MUNI
FI

PA211 Advanced Topics of Cyber Security

September 13, 2022

Pavel Čeleda, Jan Vykopal et al.

Course organization

Instructors

- The course is toughed by **CYBERSEC lab** – cybersec.fi.muni.cz
- **Pavel**, **Honza** (Jan), Lukáš, Daniela and others

Goal of this course

- The objective of the course is to **cover specific knowledge** and skills required for the work as:
  - Cyber defense infrastructure support specialist (PR-INF-001),
  - Systems security analyst (OM-ANA-001),
  - Vulnerability assessment analyst (PR-VAM-001),
  - and many other non-formally defined DevOps positions.

- Defined by the **NICE** cybersecurity workforce **framework**
  - https://niccs.cisa.gov/workforce-development/cyber-security-workforce-framework/

- Advanced hands-on **cybersecurity** course for **master** students

What you will learn

On successfully **completing the course** you will be able to:

- conduct **vulnerability scans** and **recognize vulnerabilities**,
- conduct **penetration testing** on enterprise network and applications,
- apply selected **countermeasures to harden** (secure) networks, operating systems, and applications.

Topics not covered in this course

- **Introduction** course:
  - **PV210** Cybersecurity in an Organization
- **Advanced** and specialized courses:
  - **PV276** Seminar on Simulation of Cyber Attacks
  - **PV279** Digital Forensics
  - **PV280** Network Forensics

Course format

- **Informal class** – make friends and share knowledge!
- 3 hours **block** – we start at **10:00** – A219
  - 1 hour **lecture** – topic introduction
  - 2 hours **seminar** – hands-on labs / tutorials to practice the lecture topic
- Individual involvement / work
- 4 x homework
- Anonymous **exit ticket** at the end of each lecture to **get feedback** and **improve** running course.

Grading

Assignments during the semester (60 %)

- **Four homeworks** – 4 x 15 = 60 points

Final exam (40 %)

- **Hands-on exam** – 40 points

To **pass** the course, you must **submit** the **homeworks** and **attend** the hands-on **exam**. The exam will be based on **labs sessions** content and **homeworks**.

Course schedule

Week	Date	Class Topic
1	13.09.2022	Course organization and motivation
2	20.09.2022	Asset management
3	27.09.2022	Vulnerability management
4	04.10.2022	Threat management
5	11.10.2022	Penetration testing – introduction
6	18.10.2022	Penetration testing – process
7	25.10.2022	Penetration testing – report
8	01.11.2022	Penetration testing – exemplary report and presentations
9	08.11.2022	Introduction to web application hardening
10	15.11.2022	OS-level, virtualization and containerization
11	22.11.2022	Access control mechanisms
12	29.11.2022	Web server and application hardening
13	06.12.2022	Course feedback session

Part I – Security operations management

- **Syllabus:** Asset, vulnerability, and threat management
- **Objectives:**
  - Introduce **selected parts** of security operations management
  - Focus **mainly on practical** skills and only on **necessary theory**
- **Learning outcomes:**
  - **Hands-on experience** with cybersecurity tools (e.g., asset inventory, and ELK stack)
  - Knowledge of selected **security operations processes**
  - Knowledge of enumerations, **knowledge bases**, and **data sources**
- **Assessment:** 1 homework

Part II – Penetration testing practice

- **Syllabus:** Process, report, and presentation
- **Objectives:**
  - **Understand the process** of authorized penetration testing
  - Focus **on the process,** not individual tools
- **Learning outcomes:**
  - **Hands-on experience** with penetration **testing of a realistic application**
  - Knowledge of a **structure of a testing report**
  - Exercising skills for **preparing report and presentation**
- **Assessment:** 1 homework – report and presentation

Part III – Hardening of OS and applications

- **Syllabus:** Web application stack hardening
- **Objectives:**
  - Introduce basic principles and best practice of system hardening
  - **Selected use case:** web application service
- **Learning outcomes:**
  - **Hands-on experience** with tools for monitoring, system configuration (e.g., Pakiti, Ansible)
  - Knowledge and practical usage of selected **access control mechanisms**
  - Knowledge of web-based attacks countermeasures, hardening of web app and servers
- **Assessment:** 2 homework(s)

Conclusion

PA211 course has just been born again

- This is the very **first run** of highly **innovated** PA211 **course**
- It will bring us a lot of **fun**
- Warning: **something may go wrong**, but we will find a way out
- We will be **learning** and **improving** as well
- We would highly appreciate your **feedback!**

Collaboration with us beyond the course

1) Write your **thesis** – bachelor, master, or Ph.D.

