CycloneDX

Authoritative

Guide to Attestations

Implement machine-readable standards to digitally transform audit and attestation workflows

OWASP

First Edition

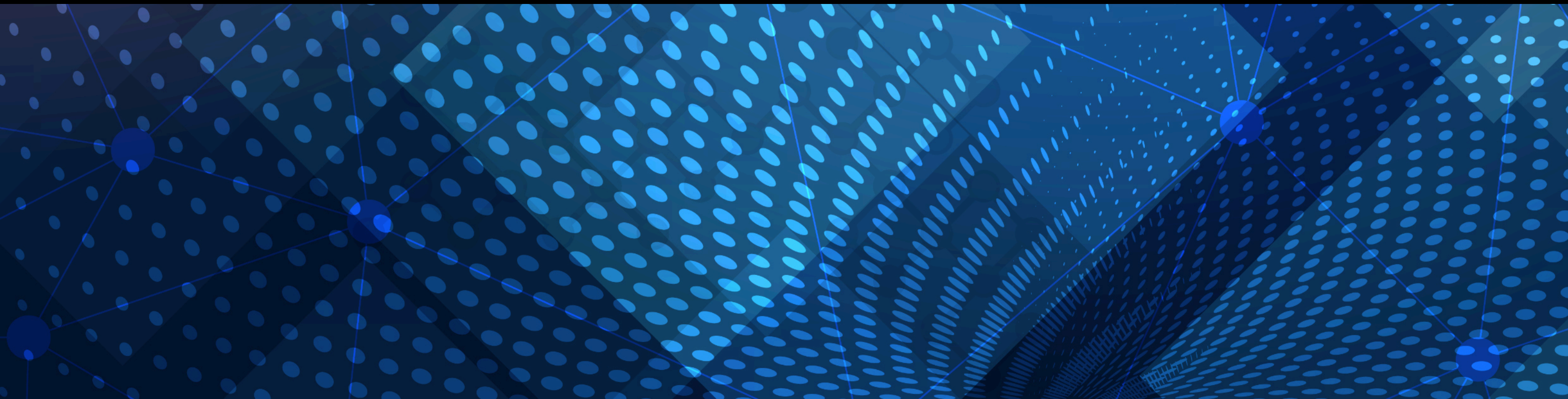

Table of Contents

About the Guide

CycloneDX is a modern standard for the software supply chain.

The content in this guide results from continuous community feedback and input from leading experts in the software supply chain security field. This guide would not be possible without valuable feedback from the CycloneDX Industry Working Group (IWG), the CycloneDX Core Working Group (CWG), the many CycloneDX Feature Working Groups (FWG), Ecma International Technical Committee 54 (TC54), and a global network of contributors and supporters.

Copyright and License

First Edition, 09 April 2024

Version	Changes	Updated On	Updated By
First Edition	Initial Release	2024-04-09	CycloneDX Feature Working Group on Attestations

Preface

Welcome to the Authoritative Guide series by the OWASP Foundation and OWASP CycloneDX. In this series, we aim to provide comprehensive insights and practical guidance, ensuring that security professionals, developers, and organizations alike have access to the latest best practices and methodologies.

At the heart of the OWASP Foundation lies a commitment to inclusivity and openness. We firmly believe that everyone deserves a seat at the table when it comes to shaping the future of cybersecurity standards. Our collaborative model fosters an environment where diverse perspectives converge to drive innovation and excellence.

In line with this ethos, the OWASP Foundation has partnered with Ecma International to create an inclusive, community-driven ecosystem for security standards development. This collaboration empowers individuals to contribute their expertise and insights, ensuring that standards like CycloneDX reflect the collective wisdom of the global cybersecurity community.

One standout example of this model is OWASP CycloneDX, which is on track to becoming an Ecma International standard through Technical Committee 54 (TC54). By leveraging the strengths of both organizations, CycloneDX is poised to become a cornerstone of security best practices, providing organizations with a universal standard for software and system transparency.

As you embark on your journey through this Authoritative Guide, we encourage you to engage actively with the content and join us in shaping the future of cybersecurity standards. Together, we can build a safer and more resilient digital world for all.

