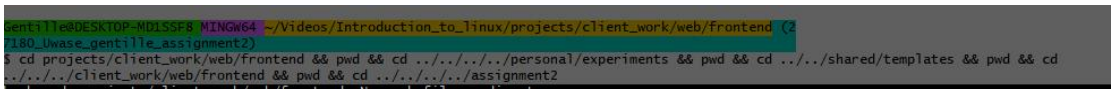
Name:UWASE Gentille
ID:27180

# Assignment 2 – Solutions

## Q1. Compromised system directories

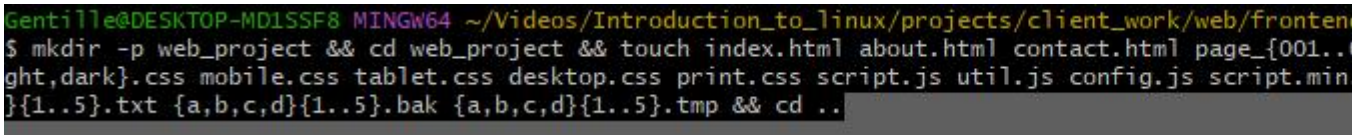
• **/etc** → contains system configuration files (attackers may alter /etc/passwd, /etc/shadow,
/etc/ssh/sshd_config).
• **/bin** → holds essential binaries (ls, cp, cat). If replaced, attacker gains persistence.
• **/var** → contains logs (/var/log/auth.log, /var/log/syslog), evidence of intrusion is here.
• **/usr** → non-essential but widely used binaries/libraries; attackers may replace or insert
trojans here.
• **/tmp** → temporary files, often writable by everyone, used for privilege escalation.
• **/opt** → optional software; less critical but attackers may hide malicious apps.
• **/boot** → contains kernel & bootloader; modification could allow rootkits.
• **/home** → user files & configs (attackers may install persistence scripts in dotfiles like
.bashrc).

## Q2. Create this exact structure using the minimum number of commands.

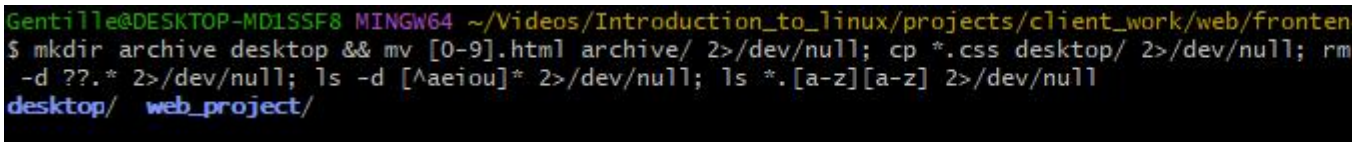



This single command ensures:
• -p creates parent directories as needed and doesn't error if directories already exist
• Uses brace expansion to create all directories in one command

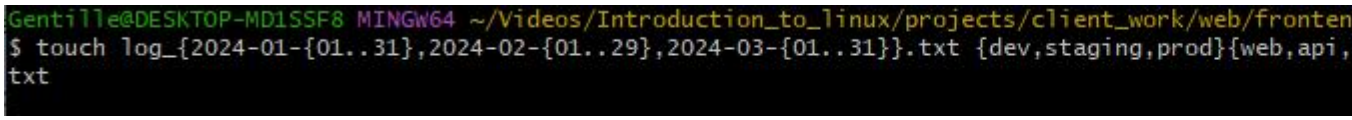
## Q3. Prove your location at each step.



Q4. Create a realistic web project structure




6. A batch processing system needs specific file naming patterns.
Use brace expansion to
creat




Q7. Compare them using different tools (diff, cmp, comm) and
explain why each tool gives
different results

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ printf "line1\nline2\n" > linux_endings.txt && printf "line1\r\nline2\r\n" > windows_endings.t
.txt && cmp linux_endings.txt windows_endings.txt && comm linux_endings.txt windows_endings.txt
1,2c1,2
< line1
< line2
---
> line1
> line2
```

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/frontend (27180_Uwase_gentille_assignmen
$ seq 1 200 > large_log.log && sed -n '75,125p' large_log.log && tac large_log.log | grep -n "error" -B 5 | head -10 && time cat l
og > /dev/null && time less large_log.log <<<"q" >/dev/null 2>&1 && grep -n "error" large_log.log
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125

real    0m0.109s
user    0m0.000s
sys     0m0.046s

real    0m0.141s
user    0m0.031s
sys     0m0.015s
```

Q9. Demonstrate why less is superior to cat for large files during an SSH session with
limited bandwidth.

