Client-side XSLT, validation and data security

Wendell Piez

National Institute of Standards and Technology (NIST)

Information Technology Laboratory (ITL)

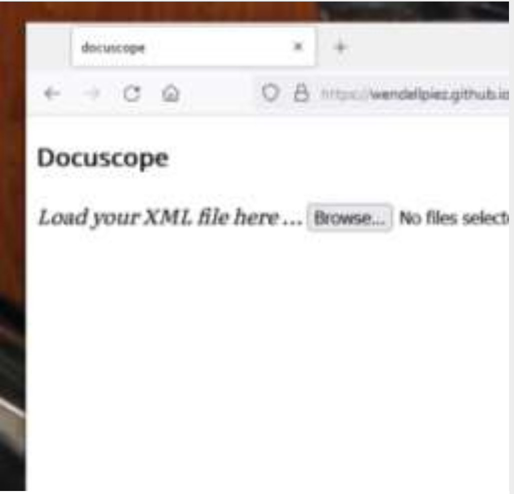
CSX – Client Side XSLT

New acronym, old thing …
… in different forms, around for a while …

What this talk is *not* about …

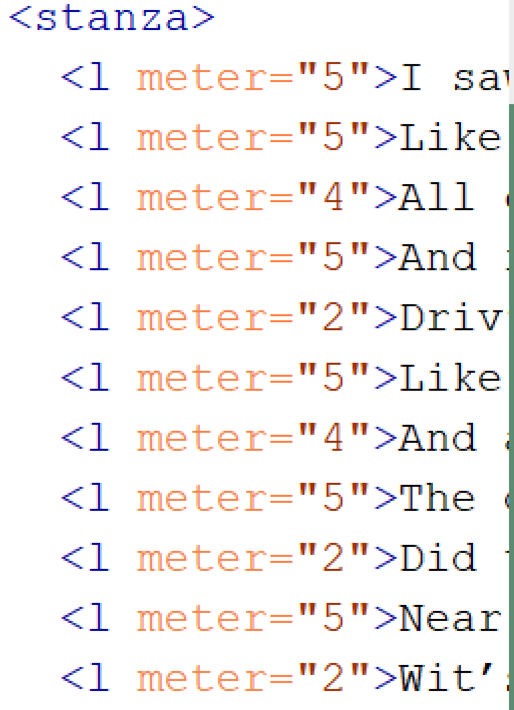
- ✗ The pros and cons of CSX
  (Or of SaxonJS in the browser)
- ✗ How you can do this
  (Although there are links)
- ✗ A claim or set of claims
  (Although there may be implications)

Two Questions



Docuscope

Load your XML file here ... Browse... No files select



```
<stanza>
  <l meter="5">I sa
  <l meter="5">Like
  <l meter="4">All
  <l meter="5">And
  <l meter="2">Driv
  <l meter="5">Like
  <l meter="4">And
  <l meter="5">The
  <l meter="2">Did
  <l meter="5">Near
  <l meter="2">Wit'
```

- What happens when the XML to be processed belongs to the user not the publisher?

- What happens as the web becomes a delivery platform for encapsulated processing logic?
  - Beyond (and by means of) http/HTML/JSON/Javascript
  - Declarative foundations
  - Distributed validation architectures

XML Jelly Sandwich demonstrations

XML-driven demonstrations using SaxonJS

Served via Github Pages

Today's demonstrations and others

Generic demonstration (any XML)

Docuscope

Document visualization and analysis

OSCAL demonstrations

OSCAL is the **Open Security Controls Assessment Language**

Supports documentary operations related to
systems security / compliance / risk management

https://wendellpiez.github.io/XMLjellysandwich/

Disclaimer: these projects have **no affiliation** with Saxonica (producers of SaxonJS) or NIST (the developer's day job) and only the developer is responsible for any errors or misrepresentations

Play along demonstrations

https://wendellpiez.github.io/XMLjellysandwich/

Download the zip file from the repository
Or try with your own XML files

Docuscope

Ever want to take a quick peek at an XML document?

Just show what's in it, no fuss?

But you want to see more than the code?

