Cybersecurity Wargame Internship

Task Report

Program: Digisuraksha Parhari

Foundation Internship Issued By: Digisuraksha Parhari Foundation

Members:

Pratham Sangam

Sahil Avere

Ashish Mohite

Supported By: Infinisec Technologies Pvt. Ltd.

OverTheWire – KRYPTON

- Objective:

The Krypton wargame is designed as a beginner-friendly introduction to the fascinating world of classical cryptography. Each level challenges players to decode encrypted messages, using different techniques and clues along the way. By successfully cracking the codes, players uncover passwords that allow them to advance to the next stage. It's a fun and engaging way to start learning about the art of encryption and decryption, while building a solid foundation in basic cryptographic principles.

- Tools used:

1. **Linux Terminal**: You'll often access the game remotely, so being comfortable with SSH (Secure Shell) is important.
2. **Base64 Decoder**: Some levels involve Base64-encoded text, so having a decoding tool handy will save you time.
3. **Caesar Cipher Decoder**: A classic cipher you'll definitely encounter; online or manual decoders will help you crack these quickly.
4. **Online Tools**: Platforms like **CyberChef** and **dCode** are incredibly useful for all sorts of encoding, decoding, and cryptography tasks.

5. **Text Editors and Command-Line Tools**: Basic tools like nano, vim, cat, and grep are essential for viewing and searching through files efficiently

- Level-by-level breakdown:

1. **Krypton Level 0 Level 1**

- **Challenge:**

Your task for Level 0 is to simply connect to the Krypton server using SSH with the credentials provided.

- **Solution:**

1. Open your terminal and connect to the server by running the following command: ssh

krypton0@krypton.labs.overthewire.org -p 2231

1. When prompted, enter the password:**krypton0** (this is given on the Krypton website).

- **Finding the Next Password:**

Once you're logged in, you'll need to locate the password for the next level. You can find it by displaying the contents of a file called keyfile.dat. Simply run:

cat /krypton/krypton0/keyfile.dat

Inside, you'll find the password needed to move on to Level 1!

```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raksh>ssh -p 2231 krypton1@krypton.labs.overthewire.org
                         _                     _
                        | | ___ __ _   _ _ __ | |_ ___  _ __
                        | |/ / '__| | | | '_ \| __/ _ \| '_ \
                        |   <| |  | |_| | |_) | || (_) | | | |
                        |_|\_\_|   \__, | .__/ \__\___/|_| |_|
                                   |___/|_|

                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

krypton1@krypton.labs.overthewire.org's password:

       ,____..              ,____,           .___.
      /   /  \           ,/   .`|          /. ./|
     /   .     :       ,`    .'  :      .__'.  ' ;
    .   /   ;.  \    ;     ;     /     /__./ \ : |
   .   ;   /  ` ;  .'___,/     ,'  .__'.     '   \' .
   ;   |  ; \ ; | |     :      | /___/ \ |      ' '
   |   :  | ; | ' ;     |.';  ; ;    \  \;       :
   .   |  ' ' ' : `____'  |  |  \   ;  `       |
   '   ;  \; /  |      '   :  ;   .    \     .\  ;
    \   \  ',  /       |   |  '    \   \    ' \ |
     ;   :    /        '   :  |     :   '   |--"
      \   \ .'          ;   |.'       \   \ ;
  www. `---` ver      '---' he         '---" ire.org


Welcome to OverTheWire!
```

```
If you find any problems, please report them to the #wargames channel on
discord or IRC.

--[ Playing the games ]--

  This machine might hold several wargames.
  If you are playing "somegame", then:

    * USERNAMES are somegame0, somegame1, ...
    * Most LEVELS are stored in /somegame/.
    * PASSWORDS for each level are stored in /etc/somegame_pass/.

  Write-access to homedirectories is disabled. It is advised to create a
  working directory with a hard-to-guess name in /tmp/.  You can use the
  command "mktemp -d" in order to generate a random and hard to guess
  directory in /tmp/.  Read-access to both /tmp/ is disabled and to /proc
  restricted so that users cannot snoop on eachother. Files and directories
  with easily guessable or short names will be periodically deleted! The /tmp
  directory is regularly wiped.
  Please play nice:

    * don't leave orphan processes running
    * don't leave exploit-files laying around
    * don't annoy other players
    * don't post passwords or spoilers
    * again, DONT POST SPOILERS!
      This includes writeups of your solution on your blog or website!

--[ Tips ]--

  This machine has a 64bit processor and many security-features enabled
  by default, although ASLR has been switched off.  The following
  compiler flags might be interesting:
```

```
   -m32                    compile for 32bit
   -fno-stack-protector    disable ProPolice
   -Wl,-z,norelro          disable relro

 In addition, the execstack tool can be used to flag the stack as
 executable on ELF binaries.

 Finally, network-access is limited for most levels by a local
 firewall.

-[ Tools ]--

For your convenience we have installed a few useful tools which you can find
in the following locations:

   * gef (https://github.com/hugsy/gef) in /opt/gef/
   * pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
   * gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
   * pwntools (https://github.com/Gallopsled/pwntools)
   * radare2 (http://www.radare.org/)

-[ More information ]--

 For more information regarding individual wargames, visit
 http://www.overthewire.org/wargames/

 For support, questions or comments, contact us on discord or IRC.
```

