## Unit 2 Progetto

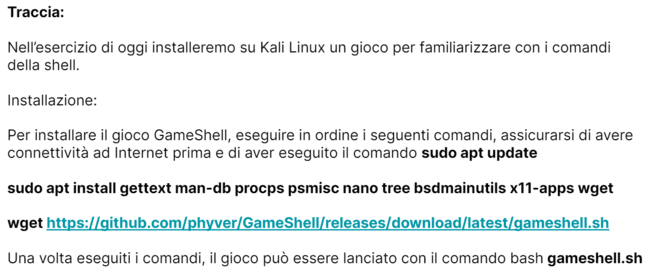


Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Per installare GameShell ho aggiornato apt e bsdmainutilis, non so se ho fatto bene ma ho anche installato mariadb server perché avevo problemi con l’installazione di GameShell.

Come da guida ho fatto partire il gioco con il comando bash gameshell.sh

# Mission 1

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Lista Comandi usati

pwd : mostra la mia posizione attuale

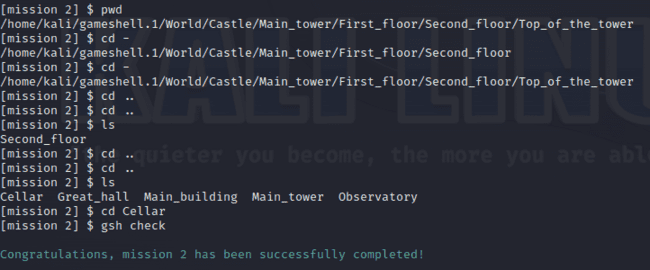
cd : mi sposta nella directory scritta

ls : mi mostra tutti i file accessibile da quella cartella

# Mission 2

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente



cd - serve per tornare nella directory precedente

cd .. invece torna indietro di directory

# Mission 3

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

cd si usa per tornare alla directory principale

cd più il percorso mi sposta all’interno dell’ultima cartella scritta

# Mission 4

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, menu

Descrizione generata automaticamente

mkdir più nome file crea una cartella all’interno della cartella in cui si è entrati

# Mission 5

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

rm più nome file serve a eliminare quei file

# Mission 6

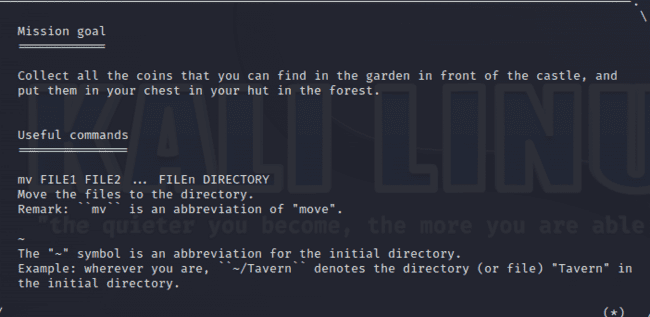


Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

mv più nome serve a spostare quei file

~/nome cartelle serve a dare il percorso

# Mission 7

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

ls mostra i percorsi all’interno ls -A mostra quelli invisibili

Premendo TAB due volte dopo aver scritto le iniziali di un file completa il nome del file

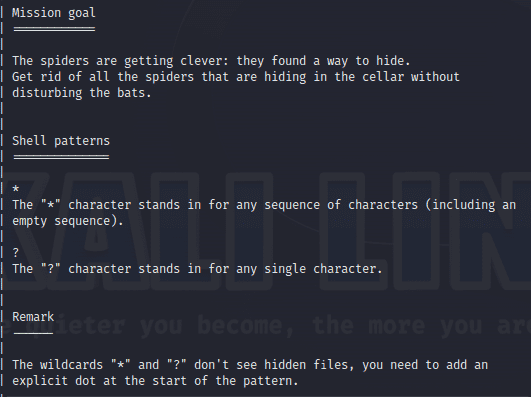
# Mission 8

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente rm \*nome file\* elimina tutti i file con lo stesso nome

# Mission 9

 Immagine che contiene testo, schermata, Carattere, menu

Descrizione generata automaticamente

Stesso procedimento di prima solo che questi ragni hanno un punto davanti

rm .\*nome file\*

# Mission 10

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, menu

Descrizione generata automaticamente

cp più nome file li copia nel percorso (da specificare)

# Mission 11

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Stessa procedimento spiegato prima, mettendo le inziali e premendo TAB mi completa il nome del file

