CHECKLIST

Ubuntu AppArmor and UFW Pre-Production Checklist

A practical checklist for validating AppArmor and UFW on Ubuntu before production rollout. Covers exposure review, profile mode checks, logging, testing, and rollback readiness.

Ubuntu AppArmor and UFW Pre-Production Checklist

Use this checklist to verify that Ubuntu host hardening with AppArmor and UFW is correctly configured, validated, and ready for production use.

---

1) Confirm the host?s intended exposure

- [ ] List the services that should run on the host.
- [ ] Document which services must be reachable from the network.
- [ ] Document which services must remain local-only.
- [ ] Identify the trusted source networks for each exposed service.
- [ ] Record any temporary exceptions and their expiration date.
- [ ] Verify that the host is not exposing any unexpected listening ports.

Check:

- [ ] ss -tulpn reviewed
- [ ] Unexpected listeners investigated
- [ ] Only intended interfaces and ports are exposed

---

2) Validate UFW policy

- [ ] Confirm UFW is installed.
- [ ] Confirm UFW is enabled.
- [ ] Review the active rule set.
- [ ] Ensure inbound rules follow a default-deny posture unless a service requires access.
- [ ] Ensure only approved ports are open.
- [ ] Ensure source restrictions are applied where possible.
- [ ] Confirm administrative access is allowed only from trusted networks.
- [ ] Remove broad allow rules that were added for troubleshooting.
- [ ] Verify there are no duplicate or conflicting rules.

Check:

- [ ] ufw status verbose reviewed
- [ ] ufw status numbered reviewed
- [ ] Open ports match the approved service list
- [ ] Allow rules are as narrow as practical

---

3) Validate AppArmor enforcement

- [ ] Confirm AppArmor is installed and active.
- [ ] Identify which profiles are loaded for the workload.
- [ ] Verify whether each relevant profile is in enforce mode or complain mode.
- [ ] Confirm any critical service has an active profile if one is expected.
- [ ] Verify that unconfined services are understood and intentional.
- [ ] Check for profile changes introduced by package updates.
- [ ] Confirm the service does not rely on permissions outside its expected scope.

Check:

- [ ] aa-status reviewed
- [ ] Loaded profiles documented
- [ ] Enforcement mode confirmed
- [ ] Unconfined processes reviewed

---

4) Test normal workload behavior

- [ ] Start or restart the service in a test window.
- [ ] Exercise all intended user paths.
- [ ] Verify inbound connectivity works only from approved sources.
- [ ] Verify the application can read required configuration and content files.
- [ ] Verify the application can write only to approved directories.
- [ ] Verify approved helper binaries and child processes still function.
- [ ] Verify expected logs are created.
- [ ] Verify the service behaves normally under representative load.

Check:

- [ ] Functional testing completed
- [ ] No unexpected failures observed
- [ ] No hidden dependency on broad local privileges

---

5) Review logs for denials and anomalies

- [ ] Review application logs for permission-related failures.
- [ ] Review kernel audit logs for AppArmor denials.
- [ ] Review firewall logs if logging is enabled.

- [ ] Correlate failures with test actions.
- [ ] Distinguish policy denials from application bugs.
- [ ] Confirm that any denial is expected and acceptable.

Check:

- [ ] No unexpected AppArmor denials during normal operation
- [ ] No blocked traffic from legitimate clients
- [ ] No false sense of health caused by local-only success

---

6) Confirm rollback and recovery options

- [ ] Document the exact commands or process for reverting UFW changes.
- [ ] Document how to relax or disable a problematic AppArmor profile.
- [ ] Ensure console or out-of-band access is available before restricting remote access.
- [ ] Save a copy of known-good firewall rules.
- [ ] Save a copy of known-good AppArmor profile changes.
- [ ] Confirm the team knows who approves emergency rollback.

Check:

- [ ] Rollback steps tested or clearly documented
- [ ] Recovery path available if remote access is blocked
- [ ] Backup of prior state stored securely

---

7) Production readiness decision

Mark the host ready only if all of the following are true:

- [ ] The service exposure is fully understood.
- [ ] UFW rules match the intended network paths.

- [ ] AppArmor profiles are loaded and in the expected mode.
- [ ] Normal workload tests passed.
- [ ] Denials were reviewed and explained.
- [ ] A rollback plan exists.
- [ ] Monitoring or alerting is in place for future denials and blocked access.

Go / No-Go:

- [ ] GO ? controls are validated and ready for production
- [ ] NO-GO ? investigate gaps before rollout

---

8) Quick verification commands

Use these as a final review before production:

---

9) Final sign-off notes

- Service name: _____
- Owner: _____
- Environment: _____
- Date: _____
- Reviewer: _____
- Notes: _____
- _____
- _____

Summary

AppArmor and UFW are complementary controls:

- UFW reduces exposure by controlling network reachability.
- AppArmor reduces impact by restricting what the process can do locally.

Use both together for a practical Ubuntu hardening baseline, then validate behavior before production use.