A Practical Guide:

The Evolution of Security in SaaS Startups

Written by Mario Kamburov, Security Professional

2026

Introduction

I've spent more than a decade in startups where everything felt possible and nothing felt optional. Crazy growth, tiny teams, ambitious founders, and CEOs driven by a vision so strong it pulled everyone into orbit.

That kind of energy is intoxicating. It makes people build faster, ship bolder, and ignore friction. Including security.

I've seen teams where innovation and creativity consistently outvoted security — not because people were careless, but because they genuinely believed the mission mattered more than anything else. And most of the time, they were right.

Until they weren't.

This book exists because there is a line – a moment where speed without guardrails stops being brave and starts being irresponsible. The problem is that startups are terrible at seeing where that line is — and even worse at knowing when to act.

I've been a founder. I've joined a maturing SaaS company later to help untangle years of "we'll fix this later." I've managed incidents, audits, automations, process failures, and people problems. I've seen what actually breaks companies — and what doesn't.

This is not a book about fear-driven security.
It's about practical security that respects innovation instead of killing it.

Along the way, I'll be blunt about:

- how security, compliance, legal, and privacy are not the same thing
- why confusing them creates theater instead of protection
- why all of this becomes harder — and more urgent — in the AI era

No security by obscurity. No buzzwords. No fake frameworks. No bullshit.

Just the realities startups eventually run into and how to deal with them before they deal with you. If you're building something ambitious, this book is here to help you protect it without slowing it down.

Who is this book for?

This book is for:

- founders building their first startup,
- people already working in startups who wonder how security should evolve,
- managers trying to establish a security mindset and culture without unnecessary bureaucracy.

It is written for teams that want **pragmatic security** and security that supports innovation instead of blocking it.

Terms and Definitions

Security

A general term covering activities, processes, and responsibilities related to protecting the organization, its data, and its critical assets (tangible and intangible). It includes areas such as cybersecurity, cloud security, product security, data security, IT security, AI security, and security compliance.

Compliance

A broad set of requirements and expectations that form the foundation of an organization's governance framework. Compliance is driven by customers, regulators, auditors, investors, and other stakeholders.

Governance

The unified way an organization defines strategy, policies, and processes, and enforces expected behavior from leadership to staff. Governance programs are typically shared between Security and Legal and together form the GRC framework.

Risk Management

The structured process of identifying, assessing, mitigating, accepting, and monitoring risks using a unified methodology overseen by leadership.

Privacy

The bridge between Security, Legal, and Engineering. Privacy connects regulatory requirements with technical implementation and human rights considerations. It typically sits within the Legal function.

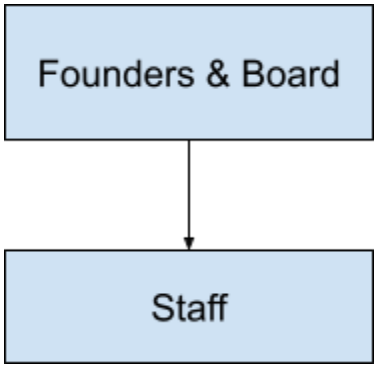
Legal

The regulatory and contractual function responsible for intellectual property, commercial terms, agreements with customers, employees, third parties, and regulators.

Stage 1

No Security. No Governance. Centralized Trust

<5 employees



All decision-making and execution are centralized around the founders, with no separation of responsibilities.

At this stage, the company is essentially a single decision-making unit. As illustrated in the diagram, all authority flows directly from the founders to the staff, without any intermediate structure or separation of responsibilities. This is not a weakness, it is a necessity. The company exists to validate an idea, find product–market fit, and survive long enough to justify further investment. Speed and flexibility outweigh almost every other concern.

The founders, usually the CEO and CTO, are the center of gravity. They make product decisions, write code, deploy to production, manage cloud accounts, administer internal tools, negotiate contracts, and interact directly with customers. There is no Segregation of Duties[1], no Governance Framework[2], and no formal accountability structure. Introducing these concepts too early would likely slow the company down without delivering meaningful protection.

