**Response to Request to Skip Security Scans & Peer Review**

If a manager requests shipping a critical feature now but automated security scans and code review are not yet complete, I'd consider it a high-risk request and approach it openly, calmly, and professionally. I'd first validate that urgency and the business situation ("I get that it's because of that client meeting and why it's important"), then simply enumerate the risk: skipping automated scans and code review makes it more likely that there will be security vulnerabilities, code regressions, and production issues that will damage customers and hurt the reputation of the firm. I'd recommend a brief, tangible plan of action that strikes a balance between fast and safe.

Specifically, I'd make one of two fast-but-safe deployment choices: (A) do a concentrated short-form analysis and limited security checks now and ship to a canary environment that sees a minuscule fraction of real traffic, or (B) ship behind a feature flag but leave it disabled for all users save a controlled test group. Both options allow the business to show the feature while containing blast radius. I'd also ask that we also immediately in-parallel run the missing automated security scans and that the team be on- call ready for rollback should trouble appear. Last, I'd document the exception request (who raised it, why, and agreed mitigations) in our change-control log and then follow up with a full post-deploy retrospective and completed scans before elevating the feature to all users.

I'd also explain to stakeholders: what we're doing, what precise mitigations, what to keep an eye out for, and rollback plan (how to turn off feature flag or reverse deployment). If there's insistence by manager, I'd go to the official change-approval owner (if in place) and seek clear, written acknowledgement that agrees to residual risk—this creates accountability.

* Alternative solutions
* Deploy to a sandbox or staging environment identical to production for the demo and use recorded demo traffic.
* Use feature flags + canary release (1–5% traffic) with active monitoring.
* Run a focused checklist: linting, unit tests, smoke tests, dependency vulnerability scan (fast tools), and peer buddy-review within 30–60 minutes.
* Postpone public demo and schedule a 30–60 minute “hotfix” window after scans complete.
* Prevention (how to avoid repeats)
* Define an emergency-release runbook with pre-approved mitigations (canary, feature flags, fast-scan checklist).
* Automate lightweight fast-scan gates in CI that run in <5 minutes for demos.
* Enforce a policy: no production deployment without at least one approval or an authorized, logged exception.
* Educate execs on risks and create a rapid sign-off flow for true emergencies.