Andrew van der Stock
Executive Director, OWASP Foundation

The Innovative History of OWASP CycloneDX

OWASP CycloneDX has carved a legacy steeped in innovation, collaboration, and a commitment to openness. OWASP continues to advance software and system transparency standards, prioritizing capabilities that facilitate risk reduction.

March 2018

OWASP CycloneDX v1.0

First general-purpose, security-focused Bill of Materials standard supporting software and hardware components. Introduced the world to Package URL for use with software security use cases.

March 2019

OWASP CycloneDX v1.1

First specification with complete pedigree support describing component lineage and the commits, patches, and diffs which make a forked version unique.

May 2020

OWASP CycloneDX v1.2

First specification to incorporate SWID (ISO/IEC 19770-2:2015) and services into inventory including data classifications, providers, and relationships between services and components.

May 2021

OWASP CycloneDX v1.3

First specification to incorporate support for composition completeness surpassing NTIA's framing of "known unknowns".

January 2022

OWASP CycloneDX v1.4

First specification to introduce vulnerability sharing and transparency, including Vulnerability Disclosure Reports (VDR) and Vulnerability Exploitability eXchange (VEX).

June 2023

OWASP CycloneDX v1.5

First specification to support AI Transparency, configuration and data components, and formulation describing how components were created, tested, trained, evaluated, and deployed.

December 2023

Ecma TC54 Established

First working group chartered with holistic supply chain goals of standardizing core data formats, APIs, and algorithms that advance software and system transparency.

April 2024

OWASP CycloneDX v1.6

First specification to support cryptographic assets for Post-Quantum Cryptography (PQC) readiness and first general-purpose attestation specification to digitally transform audit and attestation workflows.

Source: https://tc54.org/history

Introduction

CycloneDX Attestations is a modern standard for security compliance. It enables organizations to use a machine-readable format for communication about security standards, claims, and evidence about security requirements, as well as attestations to the veracity and completeness of those claims. You can think of Attestations as a way to manage "compliance as code." The Attestations project began in 2023 as part of the broader CycloneDX project.

CycloneDX Attestations is part of OWASP CycloneDX. CycloneDX is an OWASP flagship project, has a formal standardization process and governance model through Ecma Technical Committee 54, and is supported by the global information security community.

Intended Audience

CycloneDX Attestations is intended for use by:

- Software development teams that want to meet security requirements and automate security evidence generation and communication.
- Security teams that want to ensure the security and compliance of software projects being created and manage the compliance process with assessors.
- Executives who are required to attest to their compliance with security standards.
- Security assessors that want to have a standard way of processing compliance documentation and tracking compliance.
- Security tool providers that build software for managing compliance processes.
- Security standard creators that want to create a machine-readable version of their requirements.

Problem Statement

Currently, organizations use a variety of paper-based and non-standard electronic documents to communicate security requirements, evidence, and attestations. The labor required to create, process, manage, update, and track these documents is expensive, labor-intensive, and often overwhelming. There are often large gaps between what the original requirement envisioned and the argument presented by the software producer. Similarly, assessors often misinterpret requirements and focus on minutae instead of the intent of the original requirement.

The problem is widespread, and many articles explain why compliance is not the same as security. If the security requirements represented the shared security interests of all stakeholders, then security and compliance would be aligned. Unfortunately, in most cases, at least some of the requirements make no sense to apply to a product, and many critical aspects of security are not reflected in the requirements.

The root cause of these issues is a fundamental communications problem. Security requirements don't often match up well with the expected threats for a particular real-world system and its defenses. Further, security requirements are often too abstract for development organizations to clearly understand what they must do with their particular organization, processes, and technologies. The assessors who should be facilitating the interpretation of the requirements in the context of the actual system are often relegated to a strict interpretation of the words in vague, high-level requirements.

Our challenge is encouraging standards bodies, builders, and assessors to communicate effectively. All the parties need a way to ensure that the *intent* of each requirement is applied appropriately to a particular product or system and achieved with confidence.

How CycloneDX Attestations Addresses Challenge

While CycloneDX Attestations may not completely eradicate this issue, they offer a solution that significantly reduces the paperwork burden. CycloneDX Attestations fosters a more efficient and streamlined interaction process by facilitating communication in a standard, machine-readable format.