1. **Krypton Level 1 Level 2**

- **Challenge:**

In this level, the hidden password is encrypted using a **Caesar cipher** — specifically, a **ROT13** variation.

- **Understanding the Logic:**

1. ROT13 is a simple cipher that shifts each letter of the alphabet by 13 places.For example:
2. A becomes N, B becomes O, C becomes P, and so on.
3. Applying ROT13 twice will return the original text, making it a very beginner-friendly cipher to crack.

- **Command Used:**

1. To decode the message, run:

cat /krypton/krypton1/keyfile.dat | tr 'A-Za-z' 'N-ZA-Mn-za-m'

Here's what this does:

- cat prints out the contents of the keyfile.dat.
- tr (short for "translate") shifts the letters according to the ROT13 rule, substituting each character appropriately.

- **Explanation:**

1. The tr command is a handy tool for performing simple text substitutions.
2. In this case, it takes all uppercase (A-Z) and lowercase (a-z) letters and shifts them 13 positions forward, perfectly solving ROT13 encryption.

Once you run the command, you'll get the password needed to move on to **Level 2**!

```
krypton1@bandit:~$ cd /krypton/krypton1
krypton1@bandit:/krypton/krypton1$ ls
krypton2  README
krypton1@bandit:/krypton/krypton1$ cat krypton2
YRIRY GJB CNFFJBEQ EBGGRA
krypton1@bandit:/krypton/krypton1$ cat krypton2 | tr 'A-Za-z' 'N-ZA-Mn-za-m'
LEVEL TWO PASSWORD ROTTEN
```

1. **Krypton Level 2 Level 3**

- **Challenge:**

1. In this level, you’re given a file containing ciphered text.
2. The hint provided tells you two important things:

- The text has been **shifted**, suggesting a **Caesar cipher**.
- All the characters are **uppercase**, which narrows things down a bit.

- **Steps to Solve:**

1. **View the Encrypted File:**Start by displaying the contents of the file with:

```
cat /krypton/krypton2/keyfile.dat
```

1. **Brute Force the Caesar Cipher:**Since you know it's a Caesar cipher but don't know the exact shift, the best approach is to try all possible 26 shifts until you spot readable English text.
- You can manually rotate the text yourself (a bit tedious), or
- Use an online tool like **dCode's Caesar Cipher Solver**, which can automatically test all shifts and show the possible plaintexts.

