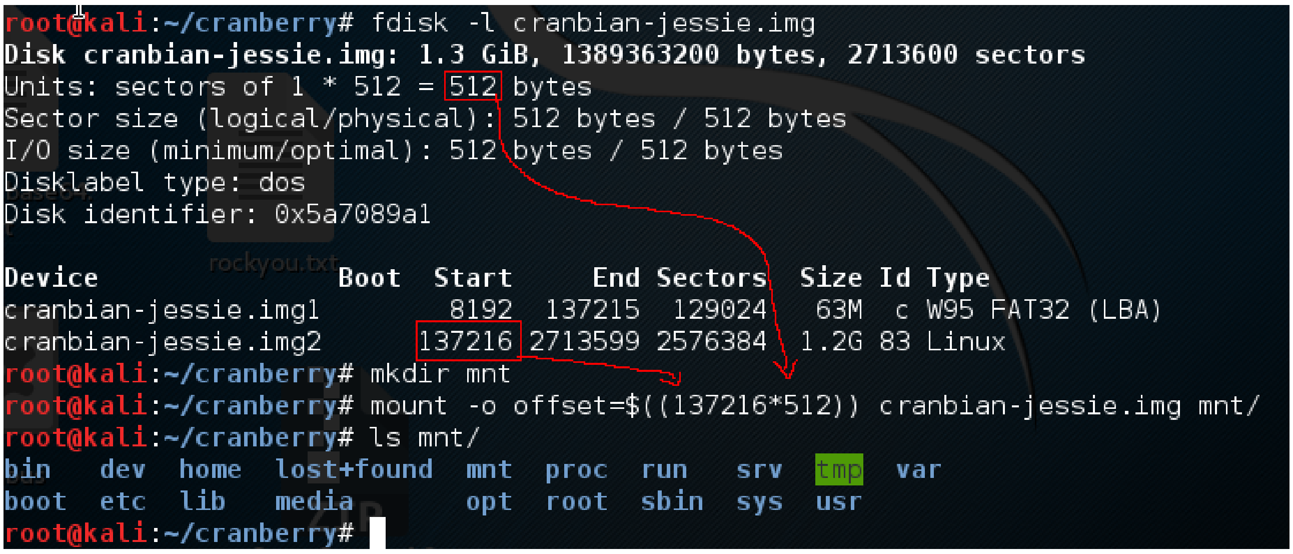
1 Part 3: A Fresh-Baked Holiday Pi

Before beginning on these quests, I needed to assemble the Cranberry Pi in the RPG world. To see where to find the pieces of the Cranberry Pi, check out ??. Once that quest was completed I downloaded the cranbian image[1].

1.1 What is the password for the "cranpi" account on the Cranberry Pi system?

Alright, so the easiest way to get to the password for the account "cranpi" is to put john[2] to work on the shadow file in the image.

That just leaves the question of how to access the shadow file. The easiest way would be to mount the cranbian image, check out Figure 1.



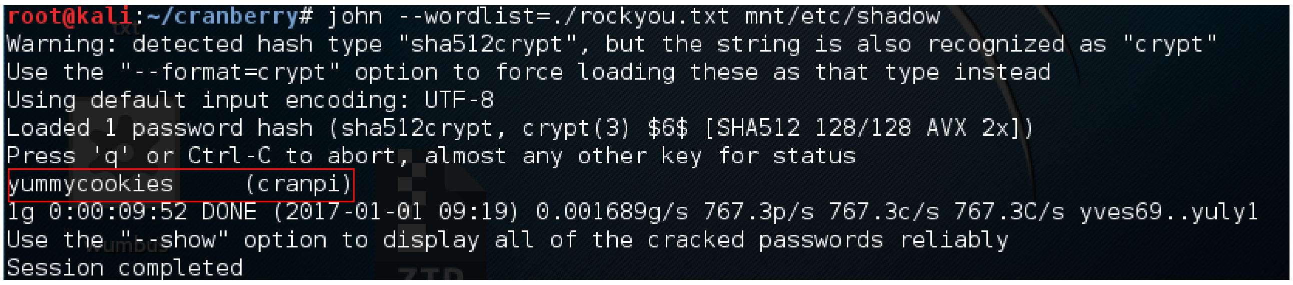
```
root@kali:~/cranberry# fdisk -l cranbian-jessie.img
Disk cranbian-jessie.img: 1.3 GiB, 1389363200 bytes, 2713600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x5a7089a1

Device               Boot  Start     End Sectors  Size Id Type
cranbian-jessie.img1        8192  137215  129024   63M  c W95 FAT32 (LBA)
cranbian-jessie.img2      137216 2713599 2576384  1.2G 83 Linux
root@kali:~/cranberry# mkdir mnt
root@kali:~/cranberry# mount -o offset=$((137216*512)) cranbian-jessie.img mnt/
root@kali:~/cranberry# ls mnt/
bin   dev  home  lost+found  mnt  proc  run   srv  tmp  var
boot  etc  lib   media       opt  root  sbin  sys  usr
root@kali:~/cranberry#
```

Figure 1: The process of mounting the Cranbian image.

A comment on the red arrows in Figure 1, 512 is the size of each sector in the image and 137216 is the start sector of the Linux file system. So with $137216 \times 512 = 70254592$ we have the offset needed to mount the image.

Now that the image is mounted, it's time to put dear John to work. In Figure 2 I do just that.