We believe:

- The use of machine-readable standards in Attestations format will encourage faster and deeper understanding by all parties.
- The Attestations claims and evidence approach will allow development organizations to articulate their compliance rationale quickly and clearly.
- The use of Attestations will enable all forms of assessors, certifiers, and accreditors to quickly evaluate compliance and provide feedback to producers.
- Attestations will enable faster compliance feedback loops and fewer surprises and delays.

Intended Use Cases

Cyclone DX Attestations provides a non-repudiatable way to communicate compliance to standards. It is intended to be used in a variety of use cases, including:

- Standard authorities - Authors of security standards that want to create a machine-readable version of their requirements. E.g., Cyclone DX may be used to represent the requirements of the OWASP Software Security Framework. NIST may use Cyclone DX to represent the requirements of CISA attestations form for federal agencies.
- Providers in highly regulated verticals - Adherence to existing regulatory and industry compliance requirements like PCI DSS, HIPAA, NIST, etc. Cyclone DX can be used to provide non-repudiatable evidence of compliance.
- Providers who want to build trust with their customers by demonstrating compliance with a specific security standard - Adherence to a specific security standard like ISO 27001, NIST 800-53, etc.
- Policy as code in Governance, Risk, and Compliance (GRC) teams - GRC teams may use Cyclone DX to represent internal policies and security standards requirements. This can be used to automate the compliance process, collect and manage evidence of compliance. This can in turn be used to estimate the risk of non-compliance and provide assurance to the board and other stakeholders.
- Evidence as code for Engineering teams - Cyclone DX makes it easy for engineering teams to collect and manage evidence of compliance with internal security standards. Engineering teams can automate the process of collecting evidence and provide assurance to the security and GRC team that they are compliant with the internal security standards.
- Consumers who want to restrict the use of software to only those that meet specific security standards - Consumers of software may use Cyclone DX to ensure that the software they are using meets specific security standards. This can be used to reduce the risk of using software that does not meet specific security standards.

Tool Support

Over time, we expect tools to emerge to manage all aspects of security attestation. As a producer, imagine being able to select appropriate standards for a project, eliminate duplication, articulate compliance rationales, automatically generate and include supporting evidence, manage reviews, and digitally sign attestations. From the assessor's point of view, imagine being able to quickly evaluate claims and evidence, easily identify changes, point out gaps, and digitally sign approvals.

Standards

In CDXA, a "standard" is just a collection of security requirements. Each "standard" has a version number, description, an owner, and a list of requirements. The requirements themselves may be very specific and concrete, with best practices, guidance, or even just principles. That's up to the standard creator. Many security standards are available in CDXA. You can see a list below.

In CDXA every requirement has:

- Identifier - should tie back to the original standard as much as possible
- Title - a short description
- Text - the actual text of the requirement
- Descriptions - an array of supplemental text that provides guidance but is not directly part of the text
- OpenCRE Identifier (where possible)
- Parent (to support a hierarchy of requirements)
- External References

Creating Your Own Standard

In CDXA, you're free to create your own security standard. It could be a subset or superset of an existing standard. There are a lot of good reasons to tailor a security standard to your particular system. But remember, you may be required to follow one or more external security standards. As we move into making claims and substantiating those claims in CDXA, you'll see how you can capture your approach to existing requirements to show compliance.

Original Standard	CDXA Version
NIST Secure Software Development Framework (SSDF)	CDXA-SSDF
PCI Secure SLC Standard	CDXA-PCI-SSLC
Build Security In Maturity Model (BSIMM)	CDXA-BSIMM
OWASP Application Security Verification Standard (ASVS)	CDXA-OWASP-ASVS
OWASP Mobile Application Security Verification Standard (MASVS)	CDXA-OWASP-MASVS
OWASP Software Component Verification Standard (SCVS)	CDXA-OWASP-SCVS

Making Attestations

CycloneDX Attestations (CDXA) represent a automated approach to enhancing security and compliance across various standards. CycloneDX facilitates the representation of any standard—whether security-focused or otherwise—along with the specific requirements associated with that standard. This capability ensures a broad applicability across different domains and compliance needs.

Building upon this foundational support, CycloneDX further empowers organizations by enabling them to make formal attestations against these defined standards and requirements. These attestations are comprised of claims that are supported by concrete evidence, providing a robust mechanism for demonstrating compliance or security posture. Importantly, CycloneDX acknowledges the complexity of real-world scenarios by also accommodating counter-claims and counter-evidence, thus fostering a comprehensive and nuanced view of compliance and security assessments.

