NAVIGATING THE PLATFORM

TABLE OF CONTENTS

INTRODUCTION

PwnIoT is a Capture the Flag (CTF) web-based platform based on pwncollege[1], which itself is based on the CTFd platform[2]. The purpose of the platform is to allow users access to specific operating systems directly on the website, in order to complete coding or administrative challenges on those systems. PwnIoT in particular focuses its challenges on Internet of Things (IoT) devices.

LOGGING IN

The default username and password after installation will be "admin" and "admin", respectively.

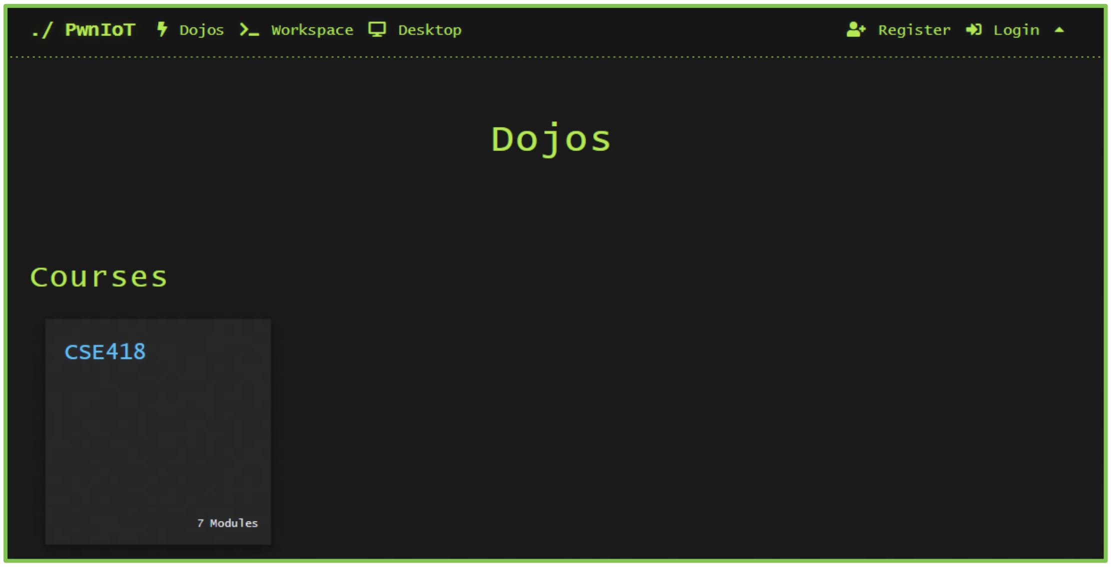
[1] https://pwn.college
[2] https://ctfd.io

NAVIGATING THE END-USER INTERFACE

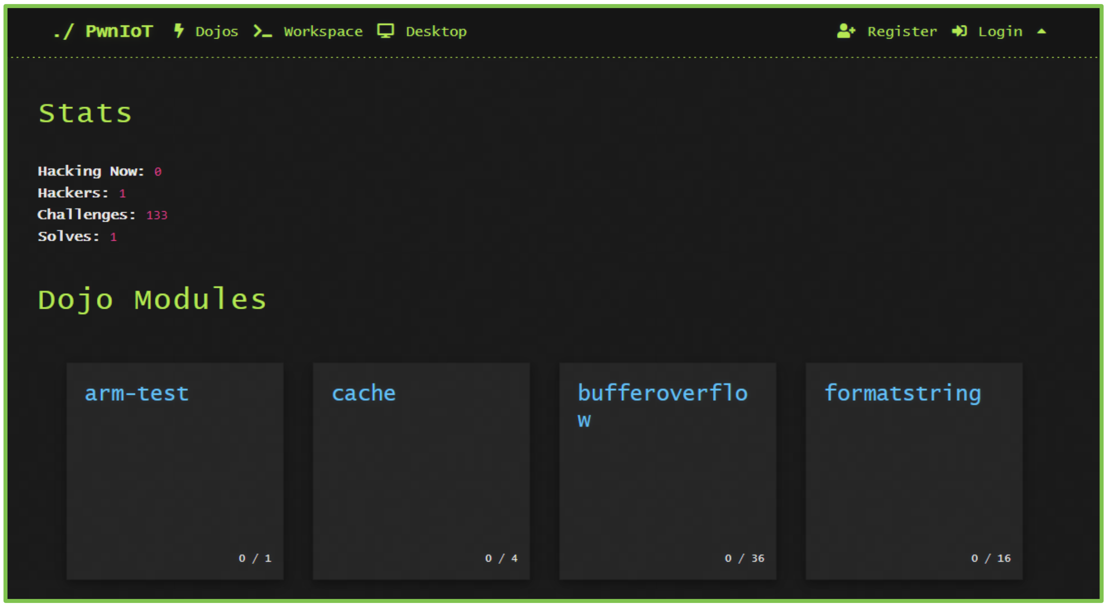
The live version can be accessed here: http://pwniot.cacti.academy