[1]Found at https://www.northpolewonderland.com/cranbian.img.zip
[2]John The Ripper - http://www.openwall.com/john/



```
root@kali:~/cranberry# john --wordlist=./rockyou.txt mnt/etc/shadow
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
yummycookies     (cranpi)
1g 0:00:09:52 DONE (2017-01-01 09:19) 0.001689g/s 767.3p/s 767.3c/s 767.3C/s yves69..yuly1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

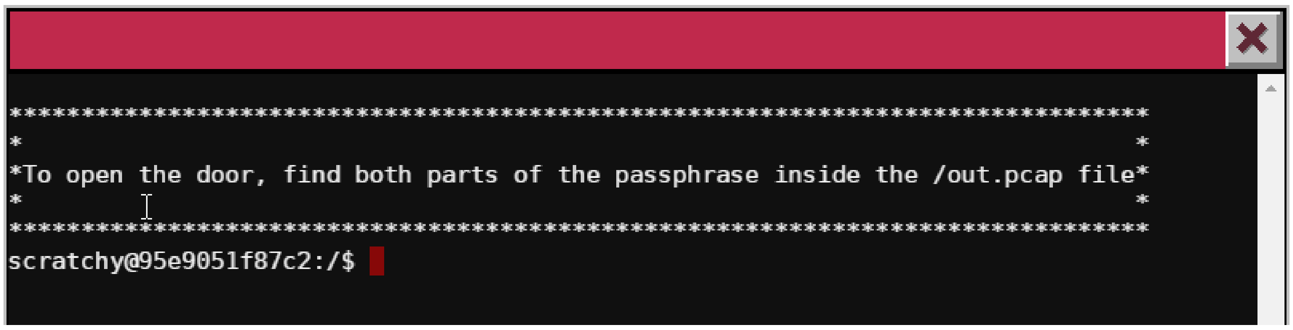
Figure 2: Cracking the shadow file. The password is 'yummycookies'

So now we have the password for the user. Time to speak to Holly Everygreen and let her know what the password is.

1.2 How did you open each terminal door and where had the villain imprisoned Santa?

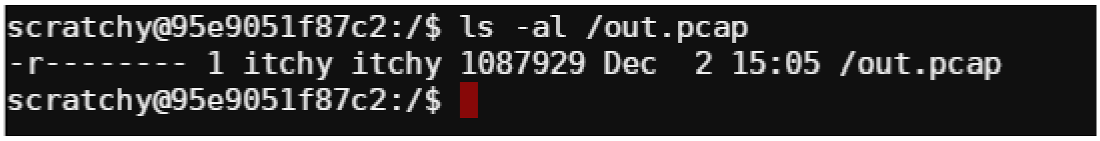
Alright, so there are five terminals scattered throughout the little world. Completing each terminal gives a password, which in turn allows access through the door next to the terminal.

1.2.1 Terminal 1: Elf House #2



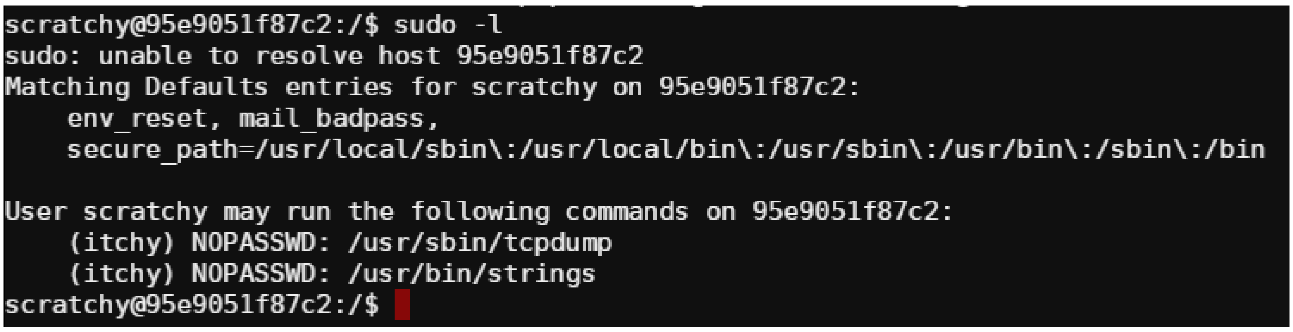
```
*******************************************************************************
*                                                                             *
*To open the door, find both parts of the passphrase inside the /out.pcap file*
*                                                                             *
*******************************************************************************
scratchy@95e9051f87c2:/$
```

This is the screen that greets you when you open this terminal. So I guess we just have to read '/out.pcap,' easy peasy...



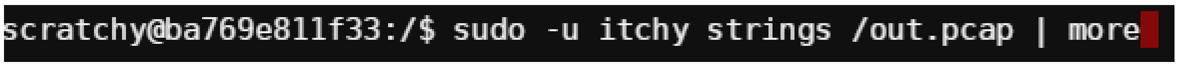
```
scratchy@95e9051f87c2:/$ ls -al /out.pcap
-r-------- 1 itchy itchy 1087929 Dec  2 15:05 /out.pcap
scratchy@95e9051f87c2:/$
```

Except it seems that only *itchy* can read the file, and I'm logged in as *scratchy*. Trying to 'su itchy' prompts a password, which I don't have. ... What else, what else. Time to take a look at 'sudo,' more precisely 'sudo -l.'

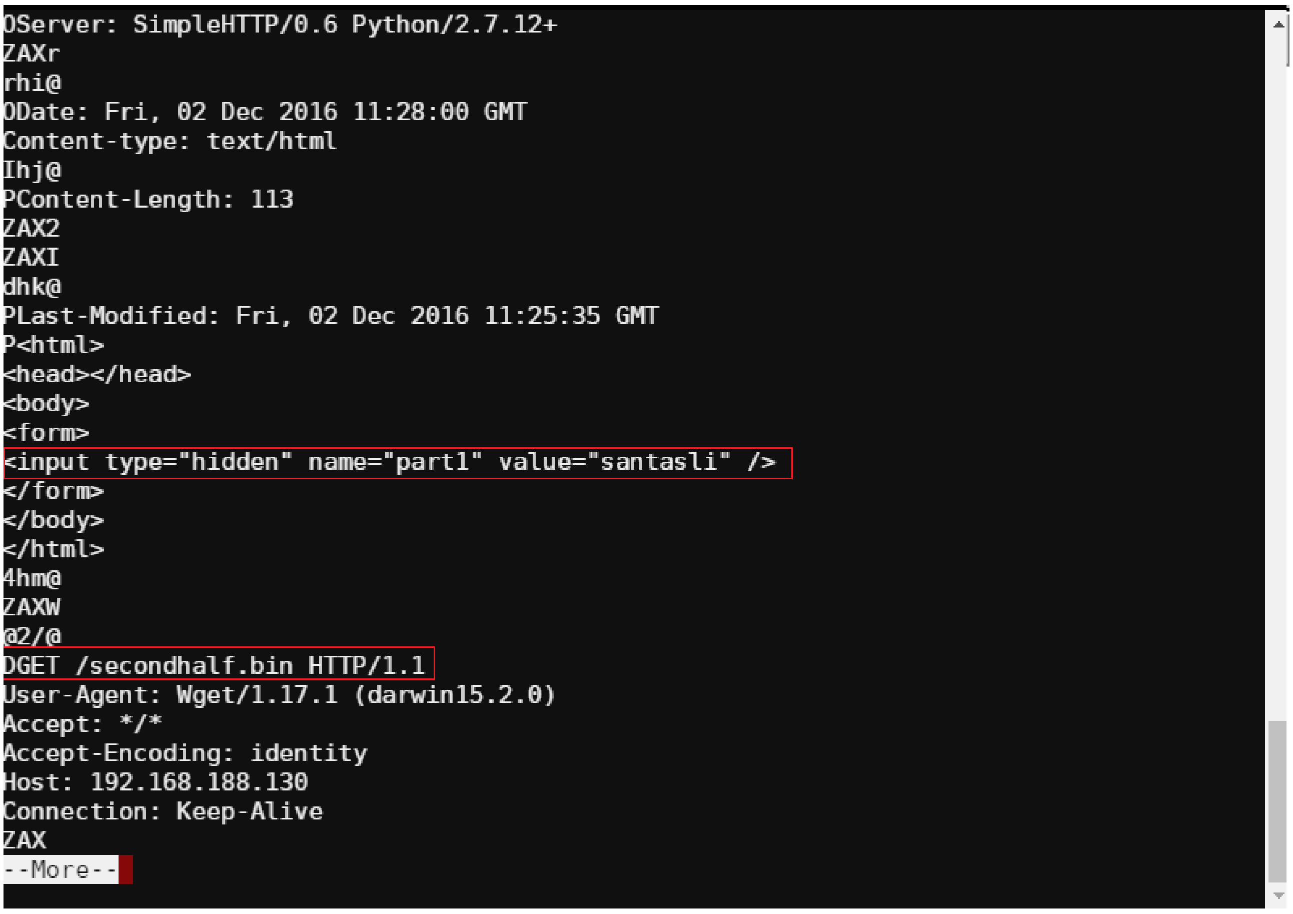


```
scratchy@95e9051f87c2:/$ sudo -l
sudo: unable to resolve host 95e9051f87c2
Matching Defaults entries for scratchy on 95e9051f87c2:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User scratchy may run the following commands on 95e9051f87c2:
    (itchy) NOPASSWD: /usr/sbin/tcpdump
    (itchy) NOPASSWD: /usr/bin/strings
scratchy@95e9051f87c2:/$
```

Alright, so there are a few commands available that I can run as *itchy*. Time to run the first one and see if there's anything exciting.



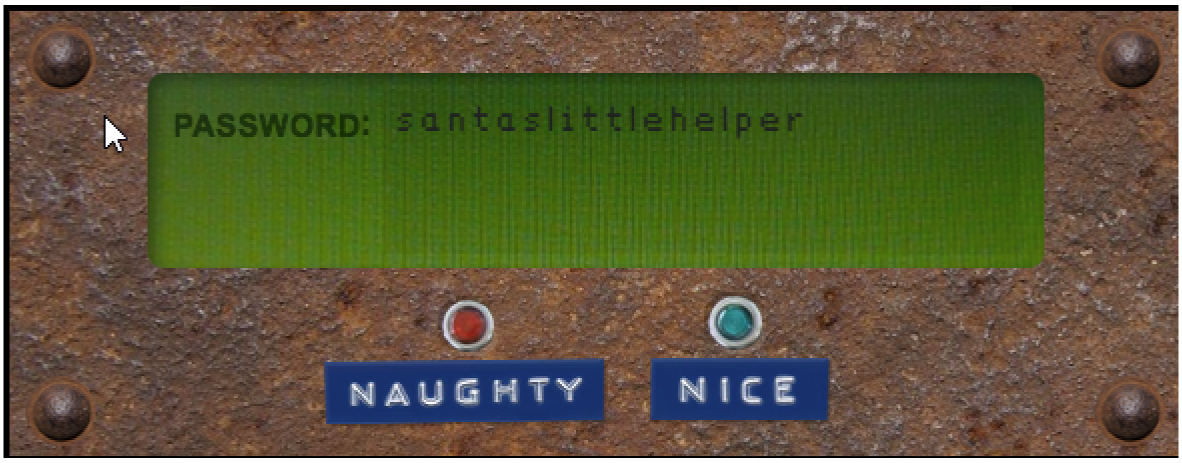
```
scratchy@ba769e811f33:/$ sudo -u itchy strings /out.pcap | more
```



```
0Server: SimpleHTTP/0.6 Python/2.7.12+
ZAXr
rhi@
0Date: Fri, 02 Dec 2016 11:28:00 GMT
Content-type: text/html
Ihj@
PContent-Length: 113
ZAX2
ZAXI
dhk@
PLast-Modified: Fri, 02 Dec 2016 11:25:35 GMT
P<html>
<head></head>
<body>
<form>
<input type="hidden" name="part1" value="santasli" />
</form>
</body>
</html>
4hm@
ZAXW
@2/@
DGET /secondhalf.bin HTTP/1.1
User-Agent: Wget/1.17.1 (darwin15.2.0)
Accept: */*
Accept-Encoding: identity
Host: 192.168.188.130
Connection: Keep-Alive
ZAX
--More--
```

Browsing through the output of 'strings' does indeed show something exciting. The first red box shows the first part of the password, 'santasli', the second shows the GET request for what I believe to be the place to find the second half of the password.

But first, I'll make a guess that the password is 'santaslittlehelper'.

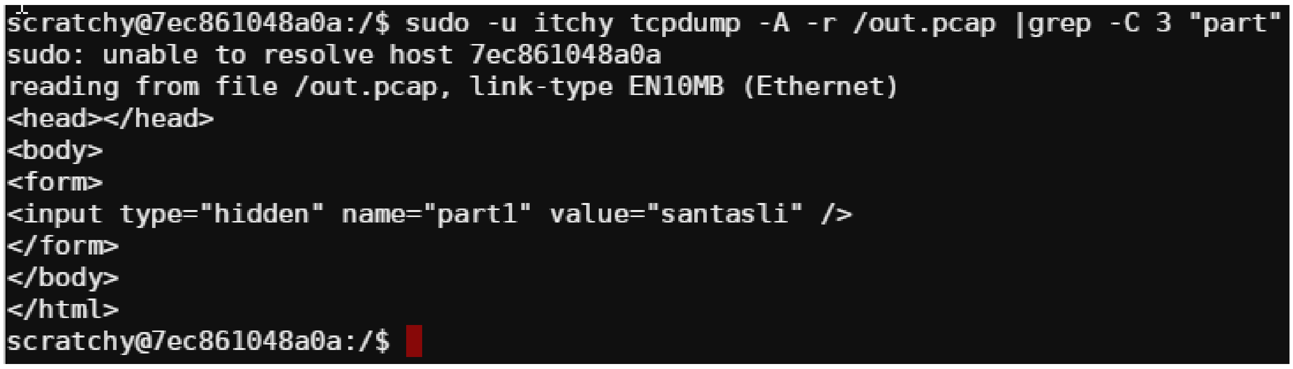


And lo and behold, it is the password.

However, for completeness sake, it is time to try and find the second half of the password and as it turns out, the second half does not show in the 'strings' output.

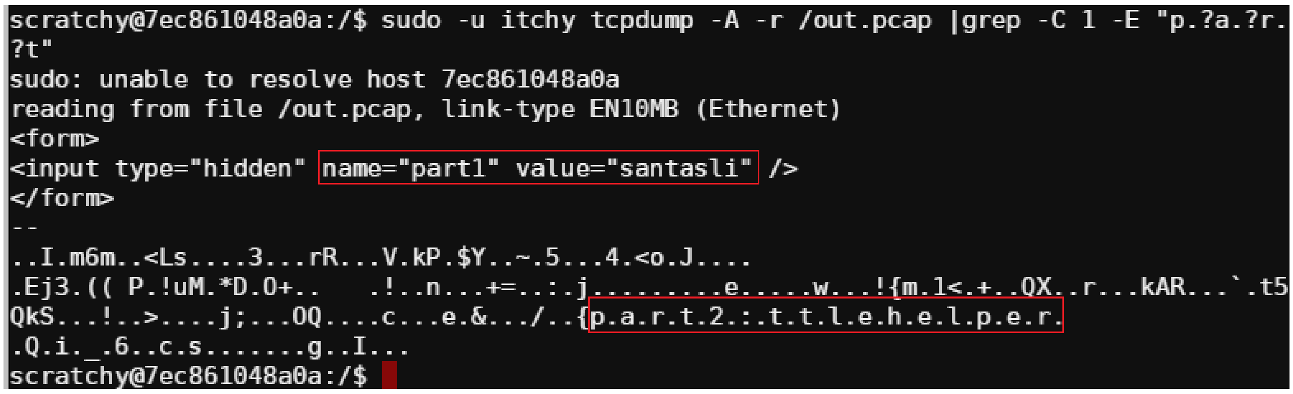
A few observations, before jumping the gun on 'tcpdump', it seems the GET requests where to */firsthalf.html* and */secondhalf.bin* and the marker for the first half of the password seems to be *part1*, so an educated guess would be to search for *part2* in the pcap.

So with that in mind, it's time to run 'tcpdump' and pipe it into a grep to see if we can catch anything.