In instances where a requirement is not being fully met, CycloneDX's versatile framework allows for the specification of mitigation strategies. This feature ensures that organizations can transparently communicate their efforts to address potential shortcomings, thereby maintaining trust and integrity in the face of compliance challenges. Through these capabilities, CycloneDX Attestations serve as a powerful tool for organizations striving to navigate the complex compliance landscape with confidence and clarity.

Claims

Claims serve as the medium through which organizations can articulate their argument for meeting a specific requirement. You can dissect the requirement into a series of claims that address some part of the overall requirement. There are numerous ways to structure your claims, but it's important to remember that the simplest and most straightforward arguments often prove to be the most effective.

For example, you may want to create a claim about each major module in a complex system. Or you might make claims about several separate aspects of a security defense. Sometimes, a single claim is enough to cover the entire requirement.

A Claim is simply a statement that captures at least one aspect of how a certain requirement has been satisfied. A claim has two key parts: a target and a predicate. Claims often restate the requirement using specific terms related to the defenses in the system.

- Target: Each claim has a target that is the subject of the claim. The target might be the specific name of an entire system, a module, a process, a team, a business unit, or a company. In many cases, the target is simply an interpretation of the requirement for the current attestation. For example, the target might be "Acme Corporation" or "The Mxyzptlk Module."

- Predicate: Each claim also has a predicate that states what is being claimed about the target. Once again, this is often a specific interpretation of the requirement that details exactly what was done to meet the requirement. For example,

The following table details the makeup of a claim.

Property	Description
bom-ref	An identifier, unique to the CDXA document, that identifies the claim.
target	The bom-ref to a target representing a specific system, application, API, module, team, person, process, business unit, company, etc... that this claim is being applied to.
predicate	The specific statement or assertion about the target.

Property	Description
mitigationStrategies	The list of bom-ref to the evidence provided describing the mitigation strategies. Each mitigation strategy should include an explanation of how any weaknesses in the evidence will be mitigated.
reasoning	The written explanation of why the evidence provided substantiates the claim.
evidence	The list of bom-ref to evidence that supports this claim.
counterEvidence	The list of bom-ref to counterEvidence that supports this claim.

For example, consider the requirement, "All developers must receive security training." An appropriate claim might be that "All members of the Acme development team have taken the HackMe Secure Coding for Java training course and received a passing grade on the final test." The target is "All members of the Acme development team," and the predicate is "have taken the HackMe Secure Coding for Java training course and received a passing grade on the final test."

```
"claims": [
  {
    "bom-ref": "claim-1",
    "target": "acme-inc",
    "predicate": "Developers have taken the HackMe Secure Coding for Java training course and received a passing grade on the final test.",
    "mitigationStrategies": [ "mitigationStrategy-1" ],
    "reasoning": "The provided evidence shows that 70% of developers have been trained this year.",
    "evidence": [ "evidence-1" ],
    "counterEvidence": [ "counterEvidence-1" ],
    "signature": {
      "algorithm": "ES256",
      "certificatePath": [ "MIIB...", "MIID..." ],
      "value": "tqIT..."
    }
  }
]
```

Substantiating Claims With Evidence

CycloneDX Attestations is a standard that enables organizations to build an argument explaining how they meet security requirements. Organizations can start this process very early in development, using attestations to plan their security strategy for meeting each requirement and capturing the general reasoning that will be used.

As the system is built and tested, the team can capture the details of their implementation with specific claims. Ultimately, they can complete the argument by generating and gathering evidence to support each claim and creating specific reasoning that explains how the evidence demonstrates the claims are well supported and that the overall requirement has been met.

Let's explore the claims, evidence and reasoning that attest to how a requirement has been satisfied. Remember, a single requirement may be supported by multiple claims and a claim may be supported by many different pieces of evidence, tied together by some reasoning.

Evidence

Once we have a specific claim, we can consider the types of evidence that might support that claim. The best evidence directly supports the target and predicate in the claim. Indirect evidence is valuable, but generally can't stand alone and will need more explanation in the reasoning section below. For example, it's better to know that the software was extensively tested for SQL Injection than it would be to know that SQL Injection is covered in the organization's secure coding guide.