After local installation (see "Installing PwnIoT Locally" documentation), navigating to http://localhost will show the local version.

DOJOS & CHALLENGES

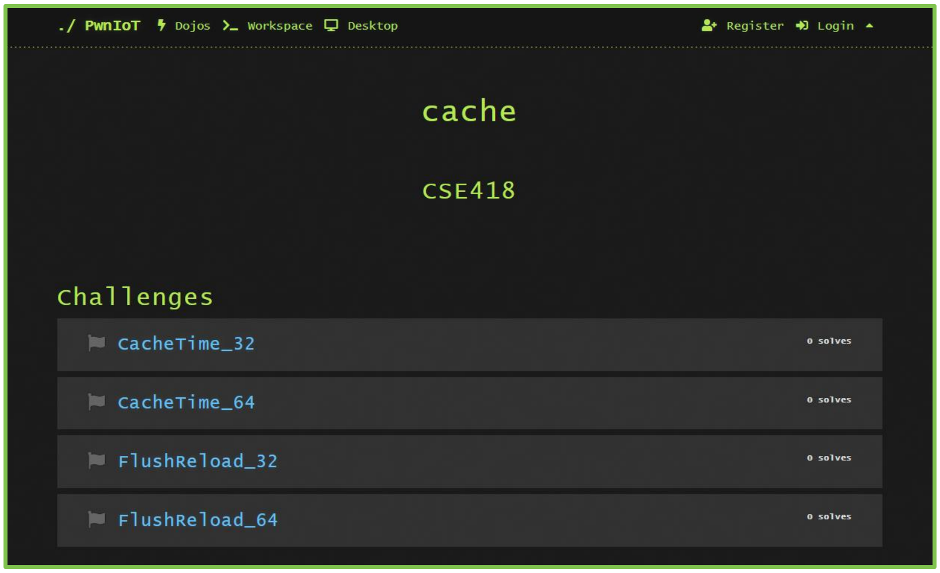


Dojos can be accessed by clicking the "Dojos" link at in the top menu bar.



Hacking Now: 0
Hackers: 1
Challenges: 133
Solves: 1

A dojo is a collection of challenge modules. Clicking on any dojo will show its modules.



