

# Labtainers Instructor Guide

Fully provisioned cybersecurity labs

December 9, 2020



# 1 Introduction

This manual is intended for use by instructors who assign and/or grade labs using Labtainers.

Labtainers provide a consistent execution environment for performing laboratory exercises, and can include execution of several different computers interconnected via virtual networks. Refer to our published papers at <https://my.nps.edu/web/c3o/labtainers> for additional information on the use of Labtainers. And see the “Lab Designer User Guide” at <https://my.nps.edu/documents/107523844/109121513/labdesigner.pdf> for information on creating and maintaining Labtainer exercises.

The easiest way to get Labtainers is to use our pre-built VM available at the Labtainer website <https://my.nps.edu/web/c3o/virtual-machine-images>. Note that any Linux system can be used as long as it supports Docker, and it can be used on *Docker Desktop* installations on Mac and Windows computers as an alternative to using the VM appliance. If Labtainers is to be used on a system other than the pre-built VM, refer to the [Labtainer Student Guide](#) for information on installing Labtainers.

Running Labtainers on servers, e.g., cloud deployments is discussed in section [2.3](#)

## 1.1 Assigning a Lab

Student instructions for using Labtainers are in the [Labtainer Student Guide](#). Students work from the `labtainer-student` directory, i.e.,

```
cd ~/labtainer/trunk/scripts/labtainer-student
```

Available labs are listed via the `labtainer` script:

```
labtainer
```

Start a lab by providing its name as an argument to the `labtainer` command. This will typically display a link to a lab manual, or will display a lab manual in one of the resulting virtual terminals. You can interact with the resulting computers just as a student would.

## 1.2 Assessing a Lab

When the student stops a lab, i.e., using `stoplab`, Labtainers creates a zip file of student artifacts (including lab reports) and then displays the path to this zip file to the student. The easiest way for the student to forward this zip file to you is by starting a browser on the Linux VM and either emailing you the zip file, or uploading the file into an LMS, (e.g., Sakai). Alternately, the student can define a shared folder in the VM and copy the zip to the host computer.

Collect all of the lab zip files from each student into your Labtainer transfer directory, which is typically at

```
~/labtainer_xfer/<labname>
```

where `labname` is the name of the lab. Do not unzip the files. Alternately student assignments can be bulk-collected from a learning management system (LMS) per [Appendix A](#) and the resulting zip would be copied into the Labtainer transfer directory. Again, do not unzip files and do not change the file names of zip files.

Instructor assessment of labs takes place from the `labtainer-instructor` directory, i.e.,

```
cd ~/labtainer/trunk/scripts/labtainer-instructor
```

Use the `gradelab` command to assess results for a given lab:

```
gradelab <labname>
```

A table of lab results with one row per student and a column for each goal will be displayed. A description of the goals follows the table. A web-based display of that data is available as described in section 1.2.2.

Note that not all labs include automated assessment. For those labs, you will see this message:

```
No automated assessment for this lab
```

Even when no automated assessment is performed, you can still observe student performance artifacts, e.g., the `.bash_history` file as described below in 1.2.1.

Use the `-r` option to get perform a fresh grading, e.g., if you've removed files from the `labtainer_xfer` directory. Sometimes zip files within the `labtainer_xfer` directory are corrupted. If error messages indicate a bad zip file, try removing it from the directory and then use the `-r` option to perform a fresh grading. Use the `-u` option to update your `gradelab` to the latest image.

Student reports (if any) are copied into

```
~/labtainer_xfer/<labname>/docs
```

on the Linux host. If LMS assignment collection is used, then student reports should be looked for in

```
~/labtainer_xfer/<labname>/reports
```

which also includes reports separately uploaded into the LMS.

### 1.2.1 Review artifact details

You can view all student results, including their original artifacts by using the `-d` flag with the `gradelab` command. This results in a virtual terminal connected to a grading container that contains all student artifacts and results. If you have not first run the `gradelab` command without the `-d` option, run `instructor.py` from within the virtual terminal to cause the zip files to be extracted. A student's home directory can then be found in

```
<student_email>/<lab>.<container>.student
```

There you will find the `.bash_history` file along with the student-created files. Student artifacts collected by the framework are found in

```
<student_email>/<lab>.<container>.student/.local/result
```

The `-d` option is also used when debugging automated assessment configuration files. You can create additional virtual terminals into the grading container by reissuing the `gradelab` command with the `-a` flag. When you are finished, or wish to stop working, type:

```
stopgrade
```

### 1.2.2 Web-base view of student assessment

An early release of a web-based tool for viewing details of student assessment is available by use of the `-w` flag with the `gradelab` command. That causes the grader container to listen on port 8008 of the Labtainer VM. You can then open a browser on that VM and go to `localhost:8008`. Alternately, to use your host machines browser, you can set port forwarding on your VM, (e.g., in VirtualBox, use Machine / Settings / Network / Advanced / Port Forwarding to set host IP 127.0.0.1:8008 to map to guest ip 0.0.0.0:8008).

