ZERO TRUST SPECTRUM

ZERO TRUST SPECTRUM · FILES | CYBOK ALIGNED | NCSC ALIGNED | NIST SP-800-207 ALIGNED

Deterministic File Assurance for Zero-Trust Programs

The Zero Trust Spectrum (ZS-1 → ZS-6) provides a CyBOK-aligned language for rating how files are inspected, sanitized, isolated, or deterministically rebuilt-complete with per-dimension scores, validation tests, and governance signals for Zero-Trust architectures.

WHO THIS IS FOR

CISOs, Zero-Trust program owners, security engineers, platform/product teams, delivery partners.

USE IT FOR

Pick the minimum assurance that still satisfies risk, usability, and compliance.

Zero-Trust Spectrum (Files) Overview

The Zero-Trust Spectrum defines six progressive assurance levels (ZS-1 → ZS-6) describing how files are verified, controlled, and evidenced within zero-trust architectures. Levels advance from detection-based trust (ZS-1/2) to deterministic rebuild and independent verification (ZS-6). The framework is aligned with NIST SP-800-207, UK NCSC Zero-Trust Architecture, and CyBOK principles and published under CC BY 4.0 for open reference and adoption.

Mission Snapshot

Purpose

Declarative framework for deciding file-handling requirements inside Zero-Trust architectures.

Scope

Email & collab · Web/SWG/ICAP · APIs · Storage · Kiosks · Air-gapped/CDS · SDK/Embedded · File import/export

Audience

Security architects, platform teams, regulated industries, product leaders building trusted data flows.

Rating Scale & Dimensions

Every ZS level is scored 0 → 4 across six dimensions so buyers can compare like-for-like claims.

Scale (0 → 4)

0 NONE / HEURISTIC

Detection or reputation only; no structural enforcement.

1 BASIC

Coarse rules or feature stripping; limited validation, best-effort only.

2 MODERATE

Isolation/flattening; limited spec awareness.

3 STRONG

Allow-only or spec-aware validation/repair with detailed logs.

4 DETERMINISTIC

Model-driven rebuild with reproducible outcomes for supported formats.

Dimensions

MECHANISM RIGOR
Strength of the control mechanism irrespective of detections.

ASSURANCE EVIDENCE
Quality of conformance proof, determinism, validators, provenance.

USABILITY FIDELITY
How much editability and workflow fidelity is preserved.

VALIDATION TEST RIGOR
Depth of acceptance tests teams can run independently.

AUDITABILITY
Chain-of-custody, tamper evidence, and investigative readiness.

THREAT COVERAGE
Active content, parser exploits, nests, obfuscation, signed content, unknowns.

ZS-Level Definitions

LEVEL	LABEL	CORE MECHANISM	DESCRIPTION
ZS-1	Pass-through	Implicit trust	Files accepted without verification; basic metadata checks only.
ZS-2	Detection	Known-threat identification	Signature/hash/pattern recognition detects known threats but cannot assure safety.
ZS-3	Isolation / Flattening	Behavioural containment	Executes or renders files in controlled environments to prevent direct interaction.
ZS-4	Sanitisation	Rule-based removal	Strips or deactivates risky features (macros, scripts, links) within the original file.
ZS-5	Transform	Rebuild from known-good template	Parses and reconstructs files to conform to safe models; removes unsafe components.
ZS-6	Deterministic Rebuild	Clean intermediary model	Files fully deconstructed and rebuilt from specification; no source bytes retained; output verified with audit evidence.

Flat-file Export / Import

Flat-file export/import is an essential control boundary. Many Zero-Trust architectures require exporting processed content to a flat, policy-governed format (XML, CSV, JSON, TXT, PDF/A) before re-ingestion or cross-domain movement.

- This ensures rebuilt data is validated, context-neutral, and verifiable—aligning with ZS-6 deterministic assurance while evidencing policy enforcement.
- ZS-5 may emit validated/sanitized output in the native format, but structural fidelity still depends on the source file.
- ZS-6 can export to neutral flat files or re-import from them, so no source-file dependency remains.
- Flat-file output suits air-gap transfers, audit trails, and ingestion by downstream trusted systems; optional flat-file import supports deterministic rebuilding from clean data structures.

Decision Quick Picks

Baseline: Recommend ZS-6 wherever formats are supported and service objectives are met. Use ZS-5 only as a documented fallback for unsupported formats/SLO exceptions, and ZS-3 isolation for executables and unstructured binaries. Performance and coverage depend on implementation—verify with PoV data.