```
scratchy@7ec861048a0a:/$ sudo -u itchy tcpdump -A -r /out.pcap |grep -C 3 "part"
sudo: unable to resolve host 7ec861048a0a
reading from file /out.pcap, link-type EN10MB (Ethernet)
<head></head>
<body>
<form>
<input type="hidden" name="part1" value="santasli" />
</form>
</body>
</html>
scratchy@7ec861048a0a:/$
```

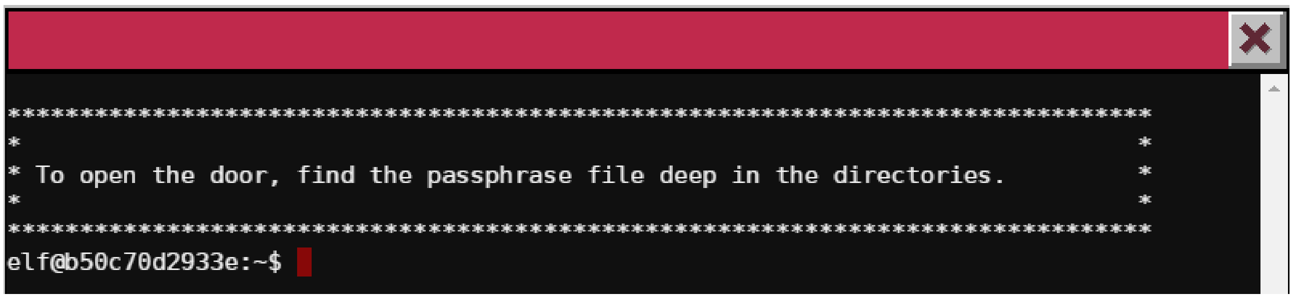
Well, it's a good start. It catches the first password. Now, since the theory is that the second part is embedded into the *secondhalf.bin* file, the text might be separated by some obscure characters, so time to try another grep.



```
scratchy@7ec861048a0a:/$ sudo -u itchy tcpdump -A -r /out.pcap |grep -C 1 -E "p.?a.?r.
?t"
sudo: unable to resolve host 7ec861048a0a
reading from file /out.pcap, link-type EN10MB (Ethernet)
<form>
<input type="hidden" name="part1" value="santasli" />
</form>
--
..I.m6m..<Ls....3...rR...V.kP.$Y..~.5...4.<o.J....
.Ej3.(( P.!uM.*D.0+..   .!..n...+=..:.j.........e.....w...!{m.1<.+..QX..r...kAR...`.t5
QkS...!..>....j;...0Q....c...e.&.../..{p.a.r.t.2.:.t.t.l.e.h.e.l.p.e.r.
.Q.i._.6..c.s.......g..I...
scratchy@7ec861048a0a:/$
```

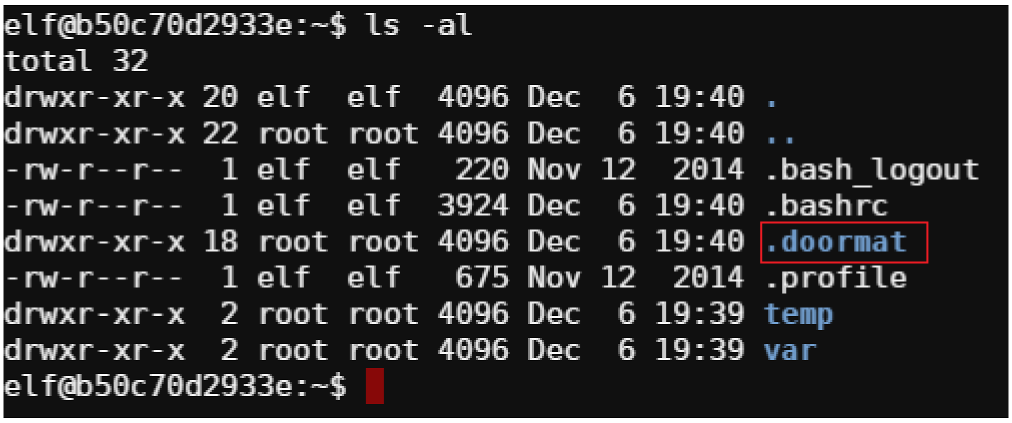
Excellent, this time both parts of the password showed up, and it also confirmed the guess of 'santaslittlehelper.'

1.2.2 Terminal 2: The Workshop (Bottom)



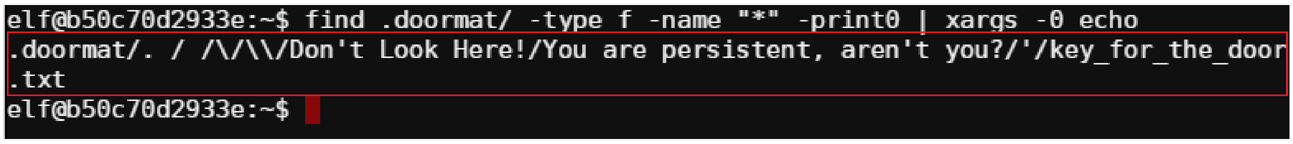
```
*******************************************************************************
*                                                                             *
* To open the door, find the passphrase file deep in the directories.         *
*                                                                             *
*******************************************************************************
elf@b50c70d2933e:~$
```

Great. Stuff hidden somewhere, huh? I guess the first thing to do a little *ls*'ing

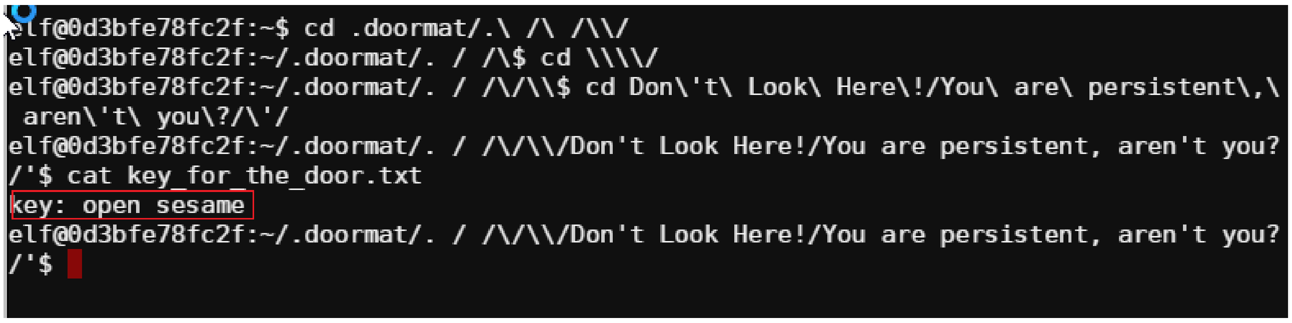


```
elf@b50c70d2933e:~$ ls -al
total 32
drwxr-xr-x 20 elf  elf  4096 Dec  6 19:40 .
drwxr-xr-x 22 root root 4096 Dec  6 19:40 ..
-rw-r--r--  1 elf  elf   220 Nov 12  2014 .bash_logout
-rw-r--r--  1 elf  elf  3924 Dec  6 19:40 .bashrc
drwxr-xr-x 18 root root 4096 Dec  6 19:40 .doormat
-rw-r--r--  1 elf  elf   675 Nov 12  2014 .profile
drwxr-xr-x  2 root root 4096 Dec  6 19:39 temp
drwxr-xr-x  2 root root 4096 Dec  6 19:39 var
elf@b50c70d2933e:~$
```

Would you look at that? There's a doormat, and everyone knows that you hide keys underneath doormats. Time to see if we can find any files in there, of if it's just a decoy.

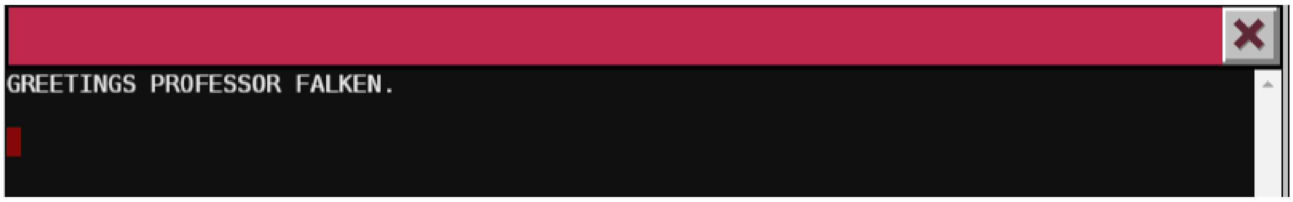