./ PwnIoT   Dojos   Workspace   Desktop   Register   Login

cache

CSE418

Challenges

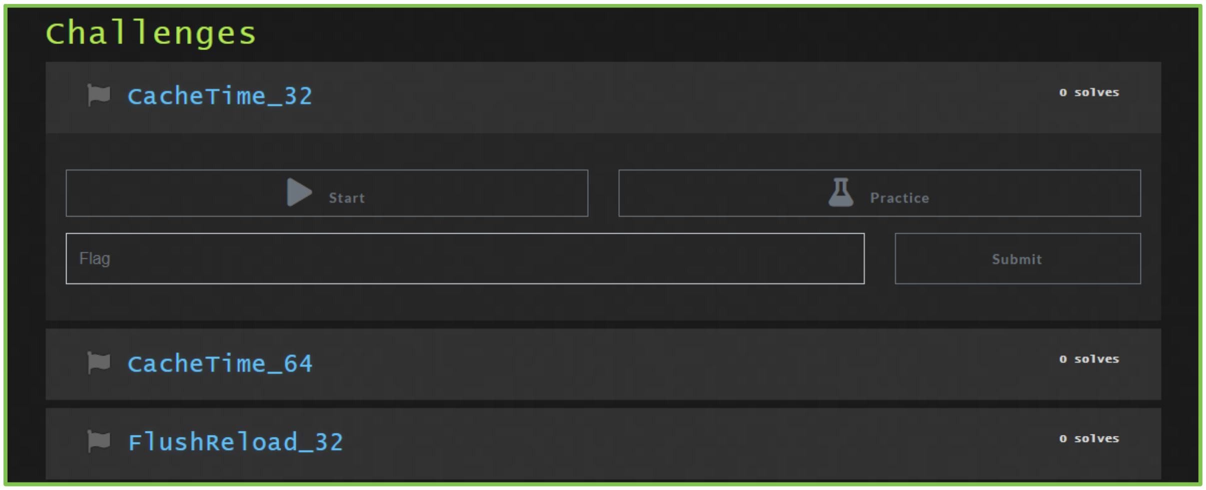
CacheTime_32   0 solves

CacheTime_64   0 solves

FlushReload_32   0 solves

FlushReload_64   0 solves

Likewise, each module is a collection of challenges. Clicking on any module will list its challenges.



Challenges

CacheTime_32   0 solves

Start   Practice

Flag   Submit

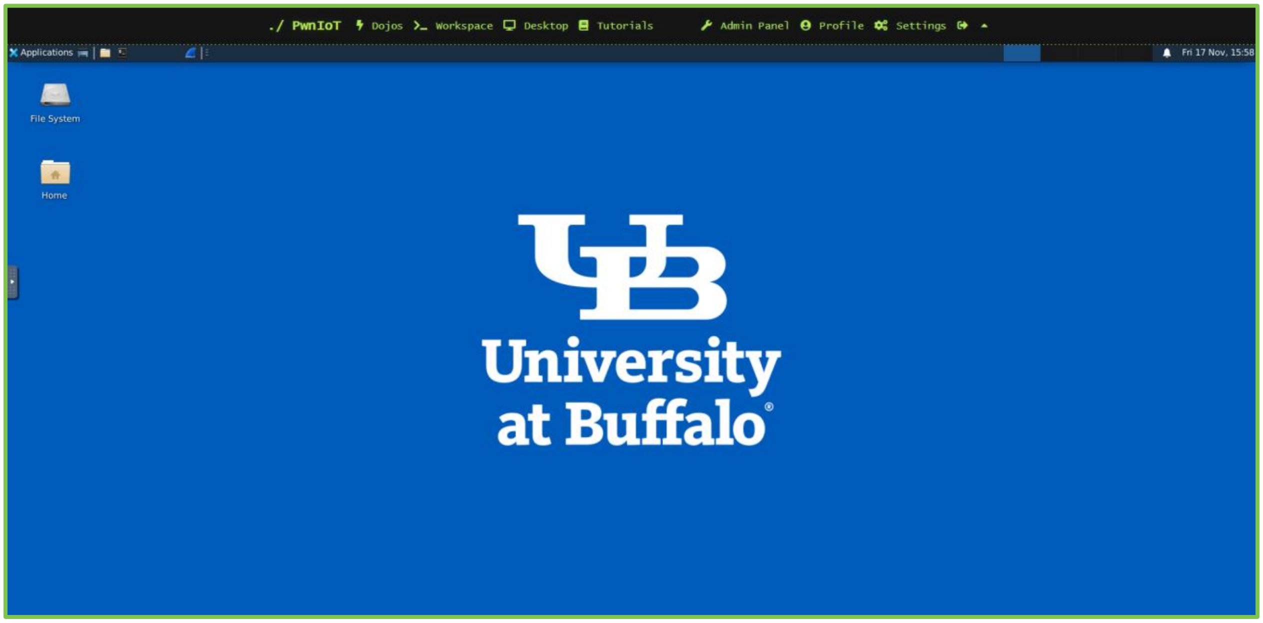
CacheTime_64   0 solves

FlushReload_32   0 solves

Clicking each challenge will display some options:

