OpenVAS / GVM — Professional Cheat Sheet

> **One-page professional reference** for **OpenVAS (now GVM — Greenbone Vulnerability Manager)**. Quick commands, architecture, setup, scanning workflows, authenticated scans, tuning, reporting, and troubleshooting for vulnerability assessments.

---

1) At-a-glance

- **Toolset:** OpenVAS is the scanner component of the Greenbone Vulnerability Management (GVM) suite. Common components: `gvmd` (manager), `gsad` (web UI / Greenbone Security Assistant), `openvas-scanner` or `gvmd-scanner`, `gvm-tools` (gvm-cli), and Feed (NVTs) updates.
- **Primary uses:** Network vulnerability scanning, authenticated checks, configuration audits, scheduled scans, and reporting.
- **Note:** terminology changed: *OpenVAS* historically; modern upstream is **GVM** (Greenbone). Use your distro's package naming (e.g., `gvm`, `openvas`).

---

2) Quick install (common Linux distros)

```
# Kali (has gvm package)
sudo apt update && sudo apt install gvm
# or (Ubuntu/Debian)
sudo apt update && sudo apt install openvas gvm gvm-tools

# Source build (advanced / when packaging lagging)
# Follow official Greenbone docs: https://greenbone.github.io/gvm
```

After install run initial setup (flush feeds, create admin user, sync NVTs):

```
sudo gvm-setup        # (Kali / packaged helper - may be distro-specific)
# or manual sequence (example)
sudo runuser -u _gvm -- gvmd --create-user=admin --password='P@ssw0rd'
sudo runuser -u _gvm -- gvmd --create-scanner="OpenVAS Scanner" --scanner-host=/
var/run/ospd/ospd-openvas.sock
sudo gvm-feed-update  # update NVTs, SCAP, CERT data
```

---

3) Core services & files

- `gvmd` — GVM manager (stores tasks, users, results).

- `ospd-openvas` / `openvas-scanner` — scanner daemon (executes NVTs).
- `gsad` — web UI (Greenbone Security Assistant) accessible via HTTPS.
- `gvm-cli` / `gvm-tools` — scriptable CLI for automation (XML over SSH or UNIX socket).
- Feed directories: `/var/lib/gvm/feeds/` (varies by distro).
- DB: PostgreSQL database used by `gvmd` (stores results, configs).

---

4) Common admin & CLI commands

```
# Check setup (Kali helper)
sudo gvm-check-setup

# Start/stop services (systemd)
sudo systemctl start ospd-openvas gvmd gsad
sudo systemctl enable ospd-openvas gvmd gsad
sudo systemctl status gvmd ospd-openvas gsad

# Update feeds
sudo gvm-feed-update   # may run a few minutes

# Create admin user (if needed)
sudo runuser -u _gvm -- gvmd --create-user=admin --password='StrongP@ss'

# List users / tasks via gvmd
sudo runuser -u _gvm -- gvmd --get-users
sudo runuser -u _gvm -- gvmd --get-tasks

# CLI automation (example: list targets)
gvm-cli socket --xml '<get_targets/>'  # requires gvm-tools and permissions
```

---

5) Basic scanning workflow (GUI)

1. **Update feeds** (NVTs/SCAP/CERT) before scans.
2. **Create target**: IP or range, credentials (SSH, SMB, SNMP, WMI), port lists, alive tests.
3. **Create credentials**: add SSH/Windows/DB credentials for authenticated checks.
4. **Create scan config**: choose Full and fast, Full and deep, or a custom policy.
5. **Create task**: select target + config, set schedule and overrides (credentials, ssh keys).
6. **Run task** and monitor.
7. **Export report**: PDF/HTML/CSV/JSON for delivery.

