This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952647

An Open-Source Cloud Testbed for Security Experimentation

Francesco Minna[1] and Fabio Massacci[1,2]

[1] Vrije Universiteit Amsterdam
[2] University of Trento

Contact

Introduction

The use of container and orchestration technologies, such as Docker and Kubernetes keeps growing every year. For the purpose of security experimentation and reproducibility of security attacks and defenses, an open-source testbed would be an important step forward. Yet, while several testbeds have been proposed in other domains (e.g., web applications testing and CTFs), a similar solution for the cloud is still missing.

To fill this gap, we propose an open-source cloud testbed that, by using Domain Specific Language (DSL) files, allows defining experimentation scenarios as configuration files. Similar to container and container images, using DSL files allows to create, share, customize, automatically deploy, and reproduce different scenarios in a user-friendly manner.

Solution Design

The design of our solution is based on the Build-it, Break-it, Fix-it (BIBIFI) approach, allowing practitioners to define custom cloud deployments (Build-it), deploy, interact, and eventually exploit applications and security tools (Break-it), and finally assess or improve the configurations (Fix-it).

Fig. 1 presents the workflow of using the testbed tool and DSL files to deploy experiment scenarios.

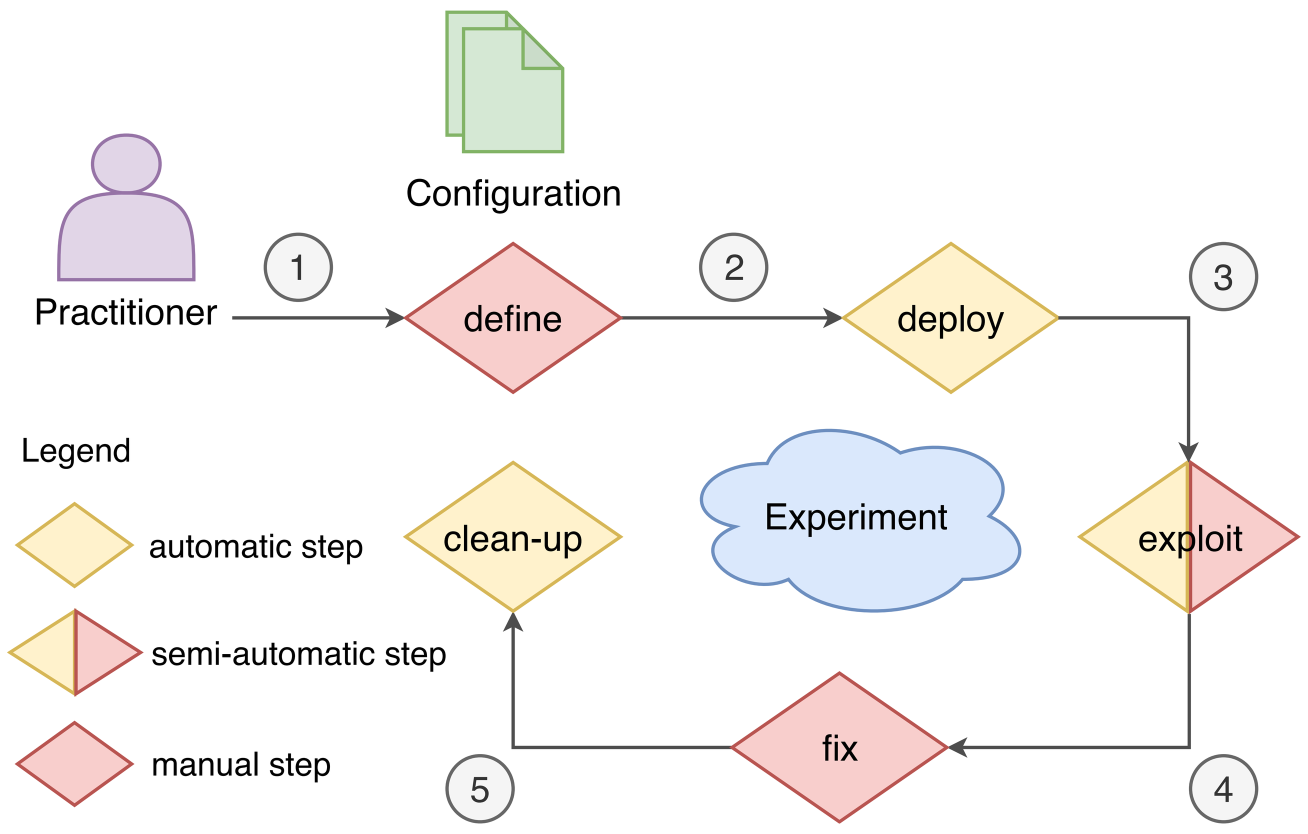
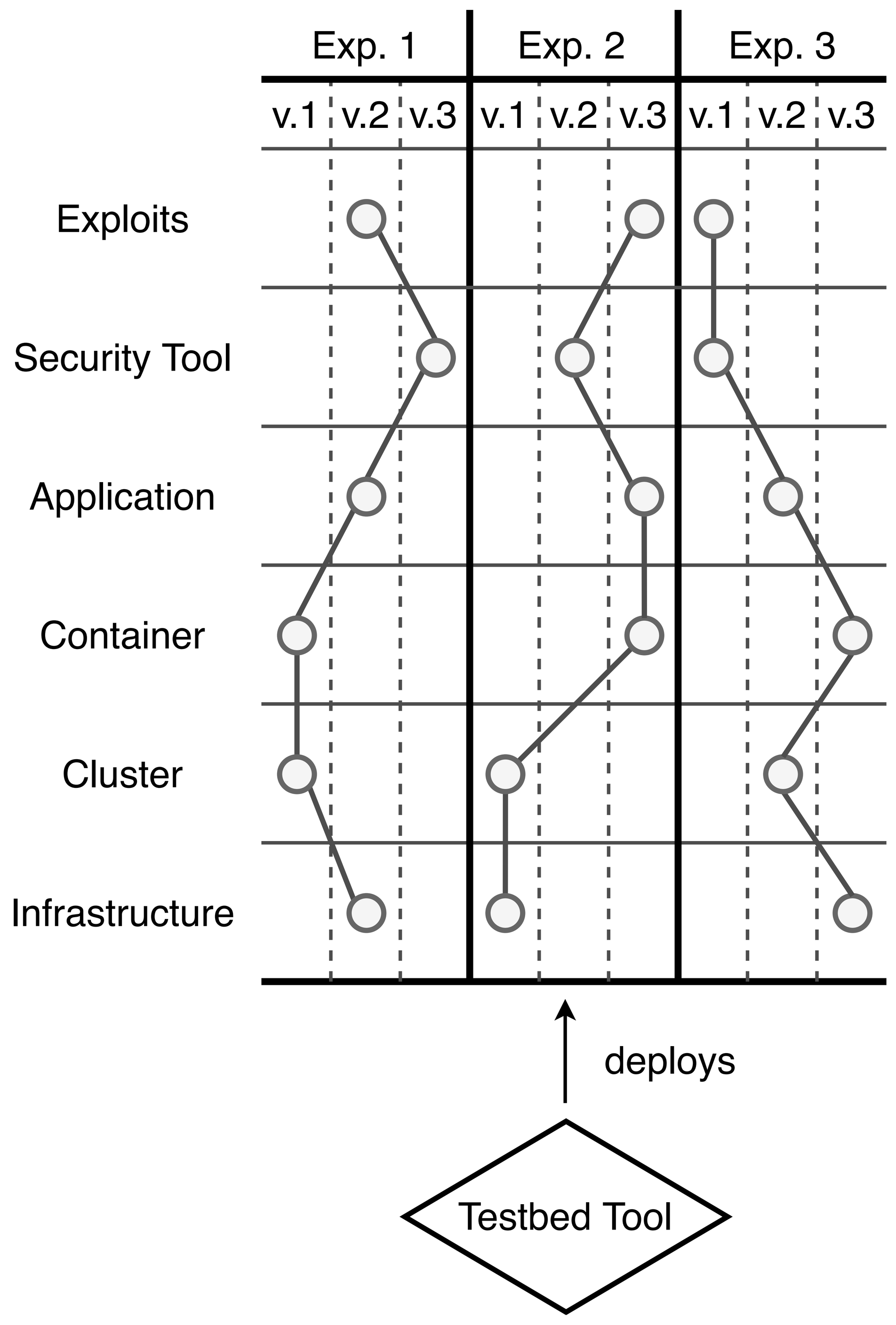