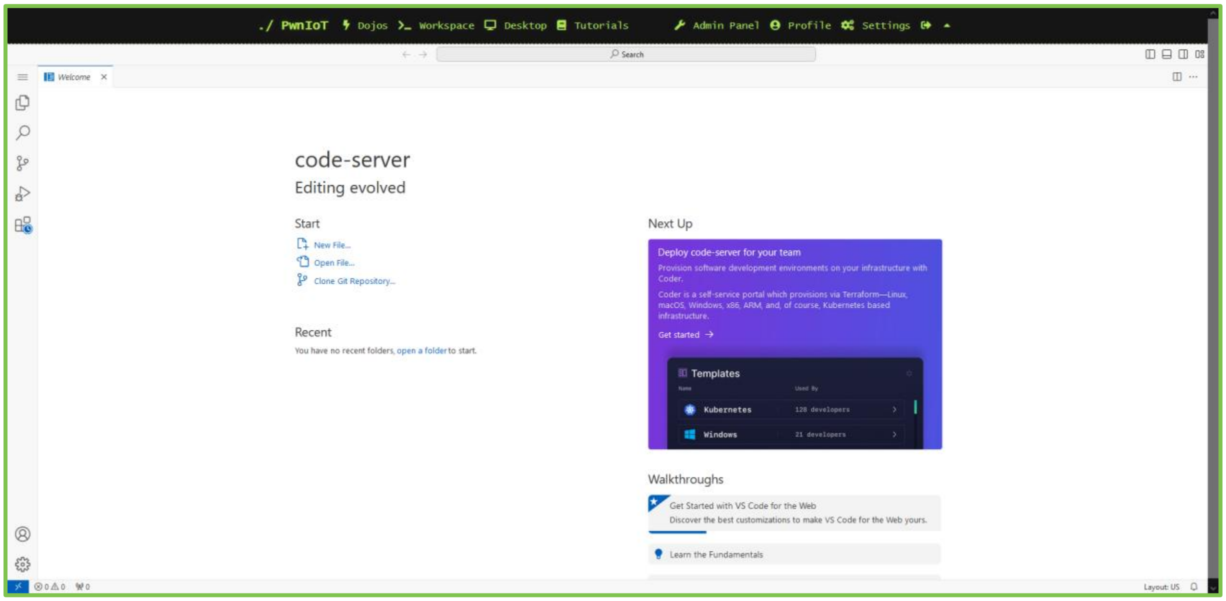
- **Start**: initiate the challenge, which will also start the challenge container. The user will have access to the container as an unprivileged user.
- **Practice**: this is the same as "Start", but the user will instead have privileged access to the container. This will allow them to run administrative commands that they otherwise would not be able to.
- **Flag**: when a challenge is completed successfully, the user is presented with a flag. Submitting the flag in this textbox will mark the challenge as complete.

CHALLENGE DESKTOP



When a challenge is started, the user may navigate to the “Desktop” link in the top menu bar. This will display the user’s current working container. It is here that the user can complete the challenge using a desktop environment. Note that the home directory of the user is shared across all challenges.

CHALLENGE WORKSPACE



code-server

Editing evolved

Recent

You have no recent folders, open a folder to start.

Deploy code-server for your team

Provision software development environments on your infrastructure with Coder.

Coder is a self-service portal which provisions via Terraform—Linux, macOS, Windows, x86, ARM, and, of course, Kubernetes based infrastructure.

