Developing a Trust Model for Security Automation Data

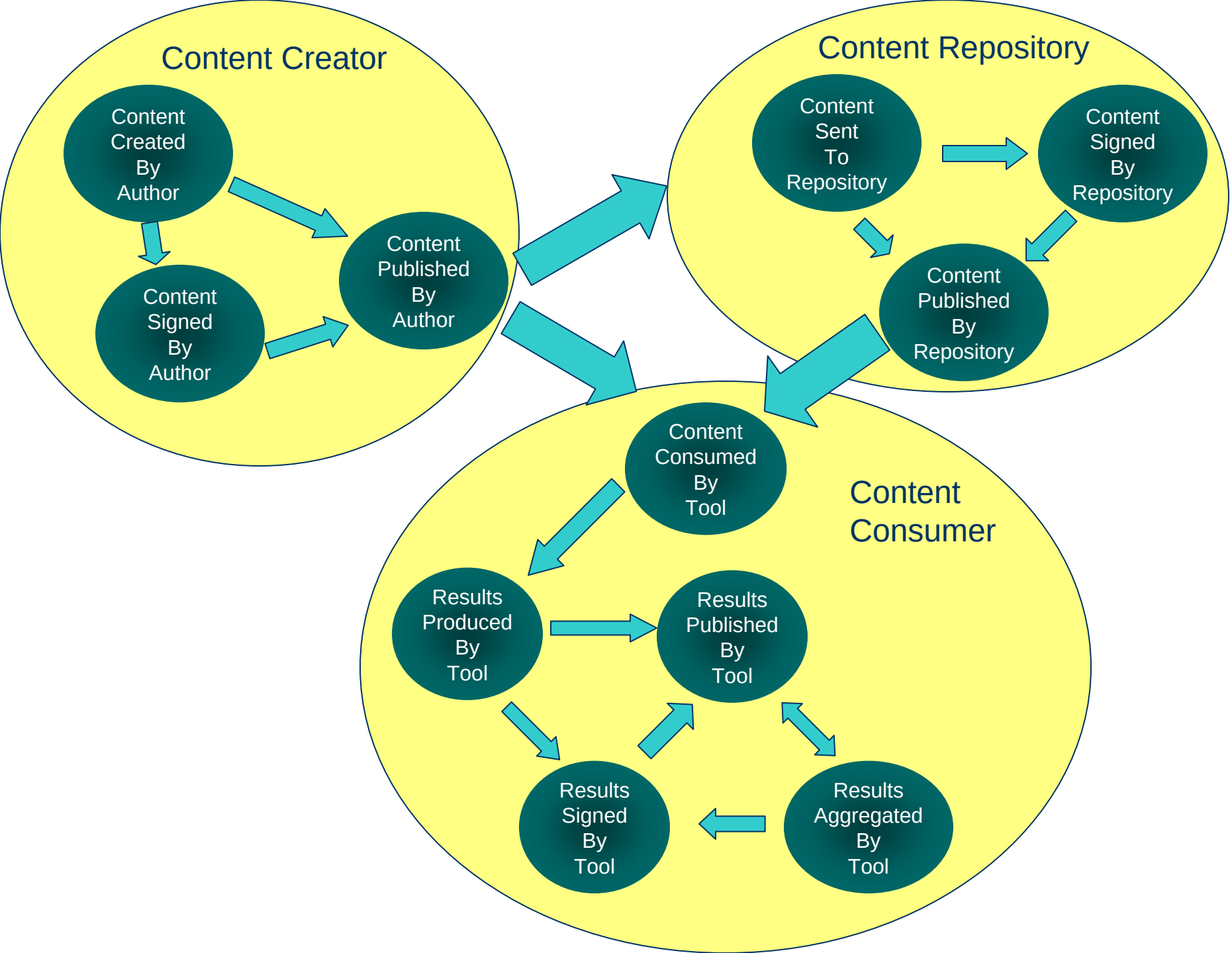
Harold Booth
NIST

Agenda

- Goals
- Use Cases
- Cryptographic Message Syntax (RFC 5652)
- XML Signature Syntax and Processing Overview
- Algorithms and Parameters
- Signature types
- Reporting
- Archiving