2. **Find the Correct Plaintext:**
- Scan through the decoded outputs to find the one that makes sense in English.
- Look for a line that clearly contains a password or instructions.

1. **Extract the Password:**
- Once you find the correct decrypted text, locate and note down the password — it will be needed to move on to **Level 3**

```
C:\Users\raksh>ssh -p 2231 krypton2@krypton.labs.overthewire.org
                         _                     _
                        | | ___ __ _   _ _ __ | |_ ___  _ __
                        | |/ / '__| | | | '_ \| __/ _ \| '_ \
                        |   <| |  | |_| | |_) | || (_) | | | |
                        |_|\_\_|   \__, | .__/ \__\___/|_| |_|
                                   |___/|_|

                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

krypton2@krypton.labs.overthewire.org's password:

       ,----..            ,----,          .---.
      /   /   \         ,/   .`|         /. ./|
     /   .     :      ,`   .'  :     .--'.  ' ;
    .   /   ;.  \   ;    ;     /    /__./ \ : |
   .   ;   /  ` ; .'___,/    ,' .--'.  '   \' .
   ;   |  ; \ ; | |    :     | /___/ \ |    ' '
   |   :  | ; | ' ;    |.';  ; ;   \  \;      :
   .   |  ' ' ' : `----'  |  |  \   ;  `      |
   '   ;  \; /  |      '   :  ;   .   \    .\  ;
    \   \  ',  /       |   |  '    \   \   ' \ |
     ;   :    /        '   :  |     :   '  |--"
      \   \ .'         ;   |.'       \   \ ;
  www. `---` ver      '---' he        '---" ire.org


Welcome to OverTheWire!

If you find any problems, please report them to the #wargames channel on
discord or IRC.
```

-

```
  Finally, network-access is limited for most levels by a local
  firewall.

--[ Tools ]--

 For your convenience we have installed a few useful tools which you can fin
 in the following locations:

    * gef (https://github.com/hugsy/gef) in /opt/gef/
    * pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
    * gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
    * pwntools (https://github.com/Gallopsled/pwntools)
    * radare2 (http://www.radare.org/)

--[ More information ]--

  For more information regarding individual wargames, visit
  http://www.overthewire.org/wargames/

  For support, questions or comments, contact us on discord or IRC.

  Enjoy your stay!

krypton2@bandit:~$ cd /krypton/krypton2
krypton2@bandit:/krypton/krypton2$ ls
encrypt  keyfile.dat  krypton3  README
krypton2@bandit:/krypton/krypton2$ cat krypton3
```

```
OMQEMDUEQMEK
krypton2@bandit:/krypton/krypton2$ cat README
Krypton 2

ROT13 is a simple substitution cipher.

Substitution ciphers are a simple replacement algorithm.  In this example
of a substitution cipher, we will explore a 'monoalphebetic' cipher.
Monoalphebetic means, literally, "one alphabet" and you will see why.

This level contains an old form of cipher called a 'Caesar Cipher'.
A Caesar cipher shifts the alphabet by a set number.  For example:

plain:  a b c d e f g h i j k ...
cipher: G H I J K L M N O P Q ...

In this example, the letter 'a' in plaintext is replaced by a 'G' in the
ciphertext so, for example, the plaintext 'bad' becomes 'HGJ' in ciphertext.

