Sandbox User Manual

SPHINX

A Universal Cyber Security Toolkit for Health-Care Industry

Table of contents

Table of figures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1 Introduction

The sandbox has 2 main functions. The first is to allow the end user to easily create different topologies in order to execute/examine VMs or other individual software components. The solution uses KVM in its core and every topology is deployed by using separate docker containers. As a result, each docker initiates the KVM and the gateway of the topology is the virtual network interface provided by docker.

The second function of the sandbox is to automatically isolate network devices that are tagged as malicious or defective. This information is provided by the SIEM and the sandbox jails the network device to the virtual/replicated environment.

2 Installation/Deployment

After downloading/cloning the git repository you build the docker image and then you start the container by using the following command:

```
> docker run -it -d --privileged -v /sys/fs/cgroup:/sys/fs/cgroup:ro --name sphinx-sandbox sphinx-sandbox
```

This command enables the docker to use KVM. Afterwards Inside the container execute and run the script:

```
> sudo chmod +x script.sh
```

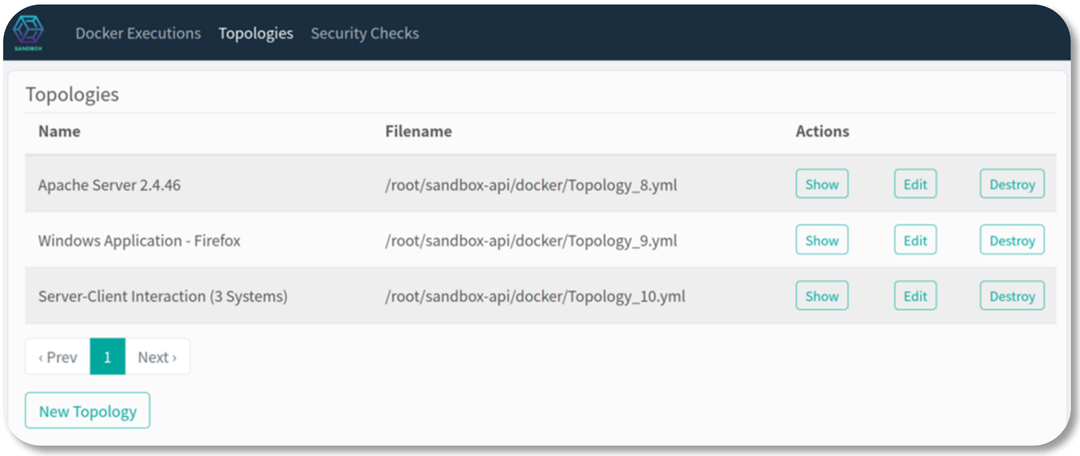
```
> ./script.sh
```

Open browser and open [docker-ip]:9090. This will allow you to access the Web UI and start installing the VMs.

If you want to execute a lightweight docker container inside a microVM then you have to clone the repository sandbox-api, build the image and start a container or by installing ruby and execute > rails server. This will start the web UI at local port 3000 by default.

3 Use Case 01: Deploy docker services or topologies

For deploying docker topologies you access the web UI or use the REST API and send a yml topology to be deployed. Actually, this endpoint receives the data and creates a filename which then is parsed periodically by the sandbox and deploys the VMs.



