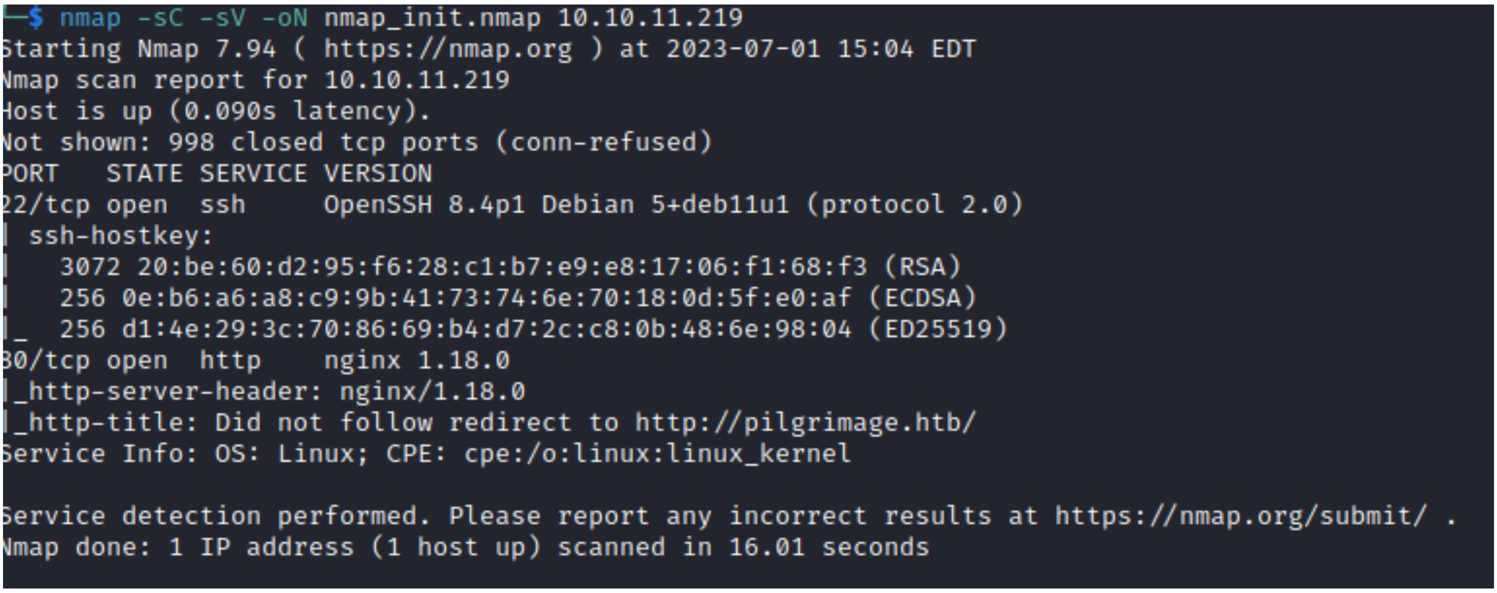
Pilgrimage

nmap

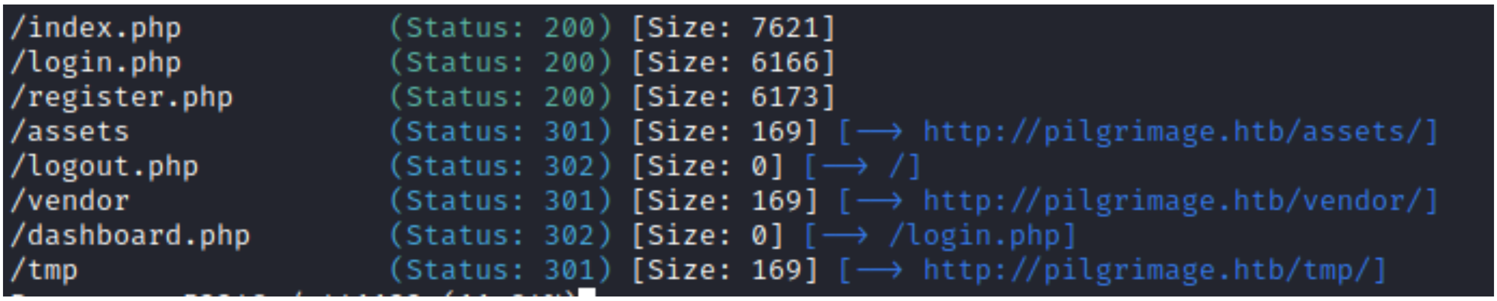


```
└─$ nmap -sC -sV -oN nmap_init.nmap 10.10.11.219
Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-01 15:04 EDT
Nmap scan report for 10.10.11.219
Host is up (0.090s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 20:be:60:d2:95:f6:28:c1:b7:e9:e8:17:06:f1:68:f3 (RSA)
|   256 0e:b6:a6:a8:c9:9b:41:73:74:6e:70:18:0d:5f:e0:af (ECDSA)
|_  256 d1:4e:29:3c:70:86:69:b4:d7:2c:c8:0b:48:6e:98:04 (ED25519)
80/tcp open  http    nginx 1.18.0
|_http-server-header: nginx/1.18.0
|_http-title: Did not follow redirect to http://pilgrimage.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.01 seconds
```

Ports 22, and 80 open. Let's look at the website

gobuster

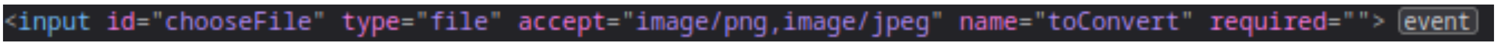


```
/index.php            (Status: 200) [Size: 7621]
/login.php            (Status: 200) [Size: 6166]
/register.php         (Status: 200) [Size: 6173]
/assets               (Status: 301) [Size: 169] [--> http://pilgrimage.htb/assets/]
/logout.php           (Status: 302) [Size: 0] [--> /]
/vendor               (Status: 301) [Size: 169] [--> http://pilgrimage.htb/vendor/]
/dashboard.php        (Status: 302) [Size: 0] [--> /login.php]
/tmp                  (Status: 301) [Size: 169] [--> http://pilgrimage.htb/tmp/]
```

Nothing too interesting

file upload

image shrinker web app



```
<input id="chooseFile" type="file" accept="image/png,image/jpeg" name="toConvert" required=""> event
```

seems to only accept png and jpeg

new findings

nmap with -vv actually reveals a .git directory

enumerating with gobuster, we find an file called index which reveals something called magick

this is imagemagick

downloading .git/magick, chmod +x, -version, gives a version