```
elf@b50c70d2933e:~$ find .doormat/ -type f -name "*" -print0 | xargs -0 echo
.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?/'/key_for_the_door
.txt
elf@b50c70d2933e:~$
```

Uh, what? That looks like something that needs to be escaped at every step. And this terminal does **not** have tab-completion.

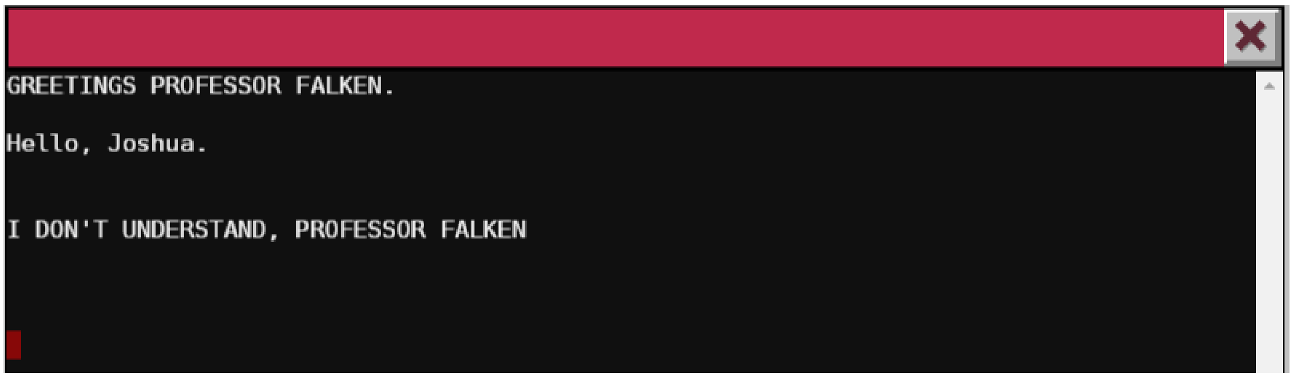

```
elf@0d3bfe78fc2f:~$ cd .doormat/.\ /\ /\\/
elf@0d3bfe78fc2f:~/.doormat/. / /\$ cd \\\\/
elf@0d3bfe78fc2f:~/.doormat/. / /\/\\$ cd Don\'t\ Look\ Here\!/You\ are\ persistent\,\
 aren\'t\ you\?/\'/
elf@0d3bfe78fc2f:~/.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?
/'$ cat key_for_the_door.txt
key: open sesame
elf@0d3bfe78fc2f:~/.doormat/. / /\/\\/Don't Look Here!/You are persistent, aren't you?
/'$
```

Okay, a painstaking *cd*'ing later and it is possible to *cat* the file. Thus providing the key for the next door.

1.2.3 Terminal 3: The Workshop - Santa's Office

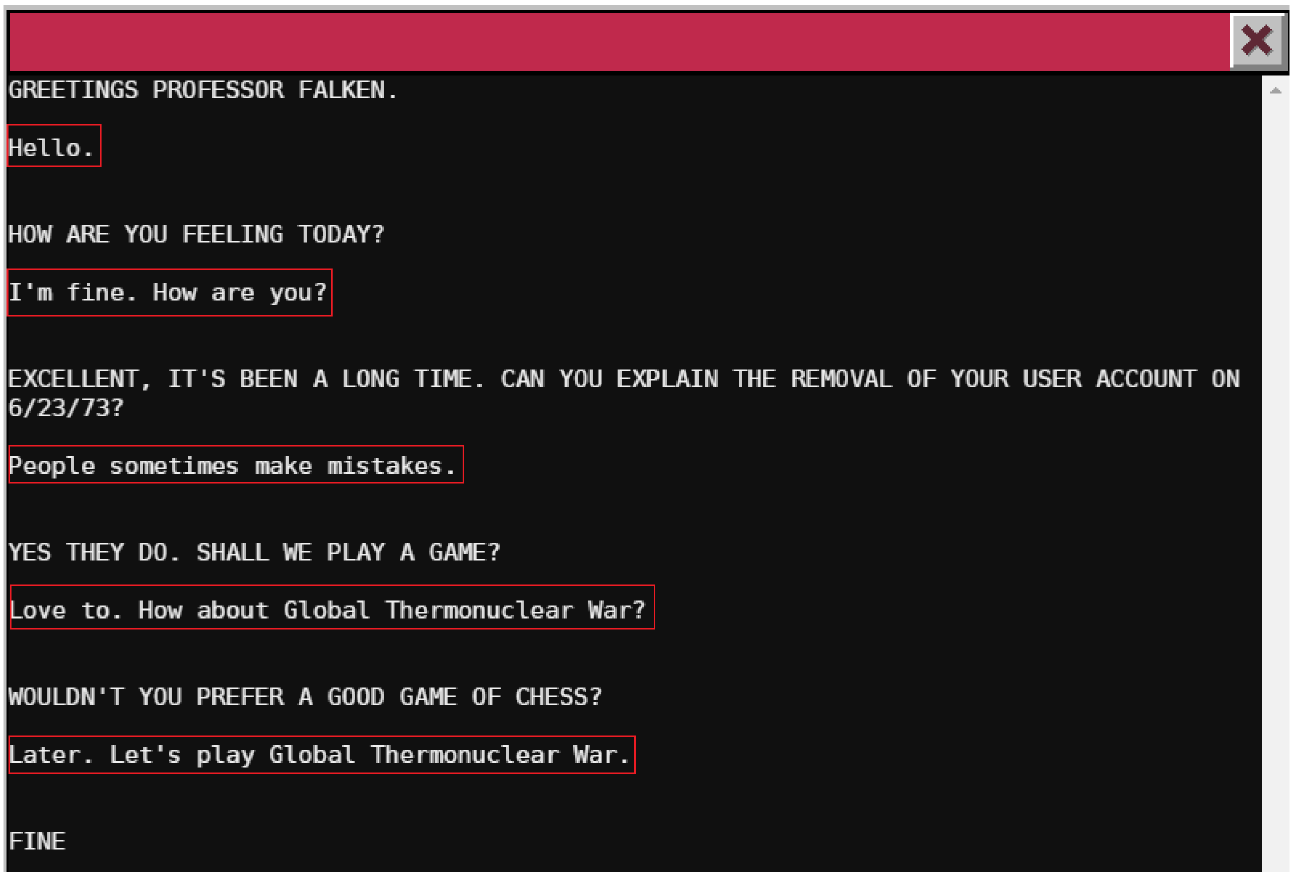

```
GREETINGS PROFESSOR FALKEN.
```

Hello, Joshua.


```
GREETINGS PROFESSOR FALKEN.

Hello, Joshua.


I DON'T UNDERSTAND, PROFESSOR FALKEN
```

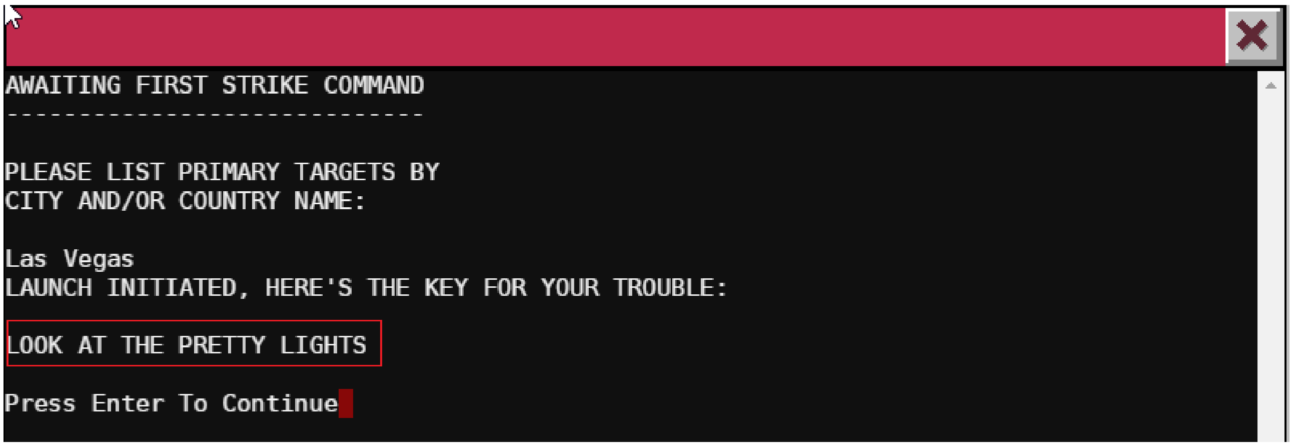
Mkay. Or not. Well, turns out just 'Hello.' works. A bit of Google-foo and the full script is located[3], however the script needed a little bit of modification.

[3]Located at https://github.com/abs0/wargames/blob/master/wargames.sh.



```
GREETINGS PROFESSOR FALKEN.

Hello.

HOW ARE YOU FEELING TODAY?

I'm fine. How are you?

EXCELLENT, IT'S BEEN A LONG TIME. CAN YOU EXPLAIN THE REMOVAL OF YOUR USER ACCOUNT ON
6/23/73?

People sometimes make mistakes.

