CHEESE Documentation

Release 0.1

CHEESE Team

Mar 02, 2021

CHEESE

CHEESE (Cyber Human Ecosystem of Engaged Security Education) is an NSF-funded initiative to develop a learning ecosystem for cybersecurity education. The project consists of the following:

- Easy to access, dynamic, web-based learning platform
- Community-contributed catalog of cybersecurity training materials
- Community-driven platform to request, develop, share, evaluate and learn about content
- Learning ecosystem that is continuously updated

CHAPTER 1

About

Cybersecurity is a highly dynamic field with newly discovered security attacks finding constant mention in news headlines. With the pervasive adoption of computing devices for activities ranging from social media to banking, travel and communication; cybersecurity is now vital in protecting personal and privileged information.

We propose to address gaps in cybersecurity training by catalyzing a broad collaborative effort of a community of cybersecurity researchers, educators, practitioners and students around open-source lessons leveraging containerized cybersecurity learning tools on a dynamic, publicly available, web-based learning platform. By presenting these learning tools exclusively through a web interface, we enable their use on a wide range of desktop operating systems. In addition to supplementing traditional cybersecurity instruction, our broader goal is to create a cybersecurity learning ecosystem that is continually updated with emerging trends in cybersecurity research as well as recently discovered security attacks.

As new security incidents are unearthed, a public resource that can provide a broad understanding of the issues involved is necessary for expanding public knowledge. By actively involving users in both the content and expansion of the platform, we can expand the reach of our platform, thus increasing its sustainability. The platform is also designed to be lightweight, portable, and flexible enough to leverage a multitude of deployment resources.

The CHEESE platform is intended for educators, students, practitioners and developers. Our emphasis on community-driven content ensures that users are actively engaged in the platform's expansion, promoting sustained use.

CHAPTER 2

Team

Baijian "Justin" Yang *Associate Professor, CIT, Purdue University*

Christine Kirkpatrick *Division Director of IT Systems and Services, San Diego Supercomputing Center*

Rajesh Kalyanam *Sr. Software Engineer, Research Computing, Purdue University*

Craig Willis *Sr. Research Programmer, National Center for Supercomputing Applications, University of Illinois at Urbana Champaign*

Mike Lambert *Research Programmer, Research Programmer, National Center for Supercomputing Applications, University of Illinois at Urbana Champaign*

CHAPTER 3

Our Approach

3.1 Open source platform

The CHEESE system builds on the Try-CybSI and National Data Service (NDS) Labs Workbench platforms. Try-CybSI provides the core containerized hands-on environments. The Labs Workbench platform provides a scalable and customizable framework for the deployment and management of containized applications.

3.2 Open source curriculum

The CHEESE curriculum will be modeled after the successful Carpentries framework. Lessons will be developed as open source modules that can easily be extended, incorporated into classroom instruction, workshops, and used for self-paced learning.

3.3 Community outreach

A goal of the CHEESE project is to develop a learning ecosystem that engages a broad community of cybersecurity educators, practitioners, and students to contribute to and benefit from the platform. The project team will engage a variety of partners from cybersecurity education communities and professional organizatinos.

3.4 Usability

The CHEESE project will evaluate the usability of the system through services provided by the Science Gateways Community Institute (SGCI).

3.5 Evaluation

The CHEESE project team will conduct formal evaluations of the content, workflow of content creation, and the impact of the provided materials on cybersecurity education through quantitative studies.

CHAPTER 4

Contributing

4.1 How to Contribute

You can contribute to the CHEESE initiative through any of the following:

- Suggest changes or fix problems by submitting an issue or creating a pull request.
- Contribute to existing hands-on scenarios or *Lessons*.
- Contribute a new hands-on scenario by developing Github repository defining Docker images and related content.
- Contribute a new lesson by developing a Github repository based on the Carpentry lesson template.
- Become a reviewer to review and approve lesson content and/or scenarios.

4.2 Hands-on Scenarios

Hands-on scenarios in CHEESEHub are implemented as one or more Docker images configured to run in Labs Workbench and demonstrate a particular technical concept.

The ARP spoofing example provides an illustration of hands-on scenario development.

Example: ArpSpoof

The ARP Spoofing scenario is a 3-component scenario with victim (client), server, and hacker containers. The victim container is a VNC-enabled Ubuntu environment with terminal, web browser, and network tools. The server is an Apache webserver with web-based terminal and example application. The hacker is a Jupyter notebook server including notebook with step-by-step instructions for performing an ARP poisoning attack.

Developing this scenario required the following:

- Github account
- Dockerhub account

- Creation of a Github repository containing the scenario materials. This includes at minimum a Dockerfile and README.md
- Automated build configuration on Dockerhub
- Creation of a set of JSON specifications in the CHEESE catalog defining each application in the scenario.

Once the scenario was developed, the author created a pull request on the CHEESE appliation catalog containing the JSON specifications, initiating the review process. The scenario was reviewed both technically and for security concepts.