- Evidence: A list of references to evidence that supports this claim. Evidence can include metrics, observations, test results, analysis, surveys, sampling, expert opinion, and more. It's important to not only present output as evidence, but also to capture the metadata about how the evidence was created, any relevant configuration details, expertise, etc...

- Counter Evidence: A list of references to evidence that contradicts this claim. Often, evidence will be created that directly or partially contradicts a claim. For example, a security testing tool discovers a gap in security defenses, such as a lack of authorization on a particular interface.

- Mitigation Strategy: If the evidence is not compelling or counter evidence is present, the producer can detail their plans for improving conformance in this section. Ideally, the strategy should detail the change or enhancement, the rough schedule, and the expected effect on the evidence supporting the claim.

The following table details the makeup of evidence.

Property	Description
bom-ref	An identifier, unique to the CDXA document, that identifies the evidence.
propertyName	The reference to the property name as defined in the CycloneDX Property Taxonomy.
description	The written description of what this evidence is and how it was created.
data	The output or analysis that supports claims.
created	The date and time (timestamp) when the evidence was created.
expires	The optional date and time (timestamp) when the evidence is no longer valid.

Property	Description
author	The author of the evidence.
reviewer	The reviewer of the evidence.

```
"evidence": [
 {
   "bom-ref": "evidence-1",
   "propertyName": "internal.com.acme.someProperty",
   "description": "Description here",
   "data": [
     {
       "name": "Name of the data",
       "contents": {
         "attachment": {
           "content": "Evidence here",
           "contentType": "text/plain"
         }
       },
       "classification": "PII",
       "sensitiveData": [ "Describe sensitive data here" ]
     }
   ],
   "created": "2023-04-25T00:00:00+00:00",
   "expires": "2023-05-25T00:00:00+00:00",
   "author": { "name": "Mary" },
   "reviewer": { "name": "Sanford" },
   "signature": {
     "algorithm": "ES256",
     "certificatePath": [ "MIIB...", "MIID..." ],
     "value": "tqIT..."
   }
 }
]
```

Reasoning

Simply providing claims and evidence is not sufficient to determine whether the claim was satisfied. We need some reasoning that explains how all the evidence, counter evidence, and mitigation strategy work together to support the claim. The best reasoning is simple and direct, and ties back to the claim as a straighforward argument.

- Reasoning: An argument that explains how the provided evidence, counter evidence, and mitigation strategy combine to support the overall claim.

Your reasoning for a claim about not being susceptible to SQL injection might tie together several pieces of evidence (and address any counter evidence). For example, a strong argument would be that "Acme corporation policy, secure coding guidelines, and training progam ensure that developers are aware of SQL injection and how to prevent it. All database access is performed through Hibernate which uses parameterized SQL queries in most cases. We test our software for all types of SQL injection using IAST and have remediated all instances discovered. Finally, we use RASP in production to detect SQL injection attacks and prevent exploitation."

Other

- External References: A claim can reference any systems, sites, and information that may be relevant, but are not included with the BOM. They may also establish specific relationships within or external to the BOM.
- Signature: Enveloped signature in JSON Signature Format (JSF).

Another Example

Claim: The software component complies with the SLSA framework.

Evidence:

- A code review report that was generated by a qualified code reviewer.
- A build log that shows that the software component was built using a secure build system.
- A cryptographic signature of the software component, generated using a digital certificate issued by a trusted certificate authority.

Reasoning: The evidence in this example provides support for the claim that the software component complies with the SLSA framework. The code review report provides evidence that the source code of the software component has been reviewed by a qualified person. The build log provides evidence that the software component was built using a secure build system. The cryptographic signature provides evidence that the component used was verified by a trusted entity.

Documenting Non-Conformance

Non-conformance of requirements occurs when something - a product, service, process, or system - fails to meet its intended specifications or established regulations. This deviation can be minor, like a typo in a document, or major, like a safety breach. Regardless of severity, non-conformance carries risks, impacting quality, performance, and potentially safety. Identifying and addressing it promptly is crucial. Many organizations document non-conformance as part of a risk management process.

Conformance

CycloneDX Attestations documents conformance through an attestation that maps requirements, claims, counterClaims and a conformance. The conformance documents the ability of the claims to satify the requirements.