- New: an opportunity of **Ph.D. trial** during your **master's** degree

2) Get a **paid job**

- Join our research and **development projects**

3) Engage in cybersecurity **community activities**

- Create **technical challenges**

For more details see – **https://muni.cz/go/cybersec**

Lab session

Lab session organization

1. **Familiarization** with Vagrant and a sandbox **at computers in A219**
2. **Installation** of Vagrant and VirtualBox at **own hardware**

Optional today, but recommended for further labs and homeworks.

Feel free to leave if you are familiar with these tools!

PCs at school vs. own devices

Do you prefer using your own hardware?

Learning objectives

At the end of this lab session, you will be able to

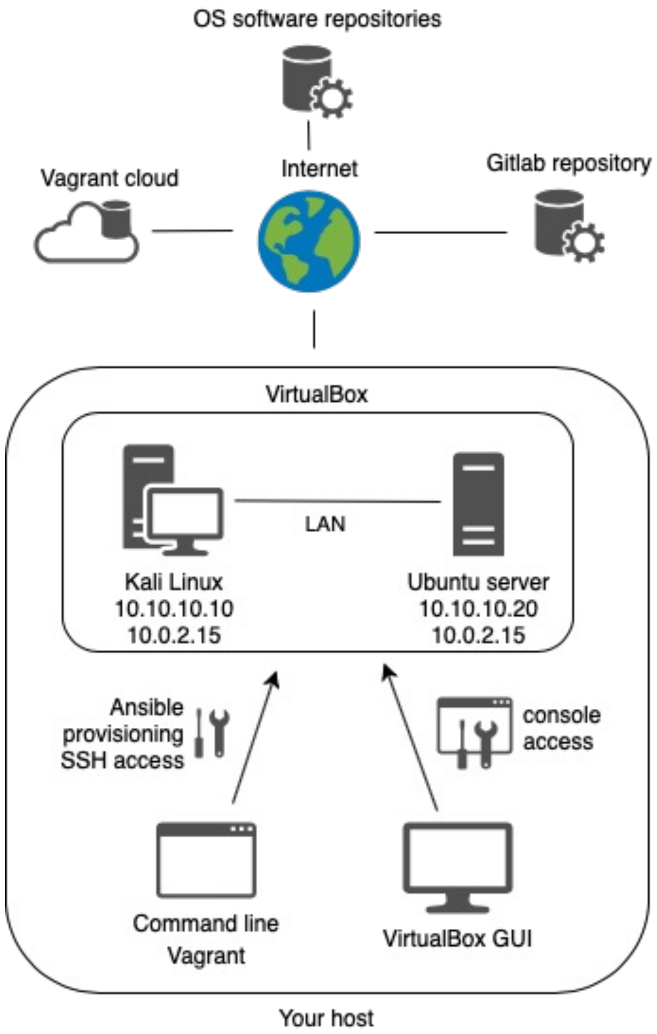
- **set up a virtual network environment** (sandbox) at your computer,
- **access the sandbox** and its hosts via SSH from both host and guest machines,
- **troubleshoot** the sandbox and services,
- **use** the sandbox **at your own machine**.

Familiarization with Vagrant and a sandbox at computers in A219

Building blocks of our lab session

- VirtualBox
- Vagrant
- Vagrant boxes (such as Kali Linux)

Sandbox – virtual network environment



- Each of you will use a local sandbox with virtual machines
- VirtualBox hosts virtual machines (VMs) accessible from your host
- Vagrant controls VirtualBox and configures VMs using `Vagrantfile`
- Vagrant provides SSH access to VMs with this command:

  `vagrant ssh <name of the VM>`
- Vagrant accesses each VM using its first network interface with IP 10.0.2.15; this interface is also used for communication of the VM with the Internet
- VMs can also be accessed directly using console in VirtualBox GUI

Let‘s start in A219 – preparation

- VirtualBox and Vagrant are already installed at PCs in A219.
- Log in and open Terminal.
- Run this script:

  ```
  pa211_setup
  ```

  It optimizes handling of Vagrant boxes, large files with images of operating systems. This script is not needed at your own PC.
- Clone a repository with the sandbox (virtual environment) you will use next week:

  ```
  git clone https://gitlab.fi.muni.cz/cybersec/pa211/management.git
  ```

Start the sandbox

- Once you cloned the repository, change directory to `dist` directory:

```
cd management/dist
```

- There is a file named `Vagrantfile`, which defines the sandbox:

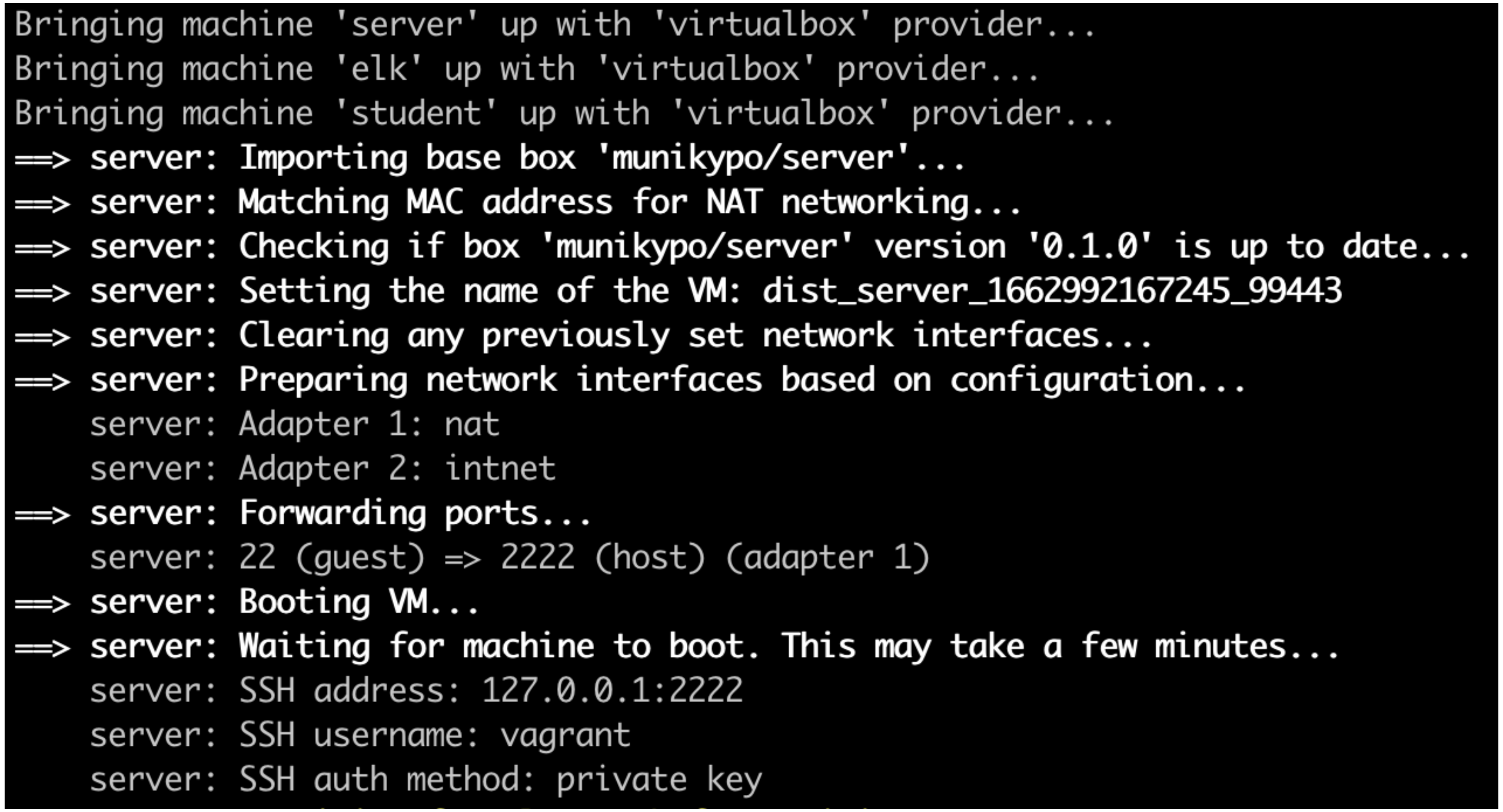
```
ls Vagrantfile
```

- Start the sandbox by this command:

```
vagrant up
```

Be sure, you’re in the dist directory and not in the root of the management repository.

Sandbox is starting, please wait…



```
Bringing machine 'server' up with 'virtualbox' provider...
Bringing machine 'elk' up with 'virtualbox' provider...
Bringing machine 'student' up with 'virtualbox' provider...
==> server: Importing base box 'munikypo/server'...
==> server: Matching MAC address for NAT networking...
==> server: Checking if box 'munikypo/server' version '0.1.0' is up to date...
==> server: Setting the name of the VM: dist_server_1662992167245_99443
==> server: Clearing any previously set network interfaces...
==> server: Preparing network interfaces based on configuration...
    server: Adapter 1: nat
    server: Adapter 2: intnet
==> server: Forwarding ports...
    server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Booting VM...
==> server: Waiting for machine to boot. This may take a few minutes...
    server: SSH address: 127.0.0.1:2222
    server: SSH username: vagrant
    server: SSH auth method: private key
```

Sandbox is starting, please wait… II

- This sandbox consists of three machines in the same local network.
- Booting and configuring takes about 15 minutes.
- If everything is OK, you will not see any red error messages at the output.
- Sometimes it may fail – for various reasons, see the troubleshooting part.

