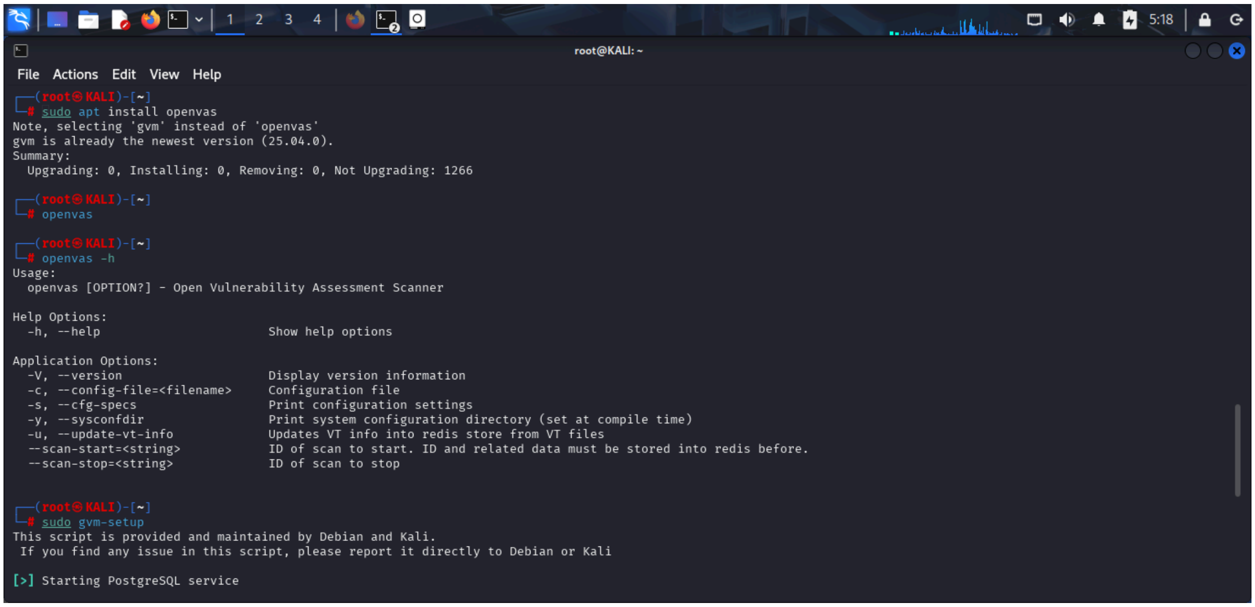
Lab 33: Network Vulnerability Scanning with OpenVAS (GVM)

Lab Number: 33
Tool Used: OpenVAS / Greenbone Vulnerability Manager (GVM)
Platform: Kali Linux
Objective: Learn to perform a quick vulnerability scan on a network using OpenVAS (GVM)

🧪 Task 1: Installing OpenVAS (GVM)