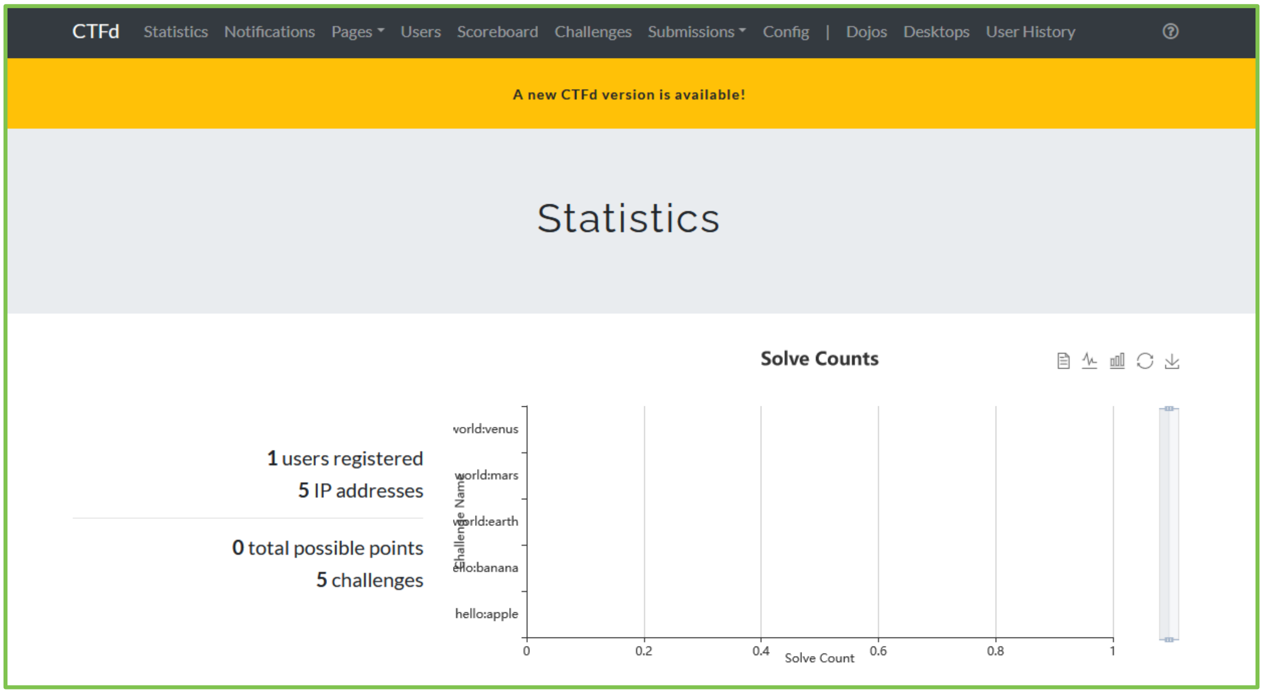
Get Started with VS Code for the Web

Discover the best customizations to make VS Code for the Web yours.

In addition to the desktop, clicking the “Workspace” link in the top menu bar will show the user an embedded VSCode instance that is connected to the challenge container, including via the terminal. This allows the user to complete the challenge without using the desktop environment.

NAVIGATING THE ADMIN PANEL

Log in as an administrator (see Logging In) by clicking "Login" in the top menu bar.



A new CTFd version is available!

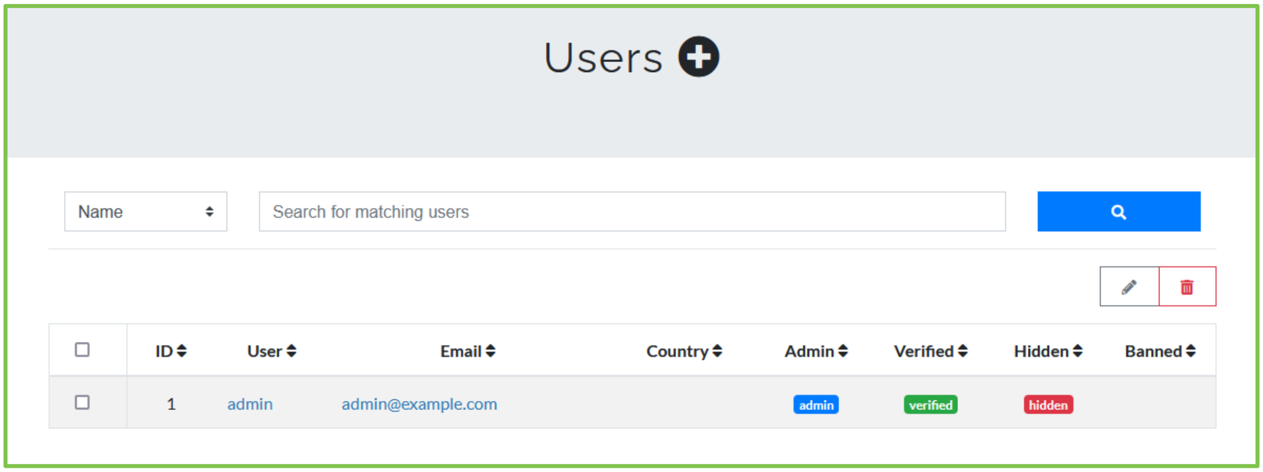
Statistics

1 users registered
5 IP addresses

0 total possible points
5 challenges

After logging in, click the "Admin Panel" link that is now available in the top menu bar. This will take you to the home page of the admin panel. The primary sections useful for development will be within the "Users", "Config", and "Desktop" pages.