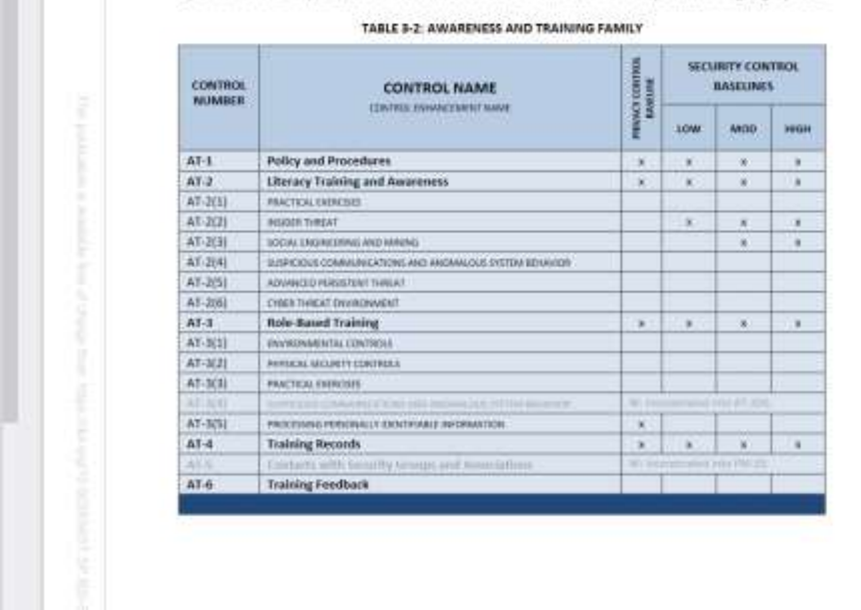
oscal/baseline-matrix

Showing our data looking pretty

Showing how we can make our data look pretty

Showing how what is "easy" is relative

Screen shot from PDF document as published

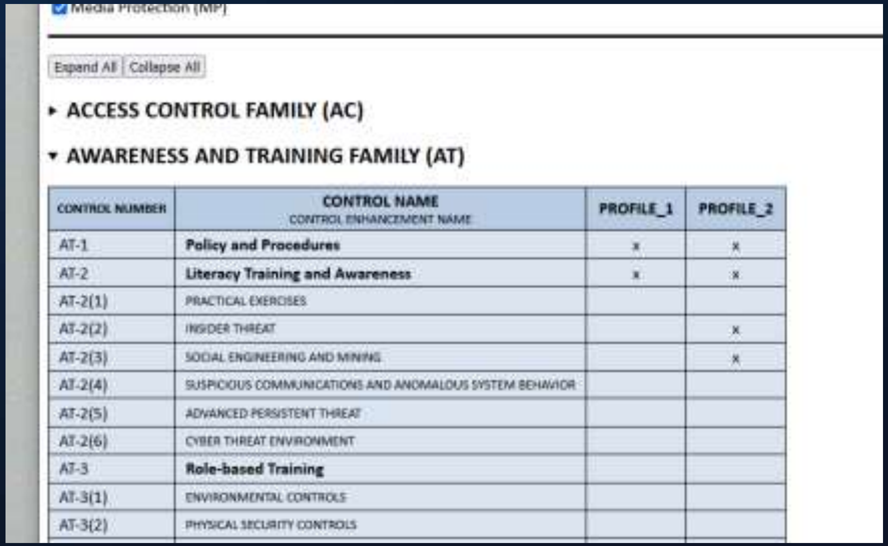


control or control enhancement that has been withdrawn from the control catalog is indicated by a "W" and an explanation of the control or control enhancement disposition in light gray text.

TABLE 3-2: AWARENESS AND TRAINING FAMILY

CONTROL NUMBER	CONTROL NAME CONTROL ENHANCEMENT NAME	PRIVACY CONTROL BASELINE	SECURITY CONTROL BASELINES		
			LOW	MOD	HIGH
AT-1	**Policy and Procedures**	x	x	x	x
AT-2	**Literacy Training and Awareness**	x	x	x	x
AT-2(1)	PRACTICAL EXERCISES				
AT-2(2)	INSIDER THREAT		x	x	x
AT-2(3)	SOCIAL ENGINEERING AND MINING			x	x
AT-2(4)	SUSPICIOUS COMMUNICATIONS AND ANOMALOUS SYSTEM BEHAVIOR				
AT-2(5)	ADVANCED PERSISTENT THREAT				
AT-2(6)	CYBER THREAT ENVIRONMENT				
AT-3	**Role-Based Training**	x	x	x	x
AT-3(1)	ENVIRONMENTAL CONTROLS				
AT-3(2)	PHYSICAL SECURITY CONTROLS				
AT-3(3)	PRACTICAL EXERCISES				
AT-3(4)	SUSPICIOUS COMMUNICATIONS AND ANOMALOUS SYSTEM BEHAVIOR	W: Incorporated into AT-2(4).			
AT-3(5)	PROCESSING PERSONALLY IDENTIFIABLE INFORMATION	x			
AT-4	**Training Records**	x	x	x	x
AT-5	Contacts with Security Groups and Associations	W: Incorporated into PM-15.			
AT-6	**Training Feedback**				

