TRUSTED CONTAINER INTEGRITY, FOR THE UNTRUSTED WORLD

Portable File Identity (PFI) Protection for ZIP Archives

Abstract v1.2

ZIP files lack intrinsic identity. VeraFile's Portable File Identity (PFI) embeds a Validation Integrity Key (VIK) into each component of the ZIP file, enabling continuous, keyless, and tamper-evident integrity validation.

VeraSec Technologies

Brian Hajost
bhajost@verasec.net

About VeraSec Technologies

VeraSec Technologies is a pioneer in Artifact Identity Security (AIS), focused on moving the security context from the infrastructure directly to the file itself. While traditional security models rely on centralized systems to establish and maintain trust, VeraSec's patent-pending technologies empower files to carry their own verifiable identity - preparing them for an actual Zero Trust world.

Operating at the forefront of Portable File Identity (PFI) technology, VeraSec invented the Validation Integrity Key (VIK) process. This innovation patches a fundamental vulnerability in traditional PKI that has persisted for decades. By cryptographically binding a file's content and metadata into a compact, human-readable identifier embedded directly within the artifact, VeraSec helps organizations set up durable, portable trust in every digital asset, signed or unsigned - wherever it travels.

About The Author

Brian Hajost is a cybersecurity entrepreneur and technologist with more than 30 years of experience in information security, compliance automation, and trusted computing systems.

As the founder of SteelCloud, he pioneered the automation of DISA STIG and CIS benchmark compliance for the DoD, federal agencies, and enterprise customers. Brian holds 15 patents in information technology and mobile security. In 2022 he was named one of the "Top 10 Most Influential People in Cyber Security" by CIO Views.

His current work focuses on solving what he describes as the Artifact Identity Problem - the absence of durable identity for files as they move across distributed infrastructures and untrusted networks.

Establishing Trusted Container Integrity

VeraFile Protection for ZIP Archives in Zero Trust Environments

Executive Summary

ZIP archives function as the shipping containers of the digital economy - encapsulating software builds, system logs, and high-value legal and financial records as they move across enterprise systems, partner ecosystems, and cross-domain boundaries. Yet the ZIP format was engineered for portability, not provenance. It has no inherent, deterministic identity; identical payloads can be represented through multiple structurally valid, but cryptographically divergent, encodings.

Within a Zero Trust architecture, this non-canonical flexibility creates a material integrity gap. Organizations have made significant progress securing the transport layer (encryption) and the identity layer (ICAM/PKI), but the artifact itself - the container carrying the data, remains fundamentally unverified. A ZIP file can be modified, repackaged, or reconstituted without leaving durable, portable evidence of tampering – even if the file has been PKI signed.

VeraFile closes this gap by introducing a containerized integrity model based on Validation Integrity Keys (VIKs). Each archive is cryptographically bound to its contents, structure, and identity, producing a persistent, human-readable integrity marker that travels with the file. This transforms ZIP archives into self-validating, tamper-evident artifacts, independent of external infrastructure, centralized registries, or key management systems.

Unlike legacy approaches, VeraFile operates without generating, storing, or relying on cryptographic secrets. This keyless model eliminates an entire class of operational and security burdens while enabling integrity validation at scale - even in uncontrolled or disconnected environments. When used alongside PKI, VeraFile establishes a higher-order trust layer: not only verifying who signed a file, but ensuring the entire container - its contents, structure, and identity, remains intact and authentic over time.

The Structural Integrity Gap: *The ZIP "Malleability" Problem*

The ZIP format is fundamentally **non-canonical** - there is no single, authoritative binary representation for a given set of files. Multiple archives can encapsulate identical payloads while producing entirely different byte-level signatures. This structural malleability is not an edge case; it is intrinsic to the ZIP format's design.

This ambiguity manifests across several dimensions:

- **Metadata Divergence** - ZIP archives maintain parallel Metadata structures (Local File Headers vs. Central Directory). These can be intentionally desynchronized, creating inconsistent interpretations between tools that parse one structure versus the other.
- **Structural Permutations -** File ordering, compression methods, timestamps, alignment padding, and extensible "extra fields" can all vary without altering the extracted content. Each variation yields a distinct binary artifact despite identical logical payloads.
- **Shadow Data Injection -** The ZIP specification permits arbitrary data in archive comments, unused regions, and non-referenced segments. These areas can be exploited to embed hidden payloads or signaling data that evade conventional inspection.

Traditional security controls focus overwhelmingly on **post-extraction validation** - hashing or scanning the decompressed files, while implicitly trusting the container. This creates a critical blind spot: the archive itself is rarely validated as a first-class security object.