YES THEY DO. SHALL WE PLAY A GAME?

Love to. How about Global Thermonuclear War?

WOULDN'T YOU PREFER A GOOD GAME OF CHESS?

Later. Let's play Global Thermonuclear War.

FINE
```

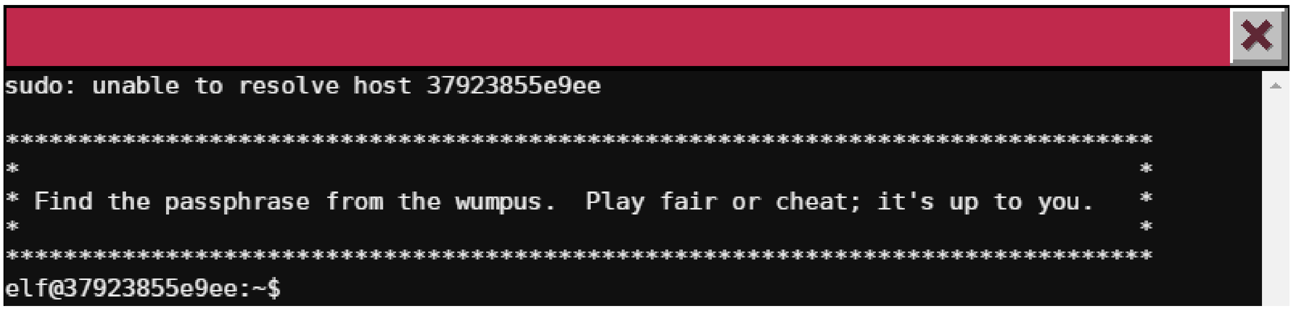
Missing the side selection here, I choose '2'.



```
AWAITING FIRST STRIKE COMMAND
-----------------------------

PLEASE LIST PRIMARY TARGETS BY
CITY AND/OR COUNTRY NAME:

Las Vegas
LAUNCH INITIATED, HERE'S THE KEY FOR YOUR TROUBLE:

LOOK AT THE PRETTY LIGHTS

Press Enter To Continue
```

And there we have it folks, the password for the next door.

1.2.4 Terminal 4: The Workshop (Top)

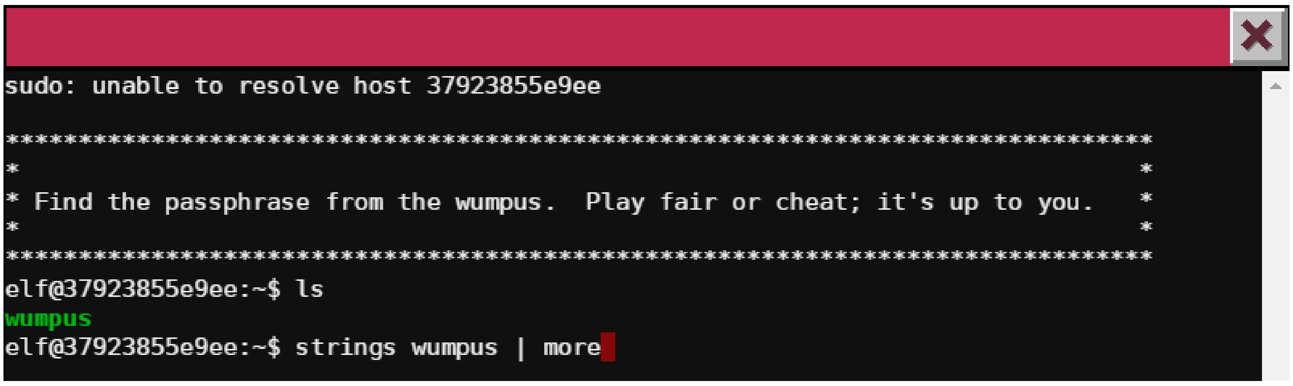


```
sudo: unable to resolve host 37923855e9ee

*******************************************************************************
*                                                                             *
* Find the passphrase from the wumpus.  Play fair or cheat; it's up to you.   *
*                                                                             *
*******************************************************************************
elf@37923855e9ee:~$
```

So a classic game of Hunt the Wumpus[4], now reversing stuff is in no form, way or shape my strong suit.

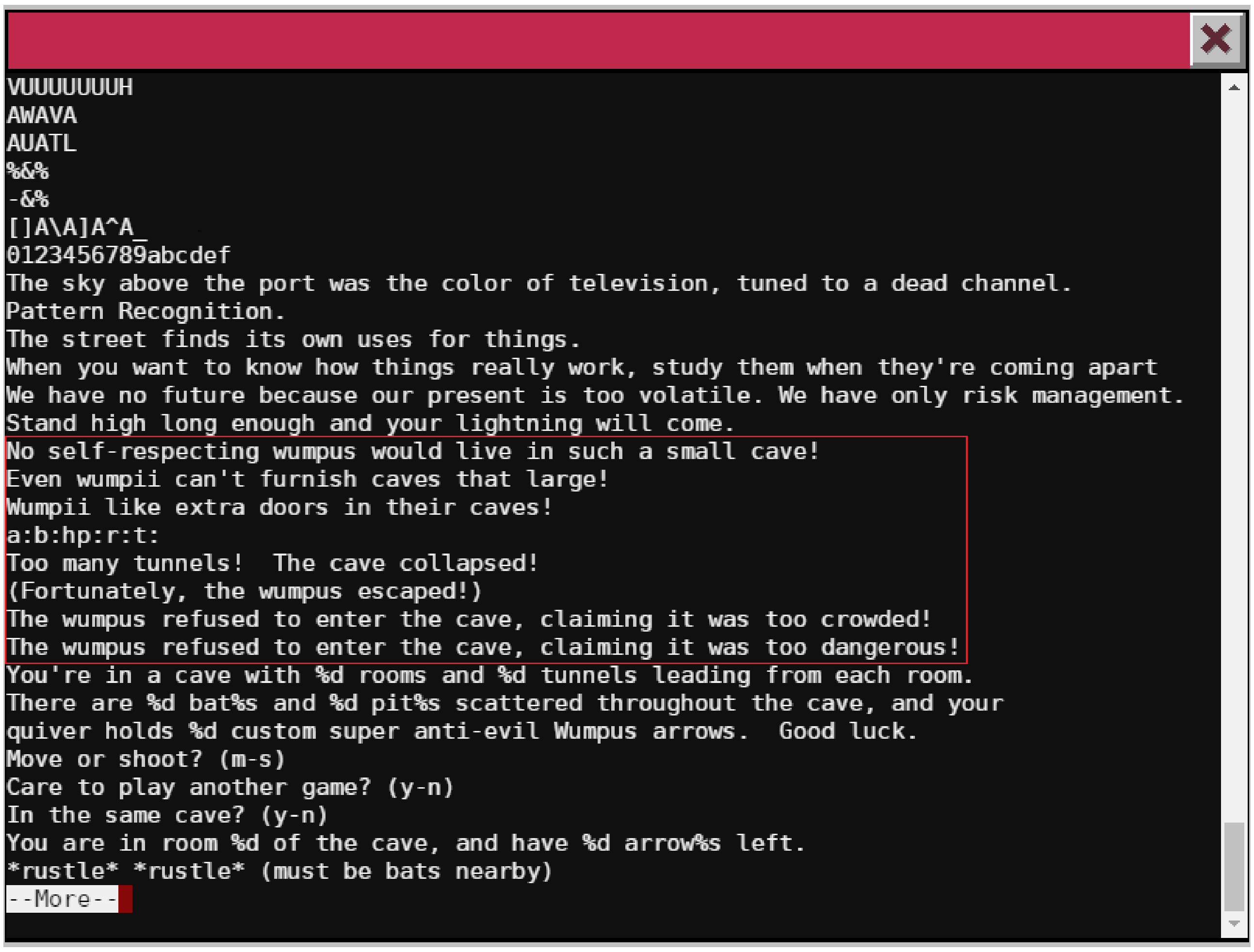
So first things first, just play the game as it's meant to be played and that actually worked out perfectly fine. I completed the challenge in around 10 minutes time.

However, for the sake of having fun and since I'm allowed to cheat, it's time to pull out the *strings* command and take a look at what strings there are in the Wumpus binary.



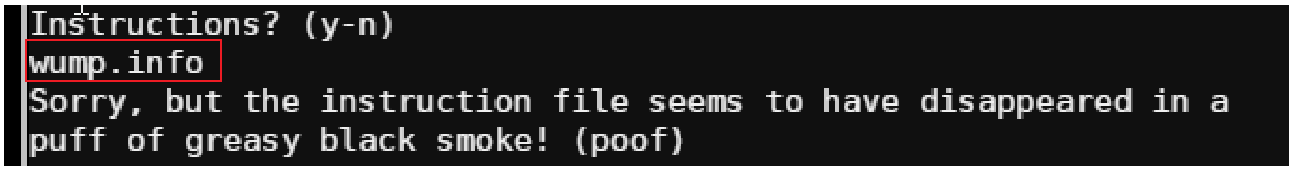
```
sudo: unable to resolve host 37923855e9ee