ImageMagick 7.1.0-49 beta

->CVE-2022-44268

https://github.com/voidz0r/CVE-2022-44268
→ we can access any file we want now.

contents of /etc/passwd

```
root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin\nbin:x:2:2:bin:/bin:/usr/sbin/nologin\nsys:x:3:3:sys:/dev:/usr/sbin/nologin\nsync:x:4:65534:sync:/bin:/bin/sync\ngames:x:5:60:games:/usr/games:/usr/sbin/nologin\nman:x:6:12:man:/var/cache/man:/usr/sbin/nologin\nlp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin\nmail:x:8:8:mail:/var/mail:/usr/sbin/nologin\nnews:x:9:9:news:/var/spool/news:/usr/sbin/nologin\nuucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin\nproxy:x:13:13:proxy:/bin:/usr/sbin/nologin\nwww-data:x:33:33:www-data:/var/www:/usr/sbin/nologin\nbackup:x:34:34:backup:/var/backups:/usr/sbin/nologin\nlist:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin\nirc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin\ngnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin\nnobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin\n_apt:x:100:65534::/nonexistent:/usr/sbin/nologin\nsystemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin\nsystemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin\nmessagebus:x:103:109::/nonexistent:/usr/sbin/nologin\nsystemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin\nemily:x:1000:1000:emily,,,:/home/emily:/bin/bash\nsystemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin\nsshd:x:105:65534::/run/sshd:/usr/sbin/nologin\n_laurel:x:998:998::/var/log/laurel:/bin/false\n
```

username we can use: emily

git repo

There is a git repo we can retrieve using gitdump

After doing so, we find an interesting file location we can read using the exploit we found earlier
/var/db/pilgrimage