Topologies

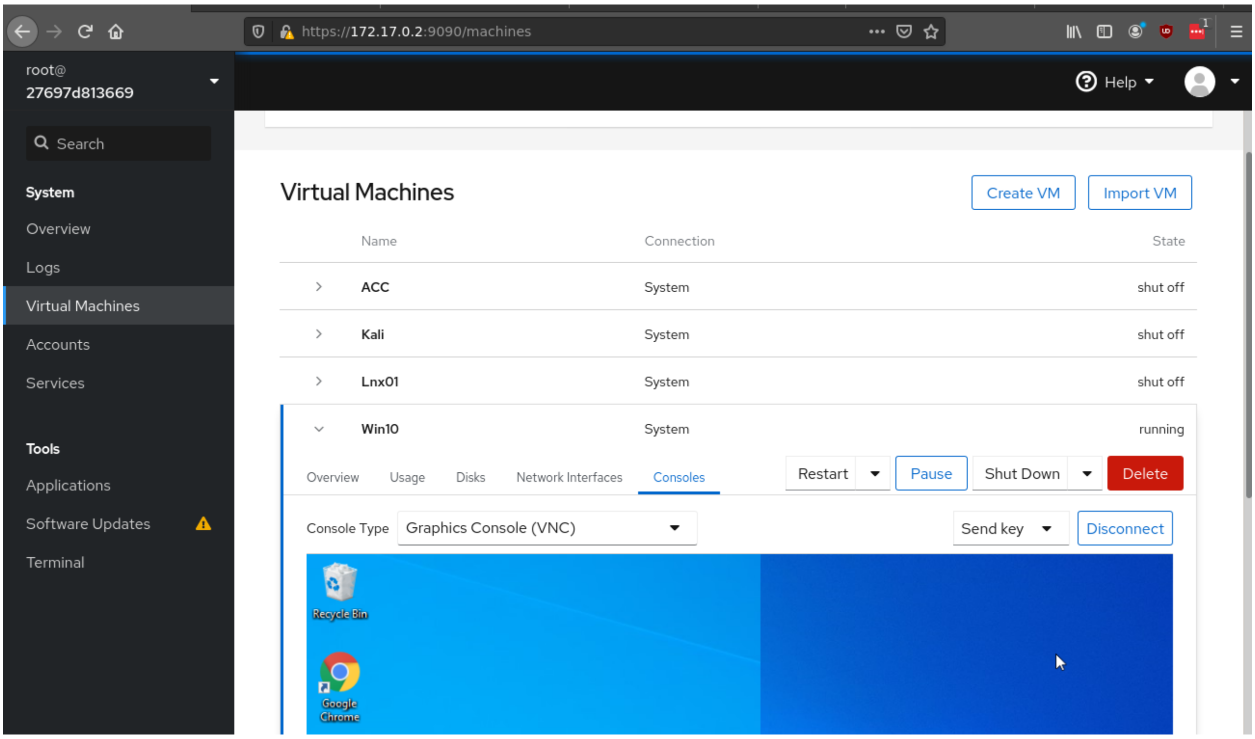
Name	Filename	Actions		
Apache Server 2.4.46	/root/sandbox-api/docker/Topology_8.yml	Show	Edit	Destroy
Windows Application - Firefox	/root/sandbox-api/docker/Topology_9.yml	Show	Edit	Destroy
Server-Client Interaction (3 Systems)	/root/sandbox-api/docker/Topology_10.yml	Show	Edit	Destroy

Figure 3.1 List of topologies to be deployed

It is possible to show and edit the yml files that have been already added. The sandbox recursively will recreate the topologies (not implemented yet). The topologies will be deployed, and the certification tasks will be installed for retrieving insights from the deployed systems. This implementation is currently deploying only Linux VMs and internally the docker containers. For having sandboxes that have Windows VMs see use case 02.

4 Use Case 02: Deploy VMs manually

Like a ESXi server or Proxmox it is possible to deploy virtual machines manually inside the sandbox. An existing topology is there already, and it is possible to clone the whole topology in another docker container. Therefore, every network communication will fall behind the dockers' virtual gateway, separating the VLANs that are created from KVM from each other. Each of the deployed sandboxes can be accessed from a WebUI (**Figure 4.1**).



https://172.17.0.2:9090/machines

root@
27697d813669

Virtual Machines

Name	Connection	State
ACC	System	shut off
Kali	System	shut off
Lnx01	System	shut off
Win10	System	running

Console Type: Graphics Console (VNC)