*******************************************************************************
*                                                                             *
* Find the passphrase from the wumpus.  Play fair or cheat; it's up to you.   *
*                                                                             *
*******************************************************************************
elf@37923855e9ee:~$ ls
wumpus
elf@37923855e9ee:~$ strings wumpus | more
```

[4]Wiki page at https://en.wikipedia.org/wiki/Hunt_the_Wumpus



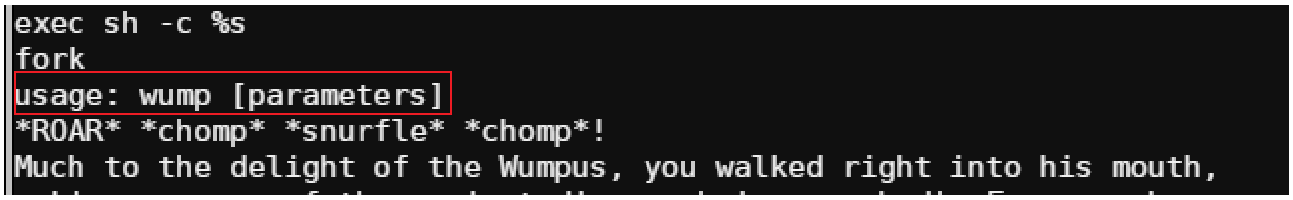
```
VUUUUUUUH
AWAVA
AUATL
%&%
-&%
[]A\A]A^A_
0123456789abcdef
The sky above the port was the color of television, tuned to a dead channel.
Pattern Recognition.
The street finds its own uses for things.
When you want to know how things really work, study them when they're coming apart
We have no future because our present is too volatile. We have only risk management.
Stand high long enough and your lightning will come.
No self-respecting wumpus would live in such a small cave!
Even wumpii can't furnish caves that large!
Wumpii like extra doors in their caves!
a:b:hp:r:t:
Too many tunnels!  The cave collapsed!
(Fortunately, the wumpus escaped!)
The wumpus refused to enter the cave, claiming it was too crowded!
The wumpus refused to enter the cave, claiming it was too dangerous!
You're in a cave with %d rooms and %d tunnels leading from each room.
There are %d bat%s and %d pit%s scattered throughout the cave, and your
quiver holds %d custom super anti-evil Wumpus arrows.  Good luck.
Move or shoot? (m-s)
Care to play another game? (y-n)
In the same cave? (y-n)
You are in room %d of the cave, and have %d arrow%s left.
*rustle* *rustle* (must be bats nearby)
--More--
```

So straight away, this looks fishy. Is it possible to change the number of caves, bats etc., in the game? And what's that *a:b:hp:r:t:* rubbish about? Let's keep going through the output.



```
Instructions? (y-n)
wump.info
Sorry, but the instruction file seems to have disappeared in a
puff of greasy black smoke! (poof)
```

So it seems as if there once were a file containing some sort of information about the game, but who knows.



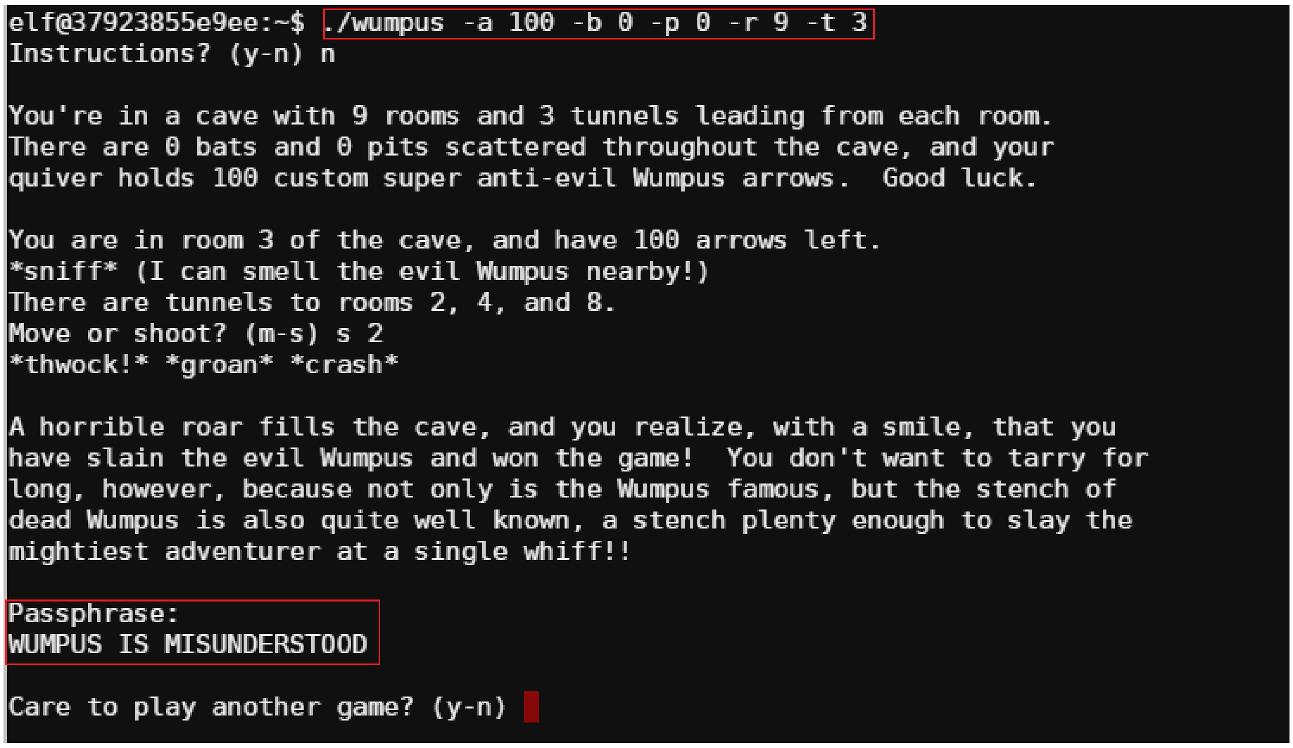
```
exec sh -c %s
fork
usage: wump [parameters]
*ROAR* *chomp* *snurfle* *chomp*!
Much to the delight of the Wumpus, you walked right into his mouth,
```

Now, this confirms the suspicion about the game taking arguments of sorts. The arguments might be related to the odd string above.

After searching google for "a:b:hp:r:t:" I found *wumpus.c*[5] and this file seems to let me know what the arguments are, and what they stand for.

- a - Amount of arrows the player can carry.
- b - Amount of bats in the game.
- h - Hard mode, no thanks.
- p - Amount of pits.
- r - Amount of rooms
- t - Amount of tunnels.

So, with this information I'm ready to play the game.

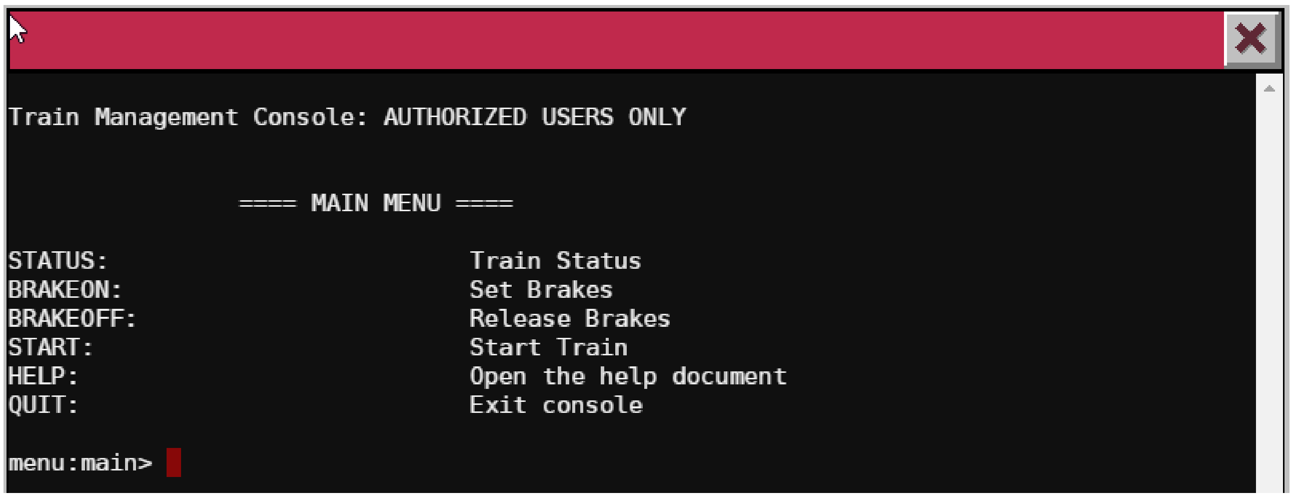


```
elf@37923855e9ee:~$ ./wumpus -a 100 -b 0 -p 0 -r 9 -t 3
Instructions? (y-n) n

You're in a cave with 9 rooms and 3 tunnels leading from each room.
There are 0 bats and 0 pits scattered throughout the cave, and your
quiver holds 100 custom super anti-evil Wumpus arrows.  Good luck.

You are in room 3 of the cave, and have 100 arrows left.
*sniff* (I can smell the evil Wumpus nearby!)
There are tunnels to rooms 2, 4, and 8.
Move or shoot? (m-s) s 2
*thwock!* *groan* *crash*

A horrible roar fills the cave, and you realize, with a smile, that you
have slain the evil Wumpus and won the game!  You don't want to tarry for
long, however, because not only is the Wumpus famous, but the stench of
dead Wumpus is also quite well known, a stench plenty enough to slay the
mightiest adventurer at a single whiff!!