# Mission 12 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamenteImmagine che contiene testo, schermata, Rettangolo, design Descrizione generata automaticamente Immagine che contiene testo, schermata, Carattere, menu Descrizione generata automaticamente

ls – l oppure ls -altr comando per vedere in file in cartella con dettagli

cat più nome file stampa l’immagine

# Mission 13

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

cal apre il calendario

# Mission 14

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

alias mi crea un comando, quindi ora scrivendo la è come se inserissi il comando ls -A

# Mission 15

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene schermata, testo, software, Software multimediale

Descrizione generata automaticamente

nano più percorso file più nome .txt creo direttamente e scrivo all’interno

# Mission 16

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Con questo esercizio ho creato una scorciatoia per modificare il file journal.txt

Ovunque io mi trovi nel castello scrivendo journal, mi apre il txt.

alias journal =”nano ~/Forest/Hut/Chest/journal.txt”

# Mission 17

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Per completare questa missione avevo 20 secondi, dovevo entrare nella cartella del ragno regina ed eliminare solo il file della regina senza toccare il pipistrello. Per farlo velocemente ho usato il comando TAB

# Mission 18 Immagine che contiene testo, schermata, Carattere, design Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Non sono riuscita a fare gli screen, comunque con il comand xeyes mi apriva un programma che mostravano due occhi che seguivano il movimento del mio mouse. Premendo control -c mi chiudeva questo programma

# Mission 19 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Ho spostato manualmente il file dalla cartella

# Mission 20 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

# Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

# 

Ho usato il comando tree per vedere l’albero dal terminale e tramite il comando mv seguendo il percorso ho spostato il coin

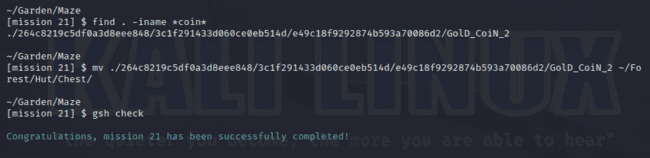
# Mission 21 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, informazione

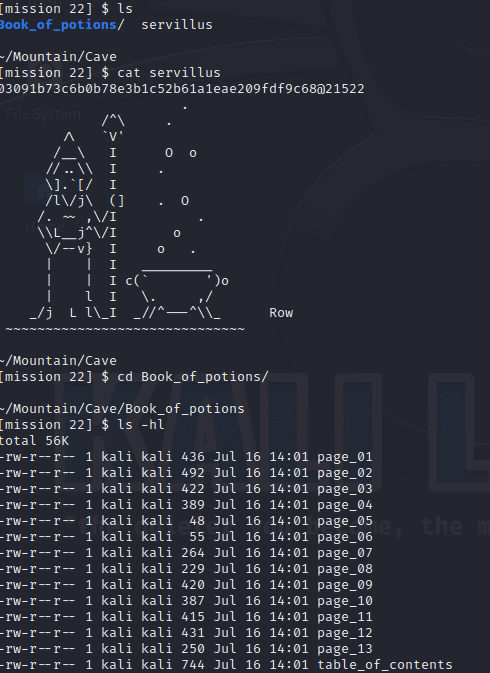
Descrizione generata automaticamente

find mi segnala tutti i file presenti, ed erano troppi

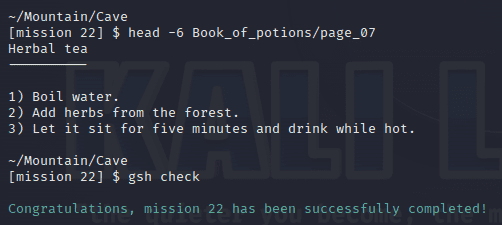
così uso find . -iname \*nome file\* e mi trova il file specifico da me richiesto (la i prima di name sta per case-insensitive, dato che con name non lo trovavo in quanto coin era scritto CoiN)



# Mission 22 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

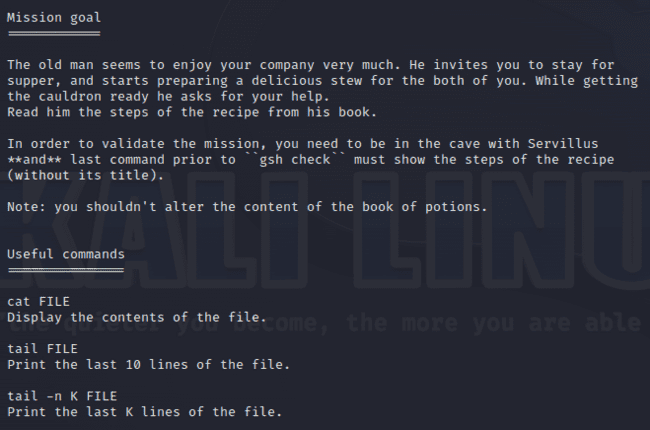
Immagine che contiene testo, schermata, Carattere

