Audio Forensics — Student Lab

Find the Hidden File — Kali Linux / Audacity Edition

Carroll Bell Education Center
Instructor: Daniel Peña

⚠ Ethics (must read)

Only run these exercises on instructor-provided files and on lab machines. Do not attack or attempt to crack files or systems you do not have explicit permission to test. If you are unsure, ask your instructor.

Goal

You are given two audio files: harlandale1.wav and harlandale2.wav. One contains a hidden message. Use command-line tools and Audacity to determine which file contains hidden data, discover the passphrase with a controlled dictionary attack, and (after instructor approval) extract the secret.

Files Provided

• harlandale1.wav
• harlandale2.wav
• Audacity-x86_64.AppImage (portable — run with ./Audacity-x86_64.AppImage)
• student_attack.sh (dictionary attack script) — do NOT change TARGET value (you will create this script in Step 6)

Student Steps

Follow the steps below. Record outputs, take screenshots where requested, and submit your findings as instructed.

1) Basic file info — baseline

Run these commands and save the outputs/screenshots:

```
file harlandale1.wav
file harlandale2.wav
```

2) Quick size check

Check file sizes (record the values; don't jump to conclusions):

```
ls -lh harlandale1.wav harlandale2.wav
```

3) Audacity + hexdump comparison (first-look)

Open both files in Audacity and compare them visually (waveform and spectrogram). Then inspect the first bytes of each file using hexdump. At this stage, you will probably conclude the files look identical — that is intentional.

A. Audacity visual comparison — steps

1. Run: ./Audacity-x86_64.AppImage
2. File → Import → Audio… → select both files
3. Switch each track between Waveform and Spectrogram
4. Zoom in (Ctrl + 1) and inspect silent areas and track endings
5. Take screenshots of waveform and spectrogram for both files

B. Quick hexdump head (first bytes)

```
hexdump -C harlandale1.wav | head -n 40 > h1_head.txt
hexdump -C harlandale2.wav | head -n 40 > h2_head.txt
nl -ba h1_head.txt
nl -ba h2_head.txt
```

❖ Stop here and form an initial hypothesis

Based on the waveform, spectrogram, and the first-bytes hexdump, write a short hypothesis answering: Do you think the two files are identical or different? List the three strongest reasons supporting your view.

1. _____
2. _____
3. _____

4) Deeper forensic checks (reveal the difference)

Now run deeper checks that will expose differences even if the files looked identical earlier.

A. File sizes (again)

```
ls -lh harlandale1.wav harlandale2.wav
```

B. Hashes (cryptographic proof)

```
sha256sum harlandale1.wav harlandale2.wav
```

C. Byte-by-byte compare (find differing offsets)

```
cmp -l harlandale1.wav harlandale2.wav | head -n 40
```

D. Inspect bytes around an offset (example)

```
hexdump -C -s 10200 -n 160 harlandale1.wav | sed -n '1,40p'
hexdump -C -s 10200 -n 160 harlandale2.wav | sed -n '1,40p'
```

Write which command gave the most convincing evidence that the files are different:

_____

5) Final hypothesis

Using all evidence (visual, hexdump head, file size, hashes, cmp) decide which file contains the hidden message and explain your reasoning (3 points):

6) Controlled dictionary attack (find the passphrase)

Create a small wordlist (keep it short so it finishes quickly):

```
cat > small_wordlist.txt <<'W'
cyberclass
Harlandale
Password123
Flag2025
harlandale1
student1
schoolrocks
linsa
W
```

Clean the wordlist to remove hidden characters:

```
tr -d '\r' < small_wordlist.txt > small_wordlist.clean.txt
mv small_wordlist.clean.txt small_wordlist.txt
```

Now create the student attack script file (student_attack.sh). Copy the exact script below into a file with that name.

```
#!/usr/bin/env bash
# student_attack.sh  (student-facing)
TARGET="harlandale1.wav"        # instructor-provided file (do not change)
WORDLIST="small_wordlist.txt"
TMPDIR="attack_temp"
mkdir -p "$TMPDIR"

if [ ! -f "$WORDLIST" ]; then
  echo "Missing wordlist: $WORDLIST"; exit 2
fi

echo "[*] Starting controlled dictionary attack on $TARGET using $WORDLIST"
echo "[*] This script will REPORT the working passphrase ONLY and will remove any extracted secret."

while IFS= read -r raw || [ -n "$raw" ]; do
  # normalize the candidate (remove CRs & trim)
  pw=$(printf '%s' "$raw" | tr -d '\r' | sed -e 's/^[[:space:]]\+//' -e 's/[[:space:]]\+$//')
  [ -z "$pw" ] && continue

  printf "Trying: »%s«\n" "$pw"

  OUT="$TMPDIR/out.$$"
  # attempt extraction quietly to OUT
  steghide extract -sf "$TARGET" -p "$pw" -xf "$OUT" 2>/dev/null

  # if an output file was created and is non-empty, assume success
  if [ -s "$OUT" ]; then
    printf "\n+++ SUCCESS: passphrase is -> %s +++\n\n" "$pw"
    # securely delete extracted file (never print its contents)
    shred -u "$OUT" 2>/dev/null || rm -f "$OUT"
    rm -rf "$TMPDIR"
    exit 0
  else
    rm -f "$OUT" 2>/dev/null
    echo "Tried: $pw  (no)"
  fi
done < "$WORDLIST"

echo
echo "No passphrase found in $WORDLIST"
exit 1
```

Make the script executable and run it:

```
chmod +x student_attack.sh
./student_attack.sh | tee attack_attempts.log
```

Save attack_attempts.log and include the successful passphrase line in your submission (if found).

7) Extraction (teacher-controlled)

After you submit your hypothesis, ask the instructor for the discovered passphrase. With the passphrase you can extract the secret yourself using steghide, or the instructor may extract and reveal it during class.

```
steghide extract -sf <suspected_file.wav> -p 'THEPASS' -xf secret.txt
cat secret.txt
```

Submission (what to turn in)

• attack_attempts.log (or the passphrase line)
• Audacity screenshots (waveform + spectrogram)
• Outputs of: file, ls -lh, sha256sum, cmp -l (copy/paste or screenshots)
• Initial hypothesis, final hypothesis, and a short forensic note (1 page max)

Hints

• Steganography (LSB) often leaves no visible/audio sign — waveforms and spectrograms can look identical.
• Hashes are the quickest cryptographic proof of file modification.
• Use cmp -l to find exact differing byte offsets and then inspect nearby bytes with hexdump -C -s -n 64.

Good luck — document your evidence carefully and think like a forensic analyst.