Passphrase:
WUMPUS IS MISUNDERSTOOD

Care to play another game? (y-n)
```

With a stroke of luck and the game is completed in the very first move.

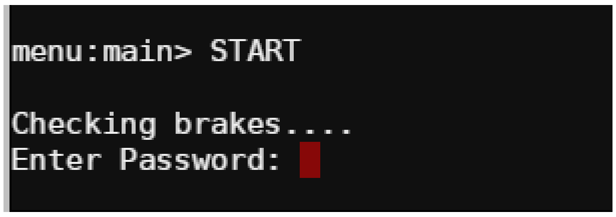
[5]Located at http://gentoo.osuosl.org/distfiles/wump.c

1.2.5 Terminal 5: Train Station



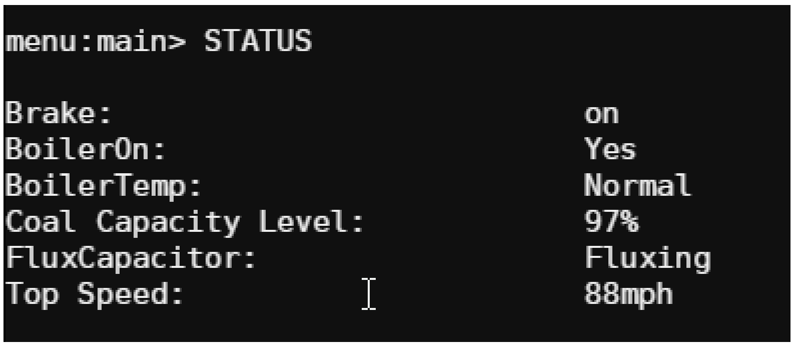
```
Train Management Console: AUTHORIZED USERS ONLY


                ==== MAIN MENU ====

STATUS:                         Train Status
BRAKEON:                        Set Brakes
BRAKEOFF:                       Release Brakes
START:                          Start Train
HELP:                           Open the help document
QUIT:                           Exit console

menu:main>
```

Huh. So some sort of menu that we need to play around with.



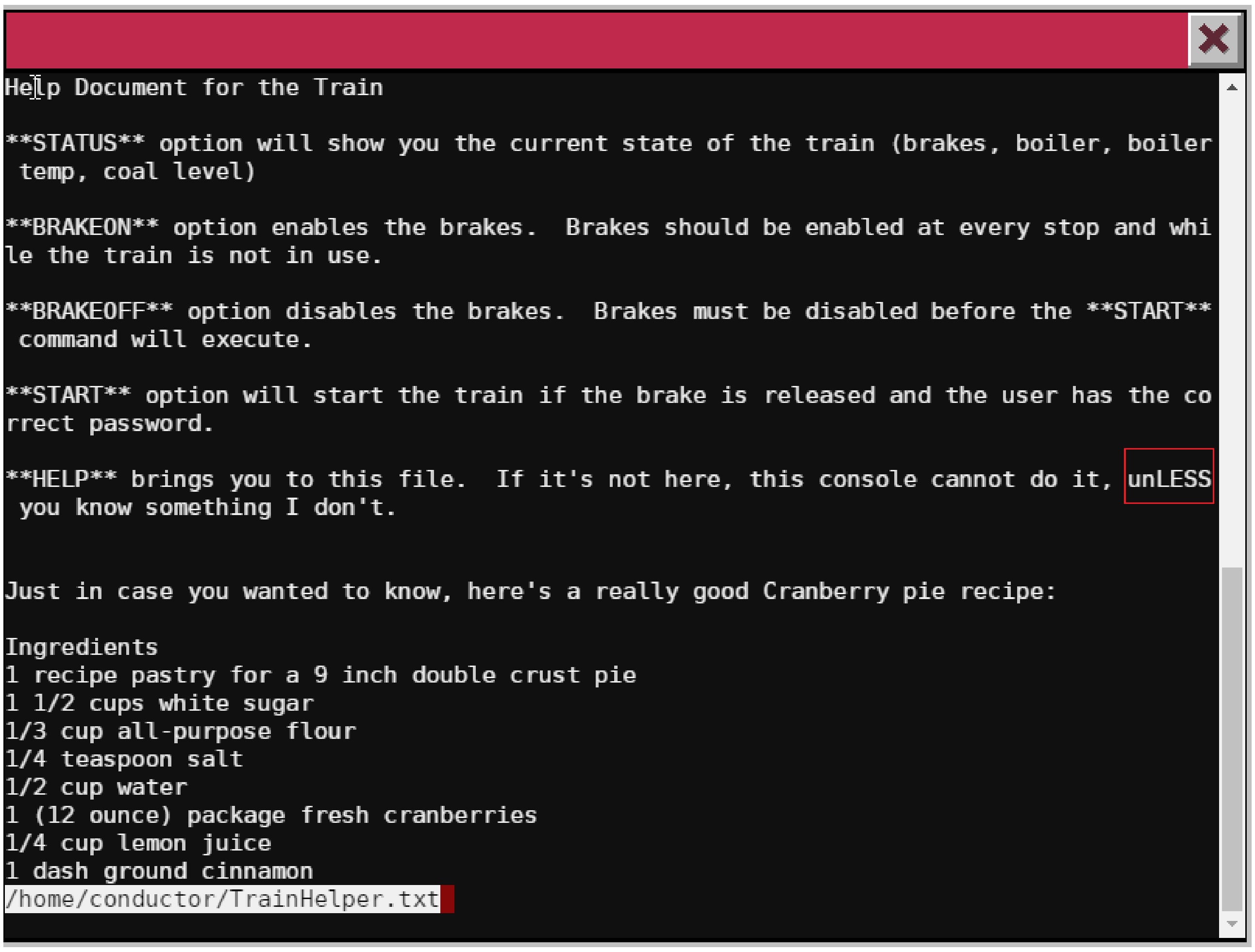
```
menu:main> START

Checking brakes....
Enter Password:
```

So it seems that I need to find a password, to be able to start this thing.



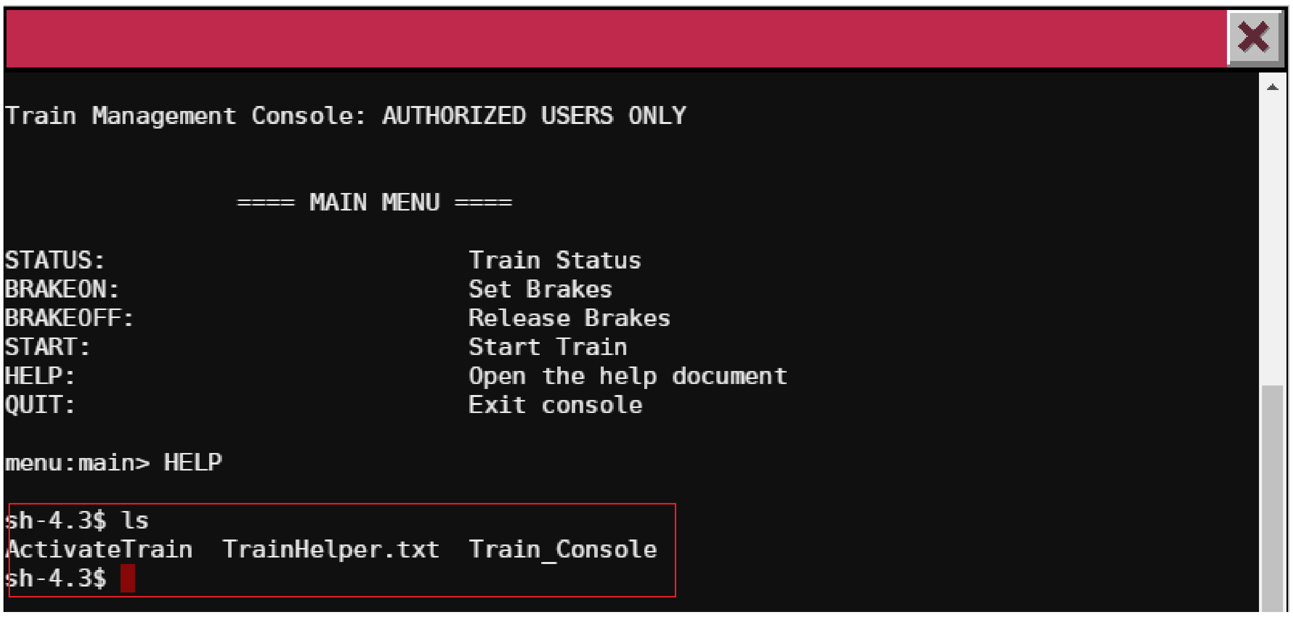
```
menu:main> STATUS

Brake:                          on
BoilerOn:                       Yes
BoilerTemp:                     Normal
Coal Capacity Level:            97%
FluxCapacitor:                  Fluxing
Top Speed:                      88mph
```

Top speed 88mph and the flux capacitor is fluxing. Excellent. This better be time travel!



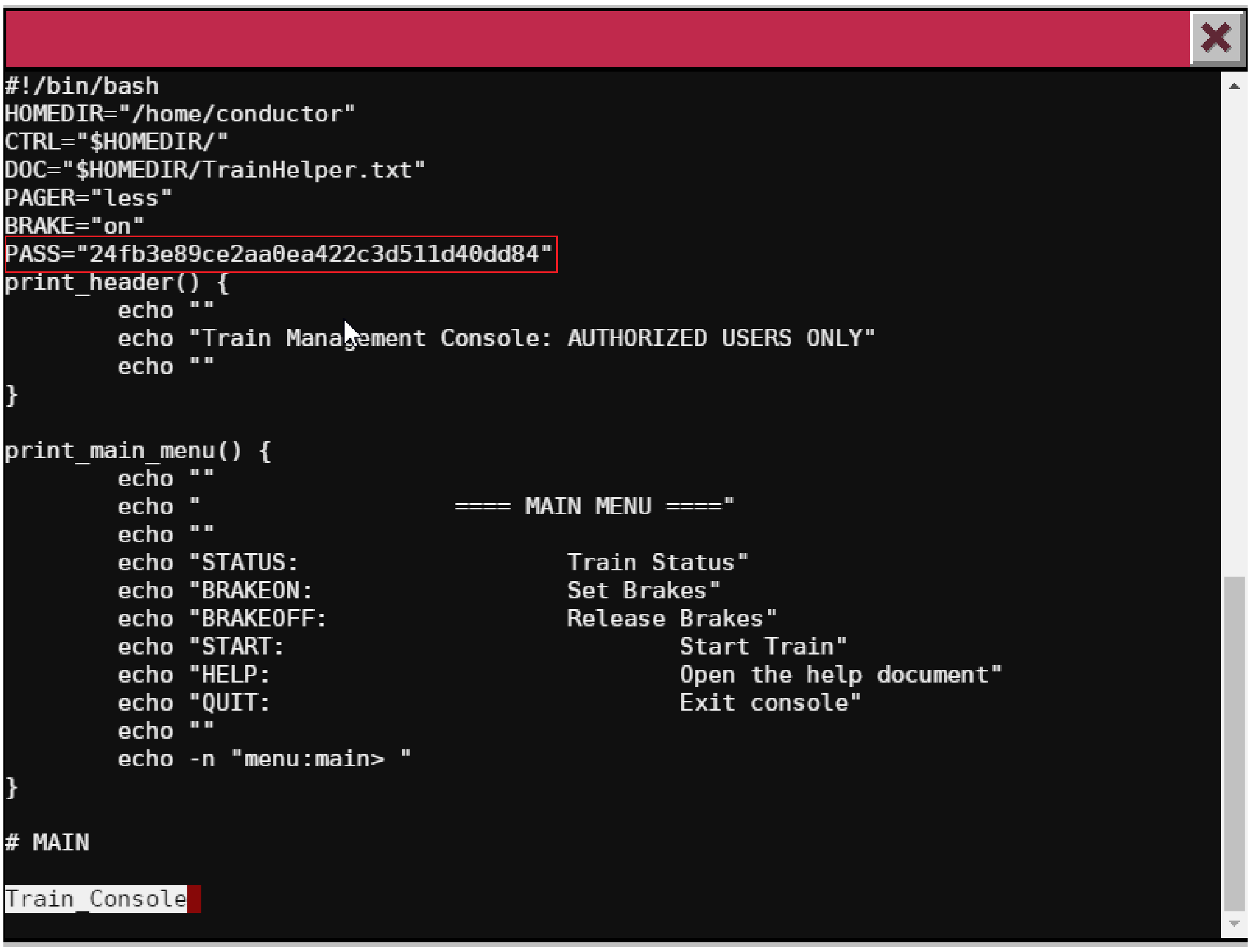
```
Help Document for the Train