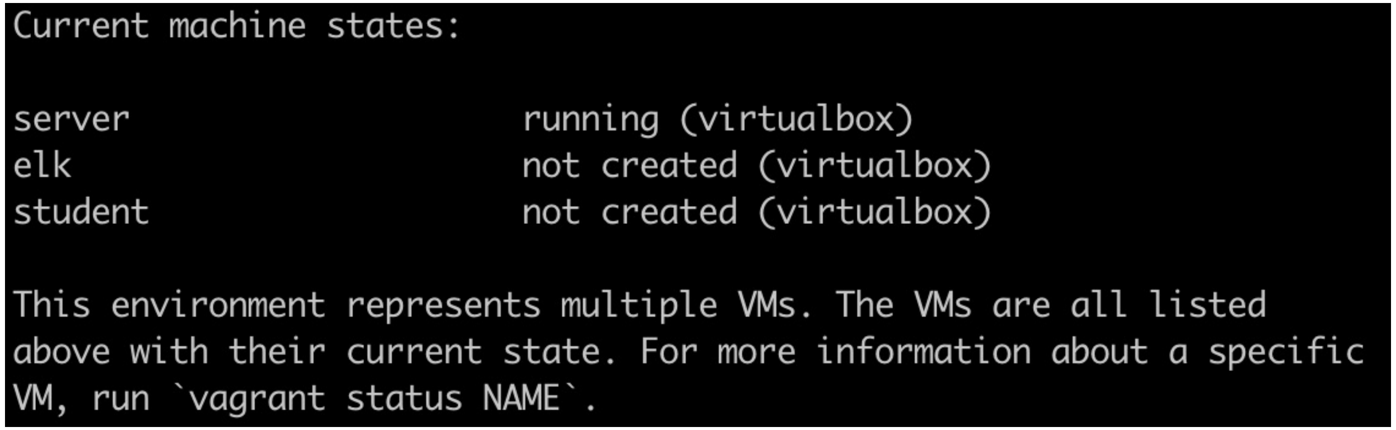
Check the status of the sandbox

- Open a new terminal window in the same working directory

  `management/dist`

- Check the status of the machines:

  `vagrant status`



```
Current machine states:

server                    running (virtualbox)
elk                       not created (virtualbox)
student                   not created (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
```

Connect to the student machine

When you sandbox is up and running, connect to the `student` machine.

There are two options:

1. command-line access using SSH
2. access to graphical interface using VirtualBox console

Connect to the student machine via SSH

1. Run in the directory with a sandbox:

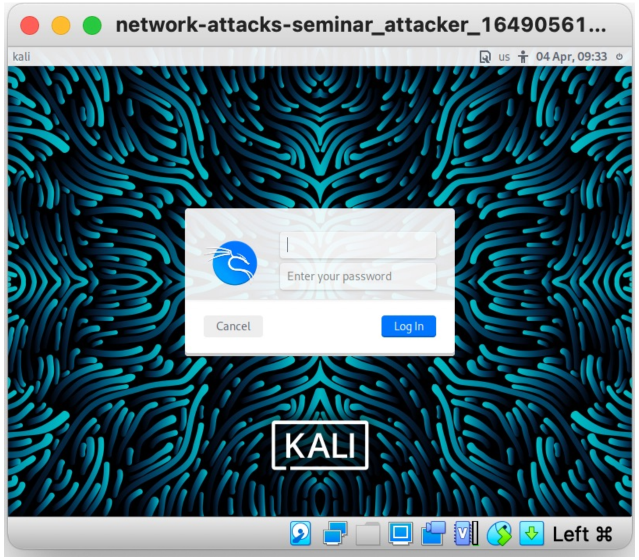
   ```
   vagrant ssh student
   ```

   You are logged in as user `vagrant.`

2. Change user to kali using `su kali` and type `kali` as password.

Connect to the student machine via console

1. Switch to a new window with login screen (a VirtualBox console).
2. Enter `kali` as username and `kali` as password
3. For unknown reason, you may need to log in twice for the very first time.

