CSE598i - Web 2.0 Security
XML/XSLT

(Presenter here)
Spring 2010
Extensible Markup Language

- XML is a self-describing data format used to create all sorts of documents underlying almost all recent information systems.
  - Standardized (1.0, 2.0 rumored)
  - Many, many libraries & functions to choose from your favorite language
- Structure: Every document has a ‘schema’ defined by the document type definition (DTD)
  - In many cases may be omitted because it is (self describing)
  - DTDs become the data structures used to pass between systems and organizations
<xml version="1.0" encoding="UTF-8"?>
<nsf-report.dtd>
<!DOCTYPE NSFREPORT SYSTEM "nsf-report.dtd">
<NSFREPORT>
  <TITLE>CAREER: Realizing Practical High Assurance</TITLE>
  <AWARDID>CNS-0643907</AWARDID>
  <PI>
    <NAME>Patrick McDaniel</NAME>
    <ORGANIZATION>Pennsylvania State University</ORGANIZATION>
  </PI>
  <YEAR>2007</YEAR>
  <YEAR>2008</YEAR>
  <YEAR>2009</YEAR>
  <YEAR>2010</YEAR>
  <YEAR>2011</YEAR>
</NSFREPORT>

<nsf-report.dtd>
<!ELEMENT NSFREPORT (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT AWARDID (#PCDATA)>
<!ELEMENT PI (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT ORGANIZATION (#PCDATA)>
<!ATTLIST ORGANIZATION university|laboratory #REQUIRED>
<!ELEMENT YEAR (#PCDATA)>
xmllint

- A tool to validate your XML structure is valid (e.g., consistent with the DTD.
- The most often used with the -valid option, e.g.,

```
% xmllint --valid career.xml
```

- This will identify any errors associated with the format of the XML and its representation of the DTD.
- Note: It is really, really useful to use this with your XML documents before using them in a system ... problems with the XML often lead to unpredictable behavior later.
  - e.g., XSLT processing yields no text or missing fields.
XSLT (Markup)

- Converts your XML into anything you want, e.g., HTML, XML, Latex, ASCII, ....
- Conceptually, a recursive query on the XML document that produces the desired output.
- Collections of callable (directly or indirectly) templates for elements of the tree ...
- Two ways of calling XML.

1. In browser, load XML page with the following line

   `<?xml-stylesheet type="text/xsl" href="/nsf-report.xsl"?>`

2. Using the XSLT command line tool

   `xsltproc nsf-report.xsl career.xml > output.html`
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" omit-xml-declaration="yes"/>
  <xsl:template match="/NSFREPORT">
    <html>
      <head>
        <link href="nsf-report.css" rel="stylesheet" type="text/css"/>
        <title><xsl:value-of select="TITLE"/> (<xsl:value-of select="AWARDID"/>)</title>
      </head>
      <body bgcolor="pink">
        <h1><xsl:value-of select="TITLE"/> (<xsl:value-of select="AWARDID"/>)</h1>
        <ol>
          <xsl:apply-templates select="YEAR"/>
        </ol>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="YEAR">
    <li/><xsl:value-of select="."/>
  </xsl:template>
</xsl:stylesheet>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link href="nsf-report.css" rel="stylesheet" type="text/css">
<title>CAREER: Realizing Practical High Assurance (CNS-0643907)</title>
</head>
<body bgcolor="pink">
<h1>CAREER: Realizing Practical High Assurance (CNS-0643907)</h1>
<p><b>PI</b>: Patrick McDaniel,  
(Pennsylvania State University)</p>
<ol>
  CAREER: Realizing Practical High Assurance  
CNS-0643907  
  Patrick McDaniel  
  Pennsylvania State University  
  2007  
  2008  
  2009  
  2010  
  2011
</ol>
</body>
</html>
Other XSLT features ...

- variables (constants)
- for-each
- named templates
- family relations “preceding-sibling” (watch out)
- value oriented selects
  - select="PROJSTAFF[@parttype = 'srpersonnel']"
- if tests
  - <xsl:if test="STARTPAGE != "">