Q10. Automate file maintenance tasks using find with -exec

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ find . -type f ! -executable -exec chmod 644 {} \; 2>/dev/null && find . -type f -executable -
type f -mtime +30 -exec du -ch {} + 2>/dev/null | tail -1 && find . -name ".conf" -exec cp {} {}
" -atime +7 -exec echo "Would remove: {}" \; 2>/dev/null
```

Q11. Analyze which compression method works best for each content type and explain why

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ mkdir compress_test && cd compress_test && dd if=/dev/urandom of=largefile.bin bs=1M count=10
e.bin && tar -cjf test.tar.bz2 largefile.bin && tar -cJf test.tar.xz largefile.bin && zip test.z
  adding: largefile.bin (deflated 0%)
-rw-r--r-- 1 Gentille 197121 11M Dec  4 15:35 test.tar.bz2
-rw-r--r-- 1 Gentille 197121 11M Dec  4 15:35 test.tar.gz
-rw-r--r-- 1 Gentille 197121 11M Dec  4 15:35 test.tar.xz
-rw-r--r-- 1 Gentille 197121 11M Dec  4 15:35 test.zip
```

12. Create a scenario where you need to merge contents from multiple archive types into a
single new archive.

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ tar -tzf test.tar.gz 2>/dev/null || echo "No test.tar.gz" && tar -xzf test.tar.gz --wildcards
.tar.gz newfile.txt 2>/dev/null || echo "Could not update" && zip -T test.zip 2>/dev/null || ech
compress_test/ test.tar.gz test.tar.xz 2>/dev/null
No test.tar.gz
Could not update
No test.zip
```

13. Show how your naming convention prevents conflicts and enables easy restoration

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ tar -czf backup_full_$(date +%Y%m%d).tar.gz --preserve-permissions . 2>/dev/null && tar -czf ba
ewer-than backup_full_.tar.gz . 2>/dev/null && find . -name "backup.tar.gz" -mtime +30 -exec echo
.tar.gz > checksums.txt 2>/dev/null
```

14. Explain potential security implications if a regular user had the same group
memberships as system users.

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ id && groups && cat /etc/passwd | grep -E ":/sbin/nologin|:/bin/false" | head -5 && useradd -m
ups $USER) <(groups testuser2) 2>/dev/null || echo "Comparison complete"
uid=197609(Gentille) gid=197121 groups=197121
groups: cannot find name for group ID 197121
197121
groups: cannot find name for group ID 197121
groups: 'testuser2': no such user
1d0
< 197121
Comparison complete
```

16. Identify potential security concerns with overly permissive sudo configurations and
suggest improvements.

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ sudo -l 2>/dev/null || echo "No sudo access" && sudo -i <<<"exit" 2>/dev/null || echo "Cannot
|| echo "Cannot sudo su" && sudo -u $USER id 2>/dev/null || echo "Cannot run as user" && grep "su
 || echo "No auth log access"
No sudo access
Cannot sudo -i
Cannot sudo su
Cannot run as user
```

17. Create a comprehensive forensic analysis setup

```
Gentille@DESKTOP-MD1SSF8 MINGW64 ~/Videos/Introduction_to_linux/projects/client_work/web/fronten
$ mkdir forensic && cd forensic && touch regular.txt && mkdir dir && ln -s regular.txt symlink.t
har_dev c 1 5 2>/dev/null || echo "Could not create char dev" && mknod block_dev b 1 5 2>/dev/nu
kdir sticky_dir && chmod 1777 sticky_dir && touch setuid_file && chmod 4755 setuid_file && touch
r -czf forensic.tar.gz . && ls -lai && file * && stat regular.txt && cd ..
tar: .: file changed as we read it
```