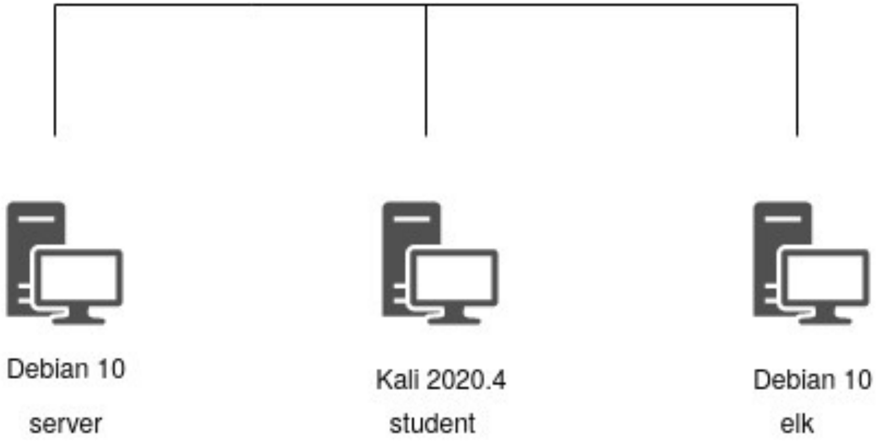


Check networking

1. Switch back to a terminal in CLI or open Terminal in GUI.
2. Check whether you can reach other machines from the student machine:

```
ping elk
ping server
```

All machines must be reachable.



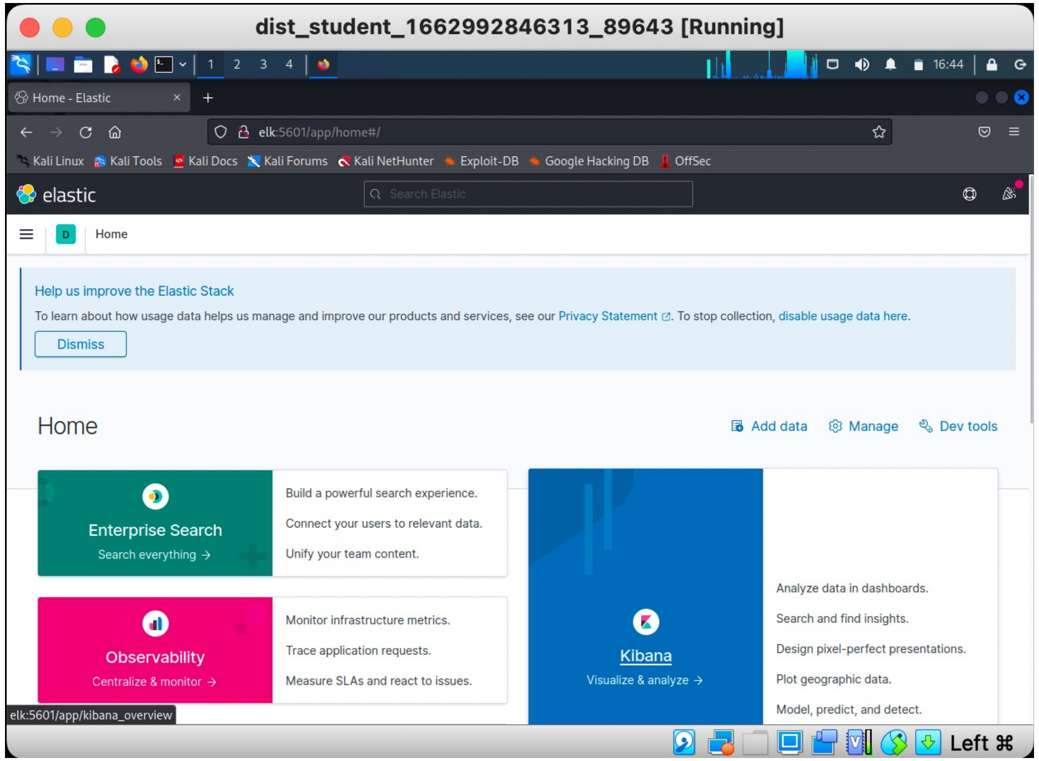
Check network services

- Elk machine provides ELK Stack.
- Student machine hosts Netbox tool.
- Both tools will be used next week.
- You will interact with a web interface of both services.

Check ELK is running

1. Open Firefox at student
2. Visit `http://elk:5601/` using Firefox.
3. You should see Kibana web interface.

If not, check whether the ELK is up in the troubleshooting part.



dist_student_1662992846313_89643 [Running]

elk:5601/app/home#/

Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

Home

Enterprise Search — Search everything →

Build a powerful search experience.
Connect your users to relevant data.
Unify your team content.

Observability — Centralize & monitor →

Monitor infrastructure metrics.
Trace application requests.
Measure SLAs and react to issues.

