

Report Generator

A lightweight script to compose an HTML Report

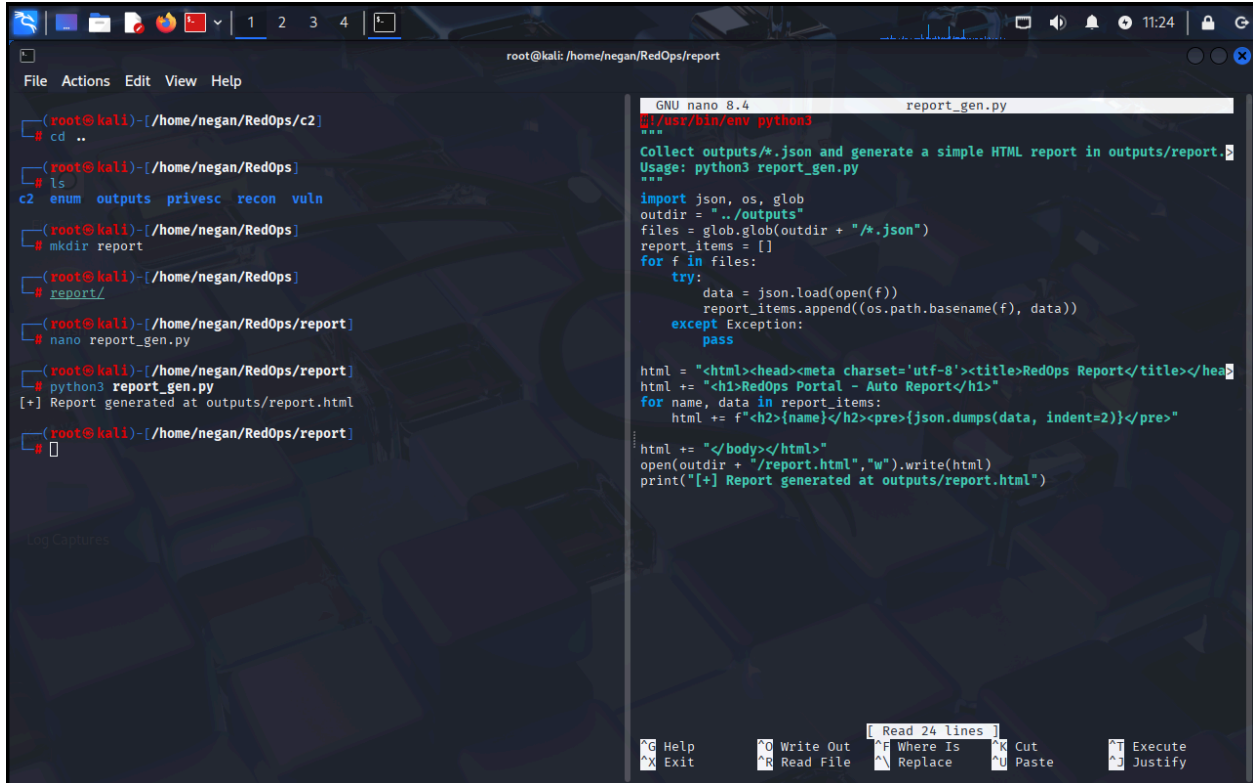
Python :

```
#!/usr/bin/env python3
"""
Collect outputs/*.json and generate a simple HTML report in outputs/report.html
Usage: python3 report_gen.py
"""
import json, os, glob
outdir = "../outputs"
files = glob.glob(outdir + "/*.json")
report_items = []
for f in files:
    try:
        data = json.load(open(f))
        report_items.append((os.path.basename(f), data))
    except Exception:
        pass

html = "<html><head><meta charset='utf-8'><title>RedOps Report</title></head><body>"
html += "<h1>RedOps Portal - Auto Report</h1>"
for name, data in report_items:
    html += f"<h2>{name}</h2><pre>{json.dumps(data, indent=2)}</pre>"

html += "</body></html>"
open(outdir + "/report.html", "w").write(html)
print("[+] Report generated at outputs/report.html")
```

Working :



The screenshot shows a Kali Linux terminal window with a dark theme. The terminal is running a series of commands to set up a directory and execute a Python script. The output of the script is displayed in the terminal. The script, named `report_gen.py`, is located in the `/home/negan/RedOps/report` directory. It uses the `python3` interpreter and the `report_gen.py` script to generate a report. The report is saved as `report.html` in the `outputs` directory. The terminal output shows the following commands and their results:

```
(root@kali)-[/home/negan/RedOps/c2]
# cd ..
(root@kali)-[/home/negan/RedOps]
# ls
c2  enum  outputs  privesc  recon  vuln
# mkdir report
(root@kali)-[/home/negan/RedOps]
# report/
(root@kali)-[/home/negan/RedOps/report]
# nano report_gen.py
(root@kali)-[/home/negan/RedOps/report]
# python3 report_gen.py
[+] Report generated at outputs/report.html
(root@kali)-[/home/negan/RedOps/report]
#
```

The Python script `report_gen.py` is as follows:

```
#!/usr/bin/env python3
"""
Collect outputs/*.json and generate a simple HTML report in outputs/report.
Usage: python3 report_gen.py
"""
import json, os, glob
outdir = "../outputs"
files = glob.glob(outdir + "/*.json")
report_items = []
for f in files:
    try:
        data = json.load(open(f))
        report_items.append((os.path.basename(f), data))
    except Exception:
        pass

html = "<html><head><meta charset='utf-8'><title>RedOps Report</title></head>"
html += "<h1>RedOps Portal - Auto Report</h1>"
for name, data in report_items:
    html += f"<h2>{name}</h2><pre>{json.dumps(data, indent=2)}</pre>"

html += "</body></html>"
open(outdir + "/report.html", "w").write(html)
print("[+] Report generated at outputs/report.html")
```

The terminal window also shows a menu at the bottom with the following options:

- Help
- Exit
- Write Out
- Read File
- Where Is
- Replace
- Cut
- Paste
- Execute
- Justify