Command Used:

```
sudo apt install openvas
```

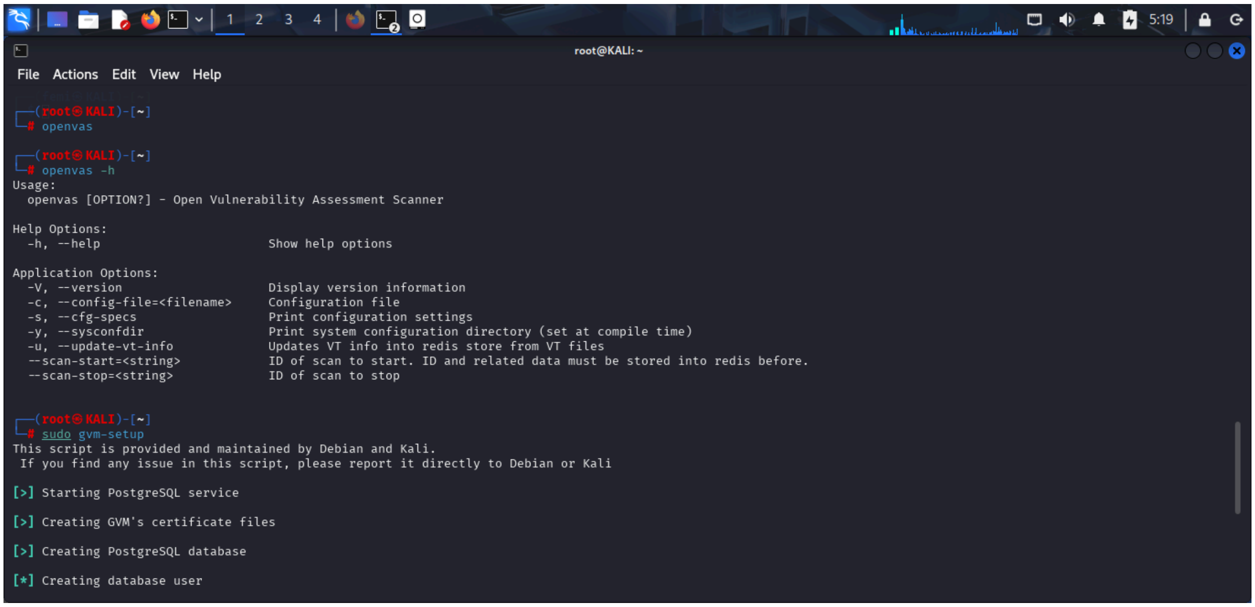


```
┌──(root㉿KALI)-[~]
└─# sudo apt install openvas
Note, selecting 'gvm' instead of 'openvas'
gvm is already the newest version (25.04.0).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1266

┌──(root㉿KALI)-[~]
└─# openvas

┌──(root㉿KALI)-[~]
└─# openvas -h
Usage:
  openvas [OPTION?] - Open Vulnerability Assessment Scanner

Help Options:
  -h, --help                       Show help options

Application Options:
  -V, --version                    Display version information
  -c, --config-file=<filename>     Configuration file
  -s, --cfg-specs                  Print configuration settings
  -y, --sysconfdir                 Print system configuration directory (set at compile time)
  -u, --update-vt-info             Updates VT info into redis store from VT files
  --scan-start=<string>            ID of scan to start. ID and related data must be stored into redis before.
  --scan-stop=<string>             ID of scan to stop


┌──(root㉿KALI)-[~]
└─# sudo gvm-setup
This script is provided and maintained by Debian and Kali.
 If you find any issue in this script, please report it directly to Debian or Kali

[>] Starting PostgreSQL service
```

-
- ✅ **Result:** Installed OpenVAS package successfully on Kali Linux.

Verification Commands:

```
openvas
openvas -h
```



```
┌──(root㉿KALI)-[~]
└─# openvas

┌──(root㉿KALI)-[~]
└─# openvas -h
Usage:
  openvas [OPTION?] - Open Vulnerability Assessment Scanner

Help Options:
  -h, --help                      Show help options

Application Options:
  -V, --version                   Display version information
  -c, --config-file=<filename>    Configuration file
  -s, --cfg-specs                 Print configuration settings
  -y, --sysconfdir                Print system configuration directory (set at compile time)
  -u, --update-vt-info            Updates VT info into redis store from VT files
  --scan-start=<string>           ID of scan to start. ID and related data must be stored into redis before.
  --scan-stop=<string>            ID of scan to stop


┌──(root㉿KALI)-[~]
└─# sudo gvm-setup
This script is provided and maintained by Debian and Kali.
 If you find any issue in this script, please report it directly to Debian or Kali

[>] Starting PostgreSQL service

[>] Creating GVM's certificate files

[>] Creating PostgreSQL database

[*] Creating database user
```

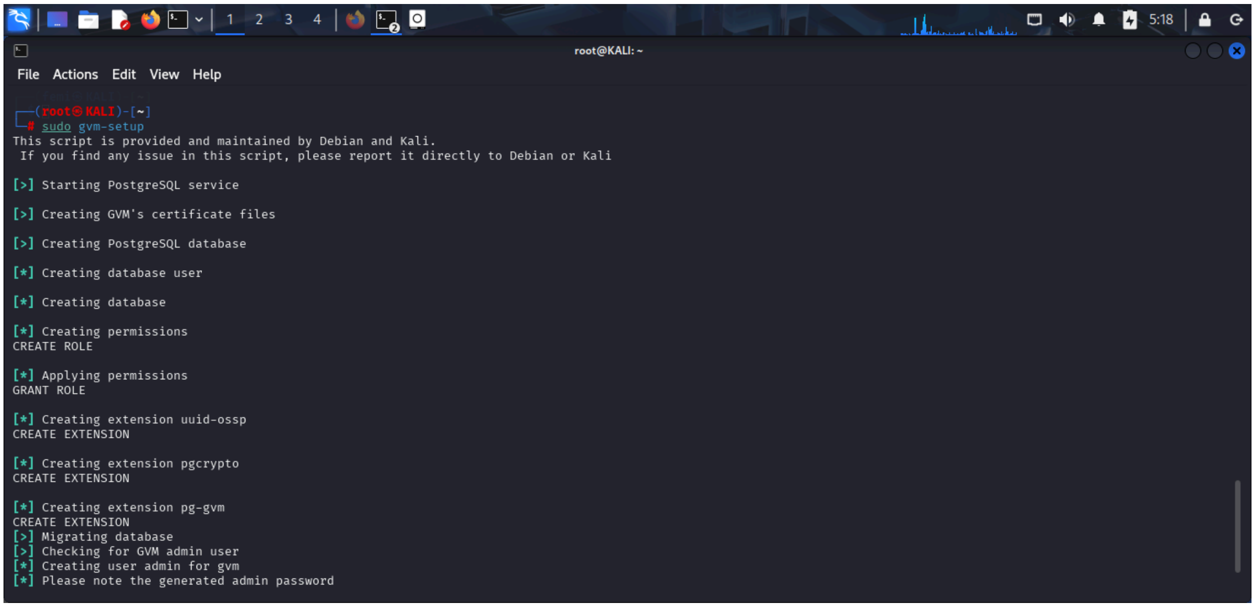
- ✅ **Result:** OpenVAS command recognized; help screen displayed successfully, confirming the installation.