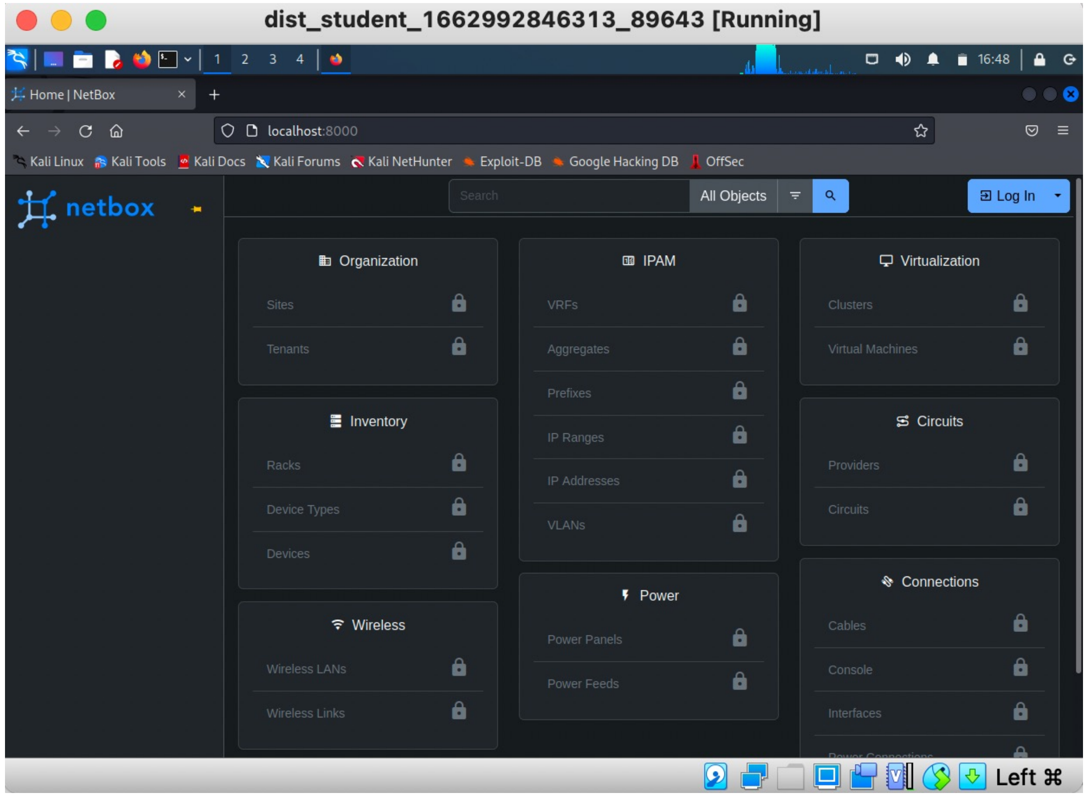
Kibana — Visualize & analyze →

Analyze data in dashboards.
Search and find insights.
Design pixel-perfect presentations.
Plot geographic data.
Model, predict, and detect.

elk:5601/app/kibana_overview

Check Netbox is running

1. Open Firefox at student
2. Visit `http://localhost:8000/` using Firefox.
3. You should see Netbox web interface.



If not, check whether Netbox is up in the troubleshooting part.

Leaving your sandbox

- If you would like to stop the sandbox, you can power it off or destroy it.
- To power off your sandbox, run `vagrant halt` in the directory with the sandbox files.
- To completely delete your sandbox and start from scratch the next time, run `vagrant destroy.`
- In any case, start your sandbox with `vagrant up` next time.
- If you use computers in **A219**, run **`vagrant destroy`**.

