

# Ataque de emanación acústica: KeyTap



**Alumno:** *José Ángel Dorado González*  
**Profesor:** *Juan Antonio Romero del Castillo*  
**Asignatura:** *Seguridad informática*

# ÍNDICE

<b>Introducción</b>	<b>2</b>
<b>Funcionamiento kbd-audio</b>	<b>2</b>
<b>Instalación</b>	<b>4</b>
<b>Ejecución KeyTap</b>	<b>6</b>
<b>Ejecución KeyTap2</b>	<b>11</b>
<b>Videos</b>	<b>12</b>

## Introducción

En este documento vamos a hablar sobre los ataques de emanación acústica del teclado, el objetivo principal de estos ataques es aprovechar el sonido que se produce al presionar las teclas del teclado para lograr adivinar el contenido del texto que se está escribiendo.

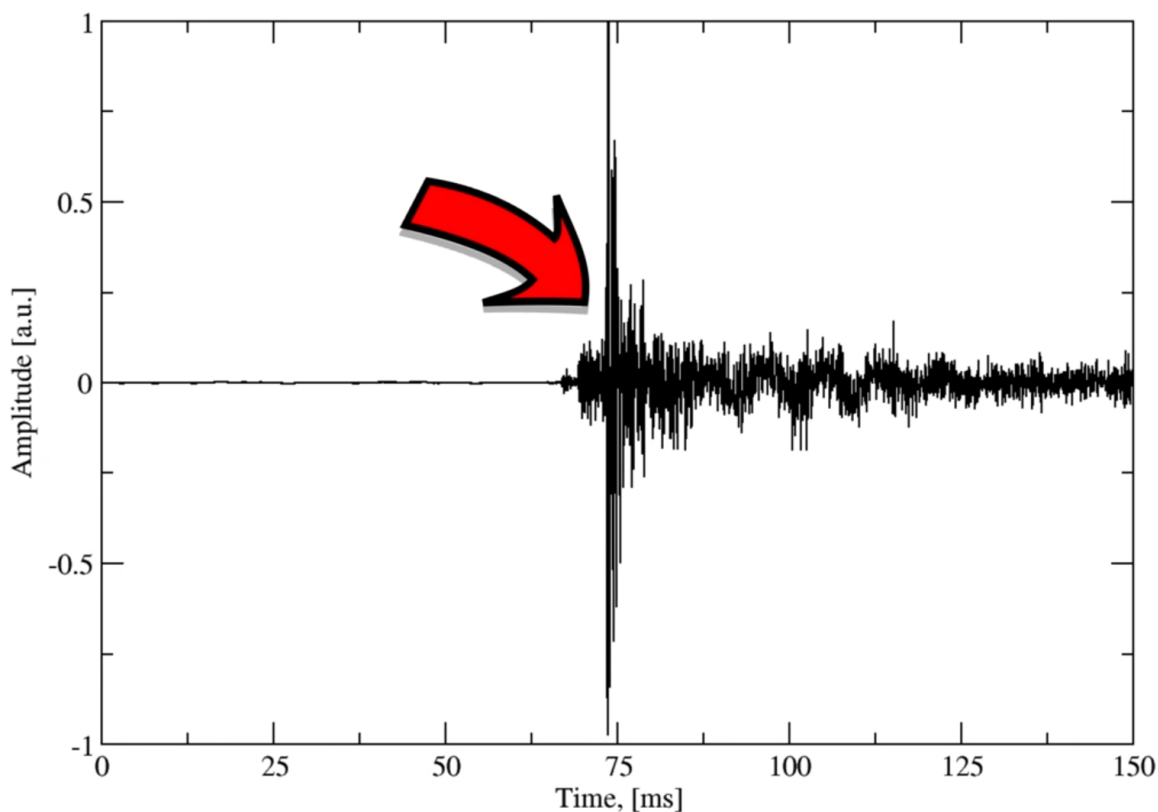
Para esta prueba utilizaremos una herramienta llamada kbd-audio y se encuentra disponible en GitHub en el siguiente enlace: <https://github.com/ggerganov/kbd-audio>