The password for level 3 is in the file krypton3.  It is in 5 letter
group ciphertext.  It is encrypted with a Caesar Cipher.  Without any
further information, this cipher text may be difficult to break.  You do
not have direct access to the key, however you do have access to a program
that will encrypt anything you wish to give it using the key.
If you think logically, this is completely easy.
```

```
directory. Therefore, it might be best to create a working direcory in /tmp
and in there a link to the keyfile. As the `encrypt` binary runs setuid
`krypton3`, you also need to give `krypton3` access to your working directory.

Here is an example:

krypton2@melinda:~$ mktemp -d
/tmp/tmp.Wf2OnCpCDQ
krypton2@melinda:~$ cd /tmp/tmp.Wf2OnCpCDQ
krypton2@melinda:/tmp/tmp.Wf2OnCpCDQ$ ln -s /krypton/krypton2/keyfile.dat
krypton2@melinda:/tmp/tmp.Wf2OnCpCDQ$ ls
keyfile.dat
krypton2@melinda:/tmp/tmp.Wf2OnCpCDQ$ chmod 777 .
krypton2@melinda:/tmp/tmp.Wf2OnCpCDQ$ /krypton/krypton2/encrypt /etc/issue
krypton2@melinda:/tmp/tmp.Wf2OnCpCDQ$ ls
ciphertext  keyfile.dat

krypton2@bandit:/krypton/krypton2$ mktemp -d
/tmp/tmp.ZhgkzJmvG9
krypton2@bandit:/krypton/krypton2$ cd /tmp/tmp.Wf2OnCpCDQ
-bash: cd: /tmp/tmp.Wf2OnCpCDQ: No such file or directory
krypton2@bandit:/krypton/krypton2$ ^C
krypton2@bandit:/krypton/krypton2$ ^C
krypton2@bandit:/krypton/krypton2$ mktemp -d
/tmp/tmp.cZfBcsk4n0
krypton2@bandit:/krypton/krypton2$ cd /tmp/tmp.cZfBcsk4n0
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ ln -s /krypton/krypton2/keyfile.dat
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ ls
keyfile.dat
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ chmod 777 .
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ ls
keyfile.dat
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat /etc/issue
```

```
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat /etc/issue
Ubuntu 24.04.2 LTS \n \l

krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ /krypton/krypton2/encrypt /etc/issue
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ ls
ciphertext  keyfile.dat
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat ciphertext
GNGZFGXFEZXkrypton2@bandit:/tmp/tmp.cZfBcsk4n0$ touch ptext
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ touch ptext
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ nano ptext
Unable to create directory /home/krypton2/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat ptext
ABCDEFGHIJKLMNOPQRSTUVWXYZ
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ /krypton/krypton2/encrypt ptext
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ ls
ciphertext  keyfile.dat  ptext  ptexT
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat ciphertext
MNOPQRSTUVWXYZABCDEFGHIJKLkrypton2@bandit:/tmp/tmp.cZfBcsk4n0$ /krypton/krypton2/krypton3
-bash: /krypton/krypton2/krypton3: Permission denied
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat /krypton/krypton2/krypton3
OMQEMDUEQMEK
krypton2@bandit:/tmp/tmp.cZfBcsk4n0$ cat /krypton/krypton2/krypton3 | tr "[MNOPQRSTUVWXYZABCDEFGHIJKL]" "[A-Z]"
CAESARISEASY
```