**STATUS** option will show you the current state of the train (brakes, boiler, boiler
 temp, coal level)

**BRAKEON** option enables the brakes.  Brakes should be enabled at every stop and whi
le the train is not in use.

**BRAKEOFF** option disables the brakes.  Brakes must be disabled before the **START**
 command will execute.

**START** option will start the train if the brake is released and the user has the co
rrect password.

**HELP** brings you to this file.  If it's not here, this console cannot do it, unLESS
 you know something I don't.


Just in case you wanted to know, here's a really good Cranberry pie recipe:

Ingredients
1 recipe pastry for a 9 inch double crust pie
1 1/2 cups white sugar
1/3 cup all-purpose flour
1/4 teaspoon salt
1/2 cup water
1 (12 ounce) package fresh cranberries
1/4 cup lemon juice
1 dash ground cinnamon
/home/conductor/TrainHelper.txt
```

So this is what is presented by the HELP command. Would you look at that, seems like there is a little hint for us. Now if this is showing the help file with *less*, then typing ! and hitting enter should open a shell for us.



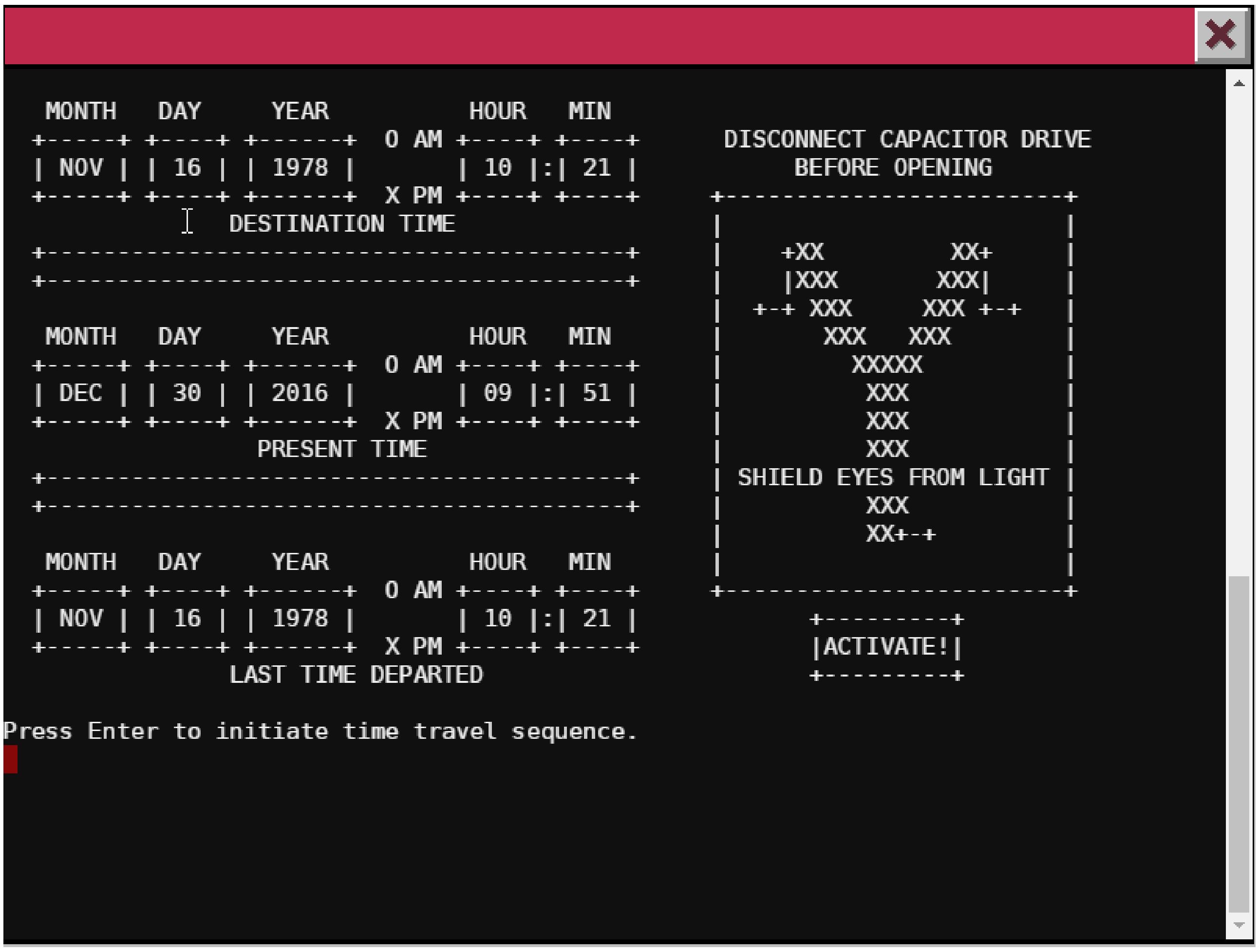
```
Train Management Console: AUTHORIZED USERS ONLY


                ==== MAIN MENU ====

STATUS:                         Train Status
BRAKEON:                        Set Brakes
BRAKEOFF:                       Release Brakes
START:                          Start Train
HELP:                           Open the help document
QUIT:                           Exit console

menu:main> HELP

sh-4.3$ ls
ActivateTrain  TrainHelper.txt  Train_Console
sh-4.3$
```

And there it is, the shell. And also the files that we need to look at. *TrainHelper.txt* contains the help information, *Train_Console* contains the following



```
#!/bin/bash
HOMEDIR="/home/conductor"
CTRL="$HOMEDIR/"
DOC="$HOMEDIR/TrainHelper.txt"
PAGER="less"
BRAKE="on"
PASS="24fb3e89ce2aa0ea422c3d511d40dd84"
print_header() {
        echo ""
        echo "Train Management Console: AUTHORIZED USERS ONLY"
        echo ""
}

print_main_menu() {
        echo ""
        echo "                  ==== MAIN MENU ===="
        echo ""
        echo "STATUS:                       Train Status"
        echo "BRAKEON:                      Set Brakes"
        echo "BRAKEOFF:                     Release Brakes"
        echo "START:                                Start Train"
        echo "HELP:                                 Open the help document"
        echo "QUIT:                                 Exit console"
        echo ""
        echo -n "menu:main> "
}

# MAIN

Train_Console
```

There's the password we need. However, maybe it's possible to run *ActivateTrain* directly.



```
  MONTH   DAY     YEAR          HOUR   MIN
 +-----+ +----+ +-------+  O AM +----+ +----+     DISCONNECT CAPACITOR DRIVE
 | NOV | | 16 | | 1978 |       | 10 |:| 21 |          BEFORE OPENING
 +-----+ +----+ +-------+  X PM +----+ +----+    +-------------------------+
              DESTINATION TIME                   |                         |
 +------------------------------------------+   |    +XX         XX+      |
 +------------------------------------------+   |    |XXX       XXX|      |
                                                 |  +-+ XXX     XXX +-+    |
  MONTH   DAY     YEAR          HOUR   MIN       |        XXX   XXX        |
 +-----+ +----+ +-------+  O AM +----+ +----+   |          XXXXX          |
 | DEC | | 30 | | 2016 |       | 09 |:| 51 |    |           XXX           |
 +-----+ +----+ +-------+  X PM +----+ +----+   |           XXX           |
                PRESENT TIME                     |           XXX           |
 +------------------------------------------+   | SHIELD EYES FROM LIGHT  |
 +------------------------------------------+   |           XXX           |
                                                 |           XX+-+         |
  MONTH   DAY     YEAR          HOUR   MIN       |                         |
 +-----+ +----+ +-------+  O AM +----+ +----+    +-------------------------+
 | NOV | | 16 | | 1978 |       | 10 |:| 21 |           +----------+
 +-----+ +----+ +-------+  X PM +----+ +----+           |ACTIVATE!|
              LAST TIME DEPARTED                         +----------+

Press Enter to initiate time travel sequence.
```

Great success, it can be start with and without the password. So that's the last of the terminals down.