Operational note: Modern ZS-6 rebuild engines routinely deliver sub-100 ms latency for mid-size documents and scale horizontally in containerized deployments.

- **Performance ≠ level:** Throughput/latency are architecture and tuning dependent; a well-engineered ZS-6 can outperform a poorly tuned ZS-2/5.
- **Format coverage is product-specific:** Many ZS-6 offerings support dozens of formats—validate rebuild vs sanitize vs bypass per format.
- **Policy fallback is about coverage/SLOs:** Use ZS-5 fallback only when a format lacks rebuild support or an SLO demands it; use ZS-3 for binaries/installer classes that cannot be rebuilt.
- **Supplier/customer uploads & data rooms:** Target ZS-6; fallback ZS-5 only for unsupported formats. External risk + audit trail means deterministic outputs; optional flat-file exports feed downstream apps safely.
- **Exec / finance / legal attachments:** Target ZS-6 for Office/PDF/images; fallback ZS-5. Sensitive workflows demand evidencing plus fidelity.
- **General collaboration (M365/Workspace, SharePoint/Drive):** Target ZS-6 for all supported formats; fallback to ZS-5 only for unsupported or custom types. ZS-3 isolation reserved for executables/archives that cannot be rebuilt.
- **Web downloads & SWG:** Docs/images/media → ZS-6 if supported & SLOs met; else ZS-5. Unknown binaries/installers → ZS-3 isolation. Tier high-volume CDN flows by MIME, size, and reputation.

- **Cross-domain / air-gap / OT/classified:** Transfer path at ZS-6 with strict policy + reporting plus optional flat-file export/import; review path via ZS-3 read-only previews.
- **Storage & archival remediation:** Enable flat-file export mode for long-term governance so structural metadata and policy outcomes persist apart from original content.
- **Automation & integration pipelines (API/SDK):** Require flat-file interchange (XML/JSON schema) for validated objects feeding analytics, ETL, SIEM, or DLP workflows.
- **Developer tools & installers:** Docs/packages → ZS-5 (ZS-6 where supported). Executables → ZS-3 isolation plus repo/code-sign controls.

Classification Criteria

DIMENSION	ZS-1	ZS-2	ZS-3	ZS-4	ZS-5
Content verification	None	Partial (signatures)	Partial (behavioural)	Moderate	Full
Structural enforcement	None	None	Partial	Moderate	Full (softwa
Policy conformance	None	Basic	Contextual	Enforced rules	Full
Audit evidence	None	Minimal	Limited logs	Moderate	Comprehen
Fidelity / usability	Full	Full	Read-only	High	High

Evidence expectations: Classifications above ZS-4 require conformance testing, proof no source-file bytes persist, policy enforcement traceability, validation/verification steps, and reproducible audit trails.

Zero Trust Spectrum Levels

ZS-1 · BASELINE

Pass-through / Uninspected

MECHANISM RIGOR

0 NONE / HEURISTIC

Allow/forward or AV-only; enforcement is purely detection based.

ASSURANCE EVIDENCE

0 NONE / HEURISTIC

No structural proof, determinism, or provenance beyond AV verdicts.

USABILITY FIDELITY

4 DETERMINISTIC

Full editability and zero latency because files are untouched.

VALIDATION TEST RIGOR

0 NONE / HEURISTIC

Malformed or macro-laden samples still pass, revealing lack of control.

AUDITABILITY

0 NONE / HEURISTIC

Only coarse pass/fail telemetry; no tamper-evident chain of custody.

THREAT COVERAGE

ZS-2 · POLICY

Rule-based Sanitization

MECHANISM RIGOR

1 BASIC

Policy strips macros/scripts/links yet leaves underlying structure untouched.

ASSURANCE EVIDENCE

1 BASIC

Evidence is limited to policy-hit logs; little spec validation.

USABILITY FIDELITY

3 STRONG

High usability apart from removed risky features.

VALIDATION TEST RIGOR

1 BASIC

Seed macros/OLE to verify removal plus reason codes.

AUDITABILITY

1 BASIC

Lists of removed constructs, but minimal per-object provenance.

THREAT COVERAGE

1 BASIC

0 NONE / HEURISTIC

No mitigation for macros, parser exploits, nesting, obfuscation, or unknowns.

Handles common active content yet struggles with parser or obfuscated attacks.

ZS-3 · ISOLATION

Flatten / Isolate

MECHANISM RIGOR

2 MODERATE

Static rendering or remote isolation keeps active content away from endpoints.

ASSURANCE EVIDENCE

2 MODERATE

Isolation telemetry shows containment but not rebuild-grade determinism.