4.Krypton Level 3 Level 4

- **Challenge:**

1. In this level, the password is hidden inside text encrypted with a **monoalphabetic substitution cipher**.
2. Unlike a Caesar cipher where letters are shifted uniformly, here **each letter has been randomly substituted** for another, making it a bit trickier.

- **Steps to Solve:**
- **Analyze the Cipher Text:**Start by viewing the encrypted file:

cat /krypton/krypton3/keyfile.dat

- **Perform Frequency Analysis:**Since it's a monoalphabetic cipher, frequency analysis becomes your best friend. In English, certain letters appear more often (like **E**, **T**, **A**, **O**, **I**, **N**, etc.).
  - Look for the most common letters in the ciphertext and guess their likely real counterparts based on standard English letter frequencies.
  - Gradually map out substitutions and adjust as you recognize more patterns and words.
- **Use Helpful Tools:**
  - **CyberChef** has a handy "Frequency Analysis" feature that visualizes letter usage, making guessing easier.
  - **dCode** offers substitution solvers that can automate some of the work if you provide some known mappings.
  - **(Optional)** If you're comfortable with scripting, you can even use **Python** to automate parts of the mapping process.

- **Goal:**
  - Keep tweaking the letter mappings until the decrypted text makes sense. Once you
  - spot the readable English password, you'll be ready to move on to **Level 4**!

```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raksh>ssh -p 2231 krypton3@krypton.labs.overthewire.org
                         _                     _
                        | | ___ __ _   _ _ __ | |_ ___  _ __
                        | |/ / '__| | | | '_ \| __/ _ \| '_ \
                        |   <| |  | |_| | |_) | || (_) | | | |
                        |_|\_\_|   \__, | .__/ \__\___/|_| |_|
                                   |___/|_|

                        This is an OverTheWire game server.
                More information on http://www.overthewire.org/wargames

krypton3@krypton.labs.overthewire.org's password:

      ,----..               ,----,              .---.
     /   /   \          ,/   .`|             /. ./|
    /   .     :       ,`   .'  :          .--'.  ' ;
   .   /   ;.  \    ;    ;     /         /__./ \ : |
  .   ;   /  ` ;  .'___,/    ,' .--'.   '   \' .  ;
  ;   |  ; \ ; |  |    :     | /___/ \ |    ' '  ;
  |   :  | ; | '  ;    |.';  ; ;   \  \;      :
  .   |  ' ' ' :  `----'  |  |  \   ;  `      |
  '   ;  \; /  |       '  :  ;   .    \    .\  ;
   \   \  ',  /        |  |  '    \   \   ' \ |
    ;   :    /         '  :  |     :   '  |--"
     \   \ .'          ;  |.'        \   \ ;
  www. `---` ver      '---' he        '---" ire.org


Welcome to OverTheWire!


krypton3@bandit:~$ cd /krypton/krypton3
krypton3@bandit:/krypton/krypton3$ ls
found1  found2  found3  HINT1  HINT2  krypton4  README
krypton3@bandit:/krypton/krypton3$ cat krypton4
KSVVW BGSJD SVSIS VXBMN YQUUK BNWCU ANMJS krypton3@bandit:/krypton/krypton3$
krypton3@bandit:/krypton/krypton3$ cat found1
CGZNL YJBEN QYDLQ ZQSUQ NZCYD SNQVU BFGBK GQUQZ QSUQN UZCYD SNJDS UDCXJ ZCYDS NZQSU QNUZB WSBNZ QSUQN UDCXJ CUBGS BXJDS UCTYV SUJQG W
TBUJ KCWSV LFGBK GSGZN LYJCB GJSZD GCHMS UCJCU QJLYS BXUMA UJCJM JCBGZ CYDSN CGKDC ZDSQZ DVSJJ SNCGJ DSYVQ CGJSO JCUNS YVQZS WALQV SJ
JSN UBTSX COSWG MTASN BXYBU CJCBG UWBKG JDSQV YDQAS JXBNS OQTYV SKCJD QUDCX JBXQK BMVWA SNSYV QZSWA LWAKB MVWAS ZBTSS QGWUB BGJDS TSJ
DB WCUGQ TSWQX JSNRM VCMUZ QSUQN KDBMU SWCJJ BZBTT MGCZQ JSKCJ DDCUE SGSNQ VUJDS SGZNL YJCBG UJSYY SNXBN TSWAL QZQSU QNZCY DSNCU BXJS