Troubleshooting

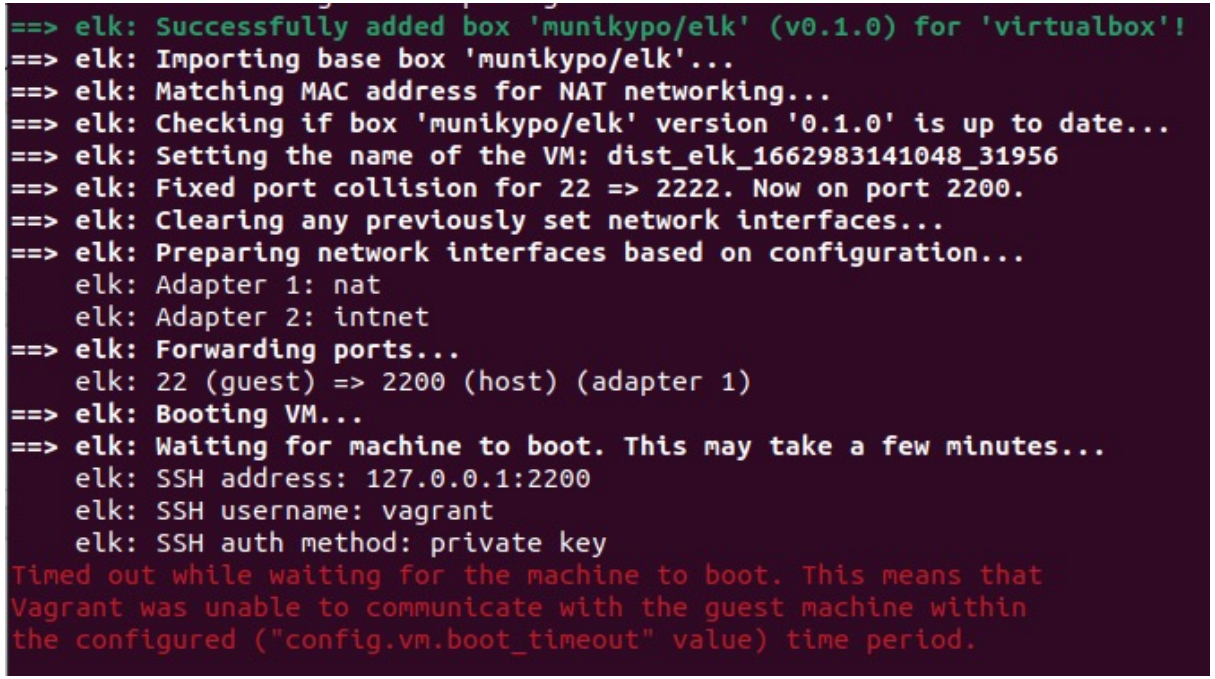
Generic approach

- **Repeat some steps several times before giving up**
- Check machine status
- Check and re-configure networking
- Check and re-configure service locally at particular machine
- Prerequisites:

  Linux networking and sysadmin skills, basic Docker commands

Troubleshooting Vagrant

- Vagrant may fail booting up a machine:



```
==> elk: Successfully added box 'munikypo/elk' (v0.1.0) for 'virtualbox'!
==> elk: Importing base box 'munikypo/elk'...
==> elk: Matching MAC address for NAT networking...
==> elk: Checking if box 'munikypo/elk' version '0.1.0' is up to date...
==> elk: Setting the name of the VM: dist_elk_1662983141048_31956
==> elk: Fixed port collision for 22 => 2222. Now on port 2200.
==> elk: Clearing any previously set network interfaces...
==> elk: Preparing network interfaces based on configuration...
    elk: Adapter 1: nat
    elk: Adapter 2: intnet
==> elk: Forwarding ports...
    elk: 22 (guest) => 2200 (host) (adapter 1)
==> elk: Booting VM...
==> elk: Waiting for machine to boot. This may take a few minutes...
    elk: SSH address: 127.0.0.1:2200
    elk: SSH username: vagrant
    elk: SSH auth method: private key
Timed out while waiting for the machine to boot. This means that
Vagrant was unable to communicate with the guest machine within
the configured ("config.vm.boot_timeout" value) time period.
```

Troubleshooting Vagrant – destroy

- If machine booting or initial configuration by Vagrant fails, run:

```
vagrant destroy <name_of_the_machine>
```

- After that, give it another try:

```
vagrant up <name_of_the_mahcine>
```

Troubleshooting Vagrant – provision

- If you see „failed“ in red after the initial configuration, try re-provision the software and configuration first:

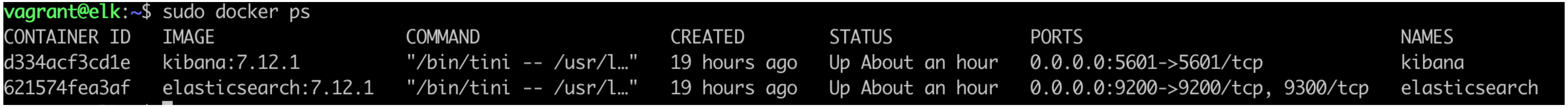
  ```
  vagrant provision <name_of_the_machine>
  ```

- If it does not help, run `vagrant destroy` and up again.