Fig. 2 shows an example of deploying different experiments with different configurations each.



Implementation Proposal

Based on prior work, Tab. 1 provides a list of tools and technologies that can be used to set up each layer of cloud deployments, namely, infrastructure, cluster, container, and code (divided into applications, tools, and exploits).

Layer	Examples	Alternatives
Exploits		`Ansible`, `Bash script`, `ASL`, `MAL`
Security Tools	`Helm charts`, `YAML files`	`Bash script`
Applications	`Helm charts`, `YAML files`	`Bash script`
Containers	`Docker`, `containerd`, `CRI-O`	`Mirantis`
Cluster	`Kubernetes`	`OpenShift`, `Mesos`
Infrastructure	`Vagrant`, `Ansible`	`Terraform`, `Packer`

Fig. 3 shows an example of a JSON file representing the configuration of an experiment.

```
"infrastructure": {
    "platform": "bare-metal",
    "provider": "vmware_workstation"
},

"cluster": {
    "orchestrator": "kubernetes",
    "version": "1.23.0",
    "cni": "Cilium"
},

"container": {
    "engine": "cri-o",
    "version": "1.18.6"
},

"misconfiguration": {},

"application": {
    "name": "charts/piggymetrics",
    "namespace": "piggymetrics"
},

"security-tool": {
    "name": "falco falcosecurity/falco",
    "namespace": "security-tools"
}

"exploit": {}
```

Impact and Contributions

- Multi-platform and multi-layer support (e.g., infrastructure and orchestration)
- Experiments-as-code (i.e., repeatability and reproducibility of experiments)
- DSL to replicate defence and attack scenarios in the cloud (complex and multi-step payloads)
- Digital twin for risk assessment

References

[1] K. Documentation. Overview of cloud native security. URL `https://kubernetes.io/docs/concepts/security/overview/#the-4c-s-of-cloud-native-security`. (Accessed on: 09/05/2022).

[2] C. Leitner. Vienna university of economics and business (wu) poster template. URL `https://www.overleaf.com/latex/templates/vienna-university-of-economics-and-business-wu-poster-template/zwkcztjcrhfk`. (Accessed on: 09/05/2022).

[3] J. Parker, M. Hicks, A. Ruef, M. L. Mazurek, D. Levin, D. Votipka, P. Mardziel, and K. R. Fulton. Build it, break it, fix it: Contesting secure development, apr 2020. ISSN 2471-2566. URL `https://doi.org/10.1145/3383773`.