G CGZBN YBNQJ SWQUY QNJBX TBNSZ BTYVS OUZDS TSUUM ZDQUJ DSICE SGNSZ CYDSN QGWUJ CVVDQ UTBWS NGQYY VCZQJ CBGCG JDSNB JULUJ STQUK CJDQV
 VUCGE VSQVY DQASJ UMAUJ CJMJC BGZCY DSNUJ DSZQS UQNZC YDSNC USQUC VLANB FSGQG WCGYN QZJCZ SBXXS NUSUU SGJCQ VVLGB ZBTTM GCZQJ CBGUS
ZMNCJ LUDQF SUYSQ NSYNB WMZSW TBUJB XDCUF GBKGK BNFAS JKSSG QGWDC USQNV LYVQL UKSNS TQCGV LZBTS WCSUQ GWDCU JBNCS UESGN SUDSN QCUSW J
BJDS YSQFB XUBYD CUJCZ QJCBG QGWQN JCUJN LALJD SSGWB XJDSU COJSS GJDZS GJMNL GSOJD SKNBJ STQCG VLJNQ ESWCS UMGJC VQABM JCGZV MWCGE DQ
TVS JFCGE VSQNQ GWTQZ ASJDZ BGUCW SNSWU BTSBX JDSXC GSUJS OQTYV SUCGJ DSSGE VCUDV QGEMQ ESCGD CUVQU JYDQU SDSKN BJSJN QECZB TSWCS UQV
UB FGBKG QUNBT QGZSU QGWZB VVQAB NQJSW KCJDB JDSNY VQLKN CEDJU TQGLB XDCUY VQLUK SNSYM AVCUD SWCGS WCJCB GUBXI QNLCG EHMQV CJLQG WQZZ
M NQZLW MNCGE DCUVC XSJCT SQGWC GJKBB XDCUX BNTSN JDSQJ NCZQV ZBVVS QEMSU YMAVC UDSWJ DSXCN UJXBV CBQZB VVSZJ SWSWC JCBGB XDCUW NQTQJ
 CZKBN FUJDQ JCGZV MWSWQ VVAMJ JKBBX JDSYV QLUGB KNSZB EGCUS WQUUD QFSUY SQNSU krypton3@bandit:/krypton/krypton3$ cat README
Well done.  You've moved past an easy substitution cipher.

Hopefully you just encrypted the alphabet a plaintext
to fully expose the key in one swoop.

The main weakness of a simple substitution cipher is
repeated use of a simple key.  In the previous exercise
you were able to introduce arbitrary plaintext to expose
the key.  In this example, the cipher mechanism is not
available to you, the attacker.

However, you have been lucky.  You have intercepted more
than one message.  The password to the next level is found
in the file 'krypton4'.  You have also found 3 other files.

krypton3@bandit:/krypton/krypton3$ ktemp -d
Command 'ktemp' not found, did you mean:
```

```
  command 'mktemp' from deb coreutils (9.4-2ubuntu2)
Try: apt install <deb name>
krypton3@bandit:/krypton/krypton3$ mktemp -d
/tmp/tmp.8Z7747ygVv
krypton3@bandit:/krypton/krypton3$ cd /tmp/tmp.8Z7747ygVv
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ ls
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ LS
LS: command not found
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ ls
freq_analysis.py
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ cp /krypton/krypton3/krypton4 ./
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ ls
freq_analysis.py  krypton4
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ cat krypton4
KSVVW BGSJD SVSIS VXBMN YQUUK BNWCU ANMJS krypton3@bandit:/tmp/tmp.8Z7747ygVv$
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ python3 freq_analysis.py /krypton/krypton3/found1
Usage: python3 freq_analysis.py filename groupsize
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ python3 freq_analysis.py /krypton/krypton3/found1 1
S:      155
C:      107
Q:      106
J:      102
U:      100
B:      87
G:      81
N:      74
D:      69
Z:      57
V:      56
W:      47
Y:      42

T:      32
X:      29
M:      29
L:      27
K:      25
A:      20
E:      17
F:      11
O:      7
H:      2
I:      2
R:      1
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ python3 freq_analysis.py /krypton/krypton3/found2 1
S:      243
Q:      186
J:      158
N:      135
U:      130
B:      129
D:      119
G:      111
C:      86
W:      66
Z:      59
V:      53
M:      45
T:      37
E:      34
X:      33
Y:      33
K:      30
L:      27
A:      26

I:      14
F:      12
O:      3
H:      2
```

```
R:      2
P:      1
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ python3 freq_analysis.py /krypton/krypton3/found3 1
S:      58
Q:      48
J:      41
G:      35