The conformance as three fields.

- score: The conformance of the claim between and inclusive of 0 and 1, where 1 is 100% conformance.
- rationale: The reasoning for the conformance score.
- mitigationStrategies: The list of evidence describing the mitigation strategies.

Instead of specifying that a claim is non-conforming, the attestation will have a lower conformance score for the portion of the requirement not met. Additionally the rationale describes what the non-conform is as a text statement, and the mitigationStrategies is evidence that alleviates the risk of the non-conformance.

Score

The score can either be a binary conformance/non-conformance or a percentage between 0 and 1, where 1 is 100% conformance.

Examples:

- A requirement for all employees to complete security training. 70 of the 100 employees has completed it. The conformance score is 0.7.
- A requirement for separate of accounts for elevated permissions from user accounts. Administrators use their user accounts for all elevated actions with no separation. The conformance score is 0.0.

Mitigation Strategies

Mitigation strategies are actions taken to reduce the severity or likelihood of a negative outcome. They are an essential part of risk management by reducing risk associated with non-conformance.

CycloneDX Attestations documents mitigationStrategies as bomLinks to evidence. This evidence details the action taken to mitigate the non-conformance.

Each mitigation strategy should include an explanation of what part of the non-conformance is being addressed. This explanation should be included as part of the conformance rationale or within the description of the evidence.

Plan of Action and Milestones (POAM)

A POAM is process used in cybersecurity and risk management to document risks, and is an opportunity to strengthen or “harden” your system through carefully planned improvements.

Common contexts where POAMs are used:

- NIST Cybersecurity Framework (CSF): Organizations working towards compliance with NIST CSF may use POAMs to address identified gaps.
- Federal Information Systems Management Act (FISMA): US government agencies use POAMs to document corrective actions for security vulnerabilities.
- Defense Contract Management Agency (DCMA): Defense contractors use POAMs to demonstrate plans for achieving cybersecurity requirements.
- Internal risk management: Organizations can use POAMs to address various internal risks beyond cybersecurity.

POAMs work well with CycloneDX Attestations and reference attestation and mitigationStrategies as bomLinks.

Counter Evidence

Much like evidence is used to document supporting a claim, counterEvidence is used document evidence that contradicts a claim. This can provide verification that 100% conformance is not met.

Signing

CycloneDX supports signing to ensure the authenticity and integrity of the attestations. This is crucial for

- Verification of Origin: Signed SBOMs and attestations provide verifiable evidence of their origin, preventing unauthorized tampering or impersonation.
- Non-Repudiation: Signing prevents the signer from denying the authenticity of the signed content, ensuring accountability and traceability.
- Integrity Assurance: Signing guarantees that the signed content has not been modified or altered after it was signed, safeguarding the integrity of the information.

Signatories

The personas that could sign claims, evidence, standards, or attestations can be categorized into two main groups:

- Producers: These are the entities responsible for creating and managing the SBOMs and attestations. They could include:
  - Software developers: Developers of software components or applications are often responsible for generating claims that accurately reflect their Software Development Lifecycle (SDL).
  - Software distributors: Distributors of software packages may create SBOMs that aggregate the SBOMs of individual components included in their packages and provide assurances for the software packages in the form of assurances.
  - Build or deployment tools: Automated build or deployment tools can generate SBOMs and evidence for supporting the claims required by the developers as part of their workflow, ensuring that the SBOMs are up-to-date with the latest code changes and the claims are supported .
  - Software supply chain security tools: Security tools can analyze SBOMs to identify potential vulnerabilities, compliance issues, or other relevant information and generate evidence of security testing and validation.
- Verifiers: These are the entities responsible for validating and verifying the authenticity and integrity of SBOMs or attestations. They could include:
  - Organizations consuming software: Organizations that use or integrate software components may require attestations that meet a certain standard. (e.g: PCI-DSS certified software)
  - Regulatory bodies or auditors: Regulatory bodies or auditors may use the claims, evidence and attestation generated by the software producers as part of compliance audits or certification processes.

In some cases, a single entity may act as both a producer and a verifier of SBOMs or attestations. For example, a software development company may generate SBOMs and attestations for its own products and also verify them from its suppliers.

