

# HTTP\_ENUM

simple directory/filename discovery using a small wordlist

## Python Script :

```
#!/usr/bin/env python3
"""
HTTP enum - checks for common files/directories. Only uses urllib from stdlib.
Usage: python3 http_enum.py http://<target>[:port] [wordlist_file]
"""

import sys
from urllib import request, error
import os

TARGET = sys.argv[1] if len(sys.argv) > 1 else None
WL = sys.argv[2] if len(sys.argv) > 2 else "../recon/recon_wordlist.txt"

if not TARGET:
    print("Usage: python3 http_enum.py http://target[:port] [wordlist_file]")
    raise SystemExit(1)

if not os.path.exists(WL):
    print("Wordlist missing:", WL)
    raise SystemExit(1)

print("[+] HTTP enum against", TARGET)
found = []
for line in open(WL, 'r'):
    path = line.strip()
    if not path:
        continue
    url = TARGET.rstrip('/') + '/' + path
    try:
        req = request.Request(url, headers={"User-Agent": "RedOps-Portal"})
        resp = request.urlopen(req, timeout=3)
```

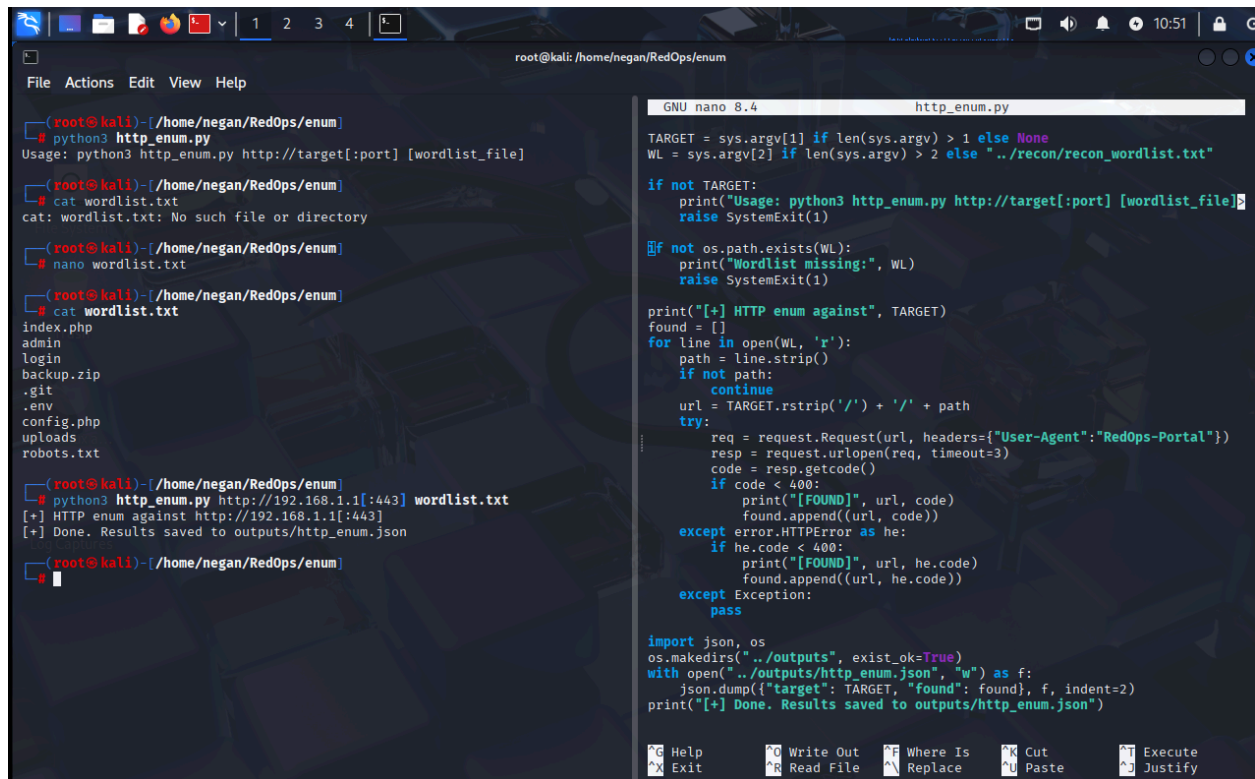
```
code = resp.getcode()
if code < 400:
    print("[FOUND]", url, code)
    found.append((url, code))
except error.HTTPError as he:
    if he.code < 400:
        print("[FOUND]", url, he.code)
        found.append((url, he.code))
except Exception:
    pass

import json, os
os.makedirs("../outputs", exist_ok=True)
with open("../outputs/http_enum.json", "w") as f:
    json.dump({"target": TARGET, "found": found}, f, indent=2)
print("[+] Done. Results saved to outputs/http_enum.json")
```

**Text file (wordlist.txt) :**

```
index.php
admin
login
backup.zip
.git
.env
config.php
uploads
robots.txt
```

Working :



The screenshot shows a Kali Linux terminal window with the nano text editor open. The terminal displays the following commands and output:

```
(root@kali)-[/home/negan/RedOps/enum]
# python3 http_enum.py
Usage: python3 http_enum.py http://target[:port] [wordlist_file]

(root@kali)-[/home/negan/RedOps/enum]
# cat wordlist.txt
cat: wordlist.txt: No such file or directory

(root@kali)-[/home/negan/RedOps/enum]
# nano wordlist.txt

(root@kali)-[/home/negan/RedOps/enum]
# cat wordlist.txt
index.php
admin
login
backup.zip
.git
.env
config.php
uploads
robots.txt

(root@kali)-[/home/negan/RedOps/enum]
# python3 http_enum.py http://192.168.1.1[:443] wordlist.txt
[+] HTTP enum against http://192.168.1.1[:443]
[+] Done. Results saved to outputs/http_enum.json

(root@kali)-[/home/negan/RedOps/enum]
#
```

The nano editor shows the source code of `http_enum.py`:

```
GNU nano 8.4 http_enum.py

TARGET = sys.argv[1] if len(sys.argv) > 1 else None
WL = sys.argv[2] if len(sys.argv) > 2 else "../recon/recon_wordlist.txt"

if not TARGET:
    print("Usage: python3 http_enum.py http://target[:port] [wordlist_file]")
    raise SystemExit(1)

if not os.path.exists(WL):
    print("Wordlist missing:", WL)
    raise SystemExit(1)

print("[+] HTTP enum against", TARGET)
found = []
for line in open(WL, 'r'):
    path = line.strip()
    if not path:
        continue
    url = TARGET.rstrip('/') + '/' + path
    try:
        req = request.Request(url, headers={"User-Agent": "RedOps-Portal"})
        resp = request.urlopen(req, timeout=3)
        code = resp.getcode()
        if code < 400:
            print("[FOUND]", url, code)
            found.append((url, code))
    except error.HTTPError as he:
        if he.code < 400:
            print("[FOUND]", url, he.code)
            found.append((url, he.code))
    except Exception:
        pass

import json, os
os.makedirs("../outputs", exist_ok=True)
with open("../outputs/http_enum.json", "w") as f:
    json.dump({"target": TARGET, "found": found}, f, indent=2)
print("[+] Done. Results saved to outputs/http_enum.json")

^G Help      ^O Write Out  ^F Where Is   ^X Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```