approach for designing reliable cyber-physical

* Corresponding author.
E-mail address: npetro@ics.forth.gr (N.E. Petroulakis).

http://dx.doi.org/10.1016/j.comnet.2016.06.028
1389-1286/

systems was given in [2]. This paper extends the original approach by developing a pattern framework in which we can evaluate and emulate S&D executable patterns on SDN-based network designs. It also presents an application framework in which S&D patterns can insert and modify flow rules through the controller to the programmable switches of SDN infrastructures.

The remainder of this paper is organized as follows. In Section 2 an overview of related work is presented. In Section 3, we present the schema of the pattern execution form. In Section 4, we introduce abstract specification instances of patterns with respect to confidentiality and availability encoded also to a rule-based reasoning language. In Section 5, we propose an implementation framework in which S&D network patterns can be applied in order to design and verify SDN network architectures. In Section 6, we emulate our proposed network patterns for the design of wireless SDN-based network architectures able to provide security against physical layer attacks and failures at design or at runtime in hostile environments. Finally, Section 7 provides conclusions and future work.

2. Related work

The main focus of network design relies on specification analysis, design, verification, and validation of systems that include hardware/software, data, procedures, and facilities. Driven from software development methodology, Model–Driven Engineering (MDE) [3] can be used to analyze certain aspects of models, synthesize various types of artifacts and design secure and dependable systems. An MDE framework for architecting wireless networks is presented in [4]. The design of a system is simplified through the modelling of design patterns. MDE applies design patterns [5,6] as solutions for reusable designs and interactions of objects by the use of formal proven properties[7]. The development of S&D patterns may benefit from the current implementations of software patterns as described in the literature in a variety of works [8–12]. The concept of component-based architecture composition is mainly applied on software components and service oriented architecture but it can be used successfully for designing networks [13,14]. Security workflow patterns, for service compositions based on enabling reasoning engines such as Drools, are also described in [15,16]. Drools enabling reasoning appeared to be also an efficient rule engine to represent our network workflow patterns. Workflow pattern for QoS aggregation for web service composition have been proposed in [17]. In our approach, executable workflow patterns are used for backward chaining for network compositions.

Especially with the softwarization of networks in SDN, design patterns can be applied in all the different layers of SDN architectures. One of major objectives of SDN is to provide Quality of Services (QoS) and on-demand services [18]. Authors in [19] present an end-to-end orchestration of IoT services using SDN-enabled edge nodes. The construction of network topologies includes also the definition of network and traffic patterns. Traffic engineering and patterns in SDN are presented in [20]. Flow policy patterns as expressed by Frenetic languages, can generate flow rules able to be installed in programmable switches of SDN networks [21]. In our approach, we can provide paths as flow rules based on the security requirements. Design patterns can also be used in northbound interfaces using RESTful API as proposed in [22]. Our proposed pattern framework is able to interact with the controller also using RESTful. Furthermore, Service Function Chaining (SFC) [23] aims to provide end-to-end security in SDN following security function compositions. Our approach is able to provide a step forward by creating dynamic security chains following a backward chaining. Finally, the concept of intent-based engineering in SDN appears to enforce security policies [24] as proposed by our S&D patterns.

3. S&D pattern schema

The design and implementation of SDN infrastructures can be based on an architectural framework where the network elements are integrated through patterns with proven capability to enable the semantic interoperability, and to preserve end-to-end and link-to-link security, privacy, and dependability. S&D patterns can be used as an instrument for designing, verifying and altering the topology of SDN networks, at design time or runtime. At design time, the procedure includes the definition of a design problem and the required S&D property that needs to be guaranteed by the SDN to be designed. In verification, an existing SDN network design (topology) and the required S&D properties are provided, and patterns are applied to analyse the former and ensure that the latter are satisfied. The analysis is based on checking if the topology of the pattern matches the network design or some part of it and that the individual components that constitute the network with the particular topology have certain properties that can guarantee end-to-end network level S&D properties. Finally, at runtime patterns are applied to alter the topology and forwarding rules of an operational network in order to ensure the satisfaction of S&D properties. The pattern specification schema is defined as follows:

Definition 1. An S&D pattern schema is an abstract structure for specifying S&D pattern which includes: (a) an abstract network topology, defining the control structure and data flows of the components of an SDN, (b) constraints that should be satisfied by the components of the network that are composed according to the structure of (a), (c) the S&D property that the network topology in (a) guarantees, and (d) an execution pattern rule.

The constituents (a)–(d) of the S&D pattern schema are discussed in more detail below.

3.1. Pattern topology

S&D patterns define generic ways of composing (i.e., establishing the connectivity between) and configuring the different and heterogeneous components that may exist at all layers of the implementation stack of an SDN. The compositions defined by S&D patterns can be both vertical and horizontal, i.e., they can involve components at the same (horizontal) or different layers (vertical) layer in the reference architecture of an SDN. To do so, S&D patterns should encode abstract and generic component interaction and orchestration protocols, enhanced (if necessary) by transformations to ensure the semantic compatibility of data or system functionality of the components that are (or need to be) composed. Furthermore, the component interaction and orchestration protocols encoded by the patterns must have an evidenced ability (i.e., an ability proven through formal verification or demonstrated through testing and/or operational monitoring) to achieve a semantically viable interoperability between their components. In SDN, components can be either hosts, forwarding devices or controllers. Paths may include single step links between two edge nodes or link compositions with at least one intermediate and two edge nodes.

In our S&D patterns so far, we have focused on the logical architecture of the network representing end-to-end connectivity, security and dependability. The basic building blocks for forming logical network topologies are the same as those identified for process workflows in [25]. As it can be seen in Fig. 1 for example, the sequence topology depicts the sequential composition of nodes in a network defines that a process is enabled after the completion of a previous one. This topology appears as the fundamental approach for building network process blocks and the diameter/tiers of a network. The multi-choice-join topology (OR-OR) provides the execution of a process to be diverged to two or more