The specific personas involved in signing CycloneDX claims, evidence, standards, or attestations will depend on the specific context and the intended use of the attestations. However, the producer and verifier roles are typically involved in establishing trust and ensuring the integrity of the information being conveyed.

Digital Signatures

CycloneDX supports multiple digital signing methods to accommodate various preferences and security requirements. These methods include:

- XML Signature (xmlsig): This is a widely used XML-based signing format that provides a standard way to sign XML documents.
- JSON Signature Format (JSF): This is a lightweight JSON-based signing format that offers a more compact and efficient alternative to XML Signature for signing CycloneDX attestations.
- Detached Signatures: Detached signatures allow the signature to be stored separately from the signed content, providing flexibility and reducing the size of the signed document.

Analog Signatures

CycloneDX supports use of analog signatures for signing CycloneDX attestations. Analog signatures are typically used for signing paper documents and are not as secure as digital signatures. However, they can be useful in some cases where digital signatures are not available or practical. CycloneDX does not mandate the use of analog signatures, and the specific format of an analog signature will vary depending on the context and the signing mechanism used. However, CycloneDX recommends the use of the following format for analog signatures:

- Signature Image: A scanned image of the signature.
- Signature Text: The text of the signature.
- Signature Date: The date when the signature was created.
- Signature Location: The location where the signature was created.
- Signature Type: The type of signature (e.g: handwritten, electronic, etc.)

Here are some specific examples of when analog signatures might be used in CycloneDX attestations:

- Verifying the identity of a software supplier: When a software supplier is required to provide an attestation to a customer, the supplier may use an analog signature to verify their identity and the authenticity of the attestation.
- Attesting to the compliance of software with regulations: An analog signature may be used to attest to the software's compliance.

In general, analog signatures are a useful way to add an extra layer of assurance to CycloneDX attestations, especially when dealing with physical documents or legacy systems.

Signing for Authenticity

CycloneDX supports signing to ensure the authenticity and integrity of the attestations. This is crucial for detecting unauthorized tampering or impersonation. Digital signing methods are used to ensure the authenticity of the attestations.

Digital Signatures

CycloneDX supports multiple digital signing methods to accommodate various preferences and security requirements - xmlsig, JSF, and detached signatures.

The following are examples of digital signatures in a CycloneDX attestation:

JSON

```
"signature": {
  "algorithm": "RS512",
  "publicKey": {
    "kty": "RSA",
    "n": "qOSWbDOGS31lv3aUZVOgqZyLVrKXXRfmxFQxEylc...",
    "e": "AQAB"
  },
  "value": "HGIX_ccdIcqmaOpkxDzKH_j0ozSHUAUyBxGpXS..."
}
```

XML

```
<attestation>
  <signature algorithm="RS512">
    <publicKey>
      <kty>RSA</kty>
      <n>qOSWbDOGS31lv3aUZVOgqZyLVrKXXRfmxFQxEylc...</n>
      <e>AQAB</e>
    </publicKey>
    <value>HGIX_ccdIcqmaOpkxDzKH_j0ozSHUAUyBxGpXS...</value>
  </signature>
</attestation>
```