Screen shot from dynamic display in Baseline Matrix



☑ Media Protection (MP)

▸ **ACCESS CONTROL FAMILY (AC)**

▾ **AWARENESS AND TRAINING FAMILY (AT)**

CONTROL NUMBER	CONTROL NAME CONTROL ENHANCEMENT NAME	PROFILE_1	PROFILE_2
AT-1	**Policy and Procedures**	x	x
AT-2	**Literacy Training and Awareness**	x	x
AT-2(1)	PRACTICAL EXERCISES		
AT-2(2)	INSIDER THREAT		x
AT-2(3)	SOCIAL ENGINEERING AND MINING		x
AT-2(4)	SUSPICIOUS COMMUNICATIONS AND ANOMALOUS SYSTEM BEHAVIOR		
AT-2(5)	ADVANCED PERSISTENT THREAT		
AT-2(6)	CYBER THREAT ENVIRONMENT		
AT-3	**Role-based Training**		
AT-3(1)	ENVIRONMENTAL CONTROLS		
AT-3(2)	PHYSICAL SECURITY CONTROLS		

oscal/import-examiner

- Like all domain languages, OSCAL relies on certain regularities
- E.g., referential integrity ("do my links work?")
  - within documents
  - between related documents
- Testable with XSLT
  - Does this OSCAL profile make sense (with respect to its imports)?
  - Is it viable in an operational context? ("will it compile?")

oscal/validator

Serves as 'schema emulator'

Enforces rule set comparable to XSD or RNG validation

XSLT is generated from OSCAL Metaschema source data

Test-of-concept work-in-progress – much to do!

CSX for systems security applications

When everything happens on the client

- We can deploy from a plain web server
- No user data is exposed outside the local system
- After the application is delivered nothing is logged

Can a decentralized network of validating nodes promote systems and data security?

Ensuring *validability of exchange artifacts.*

Questions

- Security posture of SaxonJS and its dependencies
- Authenticating XSLT – source and runtime
    - Tricky – but worse than JS libraries?
- Do we need a security assessment of Saxon?

Generalized capabilities …

```
<xsl:key match="party" name="parties-index" use="@uuid"/>

<xsl:template match="assessment-log/entry">
  <xsl:call-template name="warn-if-false">
     <xsl:with-param name="test"
       select="exists(key('parties-index', child::logged-by/@party-uuid, $authorizations))"/>
  <xsl:with-param name="msg">Logged event may not be authorized</xsl:with-param>
</xsl:template>
```

The XSLT that validates OSCAL …

```
<xsl:key match="person" name="persons-index" use="'#' || @xml:id"/>

<xsl:template match="persName">
  <xsl:call-template name="warn-if-false">
     <xsl:with-param name="test"
       select="exists(key('persons-index', @ref, $prosopography))"/>
  <xsl:with-param name="msg">Person referenced is not listed in
prosopography</xsl:with-param>
</xsl:template>
```

… is the same XSLT that validates your data

Things to do with your data

Distributed validation

- Your data, our rules set
- Third-party validation?

Distributed data conversion

- Transformation as a service
- (E.g., conference proceedings preview?)

Distributed capabilities

- Your data, your rules set(s)
- Micro-editors

Transformation Lite

- Domain-specific processing languages
- Notations describing what-have-you *(invisible XML…)*

Data description ecosystems

- Distributed validation assumes
  - *Shared rule sets*
  - *Data versatility and reusability*
- This requires
  - Declarative information models codified as standards
  - Toolkits and technologies supporting these standards
- CSX works best in combination with other XML tech
  - Works especially well with an XML database in back
- Replicability also requires the same capability be deliverable by other means
  - Unit tests!

Thank you!

All demonstrations are open source and non-proprietary

https://github.com/wendellpiez/XMLjellysandwich