Checklist

The following checklist can be used to guide scenario development. Each new scenario repository should have the following

- Dockerfile
- Top level README.md
- License (MIT or BSD-3 recommended)
- Scenario instructions either via Lessons or integrated Notebook

The README.md should have the following headings:

- Description of the scenario: What is the scenario?
- Target audience: Who is the scenario for? This may be boilerplate.
- Design and architecture: Describe why you chose the tools/methods you did and any limitations including diagrams.
- Installation and usage: At a minimum, include a link to CHEESEHub. Optionally, describe how to run directly in Docker and/or Kubernetes
- How to contribute: A link to the contributing guidelines.

4.3 Lessons

All hands-on scenarios will be accompanied by Carpentry-style lessons. Examples forthcoming.

4.4 Reviewers

The CHEESE community needs reviewers with expertise in each of the following:

- Technical aspects
- Security aspects
- Instructional/education aspects

CHAPTER 5

Developer's Guide

5.1 Local development

5.1.1 Using kubeadm-bootstrap

If you have access to an Ubuntu VM, the easiest approach to getting a development environment up is to use the `kubeadm-bootstrap` process. At this time, we've forked the original data8 repository to add support for Weave, which is required by CHEESE.

On your Ubuntu VM:

```
git clone https://github.com/nds-org/kubeadm-bootstrap
cd kubeadm-bootstrap
sudo ./install-kubeadm.bash
sudo -E ./init-master.bash weave
```

Next, clone and configure the Workbench Helm chart:

```
git clone https://github.com/nds-org/workbench-helm-chart
cd workbench-helm-chart
```

Generate a self-signed certificate:

```
./generate-self-signed-cert.sh dev.cheesehub.org
```

Customize the helm chart `values.yml`. See the Helm chart README for details. Change the `specs.repo` to:

```
specs:
  repo: "https://github.com/cheese-hub/catalog.git"
  branch: "master"
```

Workbench relies on labeled nodes. Label the node:

```
kubectl label nodes <nodename>  ndslabs-role-compute=true
```

Finally, install the chart:

```
helm install . --name=workbench --namespace=workbench
```

Once the chart is installed and services are running you should be able to register and login to your Workbench/CHEESEhub instance.

5.1.2 Using Minikube

Install and start Minikube based on the official documentation for your operating system.

```
minikube start
```

Install Helm based on the official documentation for your operating system.

Install the tiller service:

```
kubectl --namespace kube-system create sa tiller
kubectl create clusterrolebinding tiller --clusterrole cluster-admin --
↪serviceaccount=kube-system:tiller
helm init --service-account tiller
```

Install the NGINX load balancer based on the data-8 kubeadm-bootstrap repo:

```
git clone https://github.com/data-8/kubeadm-bootstrap.git
cd kubeadm-bootstrap/support && helm dep up && cd ..
helm install --name=support --namespace=support support/
```

Workbench requires wildcard DNS. You can either create entries in `/etc/hosts` as needed or install `dnsmasq`. The following instructions are for MacOS based on https://gist.github.com/petemcw/9265821:

```
brew up
brew install dnsmasq
```

Get the IP of your minikube instance:

```
minikube ip
```

Edit `/usr/local/etc/dnsmasq.conf` and add the following entry around line 80 replacing the value with your actual minikube IP:

```
address=/cheesehub.local/<minikube-ip>
```

Setup resolution for domain `cheesehub.local`:

```
$ sudo mkdir -p /etc/resolver
$ sudo tee /etc/resolver/cheesehub.local > /dev/null <<EOF
nameserver 127.0.0.1
domain cheesehub.local
search_order 1
EOF
```

Restart `dnsmasq`:

```
sudo launchctl stop homebrew.mxcl.dnsmasq
sudo launchctl start homebrew.mxcl.dnsmasq
```

Confirm DNS is working:

```
ping xyz.cheesehub.local
PING xyz.cheesehub.local (192.168.99.100): 56 data bytes
64 bytes from 192.168.99.100: icmp_seq=0 ttl=64 time=0.844 ms
```

Next, clone and configure the Workbench Helm chart:

```
git clone https://github.com/nds-org/workbench-helm-chart
cd workbench-helm-chart
```

Generate a self-signed certificate:

```
./generate-self-signed-cert.sh cheesehub.local
```

Customize the helm chart `values.yml`. At a minimum, change the `domain` to `cheesehub.local`, set `require_account_approval` to `false` and configure your SSL certs. See the Helm chart README for details. Change the `specs.repo` to:

```
specs:
  repo: "https://github.com/cheese-hub/catalog.git"
  branch: "master"
```

Workbench relies on labeled nodes. Label the node:

```
kubectl label nodes minikube  ndslabs-role-compute=true
```

Finally, install the chart:

```
helm install . --name=workbench --namespace=workbench
```

Once the chart is installed and services are running you should be able to register and login to your Workbench/CHEESEhub instance at https://www.cheesehub.local

5.2 Deploying on Jetstream

The CHEESE project uses cloud resources provided by NSF XSEDE Jetstream. This document describes how to configure a Jetstream project for use with CHEESEhub.