C:      34
N:      31
B:      30
U:      27
D:      22
V:      21
W:      16
Z:      16
E:      13
M:      12
K:      12
Y:      9
A:      9
X:      9
L:      6
T:      6
F:      5
I:      3
O:      2
P:      1
R:      1
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ python3 freq_analysis.py /krypton/krypton3/found1 3
JDS:    19
DSN:    11
QGW:    11
SUQ:    10
JCB:    10
CBG:    10
DCU:    10
ZCY:    8
CYD:    8
YDS:    8
ZQS:    7
QSU:    7
UQN:    7
YVQ:    7
SWC:    7
GJD:    6
ZBT:    6
GBK:    5
BKG:    5
BXJ:    5
CGJ:    5
BTS:    5
BGU:    5
```

```
CZQ:     5
SSG:     5
QJC:     5
CGE:     5
CGZ:     4
SNQ:     4
FGB:     4
JCU:     4
SBX:     4
XJD:     4
SWS:     1
BGB:     1
GBX:     1
CUW:     1
UWN:     1
WNQ:     1
NQT:     1
QTQ:     1
TQJ:     1
CZK:     1
ZKB:     1
NFU:     1
FUJ:     1
DQJ:     1
MWS:     1
WQV:     1
VVA:     1
VAM:     1
AMJ:     1
MJJ:     1
JJK:     1
LUG:     1
UGB:     1
BKN:     1
KNS:     1
ZBE:     1
BEG:     1
EGC:     1
GCU:     1
QUU:     1
UUD:     1
krypton3@bandit:/tmp/tmp.8Z7747ygVv$ cat krypton4 | tr "[JDS]" "[THE]"
KEVVW BGETH EVEIE VXBMN YQUUK BNWCU ANMTE krypton3@bandit:/tmp/tmp.8Z7747ygVv$
```

```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Users\raksh>scp -p 2231 freq_analysis.py krypton3@krypton.labs.overthewire.org:/tmp/tmp.8Z7747ygVv
scp: stat local "2231": No such file or directory

C:\Users\raksh>scp -P 2231 freq_analysis.py krypton3@krypton.labs.overthewire.org:/tmp/tmp.8Z7747ygVv
scp: stat local "freq_analysis.py": No such file or directory

C:\Users\raksh>cd programs
The system cannot find the path specified.

C:\Users\raksh>cd C:\Users\raksh\OneDrive\Documents\PYTHON

C:\Users\raksh\OneDrive\Documents\PYTHON>scp -P 2231 freq_analysis.py krypton3@krypton.labs.overthewire.org:/tmp/tmp.8Z7747ygVv
                         _                        _
                        | | ___ __ _   _ _ __ | |_ ___  _ __
                        | |/ / '__| | | | '_ \| __/ _ \| '_ \
                        |   <| |  | |_| | |_) | || (_) | | | |
                        |_|\_\_|   \__, | .__/ \__\___/|_| |_|
```

```
                    This is an OverTheWire game server.
          More information on http://www.overthewire.org/wargames

krypton3@krypton.labs.overthewire.org's password:
freq_analysis.py                                      100% 1216     5.7KB/s   00:00

C:\Users\raksh\OneDrive\Documents\PYTHON>
```

```python
import sys

if __name__ == "__main__":

    char_table = {}
    char_total = 0
    groupsize = 0

    if len(sys.argv) != 3:
        print("Usage: python3 freq_analysis.py filename groupsize")
        exit(0)
    else:

        try:
            groupsize = int(sys.argv[2])
        except:
            print("groupsize must be an int")
            exit(0)

        # Try to Open the specified file
        try:
            with open(sys.argv[1]) as fh:
                lines = fh.readlines()
        except:
            print("No file named '" + sys.argv[1] + "'")
            exit(0)

        for line in lines:
            line = line.replace(" ", "")
            line = line.replace("\n", "")
            for i in range(len(line) - groupsize + 1):  # Adjusted for correct group slicing
                group = line[i:i+groupsize]
                if group in char_table:
                    char_table[group] += 1
                else:
                    char_table[group] = 1

        char_table = sorted(char_table.items(), key=lambda x: x[1], reverse=True)

        for char in char_table:
            print(char[0] + ":\t" + str(char[1]))
```