Troubleshooting ELK

If the ELK is not reachable from `student`, connect to `elk` machine and check the ELK status:

1. `vagrant ssh elk`
2. `curl localhost:5601` You should see empty response (but no error).
3. `sudo docker ps` There should be two containers, both in the up status.



```
vagrant@elk:~$ sudo docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS              PORTS                              NAMES
d334acf3cd1e   kibana:7.12.1         "/bin/tini -- /usr/l…"   19 hours ago     Up About an hour    0.0.0.0:5601->5601/tcp             kibana
621574fea3af   elasticsearch:7.12.1  "/bin/tini -- /usr/l…"   19 hours ago     Up About an hour    0.0.0.0:9200->9200/tcp, 9300/tcp   elasticsearch
```

4. If any container is not up, take a closer look at its logs.

   Grab its ID (such as d334acf3cd1e for kibana) and print out logs:

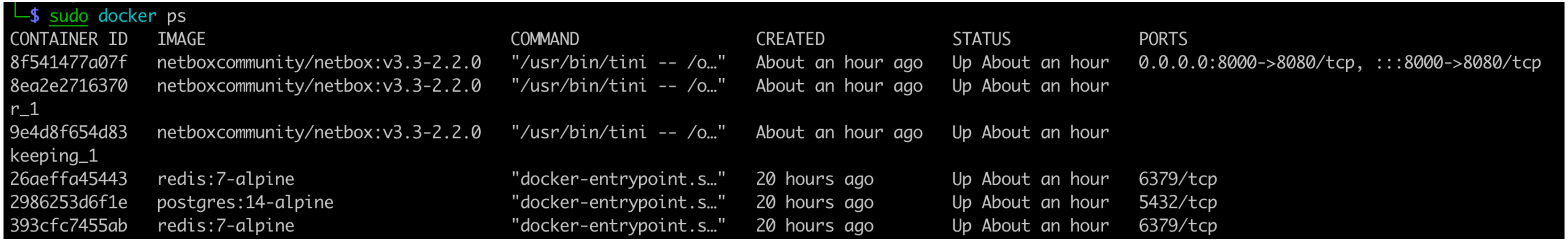
   `sudo docker logs d334acf3cd1e`

Troubleshooting Netbox

If Netbox is not reachable from `student`, check its Docker containers.

Open Terminal at student, switch to kali user, and run:

1. `sudo docker ps`   There should be six containers, all in the up status.



```
└─$ sudo docker ps
CONTAINER ID   IMAGE                               COMMAND                  CREATED              STATUS              PORTS
8f541477a07f   netboxcommunity/netbox:v3.3-2.2.0   "/usr/bin/tini -- /o…"   About an hour ago    Up About an hour    0.0.0.0:8000->8080/tcp, :::8000->8080/tcp
8ea2e2716370   netboxcommunity/netbox:v3.3-2.2.0   "/usr/bin/tini -- /o…"   About an hour ago    Up About an hour
r_1
9e4d8f654d83   netboxcommunity/netbox:v3.3-2.2.0   "/usr/bin/tini -- /o…"   About an hour ago    Up About an hour
keeping_1
26aeffa45443   redis:7-alpine                      "docker-entrypoint.s…"   20 hours ago         Up About an hour    6379/tcp
2986253d6f1e   postgres:14-alpine                  "docker-entrypoint.s…"   20 hours ago         Up About an hour    5432/tcp
393cfc7455ab   redis:7-alpine                      "docker-entrypoint.s…"   20 hours ago         Up About an hour    6379/tcp
```

2. If any container is not up, take a closer look at its logs. Grab its ID (such as 8f541477a07f) and print out logs:

   `sudo docker logs 8f541477a07f`

Troubleshooting – other known issues

- If you experience other issues, go to wiki at Known issues.

Note on preinstalled SW at FI

- Hosts `nymfe{03,05,06,08,10}` in PC hall are configured same as PCs in A219.
- You only need to run `pa211_setup` script to set path for Vagrant.

Installation of Vagrant and VirtualBox at own hardware

Recommended HW configuration

- 16 GB of RAM
- SSD drive with tens of GB free space

Linux and macOS users

1. Enable virtualization in BIOS.
2. Install VirtualBox.
   1. VirtualBox on Linux is sensitive to kernel versions. First, update the system (including the kernel), and only then install the latest Virtualbox. **IMPORTANT**: Don’t install the distro-repository version of VirtualBox. Really do install the latest version from https://www.virtualbox.org/wiki/Downloads.
   2. VirtualBox may requires x86 CPU architecture, so it may **not** work on ARM Mac.
3. Install Vagrant.

The official website should be preferred as a source. Repositories of Linux distributions could have outdated versions.

MS Windows users

1. Enable virtualization in BIOS.
2. Install VirtualBox.
3. Install Vagrant.
4. Ensure Hyper-V is disabled (Programs and Features > Turn Windows features on or off > Hyper-V)
   1. Sometimes it is not enough to disable Hyper-V in Settings; you may need to use the command `bcdedit /set hypervisorlaunchtype off` and restart the computer.
   2. Windows Update can turn Hyper-V on again, be sure to check it again after installing updates.

Let‘s install!

- We are here to help you with the process and hopefully solve issues we have already seen.

How was it today?

Please fill in an **anonymous** exit ticket:

https://muni.cz/go/pa211-22-01

MUNI

FI