USABILITY FIDELITY

1 BASIC

Mostly read-only or limited interactivity—great for triage, not authoring.

VALIDATION TEST RIGOR

2 MODERATE

Deliver active content and confirm flattened or isolated presentation.

AUDITABILITY

2 MODERATE

ZS-4 · ALLOW-ONLY

Allow-only (DDR)

MECHANISM RIGOR

3 STRONG

Positive selection copies only allow-listed structures into safe templates.

ASSURANCE EVIDENCE

3 STRONG

Template coverage and exception manifests provide repeatable assurances.

USABILITY FIDELITY

3 STRONG

High usability for modelled features; coverage gaps require new templates.

VALIDATION TEST RIGOR

3 STRONG

Inject allowed/disallowed elements and observe copy/drop evidence.

AUDITABILITY

3 STRONG

Template IDs, policy versions, and exception logs

Session logs and release/deny trails support investigations.

THREAT COVERAGE

2 MODERATE

Neutralizes active content during viewing; downloads may still carry original bytes.

support traceability.

THREAT COVERAGE

3 STRONG

Covers modelled embeds/nests/obfuscation but only where templates exist.

ZS-5 · REPAIR

Deep Parsing & Repair

MECHANISM RIGOR

3 STRONG

Spec-aware parsing plus repair/sanitise routines enforce structure; outputs can be emitted in the native format or optional flat-file exports, but still reference original structure.

ASSURANCE EVIDENCE

3 STRONG

Validation/repair logs act as conformance evidence per object.

USABILITY FIDELITY

4 DETERMINISTIC

Maintains authoring fidelity while repairing defects deterministically.

VALIDATION TEST RIGOR

ZS-6 · REBUILD

True Rebuild (Safe Blueprint)

MECHANISM RIGOR

4 DETERMINISTIC

Model-driven rebuild outputs new spec-compliant files with no unsafe bytes and can natively export/import neutral flat files (XML/JSON/CSV/PDF-A) for air-gap or analytic flows.

ASSURANCE EVIDENCE

4 DETERMINISTIC

Validators plus provenance prove deterministic results per file.

USABILITY FIDELITY

4 DETERMINISTIC

Full fidelity for supported formats while eliminating risky source bytes.

3 STRONG

Spec-violating samples should surface precise repair telemetry.

AUDITABILITY

3 STRONG

Per-object validation traces, repair counts, and policy diffs.

THREAT COVERAGE

3 STRONG

Covers macros, parser exploits, nests, and obfuscation for supported formats; optional flat-file exports aid downstream review but remain tied to sanitized source semantics.

VALIDATION TEST RIGOR

4 DETERMINISTIC

Diff input/output, review validator logs, and confirm provenance packages.

AUDITABILITY

4 DETERMINISTIC

Signed manifests, hashes, policy IDs, and tamper-evident logs per submission.

THREAT COVERAGE

4 DETERMINISTIC

Mitigates macros, parser exploits, nests, obfuscation—flat-file export/import provides a clean break for cross-domain transfer; fallbacks defined for unsupported formats.

Each tier defines how the control handles encrypted payloads, nested archives, macros, scripts, active content, media, metadata, and unsupported formats. Advance maturity when operational risk or regulatory posture demands it.

Summary: ZS-1 pass-through · ZS-2 policy sanitize · ZS-3 flatten/isolate · ZS-4 positive selection · ZS-5 deep parsing & repair · ZS-6 deterministic rebuild.

Evaluation Criteria

Functional

- Mechanism type and visibility (pass, sanitise, rebuild).
- Encrypted/password-protected handling.
- Nested/archived content support.
- Export/import to flat structured formats (CSV, XML, JSON, TXT, PDF/A).
- Ability to rebuild directly from validated flat data for deterministic ZS-6 workflows.
- Policy configuration depth per format/feature.

Governance & Interop

- Policy versioning, audit trails, and tamper-evident logs.
- DLP/SOAR/SIEM integration and open telemetry (API/syslog).
- Export sanitised/rebuilt outputs to safe zones; store or transmit clean flat files as audit artifacts and cross-domain handoffs.
- Open standards, schema alignment, and reproducible SLO reporting.

Zero Trust Spectrum · Files · Version 2025-11-10

An open reference framework for file assurance within Zero-Trust architectures.

Citable via DOI: 10.5281/zenodo.17576466

Maintained by independent contributors · ORCID 0009-0003-9242-4790

Aligned with NCSC / NIST SP 800-207 / CyBOK.