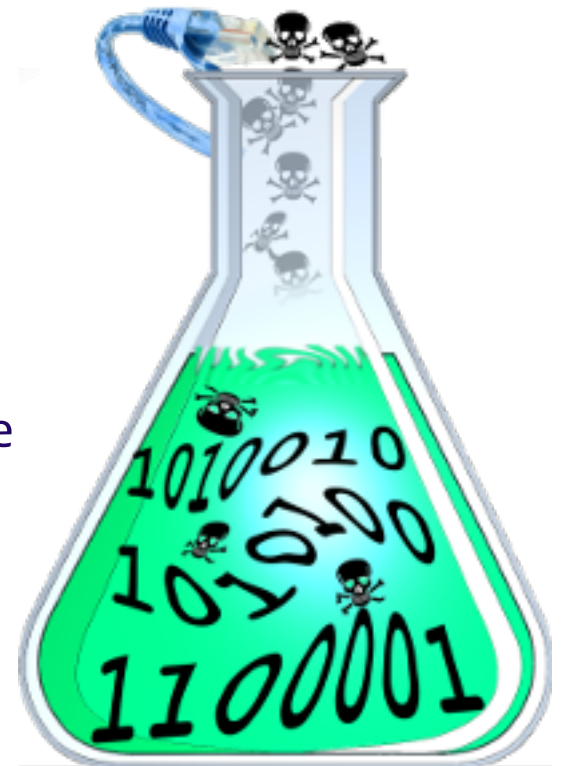




# Labtainers Cybersecurity Lab Exercises

CAE Tech Talk  
February 2022

Michael Thompson  
Department of Computer Science  
Naval Postgraduate School





## Overview – What is Labtainers?

- A collection of fully provisioned free cyber exercises
- All within a local VM or a VM on the cloud
- Docker containers for efficiency & provisioning
- With automated assessment of student progress
- Individualized labs to discourage sharing
- A framework and tools for creating and deploying new labs



## Experiential learning is desirable, but ...

- Institutional infrastructure may be absent
- Labs difficult to build and maintain
  - Overworked instructors need well-vetted labs
- Student platform diversity introduces problems
  - Different operating systems, libraries, software tools, etc.
  - Platform setup for lab distracts from learning objectives
  - Lab results vary widely due to configuration differences
- Experiential labs require exploration
  - How is this facilitated and observed?
- Students may share or reuse other work
  - Need individualized labs, but grading effort becomes large



# Labtainers Objectives

## Consistent and Fair

- Students execute labs in identical environments
- Instructors see consistent results and assess students on their work rather than environmental effects

## Parameterizable

- Labs configured so each student's work can be unique
- Labs are same level of difficulty for all students
- Uniqueness should not complicate assessment

## Support for Automatic Assessment

- Generate & collect artifacts from student work, e.g., program outputs
- Provide instructors with insight into student progress per lab goals