5.2.1 Apply for an allocation

Jestream allocations are available for research and education applications. See XSEDE Resource Allocation System for more information.

5.2.2 Jetstream Project Setup

CHEESE uses the Jetstream OpenStack user interface (UI) and API for platform deployment. New Jetstream allocations require a few preliminary steps to setup project network, subnet, and router. See the Jetstream's Setup+for+Horizon+API+User+Instances.

The basic steps include:

- Create network (named [project]-net)
- Create subnet
- Create router, attached to public network
- Add interface from router to project network
- Add security groups including remote SSH/HTTPS

Consider restricting SSH ingress to known CIDR ranges.

5.2.3 Upload Base OS Image

The CHEESE platform is currently based on Ubuntu LTS images. It is necessary to upload your own image to Jetstream. This can be done via the Horizon UI or via the OpenStack CLI.

Download the image from https://cloud-images.ubuntu.com/bionic/current/ and upload using the OpenStack CLI:

```
openstack image create --disk-format qcow2 --container-format bare \
          --file bionic-server-cloudimg-amd64.img "Ubuntu 18.04 LTS"
```

5.2.4 Create VM Instance

At this point you can create a VM instance based on the uploaded image and install CHEESEhub either as a single-node or multi-node installation.

5.2.5 Provision Kubernetes Cluster

CHEESEhub uses the kubeadm-terraform to provision Kubernetes clusters on OpenStack.

```
git clone https://github.com/nds-org/kubeadm-terraform
```

5.2.6 Setup Wildcard DNS

CHEESEhub requires wildcard DNS support for **.your.cheesehub.org*. If you do not have access to manage your own domain, contact us.

5.2.7 Setup Wildcard TLS

CHEESEHub requires a valid wildcard TLS certificate for **.your.cheesehub.org*. Free wildcard certificates are available from Let's Encrypt.

Follow these instructions to generate a valid certificate for your domain: https://opensource.ncsa.illinois.edu/confluence/display/NDS/Wildcard+Certs+via+LetsEncrypt

The certificate and key should be used in your Workbench configuration below.

5.2.8 NGINX Ingress Controller

The *kubeadm-terraform* installs an older version of the NGINX controller.

Delete the old controller:

```
helm delete --purge support
```

Create *values.yaml*:

```
controller:
  hostNetwork: true
  kind: DaemonSet
  extraArgs:
    default-ssl-certificate: workbench/ndslabs-tls-secret
  config:
    proxy-connect-timeout: "300"
    proxy-read-timeout: "300"
    proxy-send-imeout: "300"
    body-size: "64m"
    worker-shutdown-timeout: "900s"
```

Install the new controller:

```
sudo helm upgrade \
  --install support stable/nginx-ingress \
  --namespace=support \
  --version=1.17.0 \
  -f values.yaml
```

5.2.9 Install Workbench Helm Chart

CHEESEHub uses the following values:

```
name: "CHEESEHub"
domain: "hub.cheesehub.org"
support_email: <your email address>
repo: "https://github.com/cheese-hub/catalog.git"
cert: <See above>
key: <See above>
smtp.host: <Your smtp host>
smtp.port: <Your smtp port>
```

Install the Helm chart:

```
helm install . --name=workbench --namespace=workbench
```

5.2.10 Access your instance

Use *kubectl* to confirm your workbench instance is running:

```
kubectl get pods -n workbench
NAME                          READY     STATUS    RESTARTS   AGE
workbench-7cb876c6b5-tmf8m    4/4       Running   0          5h
```

Access your instance at https://www.your.cheesehub.org.

CHAPTER 6

Introduction

CHEESEHub is an easy-to-access web-based learning platform providing access to a dynamic, community contributed catalog of web-based hands-on learning environments for topics including network security, secure programming, and cryptography. Register to explore.

Features:

- Easy to access, dynamic, web-based learning platform<
- Community-contributed catalog of cybersecurity training materials
- Community-driven platform to request, develop, share, evaluate and learn about content
- Continuously updated

Example applications:

- ArpSpoof
- Heartbleed
- SQL Injection

CHAPTER 7

Lessons

Lesson	SourceSite		Maintainer(s)
Network security			Rajesh Kalyanam
Secure coding			Rajesh Kalyanam
Cryptography			Rajesh Kalyanam
Machine Learning			Rajesh Kalyanam

CHAPTER 8

User's Guide

- Quickstart
- User guide
- Contributing
- Adminsitering
- Installing

CHAPTER 9

Community

Placeholder for community bits such as

- Mailing list: https://groups.google.com/forum/#!forum/cheese-community
- Chat: https://gitter.im/cheese-hub/
- Github: https://github.com/cheese-hub/
- Contributor's guide

CHAPTER 10

Indices and tables

- genindex
- modindex
- search