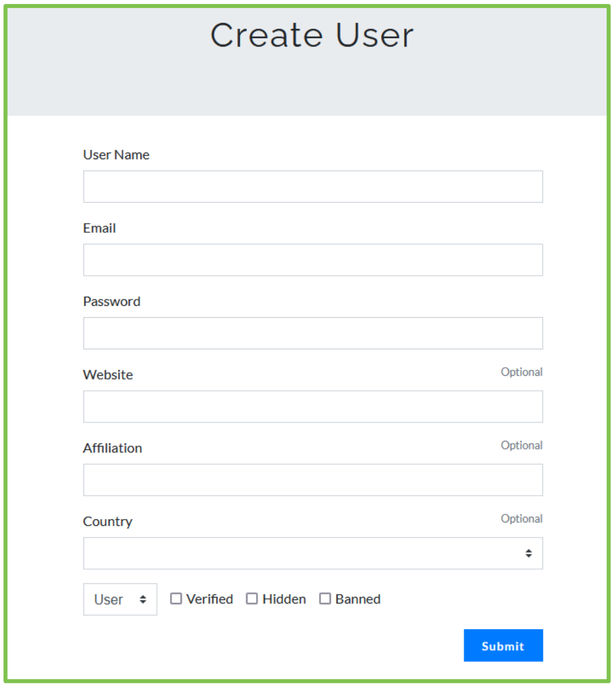
USERS



Users

□	ID	User	Email	Country	Admin	Verified	Hidden	Banned
□	1	admin	admin@example.com		admin	verified	hidden	

Clicking "Users" in the top bar will list all the users that are registered.



Create User

User Name

Email

Password

Website Optional

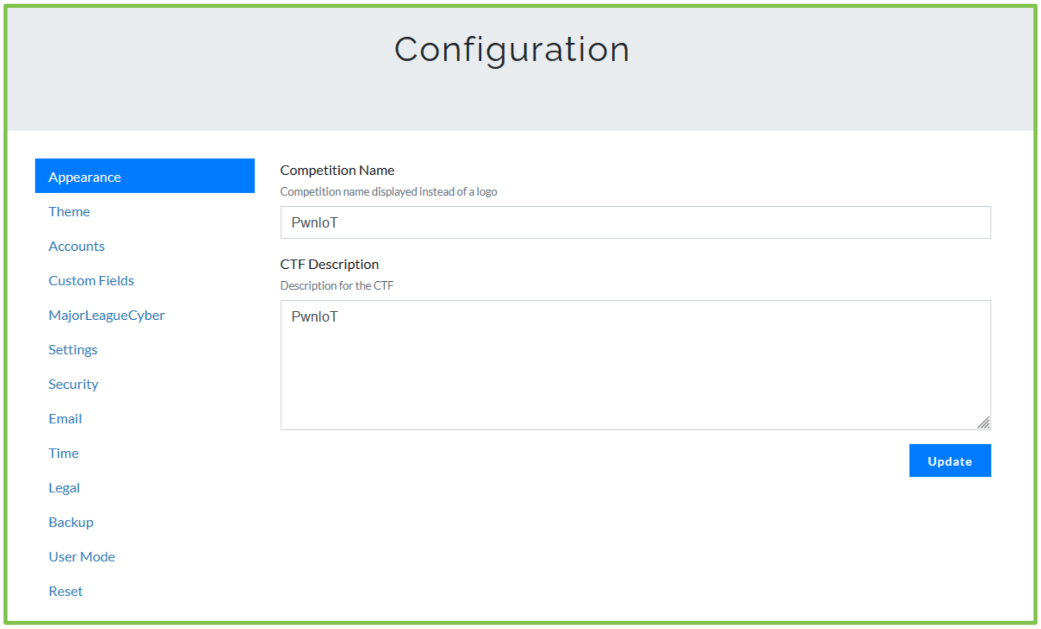
Affiliation Optional

Country Optional

User ☐ Verified ☐ Hidden ☐ Banned

Clicking the "+" icon will show the user-creation form. Creating new users can be important for testing. As well, the user may have their role set between "User" and "Admin"; an admin may add new challenges, which can be useful for additional testing.