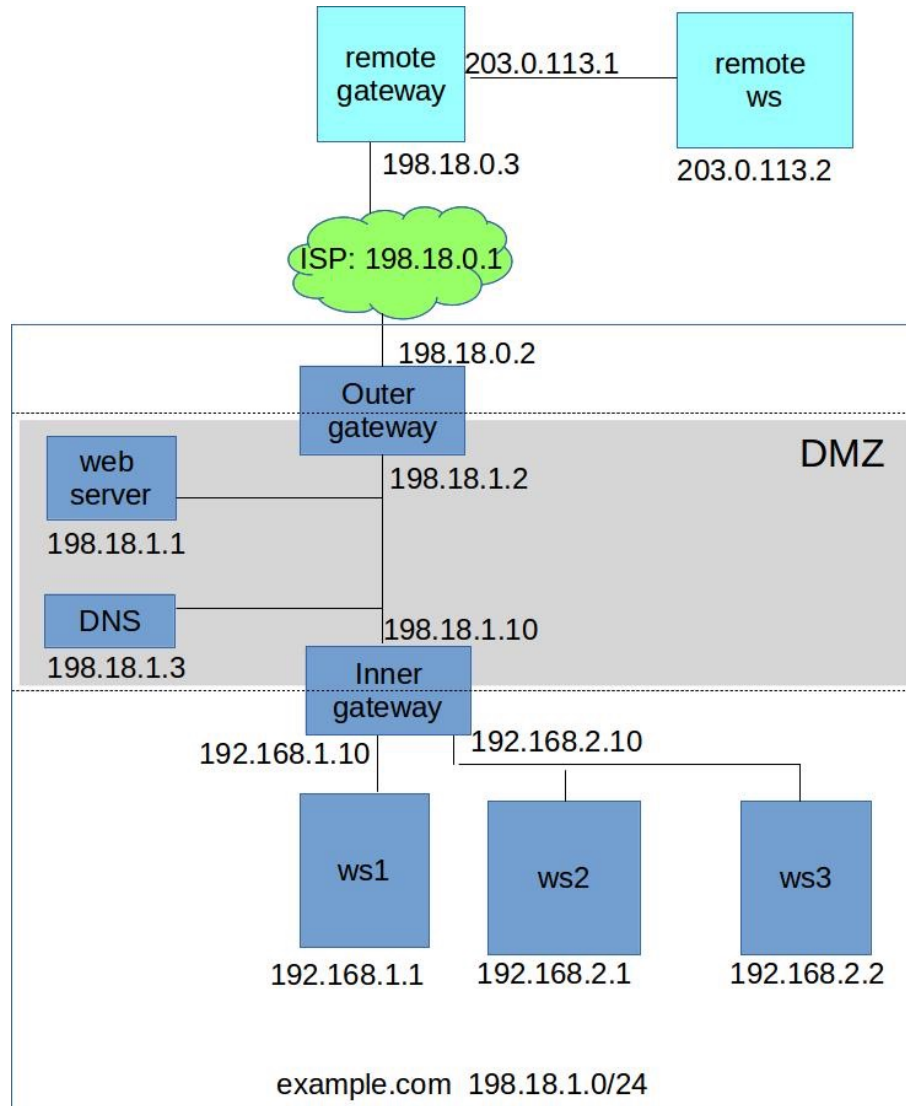


# Stand-alone Linux Cybersecurity Labs

- Multi-component network topologies
  - Packaged using Docker containers
  - Pre-configured execution environments
- Local to student's computer
  - One Linux host, (e.g., VM) runs many containers
  - No per-lab provisioning required by the student
- Cloud-based option, e.g., Azure and GCP
  - Free education accounts managed by student
  - Solves Mac M1 chip issue (won't run X86 VMs)
- Public repository of labs & open framework



# Run This Network on a Laptop





# Docker containers

- **Namespace isolation: like a VM but less overhead**
  - Commercial product with massive user base
  - Application vendors deploy for consistent environments
  - We break the model by running full Linux services
- **Student laptop can run many containers**
  - But may be bogged down by 2 or more VMs
  - Enables labs with many networked components
- **All containers share Linux kernel with host**
  - But can have distinct packages & library versions
  - Containers limited to Linux (mixed distributions)



# Automated Provisioning

- Student interacts with one Linux VM desktop
- Starting a lab pulls all necessary container images
- Containers created along with virtual networks
- Student sees multiple terminals and/or GUI apps
- Interacts with multiple computers
  - Each is fully provisioned
  - And instrumented to record results





# Parameterization

- Individualizes labs for each student (optional)
- Random number seed based on student ID
- Example: size of buffer to overflow
  - Symbolic replacement of value in source code
  - Vulnerable program compiled during first run
  - Affects offset of return address to overwrite
- *Individualizing Cybersecurity Lab Exercises with Labtainers*
  - <http://ieeexplore.ieee.org/document/8328979/>



# Automated Assessment

- Student activity and files collected as artifacts
  - Mostly transparent to students
  - *bash* hooks capture *stdin* & *stdout* into timestamped files
  - Artifacts forwarded to instructor
- Instructor tools assess student performance
  - Expected results as defined by lab designer
  - Insight into student progress and exploration
- Makes individualized labs practical



# Roles in the World of Labtainers

## Designer

SME who works with instructor to create labs based on learning objectives. Fine tunes and updates labs. May define assessment criteria.