Vision

- Allow content to be created and trusted in a consistent way by end users
  - Content Consumers
  - Content Creators
  - Tool Vendors


Content Creator
Content Created By Author
Content Signed By Author
Content Published By Author
Content Repository
Content Sent To Repository
Content Signed By Repository
Content Published By Repository
Content Consumer
Content Consumed By Tool
Results Produced By Tool
Results Published By Tool
Results Signed By Tool
Results Aggregated By Tool

Design Goals

- Consistent
  - Interoperability between products
  - Useful to any content type
  - Support community based content
- Extensible
  - The approach will likely evolve over time
    - Minimize technology lock-in
    - Incremental improvements
  - Facilitate vendor extension/innovation

A Content Management Problem

- Reusability
  - Tailoring
  - Augmentation
- Versioning
- Delivery
  - Push
  - Pull
  - Publish/Subscribe
- Provenance
  - Authentication
    - Non-repudiation
    - Integrity
  - Authorization
  - Encryption

- Source Content
  - Compositional
- Results
  - Aggregation

Near-Term Goals

- Specification of result payloads
- Establish data integrity and trusted content
  - Foster content reuse
  - Enable quality assurance processes
- Express signatures in a common format
- Provide mechanism to establish provenance of source content and produced results
- Future version of SCAP

Future Goals

- Compositionality
  - Referential
  - Tailoring
- Encryption
- Authorization

Non-Goals

- Key exchange is out of scope

Content Use Case (input)

- A content consumer needs to verify authenticity of a content stream
  - Content published by an author or authority
  - Validate that content has not been altered since publication by the author or authority
  - Consumers can establish trust with respect to content based upon identity of author or authority

Content Use Case (prior knowledge)

- Re-establish trust to content based upon prior knowledge
  - Assist with solving referential trust
  - Could be used in lieu of using identity of the author or authority

Content Quality Assurance Use Case

- An individual or organization signs content to assert confidence or trust in content
  - QA function – works in a defined environment
  - Organizational policy asserts only trusted content may be run
  - Need to maintain provenance information – who originally published
    - Traceability

Compositional Content Use Case

- A content consumer would like to know and verify that a content stream is composed of multiple source streams
  - An author may compose a data stream from multiple data streams and augment with own contribution
  - Allow reporting of results derived from a source stream to be performed independently of other source streams
  - Focus QA efforts only on augmented portion
  - Identify differences between source stream and composed stream

Results Use Case

- An organization needs results signed at the point of creation in order to verify authenticity of results
  - Results generated by a tool

Results Use Case (expanded)

- An organization needs results signed with source content identity and/or target identity at the point of creation in order to verify authenticity of produced results
  - Results created based on responses of a machine endpoint (e.g. OVAL) or individual (e.g. OCIL) – a target
  - Expanded to include identity of source content and/or target
  - Establishes identity of tool, target, and source content
  - Assumes targets have an identity capability

Aggregated Results Use Case

- Aggregation tools need to combine results and sign aggregated results
  - Maintain source data to allow consumers of aggregated data to validate findings at a later point
  - Provides traceability of aggregated results

Cryptographic Message Syntax

- IETF RFC 5652
  - PKCS #7
- Treats content as binary data
- A variety of implementations already available

XML Signature Syntax and Processing Overview

- W3C Standard
- Specialized to handle XML data
  - Canonicalization
  - Transform
- Defers to applications for validation logic
  - Public key is optional
- Hooks for X.509 Certificates
- Implemented within Java SE 6
- Other implementations?