---

6) Authenticated scans (best practice)

- **Linux/Unix:** add SSH credentials (username + private key or password) and enable `sudo -l` or `sudo su -` depending on checks.
- **Windows:** use WMI/SMB credentials (domain\user or user@domain); prefer domain accounts with least privilege required for checks.
- **Databases / web apps:** supply DB credentials for deeper checks (e.g., MySQL, MSSQL). Use separate credential entries per target or credential collection.
- **Testing approach:** always verify credentials manually (e.g., `ssh user@target`) before starting large authenticated scans.

---

7) Scan tuning & policies

- Use `Full and fast` for broad coverage; `Full and very deep` or custom for deep checks.
- **Adjust NVT families**: include/exclude families to reduce noise or target specific vectors.
- **Timeouts & port lists**: use custom port lists and timeouts for large networks to avoid long hangs.
- **Safe checks only**: enable "Safe Checks" to avoid disruptive tests when agreed with client.

---

8) Reporting & exporting

- **Export formats:** PDF, HTML, CSV, XML, TXT, JSON. Use CSV/JSON for ingestion into ticketing or reporting tools.
- **Risk filtering:** when exporting, filter by severity (High/Critical) to produce exec summaries.
- **Evidence:** attach raw HTTP responses or plugin output as needed for verification.

---

9) Automation (gvm-tools / gvm-cli)

- Install `gvm-tools` (Python) for scripted interactions: `pip3 install gvm-tools`.
- Example: run a scan via gvm-cli (socket connection):

```
# XML snippets used by gvm-cli; simple wrapper example
gvm-cli socket --xml '<start_task task_id="<TASK_UUID>"/>'
# Use Python scripts (gvm-tools examples) to create targets/tasks and fetch
reports programmatically
```

---

10) Backup & restore

- **Backup DB:** dump PostgreSQL DB (gvmd) and feed directories.
- **Export configs:** export scan configs, targets, and credentials via `gvmd` or `gvm-cli` for migration.

- **Restore flow:** re-install GVM/GVMD, restore DB, import configs, and re-import feed data if needed.

---

11) Troubleshooting & common issues

- **Feeds not updating:** check network, proxy settings, and `/var/log/gvm/gvmd.log` and `feed-update` logs.
- **Scanner not connecting to manager:** ensure `ospd-openvas` socket path matches `gvmd` scanner configuration; check permissions and systemd socket file.
- **High memory/CPU:** reduce concurrent scanners, limit max hosts per task, and tune scanner worker counts.
- **Redis / PostgreSQL errors:** check services and DB connection strings (gvmd uses PostgreSQL).
- **Authentication failures:** verify credentials manually and check time sync (Kerberos/WMI issues depend on clocks).

---

12) Performance & scaling

- For large environments, run multiple scanner workers (separate scanner hosts) and configure `gvmd` to distribute tasks.
- Use targeted port lists and segmented scanning windows (off-hours).
- Monitor resource usage and adjust parallelism in scan configs.

---

13) Security & OPSEC

- Treat credentials stored in GVM as sensitive — restrict admin access and encrypt backups.
- Use "Safe Checks" in production unless otherwise authorized.
- Coordinate scans to avoid accidental outages; schedule with stakeholders.

---

14) Useful one-liners & commands

```
# Check environment (helper)
sudo gvm-check-setup

# Update feeds
sudo gvm-feed-update

# Start services
sudo systemctl start ospd-openvas gvmd gsad

# List tasks via gvmd
sudo runuser -u _gvm -- gvmd --get-tasks
```

```
# Start a task (with gvm-cli socket)
gvm-cli socket --xml '<start_task task_id="<TASK_UUID>"/>'

# Export report (example via GUI or gvm-cli to fetch report by ID)
# Use GSAD web UI: https://<gvm-host>:9392  (default port)
```

---

15) Alternatives & complements

- **Nessus** — commercial scanner with broad plugin set.
- **OpenSCAP / Lynis** — configuration/audit focused tools.
- **Nmap + NSE** — flexible discovery and lightweight checks before full GVM scans.

This cheat sheet is intended for professional vulnerability assessors and pentesters. Always have written authorization and coordinate scans with relevant stakeholders.