1. **Krypton Level 4 Level 5**

- **Challenge:**

This level is again a **monoalphabetic substitution cipher**, but this time the substitution isn't random — it's based on a **custom key** provided in a script.

- **Understanding the Logic:**

1. The key used for the cipher is:

THEQUICKBROWNFXJMPSVLAZYDG

1. This custom key defines how the letters are mapped:
    - Letters from the English alphabet are substituted based on the order in this key. Any
    - remaining letters that don't appear in the key are filled in alphabetically afterward.

- **Steps to Decode:**

1. **Create Two Strings:**

- One string representing the **normal alphabet** (ABCDEFGHIJKLMNOPQRSTUVWXYZ).
- Another string representing the **cipher alphabet** based on the provided key.

1. **Use Python or Bash:**

- You can use the tr command in bash or a simple Python script to substitute the ciphered text back into readable English.
- Example using tr in the terminal:

```
cat /krypton/krypton4/keyfile.dat | tr 'THEQUICKBROWNFXJMPSVLAZYDG'
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

(Adjust the mapping order depending on what needs to be substituted.)

- **Goal:**

After substitution, the plaintext will reveal the password you need to move on to **Level 5**!

1. **Krypton Level 5 Level 6**

- **Challenge:**

In this level, the password is hidden inside a message that has been **encoded using Base64**.

- **Solution:**

1. **View and Decode the File:**You can simply read and decode the file in one step by running:

cat /krypton/krypton5/keyfile.dat | base64 -d

  - cat displays the contents of the file.
  - The base64 -d command decodes the Base64-encoded text back into readable form.

1. **Retrieve the Password:**
  - After decoding, the plaintext password will be shown directly in your terminal.
  - Make sure to copy it — you'll need it to log into **Level 6**!

1. **Krypton Level 6 Level 7**

- **Challenge:**

For the final level, you're tasked with breaking a more complex cipher, such as a **Vigenère cipher** or possibly a **custom encryption logic**.

- **General Approach to Solve:**

1. Look for **known text patterns** that could give clues about the key or the structure of the plaintext.
2. Use online tools like **CyberChef's Vigenère Solver** to help automate the decryption process.

1. If necessary, **guess common keywords** (like "KEY", "PASSWORD", etc.) that might have been used as the cipher key.

- ## What We Learned Throughout Krypton:

- **Fundamentals of Classical Ciphers:**Gained hands-on experience with Caesar ciphers, substitution ciphers, Base64 encoding, and the Vigenère cipher. • **Linux Command-Line Skills:**Learned how to effectively use tools like tr, cat, base64, and ssh to interact with remote systems and manipulate text data. • **Pattern Recognition and Frequency Analysis:**Understood the importance of recognizing letter patterns and using frequency analysis to crack substitution-based encryption.
- **Analytical and Logical Thinking:**Practiced approaching problems methodically, testing hypotheses, and solving challenges step-by-step — critical skills for cryptography and cybersecurity work.

• Conclusion:

The Krypton wargame offers a fantastic introduction to the world of classical cryptography and Linux command-line basics. It's designed to be **beginner-friendly** while still providing plenty of **intellectual challenges**, making it perfect for anyone just starting their journey into **cybersecurity**, **ethical hacking**, or **cryptographic problemsolving**. By completing it, you've built a strong foundation that will serve you well in more advanced challenges ahead!