XML Signature Simple Example

```
<Signature>
  <SignedInfo>
    <SignatureMethod/>
    <CanonicalizationMethod/>
    <Reference>
      <Transforms>
      <DigestMethod>
      <DigestValue>
    </Reference>
    <Reference/> etc.
  </SignedInfo>
  <SignatureValue/>
  <KeyInfo />
  <Object />
</Signature>
```

XML Signature W3C Example

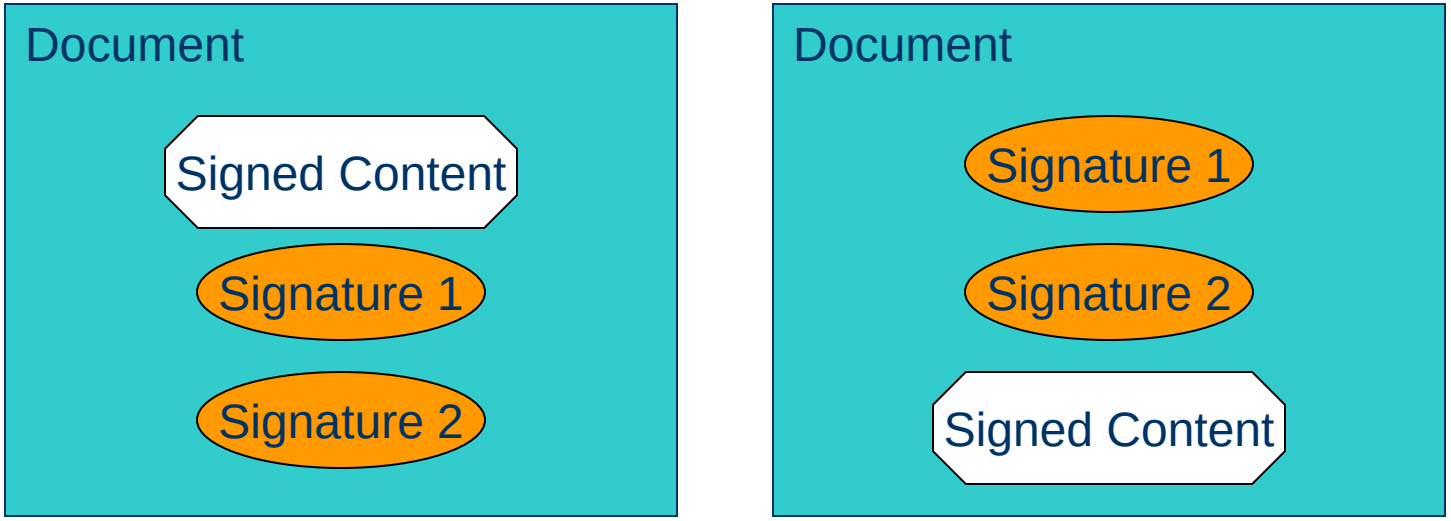
```
[s01] <Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
[s02] <SignedInfo>
[s03]  <CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
[s04]  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
[s05]  <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
[s06]  <Transforms>
[s07]   <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
[s08]  </Transforms>
[s09]  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[s10]  <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK.../DigestValue>
[s11]  </Reference>
[s12] </SignedInfo>
[s13] <SignatureValue>...</SignatureValue>
[s14] <KeyInfo>
[s15a] <KeyValue>
[s15b]  <DSAKeyValue>
[s15c]   <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
[s15d]  </DSAKeyValue>
[s15e] </KeyValue>
[s16] </KeyInfo>
[s17] </Signature>
```

Algorithms and Parameters

- Based on recommendations in FIPS 186-3
- RSA
  - 2048-bit key
  - SHA-256
  - PKCS #1.5 padding
- Elliptical Curve Digital Signature Algorithm
  - 256-bit Prime Curve
  - SHA-256

