Application Target

> **CAUTION**
> Application Target Challenges are only available on Hosted or Self-Hosted Enterprise CTFd Instances

> **CAUTION**
> Application Target Challenges are heavily beta. If your challenge workload does not currently work please reach out to us.

Application Target challenges allow users to deploy a per-account instance of an application. For example if a challenge requires that every participant must have their own instance of a given website, Application Target challenges allow for that. Each Application Target challenge instance can be deployed, redeployed, and accessed by their respective owners. Note that any user with knowledge of the URL/hostname will be able to access the challenge instance.

Defining an Application

Application Target Challenges use a subset of the `docker-compose.yml` format.

For example, the following is an example of a valid docker-compose.yml file but also a valid application spec for an Application Target challenge:

```
version: "3.9"
services:
  nginx:
    image: nginx
    ports:
      - 80
  redis:
```

```
    image: redis
```

Ports

Application Targets will take challenges whose ports are defined in the spec and map public URLs to those ports. For example the port 80 specified for `nginx` will be mapped to a subdomain on `chals.io` for Hosted CTFd instances.

Services that are internal to the application should not have any ports listed in the spec.

Per-User Flags

Application Target challenges support generating a random per-user flag and adding it into the target specification. This allows Application Target challenges to implement a type of cheating detection system where certain instances of flag sharing can be detected.

> **CAUTION**
>
> Keep in mind that it is not possible for CTFd to detect or prevent all forms of cheating. For example, we cannot reliably prevent users from sharing solutions or solving challenges for each other.

To pass the per-user flag to your application target you would provide the `$FLAG` variable in the environment section of your target specification:

```
version: "3.9"
services:
  nginx:
    image: nginx
    environment:
      - FLAG=$FLAG
    ports:
      - 80
```

During deployment of this target the application will receive a unique per-user value for the `$FLAG` variable which can then be used by the application as the flag for the challenge. This value will be accepted by CTFd as the flag to solve the answer.

Should any user attempt to submit a per-user flag belonging to a different user it will be recorded by CTFd in the Anti-Cheat page in the Admin Panel.

> **CAUTION**
>
> Because the flag is provided to the target as an environment variable, your challenge must not allow competitors to read the environment variables via command execution or by reading `/proc/<pid>/environ`.

challenge.yml

In order to use application target challenges with the ctfcli `challenge.yml` format you should set the `type` to `application_target` and set the `application_spec` and `application_name` fields under the `extra` section.

For example:

```yaml
type: application_target

# The extra field provides additional fields for data during the install/sync commands/
# Fields in extra can be used to supply additional information for other challenge type
# For example the follow extra field is for dynamic challenges. To use these following
# extra fields, set the type to "dynamic" and uncomment the "extra" section below
extra:
  application_spec: |
    version: "3.9"
    services:
      nginx:
        protocol: https
        image: nginx
        ports:
            - "80"
  application_name: testing
```

Previous
« King of the Hill

Next
Overview »

Was this page helpful? 👍 👎

Share your feedback