Offloading personal security applications to a secure and trusted network node

Offloading personal security applications to a secure and trusted network node

R. Bonafiglia[§], F. Ciaccia[¶], A. Lioy[§], M. Nemirovsky[‡‡], F. Risso[§], T. Su[§]
[§]Politecnico di Torino, Dip. Automatica e Informatica, Italy
[¶]Barcelona Supercomputing Center (BSC), Spain
[‡‡]ICREA Researcher Professor at Barcelona Supercomputing Center (BSC), Spain

***Abstract*—The current device-centric protection model against security threats has serious limitations from the final user perspective, among the other the necessity to keep each device updated with the latest security updates and the necessity to replicate all the security polices across all devices. In our model, the protection is decoupled from the users terminals and it is provided through a Trusted Virtual Domain (TVD) instantiated in future edge routers. Each TVD provides unified and homogeneous security for a single user, irrespective of the terminal employed. This paper shows a first prototype implementing this concept through a network element, called Network Edge Device, capable of running the proposed virtualized architecture and making extensive use of SDN technologies, with the aim at providing a uniform security level for the final user.**

I. Introduction

Today's classical approach to the security of personal devices requires the installation of security software directly on the devices that need to be protected. With a growing number of user terminals (laptops, smartphones, smart TVs, and more), keeping all those devices protected has become a challenge; achieving a uniform level of security across all of them would be even harder. In fact, each device needs to be configured with the proper security policies, while security patches (e.g. related to the operating system and/or the installed software) needs to be readily applied. However, this may not always be possible, as some devices (e.g. smart TVs) may not support the desired set of security properties, or security patches may not be promptly released by the software manufacturer[1]. The above problems can be mitigated by processing all the network traffic in a security device operating close to the user, possibly integrated in an home gateway or an edge router. In a nutshell, we propose to *offload security* to a *personal* Trusted Virtual Domain (TVD) running in an *Network Edge Device* (NED), in charge of executing all the user's security applications on the network traffic, regardless of the personal device in use, thus reaching *uniform network protection*. This paper introduces the main architectural components of the NED, followed by a description of a prototype developed based on the above architecture; finally a demonstration scenario of the prototype is presented.

[1]It is worth mentioning that some versions of mainstream operating systems, such as Android, Apple IOS or Microsoft Windows, are no longer covered by the proper security patches, although their penetration in the market is still far from marginal.

II. Architecture

A. The TVD

The Trusted Virtual Domain is the main component of this architecture: it is a logical container of all the user network security. Each security control processes exclusively the traffic of a single user, thus is called a Personal Security Application (PSA). It runs in an Execution Environment (EE); the EE is a placeholder for the PSA execution which should be adequately separated from other users' environment, guaranteeing traffic isolation. Multiple PSAs for the same user can run in the same EE in cascade. PSAs behaviour is monitored by another component, the Personal Security Controller (PSC) which is also in charge of configuring PSAs at startup time; the PSC runs in a separate EE. The TVD is Trusted as all the components (hardware and software) are being *attested* following the principles of Trusted Computing [1]. In this way is possible to verify that the software running on the NED is not tampered according to some *golden measurements*.

B. The TVD Manager

The TVD Manager (TVDM) is the architecture Orchestration and Management component. Once the user is authenticated in the system, the TVDM deals with the requests to instantiate a new TVD by determining the resources required for its allocation and then commanding the deployment. The TVDM performs an analysis of the required virtualization technology and the management of the network topology (physical and virtual) to support the deployment of the whole user's Service Graph (SG). The SG is the formalism that describes the service requested by the user and it is, in its simplest form, a set of cascaded PSAs active on the user's traffic. The SG is defined as the superposition of two directed graphs in which nodes represent the PSAs and arcs represent the flow of the traffic between two nodes. Arcs can be labelled with a set of packet filtering rules (e.g. OpenFlow Flowmods) that select which traffic has to flow through that arc in that direction. The TVDM is in charge of translating this abstract model into actual flow rules to be pushed in the virtual switches, thus acting as an OpenFlow controller.

C. The PSC Manager

The PSC Manager (PSCM) is the front-end component of the architecture; it includes the NED Authentication system which can be implemented either locally or using an external

authentication infrastructure (AAA+, OAuth, OpenID...). The PSCM grants an authentication token used by the rest of the infrastructure components to identify the user.