🔁 Task 2: Setting Up GVM (Greenbone Vulnerability Manager)

- **Note:** OpenVAS is now called **Greenbone Vulnerability Manager (GVM)**.

Command Used:

```
sudo gvm-setup
```



```
┌──(root㉿KALI)-[~]
└─# sudo gvm-setup
This script is provided and maintained by Debian and Kali.
 If you find any issue in this script, please report it directly to Debian or Kali

[>] Starting PostgreSQL service

[>] Creating GVM's certificate files

[>] Creating PostgreSQL database

[*] Creating database user

[*] Creating database

[*] Creating permissions
CREATE ROLE

[*] Applying permissions
GRANT ROLE

[*] Creating extension uuid-ossp
CREATE EXTENSION

[*] Creating extension pgcrypto
CREATE EXTENSION

[*] Creating extension pg-gvm
CREATE EXTENSION
[>] Migrating database
[>] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
```

-
- 🕐 **Observation:** This process downloaded a large plugin database (~20–30 minutes).
- ✅ **Result:** Setup completed successfully.
  - A default **admin password** was generated on-screen.
  - **Action Taken:** Saved the admin password into a `.txt` file for later login.

Installation Check Command:

```
sudo gvm-check-setup
```

- ✅ **Result:** Displayed `OK` across all setup checks, confirming readiness.
- **Action Taken:** Rebooted Kali Linux to finalize setup.

🚀 Task 3: Starting GVM Web Interface

Command Used:

```
sudo gvm-start
```

-

✅ **Result:** Firefox browser launched automatically with:

```
https://127.0.0.1:9392
```

-
- **Action Taken:**
    - Accepted self-signed certificate warning.
    - Logged into the **Greenbone Web UI** using:
        - **Username:** `admin`
        - **Password:** (from earlier setup)
- ✅ **Result:** Successfully accessed the **Greenbone Vulnerability Manager Dashboard**.

🌐 Task 4: Running a Quick Vulnerability Scan

- **Action Taken:**
    - Navigated to **Scans** → **Tasks**.
    - Clicked the **wand icon** on top left and selected **Advanced Task Wizard**.
- **Scan Setup:**
    - Named the scan: `Home Network Quick Scan`
    - Entered target IP/subnet: `192.168.1.0/24` *(adjust based on actual network)*

    - Clicked **Create**
- ✅ **Result:**
    - The task was successfully created.
    - The vulnerability scan started automatically.
- 🕐 **Observation:** The scan took several minutes to complete.

📊 Task 5: Viewing Scan Results

- **Action Taken:**
    - Went to **Scans** → **Results** in the top menu.
- ✅ **Result:** Displayed a detailed list of vulnerabilities discovered in the scanned network.
- **Further Action:**
    - Clicked into individual results to read:
        - Vulnerability name
        - Affected host
        - Risk level (High, Medium, Low)
        - Suggested remediation steps

✅ Conclusion / Summary

This lab demonstrated how to install, configure, and operate **OpenVAS (GVM)** to perform vulnerability scanning on a network. After setting up the tool and accessing its web interface, a **quick scan** of the home network was launched. The scan results revealed potential vulnerabilities, complete with detailed remediation recommendations.

📚 Key Skills Learned:

- Installing and configuring OpenVAS/GVM
- Launching the web dashboard via `https://127.0.0.1:9392`
- Performing a subnet-based scan using the Advanced Task Wizard
- Interpreting vulnerability results and remediation steps