## Instructor

Defines learning objectives. Works with (or is) designer. Ensures student readiness to perform labs and conducts assessments.



## Student

Performs lab exercise. Learns! Delivers results to instructor for assessment.



# Student Workflow

- Student installs a VM appliance or creates VM on the cloud
- Performs lab exercises as directed by instructor
- Artifacts are automatically collected into a zip file
- Student sends zip file to instructor, e.g., via an LMS



## Performing a Lab

```
student@ubuntu: ~/labtainer/trunk/scripts/labtainer-student
File Edit View Search Terminal Help
student@ubuntu:~/labtainer/trunk/scripts/labtainer-student$ labtainer telnetlab
pulling telnetlab.client.student from labtainers
Done with pull
pulling telnetlab.server.student from labtainers
Done with pull
Please enter your e-mail address: [dog@cat.com]
Starting the lab, this may take a moment...

The lab manual is at
file:///home/student/labtainer/trunk/labs/telnetlab/docs/telnet.pdf

You may open this manual by right clicking
and select "Open Link".

Press <enter> to start the lab
```



## Performing a Lab

```
ubuntu@client:~$  
  
ubuntu@server:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 02:42:ac:14:00:03  
          inet addr:172.20.0.3  Bcast:172.20.0.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:54 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:6865 (6.8 KB)  TX bytes:0 (0.0 B)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
ubuntu@server:~$
```



# Instructor Workflow

- Instructor collects zip files into a directory
- Runs grading tool and views table of student progress
- Optional browser-based deep dive into student activity
- Collected artifacts may include lab reports



# Automated Assessment

The screenshot shows a terminal window titled 'student@LabtainersVM: ~/labtainer/trunk/scripts/labtainer-instructor'. The user has run the command `gradelab telnetlab -wr`. The output includes a table of assessment results for various student domains and a list of actions performed during the assessment.

Student	telnetview	sshview	failed_login
alice_at_my.edu	Y	Y	Y
frank_at_beans.com	Y	Y	Y
mary_at_what.eh	Y	Y	Y
tony_at_here.there	Y	Y	Y

What is automatically assessed for this lab:  
failed\_login: Failed login as expected.  
telnetview: viewed file from telnet  
sshview: viewed file from ssh

Point your browser to <http://localhost:8008>




# Automated Assessment

Activities Firefox Web Browser Wed 14:30

localhost:8008/ localhost:8008

## Labtainers Gradelab for Lab: telnetlab



- [Table of Student Goals](#)
- [Lab Manual for telnetlab](#)
- [Labtainer Instructor Guide](#)
- [Lab Designer Guide](#)
- [Labtainers web page](#)

This is an early release version of the web-based student assessment tool. Please send feedback to Mike Thompson (mfthomps@nps.edu), or open a [GitHub issue](#)

1:35 / 2:06



# Automated Assessment

Activities Firefox Web Browser Wed 14:30

labtainer\_xfer telnetlab

localhost:8008/grades

localhost:8008/grades

[home](#)

## Student Assessment for Lab: telnetlab

Name	telnetview	sshview	failed_login
<a href="#">tony</a>	<a href="#">True</a>	<a href="#">True</a>	<a href="#">True</a>
<a href="#">mary</a>	<a href="#">True</a>	<a href="#">True</a>	<a href="#">True</a>
<a href="#">frank</a>	<a href="#">True</a>	<a href="#">True</a>	<a href="#">True</a>
<a href="#">alice</a>	<a href="#">True</a>	<a href="#">True</a>	<a href="#">True</a>