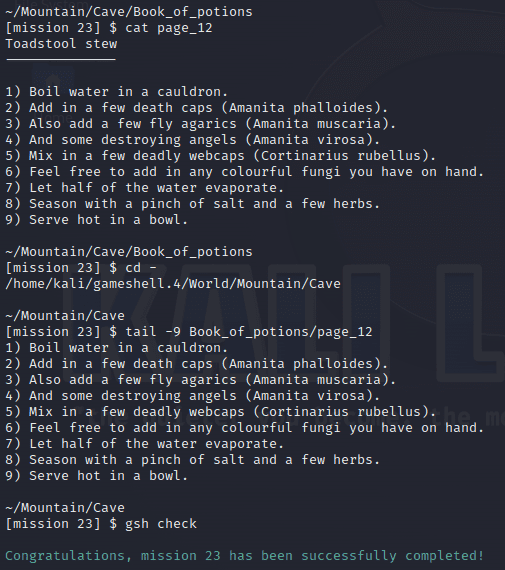
Descrizione generata automaticamente



Con il comando head – 6 ho indicato quante righe mostrare in quanto la missione mi richiedeva di stare nella Grotta e di mostrare solo titolo e ricetta del tea, senza mostrare altro.

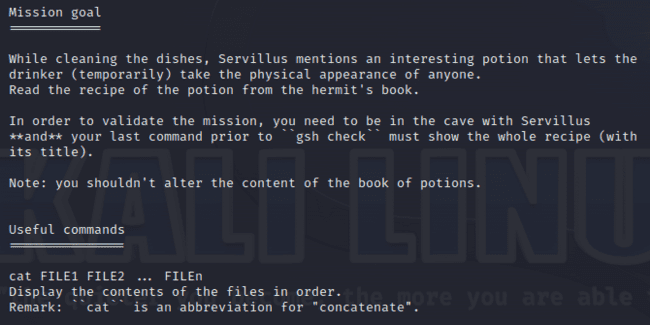
# Mission 23

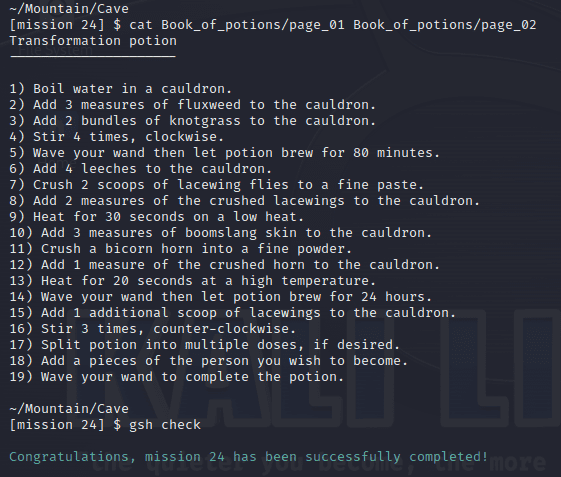




Per questa missione stesso ragionamento della precedente, ma invece del comando head (che si usa partendo dall’inizio), si è usato il comando tail (parte dal fondo)

# Mission 24





Per questo esercizio dovevo solo usare il cat

# Missione 25 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Stesso procedimento delle missioni precedenti, per completare la missione dovevo mostrare solo i passaggi senza altre scritte. Quindi ho usato il cat e poi ho richiesto con tail di mostrare 16 righe.

# Mission 26 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Per la missione stesso procedimento delle precedenti quindi tramite i comandi head and tail ho detto i parametri di quello che volevo visualizzare.

# Mission 27

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata, design

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

ps lista dei processi attivi

kill più n file distrugge il processo in questione

# Mission 28

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

kill -s TERM serve a terminare un programma

kill -s KILL è un comando più brutale per uccidere il programma

# Mission 29

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Questa missione mi chiedeva di aprire l’albero dei processi e di eliminare solo le spells del folletto e di non toccare quelle delle fata e uso il comando pstree -p $$

 kill -s TERM + i processi

Oltre agli incantesimi dovevo eliminare il carbone fatto con il comando rm -v \*coal\*

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

# Mission 30 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# Mission 31

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Immagine che contiene testo, schermata, computer

Descrizione generata automaticamente

Questo esercizio mi diceva che all’interno della libreria potevo trovare un libro di matematica, che potevo usare per risolvere le moltiplicazioni richieste Merlino. Per farlo ho cercato il libro, ho copiato il nome; e quando è partita “l’interrogazione di Merlino” ho dato il comando gsh check (che serve per completare la missione) aggiungendo < (per dare l’input) più il file copiato prima cioè Mathematic\_101. In questo modo mi ha completato in automatico le moltiplicazioni.