CONFIG



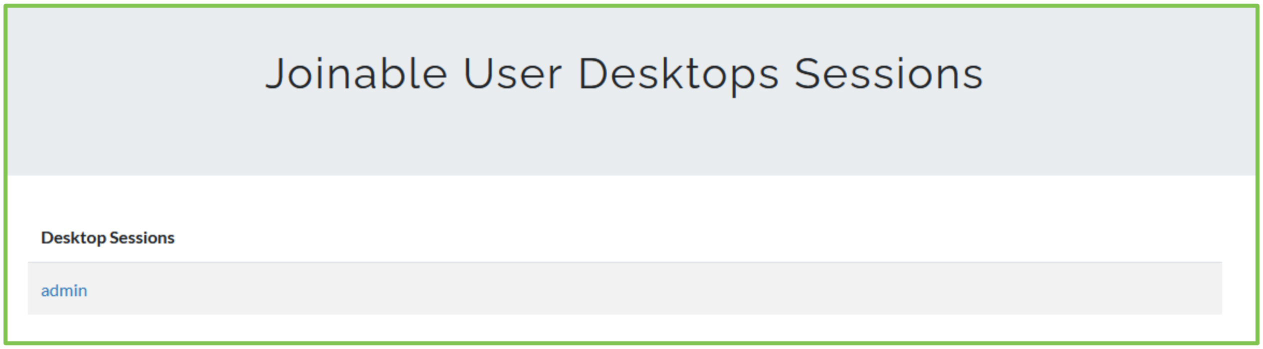
Configuration

Competition Name
Competition name displayed instead of a logo
PwnIoT

CTF Description
Description for the CTF
PwnIoT

Clicking on "Config" in the top menu bar will display all the site-wide settings for the whole platform.

DESKTOPS



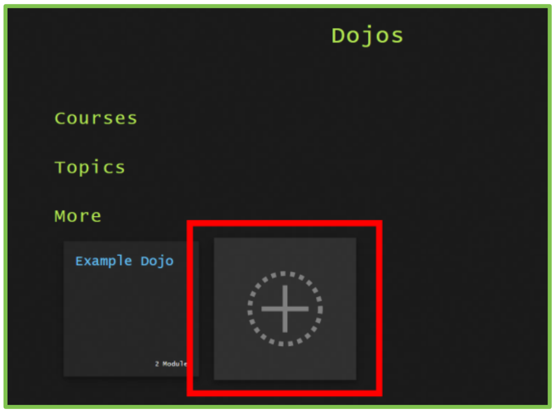
Joinable User Desktops Sessions

Desktop Sessions

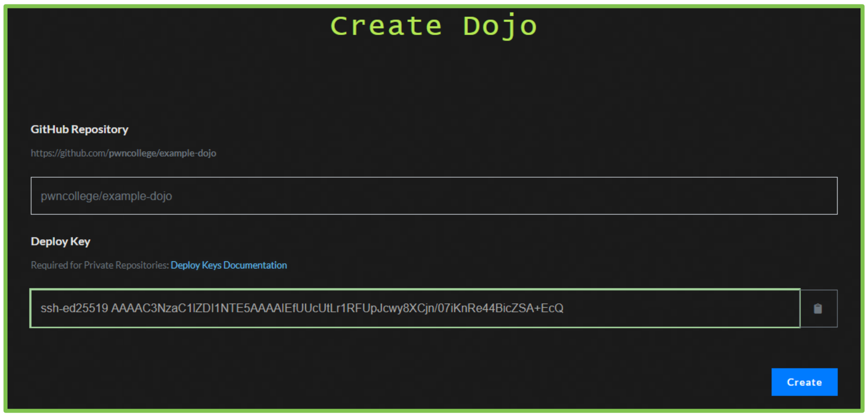
admin

Clicking on “Desktops” in the top menu bar will list all of the active challenge containers. Clicking on any one of them will log you into that container.

ADDING A CHALLENGE



As an administrator, viewing the “Dojos” page from the end-user panel (not the “Dojos” page within the admin panel) will show a “+” icon that will allow you to add a new challenge.



Create Dojo

GitHub Repository

https://github.com/pwncollege/example-dojo

pwncollege/example-dojo

Deploy Key

Required for Private Repositories: Deploy Keys Documentation

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEfUUcUtLr1RFUpJcwy8XCjn/07iKnRe44BicZSA+EcQ

Note: The default suggested input of “pwncollege/example-dojo” will fail to install, as the codebase for PwnIoT has diverged from the original pwncollege codebase. A working example can be installed with “mjkrooz/example-dojo”.

NAVIGATING THE CONTAINERS

DOCKER COMMANDS

The following Docker commands will be useful in navigating the containers.