The result is a class of **repackaging and structural manipulation attacks**, where adversaries alter the container without modifying the visible payload. These techniques can:

- Evade signature-based detection by altering binary representation
- Exploit parser inconsistencies between security tools and extraction utilities
- Enable path traversal exploits (e.g., ZIP Slip) through crafted directory structures
- Smuggle data through non-extracted regions of the archive

In effect, two ZIP files that appear identical from a content perspective may represent **materially different, and potentially malicious artifacts** at the structural level. Without deterministic container identity, there is no reliable way to distinguish between them.

Artifact Identity Security (AIS) - Canonicalization vs. PFI: *A Necessary Architectural Shift*

Canonicalization helps standardize how a ZIP archive is initially produced, but it does not solve the larger integrity problem – it is a half measure at best. Even with perfect canonicalization, protection is limited to the moment of creation. Once the archive leaves that trusted process, it can still be copied, renamed, repackaged, structurally altered, or

tampered with. Canonicalization may create consistency, but it does not create durable, portable trust.

Just as importantly, canonicalization is operationally brittle. It requires strict control over file ordering, metadata, timestamps, compression behavior, and toolchains. In distributed environments, those conditions are difficult to impossible to evaluate and enforce.

VeraFile, as a PFI solution, addresses the problem differently. Rather than trying to force ZIP files into a single normalized representation, VeraFile generates a **Validation Integrity Key (VIK)** that binds the archive's contents, structure, size, and name into a portable identity that travels with the file.

This shifts the security model from **standardizing creation** to **verifying persistence of integrity over time**. A canonically produced ZIP file may begin life in a known-good state. A VeraFile-protected ZIP can be validated - wherever it goes, however it is stored, and regardless of whether the surrounding environment is trusted.

In short, canonicalization may help produce a cleaner ZIP file. VeraFile helps prove that the cleaner ZIP file remains authentic after production.

VeraFile: *Supporting a Multi-Layered PFI Integrity Model*

VeraFile moves the "Locus of Trust" from the network infrastructure directly to the artifact. It employs a three-tiered defense-in-depth strategy:

Layer	Mechanism	Security Benefit
Atomic Level	**File VIK**	Binds content, filename, and size into a unique identity. Prevents renaming or silent substitution.
Structural Level	**Container VIK**	An aggregate identity of all constituent files and their specific order. Detects additions, deletions, or reordering.
Policy Level	**Canonical Manifest**	A "single source of truth" embedded in the archive that defines valid states and enforcement policies.

Redundancy as a Shield

VeraFile does not rely on a single point of failure. It embeds integrity data redundantly across the **Central Directory, Extra Fields, and Archive Comments.** If an attacker attempts to "strip" the integrity metadata from one section, the discrepancy between the layers triggers an immediate validation failure.

Operational Advantages of Portable File Identity (PFI) in Zero Trust

- **Elimination of Key Management Overhead** - VeraFile's VIK model is deterministic and self-contained, delivering cryptographic integrity without certificates, private keys, or supporting infrastructure. There are no CRLs, HSM dependencies, or availability requirements for external validation services. This removes a significant operational burden while enabling validation in disconnected, cross-domain, and unmanaged environments.
- **Software Supply Chain Integrity** - As software artifacts move from build to staging to production, VeraFile ensures the entire bundle remains intact. It complements code signing by extending assurance beyond the executable to the full deployment context - validating that the right files, in the right structure, are delivered without substitution, omission, or repackaging.
- **Durable Forensic Integrity** - VeraFile establishes a persistent, tamper-evident seal on archived artifacts. Any modification, whether to content, structure, or metadata such as timestamps or authorship, is immediately detectable. This enables long-term evidentiary reliability for legal, financial, and regulatory records without reliance on external trust anchors - ensuring that archived data remains audit-ready for years or decades.

Summary: - VeraFile operationalizes Zero Trust at the artifact level, delivering infrastructure-independent integrity, supply chain assurance, and long-term forensic trust with minimal friction.

Security Protocol: *Automated Archive Validation*

When a VeraFile-enabled system receives a ZIP archive, it executes a deterministic, multi-stage validation protocol before any files are extracted:

1. Structural Integrity Check - The system compares the VIKs stored in the *Extra Fields* against those in the *Archive Comments*. Any discrepancy flags the archive as Tampered.
2. Manifest Reconciliation - The embedded Canonical Manifest acts as the authoritative source of truth. The system verifies that the physically present files perfectly match the manifest's inventory.
3. Deterministic Identity Validation - A unique identity is derived for every file based on content hash, filename, and size.
    - This is compared against the VIK stored in the Manifest.
    - Individual File VIKs are aggregated to recalculate the Container VIK. If a single bit has changed, the ZIP is identified as “invalid.”

By executing this "Validate Before Use" protocol, VeraFile ensures that only known-good artifacts enter local networks.

Canonicalization improves formation. VeraFile protects identity.