# Mission 32

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

La missione mi chiedeva mettere i libri dentro il Drawer dando semplicemente l’output.

Quindi ls nome file > directory/file.txt

# Mission 33 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

L’esercizio mi chiede di fare una lista dei libri che contiene il compost gsh cioè il Glutatione, però la lista dei libri non ha la scritta gsh. Quindi con i comandi che mi dice di scrivere vado a dire di farmi una lista dei libri che contiene le lettere gsh anche non in sequenza.

Il comando usato è grep che non è nient’altro che un filtro

-l e -i servono a indicare i parametri del Glutatione (gsh)

Con 2>/dev/null restituisco l’errore

Immagine che contiene testo, schermata, Carattere, menu

Descrizione generata automaticamente

grep -l -i gsh grimoire\_\* 2>/dev/null

# Mission 34 \*\*\* Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Nella missione dobbiamo decifrare il messaggio confuso di Merlino, l’esercizio ci dice di far partire il file come un programma ./merlin e di dargli un output senza errori quindi

./merlin 2> merlin\_stdout.txt

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Immagine che contiene testo, Carattere, schermata, tipografia

Descrizione generata automaticamente

# Mission 35

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Per cambiare i permessi di una directory posso usare

chmod +rwx più nome del file per aggiungere dei permessi

chmod -rwx più nome della directory per rimuovere i permessi

chmod +x più nome del file per illudere i permessi

chmod -wx più nome del file per togliere i permessi di scrittura

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# Mission 36 Immagine che contiene testo, schermata, Carattere, design Descrizione generata automaticamente

chmod +r più nome del file aggiunge permessi di lettura

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# Mission 37

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

# Mission 38 \*\*\*

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Per trovare il rubino ho usato il comando find + -type f per indicare un file nelle directory regolare grep cerca il percorso specifico

# Mission 39 \*\*\*

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Stesso procedimento di prima più o meno

# Mission 40 \*\*

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

L’esercizio ti chiedeva di usare il comando grep togliendo la spunta boring\_object

Cosi si poteva trovare la pergamena. Con la pergamena e il comando grep togliendo la spunta paid si trovano i debiti del Re

# Mission 41 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Con grep – v “paid” più il nome \*pergamena\* indico i parametri, wc -l serve a numerare questi parametri

# Mission 42 Immagine che contiene testo, schermata, Carattere Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Gameshell mi è piaciuto, e fin da subito volevo completare tutti i livelli.   
Ho avuto qualche problema in alcune missioni, facevo errori di sintassi molto banali, che ho risolto leggendo bene la finestra del Mission goal.   
Ho avuto seri problemi nella missione 34, e dalla 38 in poi. Per questo ho cercato su internet e ho trovato un video di un francese che eseguiva gli esercizi, oltre a copiare il procedimento ho cercato di capirne il ragionamento, però non parlando il francese non ho ottenuto granché.   
Esempio nella 38 e 39, il ragazzo ha usato il comando find . -type f -exec grep -H ruby ‘{}’ \; anche leggendo la spiegazione della missione e aprendo il manuale man, non penso che sarei riuscita a finirla da sola, e ancora non le ho ben comprese.  
La 40 e la 41 una volta viste erano banali (avrei potuto farle da sola impegnandomi un po').

Mentre per la 42 non mi è chiaro i parametri messi da lui.  
tr è il comando per tradurre, quindi tr “a-z” è sensato in quanto do il parametro dalla a alla z.  
La consegna mi dice che per tradurre gli altri segreti sono state utilizzate le lettere tra la 10 e la 16, quindi dalla j alla p.  
Da sola avevo fatto vari tentativi, mettendo queste 6 lettere, ma non mi davano un messaggio leggibile, poi guardando lui, ho visto che metteva “za” tra le lettere del secondo parametro per lui erano così “n-za-o”.  
Non ho ben capito il ragionamento e la differenza dietro questi due parametri.  
Nel mio caso tr “a-z” “m-n” < secret\_message e tr “a-z” “m-za-n” < secret\_message.   
Il primo mi dava un messaggio illeggibile, mentre il secondo era la traduzione.  
Ho capito che le prime lettere “a-z” servono per dare il primo parametro dalla a alla z, e che il secondo “ m-za-n” per tradurre, ma non capisco l’utilizzo di za tra la m e la n. Forse perché era a specchio come diceva la consegna, quindi dovevo trovare due lettere che al contrario mi traducevano il messaggio? Quindi m e n erano le due lettere e za indicava dalla z alla a?