Enveloped

- Signature embedded within the document containing signed content

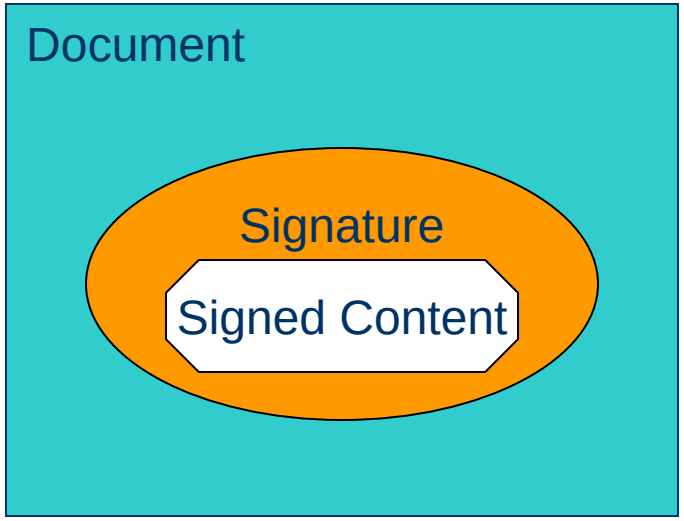


Enveloped Consequences

- Document must have a placeholder to hold the signature
  - Higher coordination costs between specifications to maintain consistency of use
- Signed/unsigned content has same content format
- Signature and content are coupled together

Enveloping

- Signature contains the signed content

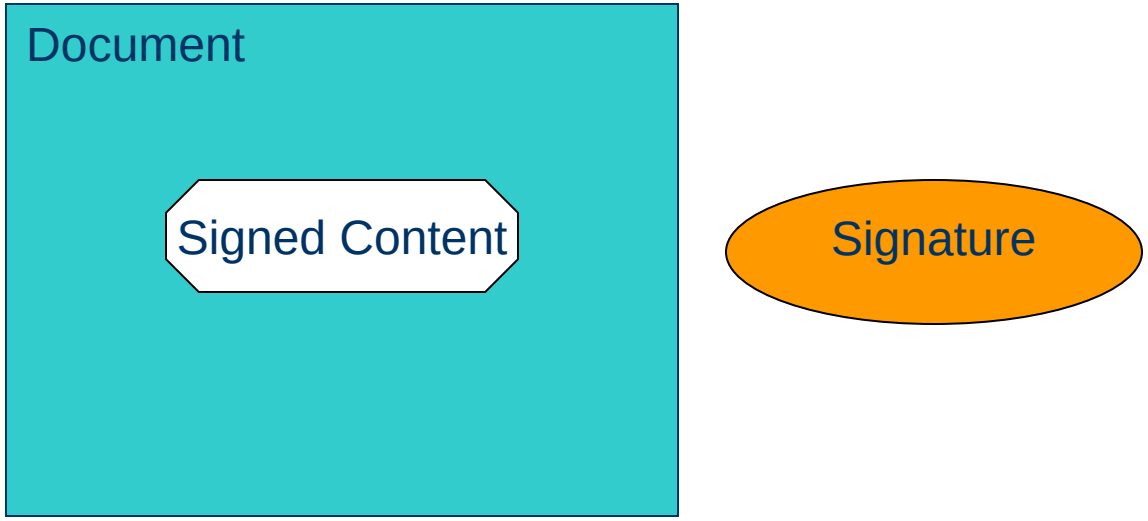


Enveloping Consequences

- Processing of document requires processing of signature syntax
  - Signed/unsigned content may have different formats
- Signature and content are coupled together

Detached

- Signatures are separate from the content



Detached Consequences

- Processing of signature and document are separated
  - Signed/Unsigned content is identical
- Signature format and content format can revision independently
- Signature and content are separated
  - Another thing to track

Reporting

- What additional information do we need to include?
  - Date
  - Tool Identity
  - Source Content
  - Target Identity
  - ?

Archiving

- Signing documents which may no longer be trusted
  - Key Expiration
  - Key Revocation
  - Weakness in crypto

Comments

emerging-specs@nist.gov

References

- XML Signature Syntax and Processing
  - http://www.w3.org/TR/xmldsig-core/
- Cryptographic Message Syntax (RFC 5652)
  - http://tools.ietf.org/html/rfc5652