This is an sqlite database which contains credentials to emily on the website

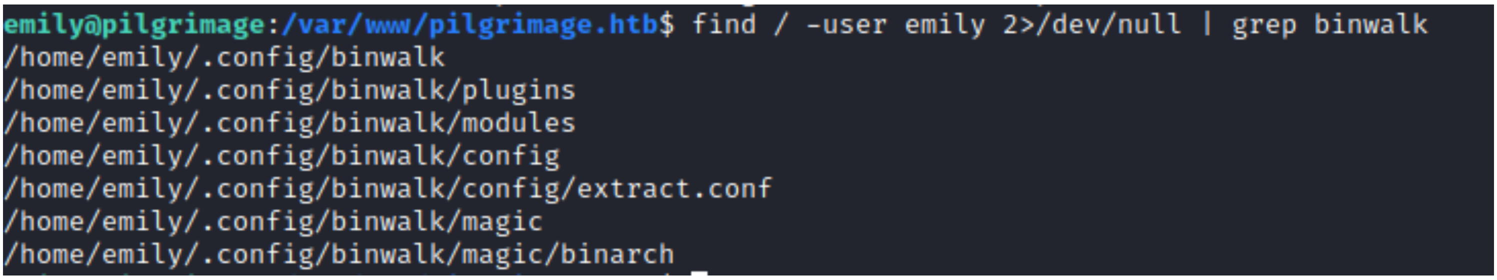
creds

for website and ssh:

emily:abigchonkyboi123

priv esc

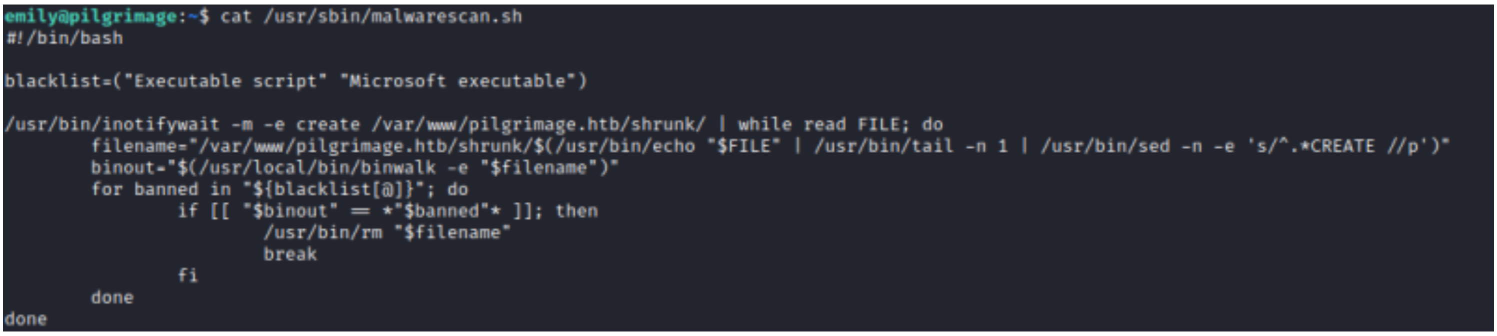
Found some unusual files owned by emily



```
emily@pilgrimage:/var/www/pilgrimage.htb$ find / -user emily 2>/dev/null | grep binwalk
/home/emily/.config/binwalk
/home/emily/.config/binwalk/plugins
/home/emily/.config/binwalk/modules
/home/emily/.config/binwalk/config
/home/emily/.config/binwalk/config/extract.conf
/home/emily/.config/binwalk/magic
/home/emily/.config/binwalk/magic/binarch
```

LINPEAS:

root runs /usr/sbin/malwarescan.sh, which is readable by us. It also uses binwalk!!



```
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
```

Every time something is created in /var/www/pilgrimage.htb/shrunk/, it is checked against a blacklist to see if it is a script/executable. If so, it is removed.

After some investigating, there is an exploit available on the version of binwalk installed on the machine:

Binwalk v2.3.2 - Remote Command Execution (RCE)

This gives us a reverse shell as root!