Example

The following is a full example of a CycloneDX attestation.

```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.6",
  "serialNumber": "urn:uuid:3e671687-395b-41f5-a30f-a58921a69b79",
  "version": 1,
  "declarations": {
    "assessors": [
      {
        "bom-ref": "assessor-1",
        "thirdParty": true,
        "organization": {
          "name": "Assessors Inc"
        }
      }
    ],
    "attestations": [
      {
        "summary": "Attestation summary here",
        "assessor": "assessor-1",
        "map": [
          {
            "requirement": "requirement-1",
            "claims": [ "claim-1" ],
            "counterClaims": [ "counterClaim-1" ],
            "conformance": {
              "score": 0.8,
              "rationale": "Conformance rationale here",
              "mitigationStrategies": [ "mitigationStrategy-1" ]
            },
            "confidence": {
              "score": 1,
              "rationale": "Confidence rationale here"
            }
          }
        ],
        "signature": {
          "algorithm": "ES256",
          "certificatePath": [ "MIIB...", "MIID..." ],
          "value": "tqIT..."
        }
      }
    ],
    "claims": [
      {
        "bom-ref": "claim-1",
        "target": "acme-inc",
        "predicate": "Predicate here",
        "mitigationStrategies": [ "mitigationStrategy-1" ],
        "reasoning": "Reasoning here",
        "evidence": [ "evidence-1" ],
        "counterEvidence": [ "counterEvidence-1" ],
        "externalReferences": [
          {
            "type": "issue-tracker",
            "url": "https://alm.example.com"
          }
        ],
        "signature": {
          "algorithm": "ES256",
          "certificatePath": [ "MIIB...", "MIID..." ],
          "value": "tqIT..."
        }
      }
    ],
    "evidence": [
      {
        "bom-ref": "evidence-1",
        "propertyName": "internal.com.acme.someProperty",
        "description": "Description here",
        "data": [
          {
            "name": "Name of the data",
            "contents": {
              "attachment": {
                "content": "Evidence here",
                "contentType": "text/plain"
              }
            },
            "classification": "PII",
            "sensitiveData": [ "Describe sensitive data here" ]
          }
        ],
        "created": "2023-04-25T00:00:00+00:00",
        "expires": "2023-05-25T00:00:00+00:00",
        "author": {
          "name": "Mary"
        },
        "reviewer": {
          "name": "Jane"
        },
        "signature": {
          "algorithm": "ES256",
          "certificatePath": [ "MIIB...", "MIID..." ],
          "value": "tqIT..."
        }
      },
      {
        "bom-ref": "counterEvidence-1",
        "propertyName": "internal.com.acme.someProperty",
        "description": "Description here",
        "data": [
          {
            "name": "Name of the data",
            "contents": {
              "attachment": {
                "content": "Counter evidence here",
                "contentType": "text/plain"
              }
            },
            "classification": "Public",
            "sensitiveData": [ "Describe sensitive data here" ]
          }
        ],
        "created": "2023-04-25T00:00:00+00:00",
        "expires": "2023-05-25T00:00:00+00:00",
        "author": {
          "name": "Mary"
        },
        "reviewer": {
          "name": "Jane"
        },
        "signature": {
          "algorithm": "ES256",
          "certificatePath": [ "MIIB...", "MIID..." ],
          "value": "tqIT..."
        }
      },
      {
        "bom-ref": "mitigationStrategy-1",
        "propertyName": "internal.com.acme.someProperty",
        "description": "Description here",
        "data": [
          {
            "name": "Name of the data",
            "contents": {
              "attachment": {
                "content": "Mitigation strategy here",
                "contentType": "text/plain"
              }
            },
            "classification": "Company Confidential",
            "sensitiveData": [ "Describe sensitive data here" ]
          }
        ],
        "created": "2023-04-25T00:00:00+00:00",
        "expires": "2023-05-25T00:00:00+00:00",
        "author": {
          "name": "Mary"
        },
        "reviewer": {
          "name": "Jane"
        },
        "signature": {
          "algorithm": "ES256",
          "certificatePath": [ "MIIB...", "MIID..." ],
          "value": "tqIT..."
        }
      }
    ],
    "targets": {
      "organizations": [
        {
          "bom-ref": "acme-inc",
          "name": "Acme Inc"
        }
      ]
    },
    "affirmation": {
      "statement": "I certify, to the best of my knowledge, that all information is correct...",
      "signatories": [
        {
          "name": "Tom",
          "role": "CEO",
          "signature": {
            "algorithm": "ES256",
            "certificatePath": [ "MIIB...", "MIID..." ],
            "value": "tqIT..."
          }
        },
        {
          "name": "Jerry",
          "role": "COO",
          "organization": {
            "name": "Acme Inc"
          },
          "externalReference": {
            "type": "electronic-signature",
            "url": "https://example.com/coo-sig.png"
          }
        }
      ],
      "signature": {
        "algorithm": "ES256",
        "certificatePath": [ "MIIB...", "MIID..." ],
        "value": "tqIT..."
      }
    },
    "signature": {
      "algorithm": "ES256",
      "certificatePath": [ "MIIB...", "MIID..." ],
      "value": "tqIT..."
    }
  },
  "signature": {
    "algorithm": "ES256",
    "certificatePath": [ "MIIB...", "MIID..." ],
    "value": "tqIT..."
  }
}
```

OWASP