The table of goals displayed in the browser includes links that allow you to view details of artifacts created by the student when performing the lab. For example, clicking on the student name displays a table of all timestamped result artifacts. That page also includes a **History** heading with links to the `.bash.history` files one each container.

Links within each goal table cell lead to pages whose content depends on the type of goals defined. For example, a goal whose value is defined by a boolean expression will lead to a table of all boolean values for each timestamp for which results are present.

Definitions of different goal types and result types can be found in the *Lab Designer Guide*. Note that you need not understand all of the displayed data in order to gain useful insight into student progress. Some of the displayed information requires an understanding of the Labtainers automated assessment configuration directives, and is made available in the displays primarily in support of those developing automated assessment for labs.

## 2 Managing Labtainer Installations and Updates

Any given Labtainers installation can be brought up to date to the latest version by using the

```
update-labtainer.sh
```

command from the `labtainer-student` directory. The current version of a Labtainer installation is seen by using:

```
labtainer -v
```

The first time any given lab exercise is started, the latest version of that lab is automatically pulled from the Docker Hub registry. Note however that any given lab is not updated by the `update-labtainer.sh` command once the lab has been started. To update a specific lab to the latest version after it has been started the previous version of that lab must be deleted using:

```
removelab.py <labname>
```

The next time the lab is started, the latest version will be retrieved from the Docker registry.

If you want to update the `labtainer.grader` docker image (and delete the previous image and grader containers) use:

```
gradelab -u <labname>
```

### 2.1 Suggestions for student workflow

A student's work on any given lab is preserved until and unless the student restarts the lab using the `"-r"` option on the `labtainer <labname> -r` command. When taking a break from work on a lab, the student can either stop the lab using `stoplab`, or simply pause the VM. However, if the student wishes to perform other Labtainer-related work on the VM, (e.g., revisit a previous lab), they should first use `stoplab` for the current lab. When they restart the lab, none of their work will be lost.

## 2.2 Deploying without the Internet

Labtainers pulls Docker images from Docker Hub when a student first runs any given lab. You can deploy Labtainers within environments that have no Internet connection by first creating your own VM template. Start with the standard Labtainers VM, and run the script at

```
$LABTAINER_HOME/setup_scripts/pull_lab.py
```

to pull images for your desired labs onto the VM. Then replicate that VM for each user, e.g., by exporting it as an appliance.

## 2.3 Deploying on servers

Labtainers can be deployed on servers and accessed by students using a web browser. This assumes you have access to suitable infrastructure and IT support. Two general approaches are:

1. Virtual Desktop Infrastructure – Use VDI products such as VMWare Horizon to run Labtainer VMs. In these environments, each student is allocated a VM, and that VM's desktop is seen by the student in the browser. Students deliver their results to instructors by starting a browser within the VM, e.g., to access an LMS or web-mail account.
2. *Headless Labtainers* – Labtainers are deployed as servers in a cloud and a *NOVNC* desktop is rendered using a web browser. Access to the Labtainer server instance is via HTTP through an SSH tunnel. Please see <https://raw.githubusercontent.com/mfthomps/Labtainers/master/headless-lite/README.md> for additional information, including a sample cloud-config file.

## 3 Customizing Labtainers

Creating new labs and modifying existing labs is described in the *Labtainers Lab Designer User Guide*. <https://nps.box.com/shared/static/12n0y4yue49xcxtpv0jybgqiiskx8pr6.pdf>

That guide also describes how to provide your students with custom versions of the lab manuals, and how to publish new labs so that they can be incorporated into your student's Labtainers instances, and shared with other educators.

# A

## LMS Assignment Collection

### A.1 Sakai

In the Sakai Assignments section, select the “In / New” entry for the appropriate assignment. The resulting page should enumerate each student who has submitted an assignment. In the upper right, click the “Download All” link, and then click the “Student submission attachment(s)” option and click the “Download” button. Copy the resulting zip into the lab transfer directory on the Linux host, i.e.,

```
~/labtainer_xfer/<labname>
```

Do not unzip the file and do not change its file name. You can then run the `gradelab <labname>` command from the `labtainer-instructor` directory. In addition to the assessment summary, any student lab reports will be available in:

```
~/labtainer_xfer/<labname>/reports/<student name>
```

Those reports will include any that the student separately uploaded into Sakai (it is important to remind students to NOT change the name of lab report documents.)