

# Banner Grabber

Banner grabber & basic SSH/HTTP header probe

Python Script :

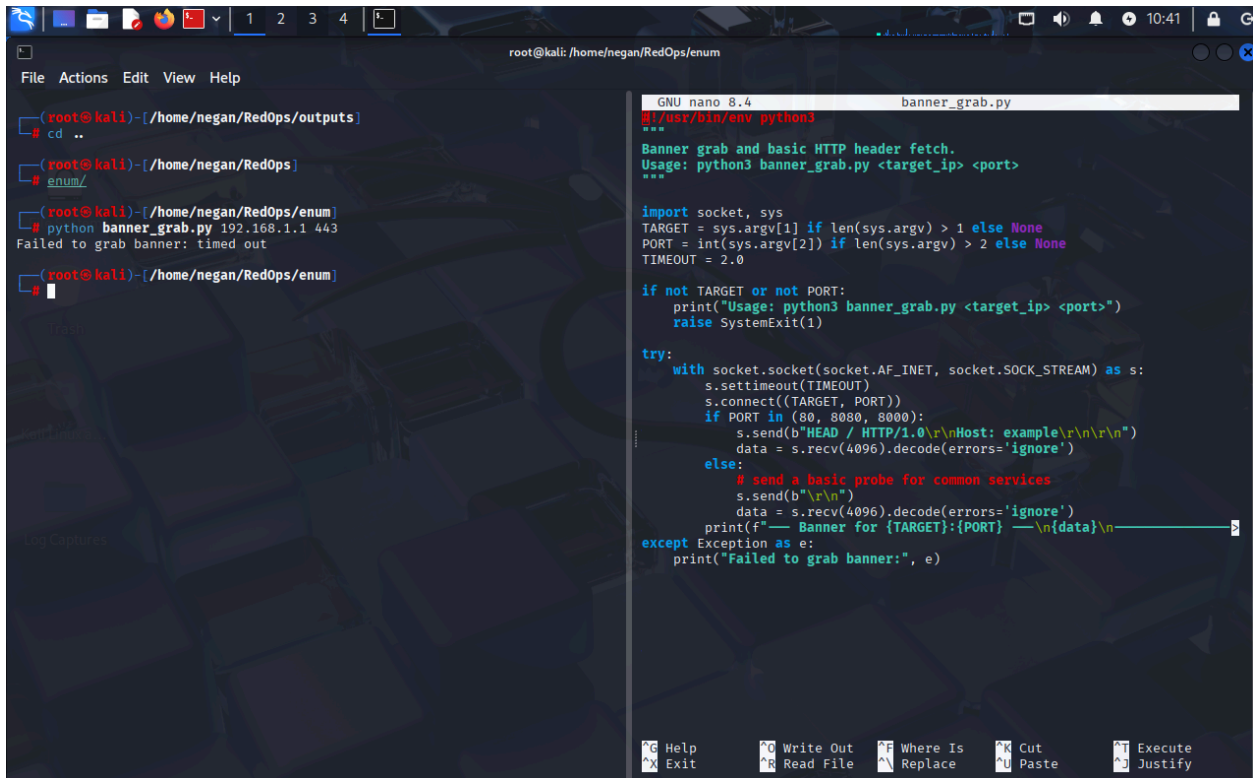
```
#!/usr/bin/env python3
"""
Banner grab and basic HTTP header fetch.
Usage: python3 banner_grab.py <target_ip> <port>
"""

import socket, sys
TARGET = sys.argv[1] if len(sys.argv) > 1 else None
PORT = int(sys.argv[2]) if len(sys.argv) > 2 else None
TIMEOUT = 2.0

if not TARGET or not PORT:
    print("Usage: python3 banner_grab.py <target_ip> <port>")
    raise SystemExit(1)

try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.settimeout(TIMEOUT)
        s.connect((TARGET, PORT))
        if PORT in (80, 8080, 8000):
            s.send(b"HEAD / HTTP/1.0\r\nHost: example\r\n\r\n")
            data = s.recv(4096).decode(errors='ignore')
        else:
            # send a basic probe for common services
            s.send(b"\r\n")
            data = s.recv(4096).decode(errors='ignore')
        print(f"--- Banner for {TARGET}:{PORT} ---\n{data}\n-----")
except Exception as e:
    print("Failed to grab banner:", e)
```

Working :



The screenshot displays a Kali Linux desktop environment with a terminal window and a nano text editor. The terminal window, titled 'root@kali: /home/negan/RedOps/enum', shows the following sequence of commands and output:

```
(root@kali)-[/home/negan/RedOps/outputs]
# cd ..

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

(root@kali)-[/home/negan/RedOps/enum]
# python banner_grab.py 192.168.1.1 443
Failed to grab banner: timed out

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

The nano editor, titled 'GNU nano 8.4 banner\_grab.py', shows the source code of the script:

```
#!/usr/bin/env python3
"""
Banner grab and basic HTTP header fetch.
Usage: python3 banner_grab.py <target_ip> <port>
"""

import socket, sys
TARGET = sys.argv[1] if len(sys.argv) > 1 else None
PORT = int(sys.argv[2]) if len(sys.argv) > 2 else None
TIMEOUT = 2.0

if not TARGET or not PORT:
    print("Usage: python3 banner_grab.py <target_ip> <port>")
    raise SystemExit(1)

try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.settimeout(TIMEOUT)
        s.connect((TARGET, PORT))
        if PORT in (80, 8080, 8000):
            s.send(b"HEAD / HTTP/1.0\r\nHost: example\r\n\r\n")
            data = s.recv(4096).decode(errors='ignore')
        else:
            # send a basic probe for common services
            s.send(b"\r\n")
            data = s.recv(4096).decode(errors='ignore')
        print(f"Banner for {TARGET}:{PORT} {data}")
except Exception as e:
    print("Failed to grab banner:", e)
```

The terminal window also shows a sidebar with icons for 'Log Captures' and 'Log Viewer'.