megaognzeyum is a netcore developed for web-based security frameworks, it is a sub-library of another project. web-based security and finding network vulnerabilities in all http-based network communications
and securing your server are our biggest needs
but you can use this python-based version for your daily operations.
terms of use:
AZENCOMPILER OPEN SOURCE FOUNDATION REASONS ARE AS FOLLOWS.
1: The software and software rights holders are not responsible in any way for illegal use.
2: When making any changes to the software, it can only be published subject to azcosf license rights.
3: All commercial rights of the software belong to azcosf, so if you want to use the source code of the software commercially, you need to get azcosf abnryzen commercial permission statement (warning: this statement means that you have established your partnership with azcosf)
4: The copyright and license rights of the software belong to azcosf
sample usage structure:

```
# example data checking

import megaognezyum

# Create an instance of RowanObject
rowan = megaognezyum.RowanObject(
    domain="example.com",
    ip_address="192.168.1.1",
    data="Sample data"
)

# Print the object's information
print("Printing RowanObject information:")
rowan.print_info()
```

The structure here is based on a new rowan object module and is intended to present and monitor all data information in the IP address. This structure is one of the basic object structures of the rowan library.

The structure here is based on a new rowan object module and is intended to present and monitor all data information in the IP address. This structure is one of the basic object structures of the rowan library. With this structure, it is possible to output data according to the specified domain IP address communication. you are wondering why is rowan called object rowan we had a brother named rowan he used to format people as objects and read their insides. I know it sounds scary when I say it like that but rowan was such a man and he developed himself like this.....
ANYWAY
rowan basic diagnostic system uses this structure for server vulnerability control:
```
# Log a security attempt
print("\nLogging a security attempt...")
rowan.log_security_attempt(
service_name="SSH",
attempt_type="Failed Login"
)
```

to test ssh server vulnerability or performance at login and other server interaction examples:

```
# List all security logs
print("\nListing security logs:")
logs = rowan.list_security_logs()
for log in logs:
    print(log)

# Track an HTTP link using the http submodule
print("\nTracking an HTTP link...")
megaognezyum.http.track_http_link(
    url="http://example.com/page",
    trigger="User Click"
)

# Start tracking SSL IPs using the pcap submodule for 10 seconds
```

```
print("\nTracking SSL IPs for 10 seconds...")
megaognezyum.pcap.monitor_ssl_ips(
    ip_address="192.168.1.1",
    duration=10
)
print("SSL IP tracking completed.")

# Start tracking SSH login attempts using the ssh submodule for 10 seconds
print("\nTracking SSH login attempts for 10 seconds...")
megaognezyum.ssh.track_ssh_attempts(
    ip_address="192.168.1.1",
    duration=10
)
print("SSH login attempt tracking completed.")

# Start tracking FTP activity using the ftp submodule for 10 seconds
print("\nTracking FTP activity for 10 seconds...")
megaognezyum.ftp.track_ftp_activity(
    ip_address="192.168.1.1",
    duration=10
)
print("FTP activity tracking completed.")

# Start monitoring general IP addresses using the pcap submodule for 10 seconds
print("\nMonitoring IP addresses for 10 seconds...")
megaognezyum.pcap.monitor_ip_addresses(
    ip_address="192.168.1.1",
    duration=10
)
print("IP address monitoring completed.")

# Optionally, read and display the logs

# Read and display the HTTPS IP tracking logs
print("\nHTTPS IP Tracking Logs:")
try:
    with open("https_ip_tracking.txt", "r") as f:
```

```
        for line in f:
            print(line.strip())
except FileNotFoundError:
    print("No HTTPS IP tracking logs found.")

# Read and display the SSH tracking logs
print("\nSSH Tracking Logs:")
try:
    with open("ssh_tracking.txt", "r") as f:
        for line in f:
            print(line.strip())
except FileNotFoundError:
    print("No SSH tracking logs found.")

# Read and display the FTP tracking logs
print("\nFTP Tracking Logs:")
try:
    with open("ftp_tracking.txt", "r") as f:
        for line in f:
            print(line.strip())
except FileNotFoundError:
    print("No FTP tracking logs found.")

# Read and display the General IP monitoring logs
print("\nGeneral IP Monitoring Logs:")
try:
    with open("general_ip_monitoring.txt", "r") as f:
        for line in f:
            print(line.strip())
except FileNotFoundError:
    print("No General IP monitoring logs found.")

# Read and display the HTTP link tracking logs
print("\nHTTP Link Tracking Logs:")
try:
    with open("http_link_tracking.txt", "r") as f:
        for line in f:
            print(line.strip())
except FileNotFoundError:
```

print("No HTTP link tracking logs found.")

```