## Funcionamiento kbd-audio

Esta herramienta emplea un algoritmo que toma de entrada un set de entrenamiento, este set de entrenamiento consiste en una grabación de audio en el que se pulsan las teclas. La idea es que con esos datos de audio el algoritmo aprenda cuál es el sonido de las diferentes teclas, para posteriormente lograr reconocerlas únicamente mediante el sonido capturado. La herramienta es capaz de predecir las pulsaciones en tiempo real, esto es debido a que se compone de las siguientes fases:

### Fase inicial de entrenamiento

Aquí se realiza la recolección de datos mediante las pulsaciones introducidas. Para ello utiliza un tiempo de 150 ms en el que se presiona la tecla, por lo que se verá tal que así:



Aquí podemos ver que se graban los 75 ms primeros, luego se produce el pico de presión marcado en el momento de presionar la tecla y posteriormente se registran otros 75 ms del sonido asociado.

En esta fase contra más formas de onda de cada tecla se registren mejor será el resultado, ya que el sonido puede variar según la fuerza de presión o la inclinación.

### Fase de aprendizaje

Aquí se efectúa un modelo de predicción, esto se podría realizar mediante aprendizaje automático, inteligencia artificial o incluso redes neuronales. En este caso el enfoque que le da esta herramienta es mucho más simple, para cada tecla se realizan tres pasos:

- Alineamiento de picos de las formas de onda recopilada
  - Esto lo hace para evitar retrasos de tiempo antes de la detección de la pulsación de las teclas
- Alineación más fina de las formas de onda basada en la métrica de similitud
  - Este enfoque está diseñado para intentar lograr una mayor precisión
- Se realiza una media
  - Un promedio ponderado simple de las formas de onda

Una vez finalizados estos pasos dará como resultado una única forma de onda para cada tecla, esto se comparará con los datos capturados en tiempo real para lograr predecir la tecla presionada más probable.

### Fase de detección de pulsaciones

En esta fase lo que hace KeyTap es detectar la pulsación de tecla que se realice en el teclado.

### Fase de predicción

En esta última fase la herramienta compara la forma de onda capturada con las que tiene ya registradas, a partir de esto intenta predecir la tecla que ha sido presionada.

## Instalación

A continuación se exponen los pasos que se deben realizar para la correcta instalación de esta herramienta:

1. Clonamos el repositorio de GitHub y accederemos a la carpeta.

```
Terminal
jose@jose-S15449 [~/.../SGI/Trabajo] ± master U:1 ? :4 X [10:45:08]
> git clone https://github.com/ggerganov/kbd-audio
Clonando en 'kbd-audio'...
remote: Enumerating objects: 992, done.
remote: Counting objects: 100% (146/146), done.
remote: Compressing objects: 100% (100/100), done.
remote: Total 992 (delta 79), reused 88 (delta 44), pack-reused 846
Recibiendo objetos: 100% (992/992), 12.65 MiB | 10.69 MiB/s, listo.
Resolviendo deltas: 100% (631/631), listo.
jose@jose-S15449 [~/.../SGI/Trabajo] ± master U:1 ? :5 X [10:45:18]
> cd kbd-audio/
/home/jose/UCO/4-Universidad/SGI/Trabajo/kbd-audio
jose@jose-S15449 [~/.../Trabajo/kbd-audio] ± master ✓ [10:45:44]
>
```