What is automatically assessed for this lab:  
failed\_login: Failed login as expected.  
telnetview: viewed file from telnet  
sshview: viewed file from ssh

[Goals configuration files](#)

1:38 / 2:06



## More than 50 Existing Labs

- Software vulnerabilities, e.g., buffer overflow
- Networking, e.g., arp-spoof, DNS-spoof, Snort
- Operations, e.g., ACLs, system logs
- Web, e.g., cross site scripting (OWASP IModules)
- Cryptography, e.g., hashing, VPNs
- Industrial control systems (PLCs)



# Find labs using keywords

```
student@LabtainersVM: ~/labtainer/labtainer-student
File Edit View Search Terminal Help
sql injection
ssh
ssh agent
ssl
symmetric keys
syn flood
syslog
tcp/ip
tcpdump
teleportation
telnet
traffic analysis
trojan horse
tshark
users
vpn
vpn gateway
vulnerability
wireshark
xsite
xsrif
student@LabtainersVM:~/labtainer/labtainer-student$ labtainer -f ssh
denyhost -- Explores the use of denyhosts to limit SSH login attempts from IP addresses.
ssh-agent -- Use of an SSH agent SSHing to remote computers without entering a passphrase for each access.
sshlab -- Use of a public/private key pair to access a server via ssh.
telnetlab -- Illustrates the telnet protocol transmission of plain-text passwords.
student@LabtainersVM:~/labtainer/labtainer-student$
```



# Finding labs with Labpacks

```
student@LabtainersVM: ~/labtainer/labtainer-student
File Edit View Search Terminal Help
ssh-agent -- Use of an SSH agent SSHing to remote computers without entering a passphrase for each access.
sshlab -- Use of a public/private key pair to access a server via ssh.
telnetlab -- Illustrates the telnet protocol transmission of plain-text passwords.
student@LabtainersVM:~/labtainer/labtainer-student$ labpack
List of installed Labpacks:
network-intro -- Introduction to basic network concepts
networks -- Network security labs
operations -- Computer security operations and administration
crypto -- Applied cryptography
access -- Authentication and access control
net-traffic -- Network traffic analysis
web-security -- Web security labs
vuln -- Software vulnerability analysis and exploitation
ics -- Industrial Control Systems / Operational Technology

usage: labpack [-h] [-a ADD] [-u] [name]

Track performance of labs in a Labpack

positional arguments:
  name                The Labpack to track.

optional arguments:
  -h, --help          show this help message and exit
  -a ADD, --add ADD  Get a Labpack from a URL.
  -u, --updates       Update Labpacks for this installation.
student@LabtainersVM:~/labtainer/labtainer-student$
```



## Lab content

- Each exercise has a lab manual for student reference
- Labs optionally support automated assessment (most do)
- Students can use automated assessment to check work
- Some lab have interactive quiz questions
- Most labs have “solution” scripts, contact me for those



## Create new labs

- Lab editing tool for GUI-based lab creation
- Provision each container in the lab
- Network definitions; configuration settings; etc.
- Creates a set of container images and config files
- Publish as “IModules” and share with others
- Or customize existing labs and or lab manuals



# Lab Editor

Labtainers



File Run Edit Help View

Lab: telnetlab

Lab running: No

Grader running: No

## Containers

Add

Copy from

client

server

## Networks

Add

SOME\_NETWORK 172.20.0.0/24

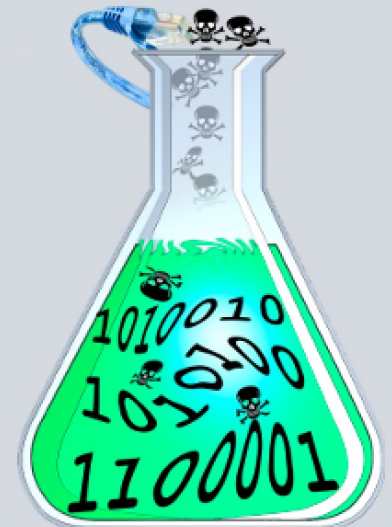
## Automated Assessment

Results

Goals

## Individualize

Parameters







# Configure a Container

Container Config: server

Edit

General Docker Hosts Other GNS3

Container: server Base: labtainer.network

User name:

Terminal quantity:

Password:

Terminal group:

Lab Gateway:

nameserver:

X11 enabled

No external gateway

No resolv.conf server

Add

Networks

Delete

OK Cancel

Labtainers

No  
ing: No

### Networks

Add

SOME\_NETWORK 172.20.0.0/24

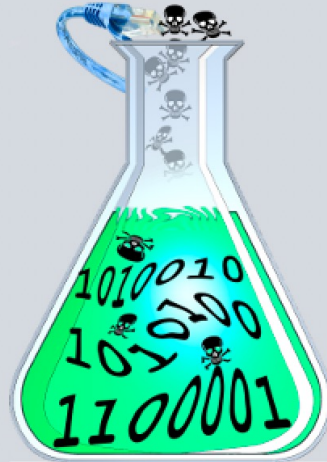
### Automated Assessment

Results

Goals

### Individualize

Parameters





# Add Packages in Dockerfile

File Edit View Search Terminal Help

```
ARG registry
FROM $registry/labtainer.network
ARG lab
ARG labdir
ARG imagedir
ARG apt_source
ARG user_name
#
#
ENV APT_SOURCE $apt_source
RUN /usr/bin/apt-source.sh
RUN apt-get update && apt-get install -y --no-install-recommends \
    telnetd

ADD $labdir/sys_$lab.tar.gz /

RUN useradd -ms /bin/bash $user_name
RUN echo "$user_name:$user_name" | chpasswd
RUN adduser $user_name sudo

USER $user_name
ENV HOME /home/$user_name
ADD $labdir/$lab.tar.gz $HOME

USER root
CMD ["/bin/bash", "-c", "exec /sbin/init --log-target=journal 3>&1"]
```

1,1

All



# Example Artifact Identification

Results for telnetlab

Create Remove All

	Result Tag	Container	File	Field Type	Field ID	Line Type	Line ID	
1	fileview	client	telnet.stdout	TOKEN	4	STARTSWITH	My string is:	^ Delete v Doc
2	sshfileview	client	ssh.stdout	TOKEN	4	STARTSWITH	My string is:	^ Delete v Doc
3	failed_login	server	/var/log/auth.log	CONTAINS	FAILED LOGIN	File		^ Delete v Doc

OK Cancel



# Resources

- Labtainers web page: <https://nps.edu/web/c30/labtainers>
  - Links to VM appliance
  - Student guide describes use of cloud based VMs
- Video tutorials on installation, use and assessment
- Student guide and Instructor guide
- Lab Designer Guide (how to build your own)
- IModules shared from other developers
- Published papers
- <https://github.com/mfthomps/Labtainers>
- Write me!



# Labtainers

[nps.edu/web/c30/labtainers](https://nps.edu/web/c30/labtainers)

## Contact

Cynthia Irvine [irvine@nps.edu](mailto:irvine@nps.edu)

Mike Thompson [mftomps@nps.edu](mailto:mftomps@nps.edu)

Department of Computer Science  
Naval Postgraduate School  
Monterey, CA 93943 U.S.A

The Labtainer framework and labs were developed with funding from the National Science Foundation and the DoD CySP program. Views expressed here do not necessarily represent those of NSF or the DoD. Many of the Labtainer exercises derive from the SEED project at Syracuse University, <http://www.cis.syr.edu/~wedu/seed/labs.html>.

CAE Tech Talk, Feb. 2022