D. Network topology

Users connect to the NED with a secure channel (it can be an Ethernet cable, a WiFi encrypted channel or even a remote encrypted connection, e.g. a VPN using IPsec). Internally the NED presents two separate networks: the data plane network and the control and management network. Their means is to separate the user traffic (data plane) from the traffic generated by the infrastructure components for management and monitoring. The two networks are managed by two virtual switches. The Data Plane switch forwards all the users' traffic; isolation of flows is guaranteed by fine-tuned rules in the switch which allows for traffic steering towards, and only to, the user's TVD.

III. The Prototype

A pure software prototype of this architecture has been developed in the context of the SECURED European project [2]. The prototype relies on customized Virtual Machines as Execution Environment for both PSAs and PSC. The host is running a Fedora Linux distribution with the KVM hypervisor together with QEMU. The NED software is running on commodity hardware (x86 Intel CPU, 8GB RAM, 1Gbit/s NIC with 2 ports). The development of the NED components leveraged on fast prototyping tools (e.g. Python language) and clean API interfaces (RESTful services). To conduct Remote Attestation an external Third Party Verifier hosting the golden measurements and attesting the NED trustability is needed. The Verifier software has been developed and deployed on a different machine using also the Python language. Functional testing has been conducted on the prototype in a use case scenario with two users connecting to the NED. Each user connecting to the NED will go through these steps:

- *Trusted Channel establishment*: a VPN tunnel using IPsec is established towards the NED. The IPsec handshake is not completed until the user terminal receives a trusted verification report from the Verifier. This report proves that the certificate used by the NED to create the secure channel is controlled by the software actually in use, and the software running in the NED is benign and in the latest version.
- *User Authentication*: the PSCM is now reachable and the user can authenticate in the SECURED system.
- *TVD instantiation by TVDM*: the authentication triggers the TVD instantiation and configuration (PSC and PSAs VMs are deployed).

Once the TVD is configured users are finally able to reach the Internet taking advantage of the NED security controls (e.g. in Fig. 1 user1 and user2 both have the firewall PSA but with different policies enforced).

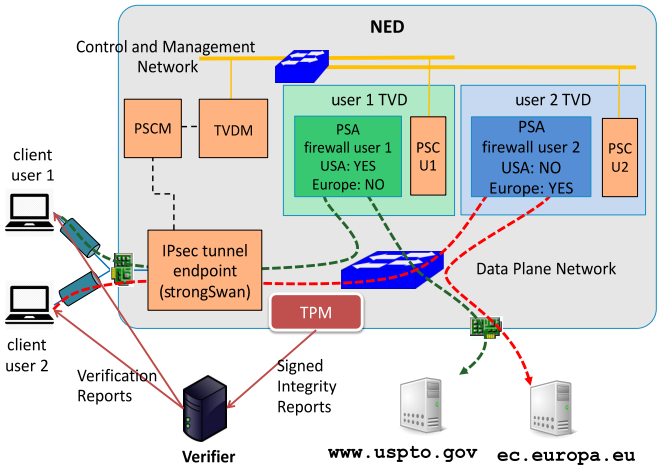

Fig. 1. A NED managing a multi-tenant scenario with different security policies enforced

IV. Conclusions and Future Work

With the prototype implementation we demonstrated the feasibility of the presented architecture. The system architecture, including also the external infrastructure architecture [3], is easily mapped to an NFV deployment and a design compatible with the ETSI standard [4] is currently under development. The concept of the Service Graph allows for complex modeling of VNFs chaining, becoming an NFV enabler. The security paradigm shift towards the network edge explores the new possibility offered by Fog Computing, a scalable architecture which will be an IoT key technology [5] [6]. Finally the Trusted Computing techniques here investigated focus on the concepts of Remote Attestation, an ongoing research topic which can provide authentic evidence about the integrity status of the platform.

Acknowledgment

The research described in this paper is part of the SECURED project [2], co-funded by the European Commission under the ICT theme of FP7 (grant agreement no. 611458).

References

[1] TCG infrastructure working group, "TCG architecture part ii: Integrity management. tcg specification version 1.0, revision 1.0."
[2] "Security at the network edge (SECURED)," http://www.secured-fp7.eu/.
[3] D. Montero, M. Yannuzzi, A. Shaw, L. Jacquin, A. Pastor *et al.*, "Virtualized security at the network edge : A user-centric approach," *IEEE Communications*, to appear, 2015.
[4] ETSI NFV ISG, "NFV white paper ver. 2," https://portal.etsi.org/nfv/nfv_white_paper2.pdf, 2013.
[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of things," in *Workshop on Mobile Cloud Computing (MCC)*, 2012, pp. 13–16.
[6] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog computing, Cloud computing, and more Fog computing," in *Int. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2014, pp. 325–329.