Security at this stage is not absent because people are careless. It is absent because trust is centralized. Everyone involved knows each other personally, works closely together, and shares context continuously. Decisions are made quickly, often informally, and almost always under pressure. This environment is highly risk-tolerant by design.

[1] **Segregation of Duties (SoD)** — A control principle that prevents a single person from having end-to-end control over a critical process. It reduces risk by separating execution, approval, and oversight so that errors or abuse cannot occur without detection. Example – approved Github Pull Request by 2nd Engineer.

[2] **Governance Framework** — A structured set of roles, processes, and rules that define how decisions are made, monitored, and enforced within an organization. Governance frameworks provide accountability and oversight, especially as organizations grow and responsibilities are distributed.

The real risk in Stage 1 is not insecurity itself, but the formation of habits and architectural decisions that later become difficult or impossible to reverse. When unrestricted access becomes normal, when shared credentials feel convenient, or when fragile technology choices are made purely for speed, these patterns tend to persist into later stages where the cost of change is much higher.

Security failures here are usually not external attacks. They are internal and structural: weak technology choices, early technical debt, and leadership behavior that normalizes bypassing basic safeguards. Statements like “I need full access to do my job” or “this is slowing us down” may be justified at the moment, but they shape expectations for the future organization.

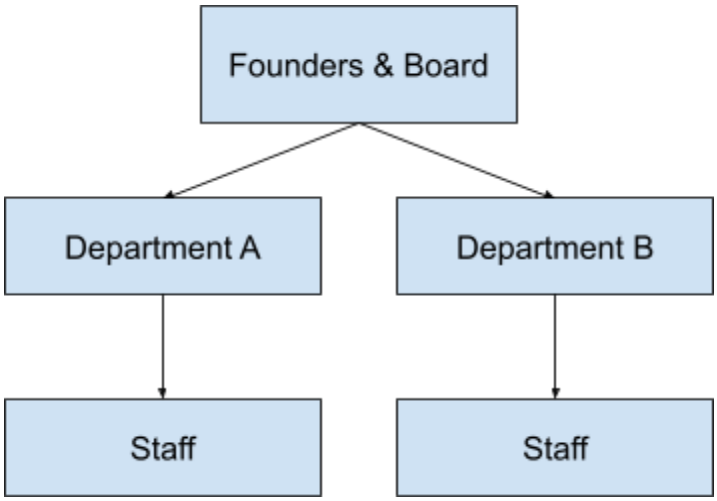
The objective of Stage 1 is not to introduce security controls, but to avoid irreversible mistakes. Founders should think in terms of reversibility. If a decision can be undone easily later, optimizing for speed is reasonable. If it is deeply embedded—such as identity architecture, data isolation, or core cloud ownership—it deserves more thought, even at this early stage.

A successful Stage 1 company exits this phase with centralized trust still intact, but with awareness of future risk. Identity is not chaotic, access can be revoked, and foundational architectural decisions have been made consciously. Security is not yet a function, but it is no longer invisible.

Stage 2

Basic Security Inside Delivery Teams

5-50 employees



Responsibilities begin to split across teams, but decision authority and oversight remain centralized.

In Stage 2, the diagram shows the first meaningful structural change: the organization splits into departments. Authority still flows from the founders, but now through the department leads to staff. This reflects a shift from informal coordination to functional ownership. Engineering, product, and operational functions begin to form, and the company can no longer rely entirely on shared context.

Security at this stage does not exist as an independent function. Instead, it is embedded within delivery teams. Responsibility for access, infrastructure, and operational hygiene is usually absorbed by roles such as infrastructure engineering, IT operations, internal support, or senior engineers with deep system knowledge. The diagram makes this clear: there is still no controlling or oversight function.

This stage often coincides with increased external exposure. The company now has customers, marketing presence, integrations, and public artifacts. As a result, it becomes more visible to opportunistic attackers and automated abuse. Security incidents at this stage are rarely sophisticated, they are the result of inconsistent identity practices, weak access hygiene, misconfigurations, or leaked credentials.

A common misunderstanding in Stage 2 is the belief that hiring a single security-minded individual solves the problem. The diagram contradicts this assumption. Security still flows