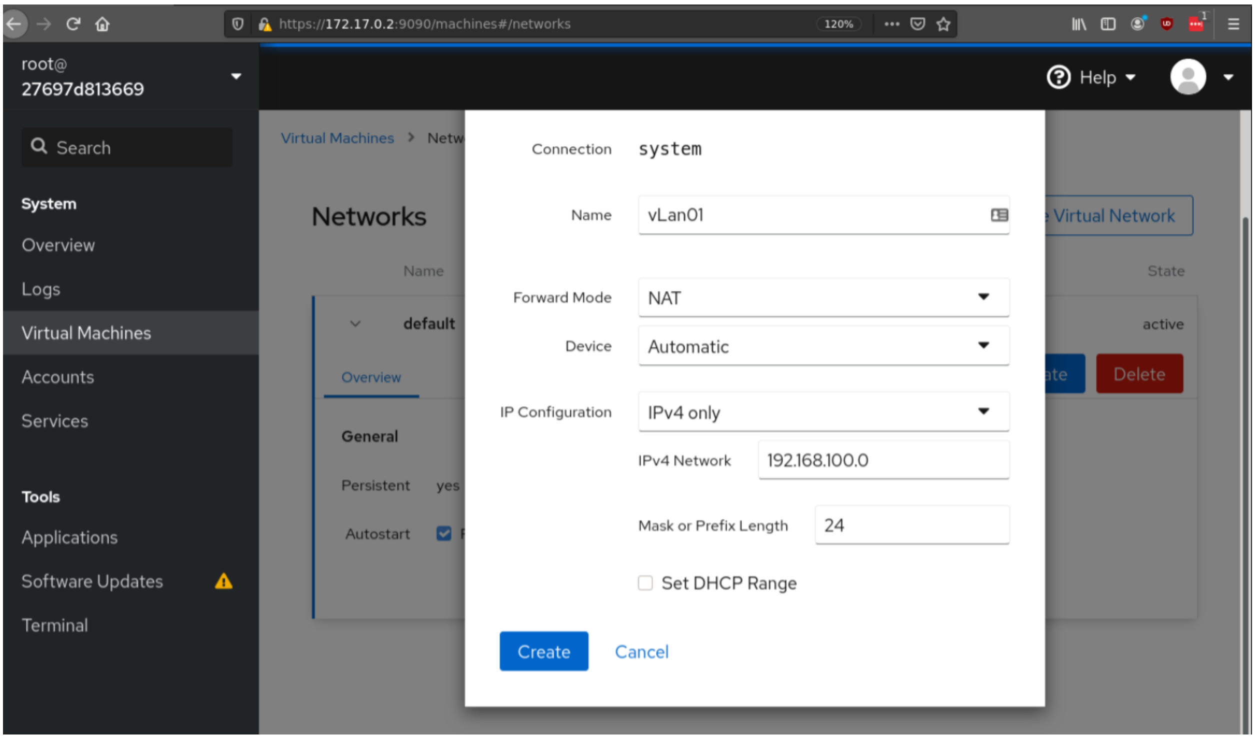
Figure 4.1 WebUI for accessing the sandbox and each of the VMs

Each of the Virtual Lans (Vlans) can be created by using the network settings and assign the VMs accordingly (**Figure 4.2**). In this approach a second part includes to install the agents which will collect the logfiles and status of the machines to closely monitor them. The following setup stages are to be done:

- Install Sandbox
- Deploy Machines or clone the existing topology
- Deploy SIEM Agents
- Deploy Wazuh Agents
- (Extra step for Linux Systems only) -> Deploy Lynis

By deploying the components, we will be able to investigate the system status and send data to the SIEM.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.



https://172.17.0.2:9090/machines#/networks

Connection	system
Name	vLan01
Forward Mode	NAT
Device	Automatic
IP Configuration	IPv4 only
IPv4 Network	192.168.100.0
Mask or Prefix Length	24

☐ Set DHCP Range

Figure 4.2 Network configuration of the systems inside the Sandbox

5 Use Case 03: Network Isolation of BYOD devices inside the Sandbox

This last use case is a core process for a various use cases in Sphinx. The sandbox has to be established as a gateway to the network that is monitored and will be act as a DHCP server. Every communication will pass from the sandbox. If anything, suspicious happens the sandbox will block and forward the traffic of the suspicious device to a virtual Lan which includes various replicated systems which might be similar to the real infrastructure. The network traffic and behavior of the suspicious device is monitored.

The sandbox is informed from the SIEM which devices are suspicious and the sandbox proceeds to applying the specific rules.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.