Comparison: *VeraFile vs. Standard PKI and VeraFile + PKI*

Public Key Infrastructure (PKI) is still the industry standard for establishing the identity of users, systems, and signing authorities. However, its trust model is **signature-centric and byte-range dependent**, which limits its ability to ensure the integrity of complex containers like ZIP archives over time.

- Standard PKI - PKI signs a specific byte sequence within a file, binding that sequence to a cryptographic identity. While this provides strong assurance of *who signed what at a point in time*, it does not inherently protect the broader artifact lifecycle. PKI-protected ZIP containers can be repackaged, renamed, partially modified, or reconstructed in ways that preserve or bypass signature expectations. In practice, PKI provides identity assurance, but only if the file has not been tampered - *PKI does not protect itself.*
- VeraFile - VeraFile establishes a deterministic, portable identity for the entire container through the Validation Integrity Key (VIK). It binds contents, structure, size, and filename into a single, human-readable integrity marker, without relying on keys, certificates, or external infrastructure. This enables persistent, environment-independent validation of the artifact as a whole, regardless of how or where it is stored or transmitted.
- Better Together, *VeraFile + PKI* - Combined, the two approaches create a composite trust model. PKI answers the question of “Who asserted trust?” while VeraFile answers “Is this still the exact artifact that was trusted?” This produces a durable

Trust Envelope, where identity, integrity, and artifact continuity are cryptographically and operationally linked across the file's entire lifecycle.

Bottom line: ***Together, they deliver end-to-end, lifecycle integrity.***

- PKI establishes *authorship*.
- VeraFile preserves *artifact truth*.

Industry-Specific Use Cases

Industry Sector	Critical Use Case	The "ZIP" Vulnerability	VeraFile Strategic Impact
Defense & Intelligence	**Tactical Edge Data Transfer**	Intelligence packages sent over degraded networks can suffer corruption or silent tampering that standard ZIP tools ignore.	**Mission Assurance:** Ensures tactical data (maps, ISR feeds) is structurally identical to the source, even in disconnected environments.
Government	**Cross-Domain Solutions (CDS)**	Adversaries can use "ZIP Slip" or metadata injection to hide malicious payloads that bypass traditional file-scrubbers.	**Structural Sanitization:** Forces a "Canonical Manifest" that rejects any hidden files or unauthorized metadata before crossing boundaries.
Healthcare	**Interoperability & HIE**	PHI moved in ZIP archives between providers lacks a "Chain of Custody" for the container itself.	**HIPAA Compliance:** Provides a deterministic audit trail, proving medical records were not reordered or substituted during transit.
Software Supply Chain	**Secure Build Distribution**	CI/CD pipelines produce ZIP artifacts where an attacker can swap a single .dll without breaking functional validity.	**Binary Provenance:** Binds every file name and size to the container identity, halting deployment if a mismatch occurs.

Conclusion

ZIP archives were originally designed for a simpler era of computing - one focused on the efficient packaging and transport of data, rather than providing a rigorous foundation for trust. In today's contested threat landscape, where data traverses complex, decentralized networks and frequently crosses sensitive trust boundaries, that historical distinction is no longer sustainable. The widespread reliance on ZIP as a primary vehicle for mission-critical data, from tactical edge intelligence and healthcare records to highly regulated software supply chain artifacts, demands a fundamental evolution in how container integrity is envisioned, established and verified.

VeraFile meets this critical need by shifting the security paradigm away from infrastructure-dependent models. Rather than relying solely on perimeter defenses or external validation services, VeraFile embeds a deterministic identity directly into the structure of the archive itself. By utilizing Validation Integrity Keys (VIKs) at both the file and container levels, it transforms the ZIP from a passive, malleable transport mechanism into an active, self-validating entity. Crucially, VeraFile achieves this continuous verification without using or managing keys, entirely eliminating the massive administrative overhead, certificate lifecycle management, and connectivity requirements traditionally associated with complex cryptographic infrastructure.

This artifact-centric approach directly supports the rigorous security postures required by modern compliance and regulatory frameworks. As organizations strive to meet stringent mandates - such as those outlined by NIST and DFARS, the ability to mathematically prove the unalterable provenance and structural integrity of data artifacts becomes paramount. VeraFile provides the deterministic auditability that these Zero Trust architectures demand, ensuring that no implicit trust is ever granted to a data container simply because it arrived via a secure channel or bypassed a network filter.

Ultimately, VeraFile closes a critical and frequently overlooked gap in secure data transport. It establishes a new standard for digital exchange: one in which trust is never assumed from context but is instead proven by the artifact itself. By ensuring that every archive carries a durable, verifiable, and infrastructure-independent identity across systems, environments, and time, VeraFile lays the groundwork for a more resilient, scalable, and genuinely secure digital future.