CHECKLIST

Ubuntu AppArmor and UFW Server Hardening Checklist

A practical checklist for validating AppArmor confinement and UFW firewall rules on Ubuntu servers before production rollout.

Ubuntu AppArmor and UFW Server Hardening Checklist

Use this checklist to confirm that AppArmor and UFW are working together to reduce exposure on an Ubuntu server. Treat network reachability and application confinement as separate controls, then verify both before production.

---

1) Scope the server and its services

- [ ] Identify every service running on the host.
- [ ] Record which services are internet-facing, internal-only, or admin-only.
- [ ] List the exact ports each service must use.
- [ ] Document the source networks allowed to reach each port.
- [ ] Note any local-only sockets, helper binaries, or backend processes the service depends on.
- [ ] Confirm whether the host runs web, SSH, database, proxy, agent, or backup workloads.

---

2) Validate the network exposure model

- [ ] Confirm that only required inbound ports are open.
- [ ] Verify that admin access is restricted to approved management networks.
- [ ] Confirm that no unused listening ports are exposed.
- [ ] Check whether any service is reachable from broader networks than intended.
- [ ] Validate that the firewall policy reflects the actual service exposure, not assumptions.
- [ ] Test connectivity from both allowed and disallowed source networks.
- [ ] Document the expected outcome for each test.

UFW verification checks

- [ ] ufw status verbose shows the intended default policy.
- [ ] Allowed rules match the documented service requirements.
- [ ] No broad allow rules exist unless they are explicitly justified.
- [ ] IPv4 and IPv6 behavior is reviewed if both are in use.
- [ ] Logging is enabled at an appropriate level for troubleshooting.

---

3) Verify AppArmor coverage

- [ ] Confirm AppArmor is installed and active on the host.
- [ ] Verify that the target service has an AppArmor profile.
- [ ] Confirm the profile is enforced for the running process.
- [ ] Check whether the profile is in complain mode, enforce mode, or not applied.
- [ ] Review profile scope for files, capabilities, signals, mounts, and network-related behavior.
- [ ] Identify any nonstandard paths, sockets, or helper programs the service needs.
- [ ] Compare profile allowances with the service?s real runtime behavior.

AppArmor verification checks

- [ ] The service label matches the expected profile.
- [ ] Legitimate file access is permitted.
- [ ] Unnecessary file access is denied.
- [ ] Only required capabilities are allowed.
- [ ] Any denials are explainable and tied to known behavior.
- [ ] Audit logs are available for review.

---

4) Test real workload behavior

- [ ] Start or restart the service with the intended firewall rules in place.
- [ ] Exercise normal application workflows.
- [ ] Confirm the service remains reachable only where expected.
- [ ] Confirm the service can read and write only the files it needs.
- [ ] Confirm the service can use required local sockets or backend connections.
- [ ] Confirm no unexpected helper binaries are spawned.
- [ ] Watch for AppArmor denials during normal operation.
- [ ] Watch for UFW drops or rejects that indicate missing rules.

Functional validation prompts

- [ ] Can users connect from approved networks only?
- [ ] Can administrators connect from the management subnet only?
- [ ] Can the service perform its primary function without broadening access?
- [ ] Do logs remain clean under expected load?

- [ ] Are any denials blocking approved operations?

---

5) Review logs and evidence

- [ ] Review system logs for firewall and confinement events.
- [ ] Review AppArmor denials and confirm whether they are expected.
- [ ] Review UFW logs for blocked connections from unapproved sources.
- [ ] Capture evidence of successful and denied access tests.
- [ ] Save the final configuration state for change tracking.
- [ ] Record any exceptions with a business or technical justification.

---

6) Tune with the smallest safe change

- [ ] Adjust only one control at a time when possible.
- [ ] If a legitimate action is denied, determine whether the fix belongs in UFW or AppArmor.
- [ ] Narrow rules rather than broadening them.
- [ ] Avoid opening extra ports to compensate for application issues.
- [ ] Avoid weakening AppArmor profiles unless the behavior is understood and approved.
- [ ] Re-test after each change.
- [ ] Confirm the issue is resolved without introducing new exposure.

---

7) Production readiness review

- [ ] All required services are reachable only from approved sources.

- [ ] All unnecessary ports are closed.
- [ ] AppArmor enforcement is confirmed for the intended services.
- [ ] Legitimate application behavior works under normal load.
- [ ] Denials are understood, documented, and not blocking production use.
- [ ] Restart behavior has been tested.
- [ ] Rollback steps are documented.
- [ ] Monitoring and log review responsibilities are assigned.

---

8) Common mistakes to avoid

- [ ] Do not treat UFW as application-aware.
- [ ] Do not assume AppArmor is effective without verifying enforcement.
- [ ] Do not change firewall rules, service settings, and AppArmor policy all at once unless necessary.
- [ ] Do not ignore unexplained denials.
- [ ] Do not leave broad allow rules in place after testing.
- [ ] Do not move to production without testing both connectivity and confinement.

---

9) Quick go/no-go decision

Use this simple decision rule:

- [ ] If the service has a clear port list and a known runtime footprint, AppArmor and UFW are a strong fit.
- [ ] If the service behavior is unstable or poorly understood, observe first and enforce later.
- [ ] If you cannot explain every allowed port and every profile exception, the server is not ready for production.

---

10) Final sign-off

Server: _____

Owner: _____

Date: _____

AppArmor enforced for target services: Yes / No

UFW rules validated: Yes / No

Production approved by: _____

Notes:

- _____
- _____
- _____