The Subtractive Exception & Security Debt Standard

A framework for managing deviations as **Temporary Insecurity**.

Exceptions are not permissions to remain insecure; they represent **documented Security Debt** that must be actively reduced. The objective is to drive **Architectural Correction** and preserve the system's integrity.

Mandatory "Subtractive" Requirements

All exception requests must include:

- **Attack Path Impact:** Explicitly identify which stage of an attack (identity, lateral movement, egress) is enabled by this deviation.
- **Mandatory Non-conductivity:** Requests must identify technical compensating controls—such as segmentation or restricted egress—that materially limit the exception's blast radius as defined by a reduction in Path Erasure Rate (PER).
- **Architectural Remediation Plan:** A defined plan to eliminate the dependency and remove the exception permanently.

The "No-Permanent-Debt" Rule

- **Temporary by Design:** All exceptions must have a mandatory expiration date.
- **Renewal = Re-decision:** Renewal requires proof that compensating controls remain effective and that remediation has progressed.
- **Escalation:** Repeated renewals without architectural change are classified as a **Governance Failure**.

---

1. Purpose

This policy defines how security exceptions are requested, evaluated, approved, and retired in a way that **reduces risk over time rather than documenting it indefinitely**.

The objective of the exception process is **not to allow permanent deviations**, but to:

- temporize unavoidable constraints,
- require meaningful compensating controls,
- drive architectural correction,
- prevent recurrence of known attack paths,
- and preserve the security system's integrity as attack paths are erased.

An exception **does not grant permission to remain insecure**. It represents **documented, time-bounded risk debt** that must be actively managed and reduced.

---

2. Guiding Principles

Exceptions are signals, not solutions
Every exception indicates a control, dependency, or architectural invariant that cannot currently be enforced.

Risk must be reduced, not deferred
An exception is only acceptable if it reduces risk relative to doing nothing while permanent remediation is pursued.

Compensation is mandatory
No exception may be approved without explicit compensating controls that materially limit attacker capability, blast radius, or exploitability.

All exceptions are temporary
Exceptions must have a defined expiration date and require active re-decision to continue.

Architecture must evolve
Every exception must include a remediation plan describing how the exception will be eliminated.

Recurrence is failure
Repeated exceptions for the same condition indicate a governance or architectural failure and require escalation.

---

3. Scope

This policy applies to:

- all security controls,

- all functional units (e.g., endpoints, identity platforms, CI/CD, cloud, backups),
- all employees, contractors, applications, and infrastructure,
- all environments (production and non-production).

---

4. Definitions

Security Exception

A formally approved, time-bounded deviation from an established security control or architectural invariant.

Compensating Control

A technical or architectural control that demonstrably reduces risk introduced by an exception (e.g., segmentation, reduced privileges, additional enforcement, high-confidence detection).

Security Debt

Residual security risk temporarily accepted with the explicit intention of elimination through architectural change.

Functional Unit

A bounded system with predictable behaviors and a defined security contract (e.g., workstations, admin plane, CI/CD).

---

5. Roles and Responsibilities

Requestor

- Initiates the exception request.
- Provides business justification and operational constraints.

Functional Unit Owner (FUO)

- Describes technical impact.
- Proposes compensating controls.
- Defines architectural remediation plan.

Business Risk Owner (BRO)

- Accountable for accepting residual risk.
- Approves business justification and risk tradeoffs.

Security Governance Owner (SGO)

- Ensures policy adherence.
- Confirms compensating controls exist.
- Tracks exception lifecycle and recurrence.

---

6. Exception Request Requirements (Mandatory)

All exception requests **must** include the following. Requests missing any section will be rejected.

6.1 Description of Deviation

- What control or invariant cannot be enforced?
- What is being allowed that would otherwise be prohibited?

6.2 Attack Path Impact

- Which attacker capability is enabled or made easier?
- Which attack stage is affected (identity, execution, lateral movement, egress, impact)?

6.3 Justification

- Why enforcement is infeasible now (technical or business constraint).
- “Convenience”, “legacy”, or “we’ve always done it this way” are not sufficient.

6.4 Mandatory Compensating Controls

Describe **explicit controls** that reduce risk while the exception exists, such as:

- reduced privileges or scope,
- segmentation or isolation,
- restricted egress,
- bounded execution context,
- additional validation or approvals,
- high-confidence detection for boundary violations.

If no compensating control can be identified, the exception must not be approved.

6.5 Residual Risk Statement

- What risk remains?
- What explicitly cannot happen due to the compensating controls?

6.6 Architectural Remediation Plan

- What must change to remove this exception permanently?
- What dependency will be eliminated?
- What control will be enforceable afterward?

6.7 Time Bound

- Start date
- Mandatory expiration date (no “until further notice”).

---

7. Approval Criteria

An exception may only be approved if **all** of the following are true:

- The risk introduced is clearly understood and documented.
- Compensating controls materially reduce feasibility of attack or impact.
- An architectural remediation plan exists.
- The business owner explicitly accepts residual risk.
- The exception is time-bounded.

An exception **must not** be approved if:

- it does not reduce risk,
- it merely documents exposure,
- it replaces engineering work with paperwork,
- or it creates permanent operational debt.

---

8. Exception Duration and Renewal

- All exceptions expire automatically.
- Renewal requires a **new decision**, not an extension.
- Evidence must be provided showing:

    - compensating controls remain effective,
    - remediation has progressed or dependencies still exist,
    - risk has not increased.

Repeated renewals without architectural change will be escalated as a governance failure.

---

9. Monitoring and Review

- All exceptions are recorded in the Risk Debt Register.
- Exceptions nearing expiration (≤30 days) are reviewed weekly.
- Exception trends are reviewed monthly:
    - total active exceptions,
    - average duration,
    - repeat exceptions by root cause.

Exception trends inform architectural prioritization.

---

10. Relationship to Alerts and Incidents

- Alerts linked to an active exception must be explicitly flagged.
- Alerts caused by unresolved exceptions **do not count as acceptable noise**.
- Repeated alerts related to an exception require reassessment of compensating controls and may force early revocation.

---

11. Policy Enforcement

Failure to comply with this policy may result in:

- exception denial,
- forced control enforcement,
- escalation to executive leadership,
- or removal of affected functionality until compliance is achieved.

---

12. Statement of Intent

Exceptions exist to protect the organization while removing insecurity—not to normalize it.
This policy ensures that deviation is temporary, risk is visible, and architecture continuously improves.