Towards Orchestration and Security of Next Generation Clouds

Roman-Valentyn Tkachuk
Blekinge Institute of Technology (BTH)
Karlskrona, Sweden
roman-valentyn.tkachuk@bth.se
Supervisors: Kurt Tutschku, BTH, kurt.tutschku@bth.se; Dragos Ilie, BTH, dragos.ilie@bth.se;

I. Security Testbeds

As we witness an increase in the frequency of security incidents, ordinary defensive mechanisms, *e. g.*, *firewalls* or *IDSs/IPSs*, must be complemented by offensive actions, *e. g.*, *penetration testing*. The reason is that many modern applications rely on multiple distributed components interconnected over the Internet and located in different administrative domains. This makes perimeter-based defenses less effective as attackers try to exploit vulnerabilities in the application logic to laterally move towards where the assets are located. The aforementioned offensive testing actions help to proactively find and patch vulnerabilities that enable this type of intrusions. However, when executed towards a production environment, these actions may disrupt or degrade the operation of the application. It would be useful if one can replicate the application within a testbed where it can be subjected to offensive actions without disturbing the production system.

Infrastructure as a Service (IaaS) solutions, *e. g.*, *OpenStack*, permit the setup and deployment of scalable environments for rapid and flexible service testing. The configuration and deployment of cloud-based systems can be done manually. However, this process can become a highly time consuming and error-prone when provisioning must be done repeatedly. In order to make provisioning process more efficient, IaaS Systems employ the idea of *Infrastructure as Code (IaC)*, which is the process of provisioning of computing infrastructure through machine-readable definition files. The advantages of IaC code have been addressed in our study [1], along with scalability demonstration of one of the tools.

While IaC makes the process of network devices provisioning more efficient, one still needs the special knowledge to write provisioning scripts and operate the cloud. In order to decrease the initial training period to use the testbed, a comprehensive interface has to be built. This interface is a part of the framework which allows us to connect to the IaaS cloud and use it as a platform for security tests while having no need to know how the cloud operates internally.

As part of the *Test Arena Blekinge* project that was carried out in Karlskrona, Sweden, we have investigated architecture and technologies suitable for automating a testbed for security. We defined the components of a security testbed framework, outlined an architecture for such testbed, detailed an specific implementation and discussed the capabilities, limitations and lessons learned from the testbed implementation for security testing. The results are presented in our study [2].

For future research we would like to investigate the possibility of federated testbed, connecting multiple IaaS clouds at a time, and having a possibility to run cross-cloud security tests.

II. Cloud-based Services Security

Cloud systems, permit deployment of new services and service chains rapidly and flexibly. However, while being exposed in the network, services are objects of constant security risk. Therefore, there is a need to develop new security mechanisms, which would protect exposed services on place in cloud.

In our team we have investigated security mechanisms that can be generic enough to be applied to majority of services that being deployed nowadays yet provide only authorized access, executed commands verification and compliance with the user license.

As part of the *H2020 Bonseyes* project (*www.bonseyes.eu*), we have developed a *proxy based architecture for Platform as a Service clouds*. The proxy is an entity that is placed on-demand, close to the service and prevents its direct access from users. The proxy is exposed to the user via *User-bound interface*. A user can connect to the service, only thorough this interface. Furthermore, the proxy connected to the service via the *Service-bound interface*. A *trust boundary* between the user and service is implemented by permitting connections only to the proxy, while having the proxy verify the eligibility of received commands and check compliance with the user license. The compelling feature of our approach is that it provides clear separation without requiring any explicit support from the service. This reduces significantly the efforts to develop services since the designer can focus on service functions only and does not need to address security mechanisms in the service.

The research on the cloud-based services security will be continued in *KKS HÖG Symphony (Supply-and-Demand-based Service Exposure using Robust Distributed Concepts)* project where research area will be extended to exposing services in *Cloud-native* and *federated environments*, addressing the security, confidentiality, privacy and provenance needs for future applications of digital societies.

References

[1] R.-V. Tkachuk, D. Ilie, and K. Tutschku, "Orchestrating future service chains in the next generation of clouds," in *Proc. SNCNW*, Luleå, Sweden, Jun. 2019.

[2] ——, "Building a framework for automated security testbeds in cloud infrastructures," in *Proc. SNCNW*, Kristianstad, Sweden, May 2020.