- List the containers that are currently running:

  ```
  docker container ls
  ```

- List the images that were created that the containers are based off of:

  ```
  docker images
  ```

- Start and stop the root container, named “pwniot.academy”:

  ```
  docker container start pwniot.academy
  docker container stop pwniot.academy
  ```

- Enter a bash shell based on the name of the container:

  ```
  docker container exec -it <container_name> bash
  ```

  e.g.

  ```
  docker container exec -it pwniot.academy bash
  ```

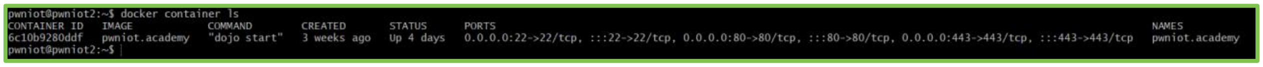
- Remove a container based on its name:

  ```
  docker container rm <container_name>
  ```

- Remove an image based on its name. This may fail if stopped containers are being used by it, but it will list those containers which can then be removed by the previous command:

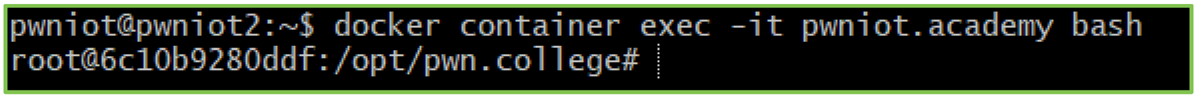
  ```
  docker rmi <image_name>
  ```

ROOT CONTAINER



```
pwniot@pwniot2:~$ docker container ls
CONTAINER ID   IMAGE            COMMAND        CREATED       STATUS      PORTS                                                                                      NAMES
6c10b9280ddf   pwniot.academy   "dojo start"   3 weeks ago   Up 4 days   0.0.0.0:22->22/tcp, :::22->22/tcp, 0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp   pwniot.academy
pwniot@pwniot2:~$
```

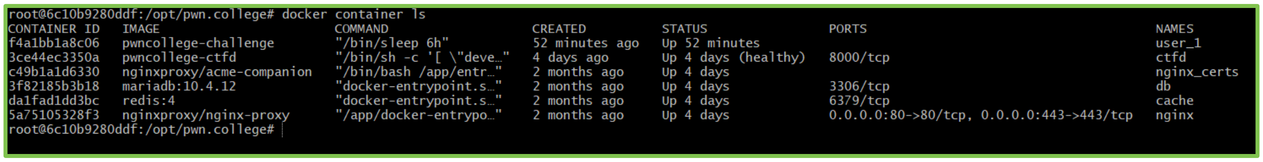
After installing locally, open a terminal. Use the `docker` command to list the containers to see that there will be a single docker container instance running.



```
pwniot@pwniot2:~$ docker container exec -it pwniot.academy bash
root@6c10b9280ddf:/opt/pwn.college#
```

A shell into the root container can be created and entered using the `exec` argument.

NESTED CONTAINERS



```
root@6c10b9280ddf:/opt/pwn.college# docker container ls
CONTAINER ID   IMAGE                        COMMAND                  CREATED          STATUS                PORTS                                      NAMES
f4a1bb1a8c06   pwncollege-challenge         "/bin/sleep 6h"          52 minutes ago   Up 52 minutes                                                    user_1
3ce44ec3350a   pwncollege-ctfd              "/bin/sh -c '[ \"deve…"   4 days ago       Up 4 days (healthy)   8000/tcp                                   ctfd
c49b1a1d6330   nginxproxy/acme-companion    "/bin/bash /app/entr…"   2 months ago     Up 4 days                                                        nginx_certs
3f82185b3b18   mariadb:10.4.12              "docker-entrypoint.s…"   2 months ago     Up 4 days             3306/tcp                                   db
da1fad1dd3bc   redis:4                      "docker-entrypoint.s…"   2 months ago     Up 4 days             6379/tcp                                   cache
5a75105328f3   nginxproxy/nginx-proxy       "/app/docker-entrypo…"   2 months ago     Up 4 days             0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp   nginx
root@6c10b9280ddf:/opt/pwn.college#
```

Within the root container are a set of additional containers that each serve a particular purpose:

- **ctfd**: this is the primary web server for the web interface. It includes the admin panel source code as well.
- **db**: the MySQL server that acts as the database for the web server.
- **cache**: a caching server running Redis.
- **nginx**: a reverse proxy based off of nginx-proxy[3] that serves as the primary entry-point to the web server.
- **nginx_certs**: an instance of Acme Companion[4] that accompanies nginx-proxy, and is used for SSL certificate generation.
- **"user_#"**: a currently-running challenge container for a particular user.

Each can be entered using their name. For example, entering the "db" container is done as follows:

```
docker exec -it db bash
```

[3] https://github.com/nginx-proxy/nginx-proxy
[4] https://hub.docker.com/r/nginxproxy/acme-companion