2. A continuación dentro de la carpeta debemos utilizar el siguiente comando *git submodule update --init*.

```
Terminal
build-vars.h.in      index-keytap3-gui-tmpl.html  record.cpp
cmake                key-average-gui.cpp         record-full.cpp
CMakeLists.txt       key-detector.cpp            scale.cpp
common.cpp           keytap2.cpp                 style.css
common-gui.cpp       keytap2-gui.cpp             subbreak2.cpp
common-gui.h         keytap2-gui-old.cpp         subbreak2.h
common.h             keytap3-app.cpp             subbreak3.cpp
compress-n-grams.cpp keytap3.cpp                 subbreak3.h
constants.h          keytap3-gui.cpp             subbreak.cpp
data                keytap3-multi.cpp           subbreak.h
dr_wav.h             keytap.cpp                  test-subbreak3.cpp
generate-clusters.cpp keytap-gui.cpp              view-full-gui.cpp
guess-qp2.cpp        LICENSE                     view-gui.cpp
guess-qp.cpp         non-exact-subbreak2.cpp
imconfig-vtx32.h     non-exact-subbreak.cpp
jose@jose-S15449 [~/.../Trabajo/kbd-audio] ± master ✓ [10:47:01]
> git submodule update --init
Submódulo 'imgui' (https://github.com/ocornut/imgui) registrado para ruta 'imgui'
Clonando en '/home/jose/UCO/4-Universidad/SGI/Trabajo/kbd-audio/imgui'...
Ruta de submódulo 'imgui': check out realizado a '3f26a07ee1813cecaa87253436149e28fc11dc4e'
jose@jose-S15449 [~/.../Trabajo/kbd-audio] ± master ✓ [10:47:30]
>
```

- Ahora crearemos una carpeta llamada build y accederemos a ella

```
Terminal
jose@jose-S15449 [~/.../Trabajo/kbd-audio] ± master ✓ [10:49:40]
> mkdir build
mkdir: se ha creado el directorio 'build'
jose@jose-S15449 [~/.../Trabajo/kbd-audio] ± master ✓ [10:49:45]
> cd bui
build/      build-em/
jose@jose-S15449 [~/.../Trabajo/kbd-audio] ± master ✓ [10:49:45]
> cd build
/home/jose/UCO/4-Universidad/SGI/Trabajo/kbd-audio/build
jose@jose-S15449 [~/.../kbd-audio/build] ± master ✓ [10:49:49]
>
```

- Una vez creada la carpeta debemos usar el comando *cmake ..* y posteriormente el comando *make*. Para este paso hay que tener instalado make.

```
Terminal
CMake Error at CMakeLists.txt:90 (message):
  Aborting

-- Configuring incomplete, errors occurred!
See also "/home/jose/UCO/4-Universidad/SGI/Trabajo/kbd-audio/build/CMakeFiles/CMakeOutput.log".
jose@jose-S15449 [~/.../kbd-audio/build] ± master U:1 X [12:18:57]
1 > cmake ..
CMake Deprecation Warning at CMakeLists.txt:1 (cmake_minimum_required):
  Compatibility with CMake < 2.8.12 will be removed from a future version of
  CMake.

  Update the VERSION argument <min> value or use a ...<max> suffix to tell
  CMake that the project does not need compatibility with older versions.

CMake Warning (dev) at /usr/share/cmake-3.22/Modules/FindOpenGL.cmake:315 (message):
  Policy CMP0072 is not set: FindOpenGL prefers GLVND by default when
  available. Run "cmake --help-policy CMP0072" for policy details. Use the
  cmake_policy command to set the policy and suppress this warning.

  FindOpenGL found both a legacy GL library:

    OPENGGL_gL_LIBRARY: /usr/lib/x86_64-linux-gnu/libGL.so

  and GLVND libraries for OpenGL and GLX:

    OPENGGL_opengl_LIBRARY: /usr/lib/x86_64-linux-gnu/libOpenGL.so
    OPENGGL_gL_LIBRARY: /usr/lib/x86_64-linux-gnu/libGLX.so

  OpenGL_GL_PREFERENCE has not been set to "GLVND" or "LEGACY", so for
  compatibility with CMake 3.10 and below the legacy GL library will be used.
Call Stack (most recent call first):
  CMakeLists.txt:77 (find_package)
This warning is for project developers. Use -Wno-dev to suppress it.

-- Could NOT find FFTW (missing: FFTW_LIBRARIES FFTW_INCLUDE_DIRS)
CMake Warning at CMakeLists.txt:94 (message):
  FFTW library not available. Some targets will not be built

-- Configuring done
-- Generating done
-- Build files have been written to: /home/jose/UCO/4-Universidad/SGI/Trabajo/kbd-audio/build
jose@jose-S15449 [~/.../kbd-audio/build] ± master ✓ [12:19:15]
> make
[ 2%] Building CXX object CMakeFiles/Core.dir/common.cpp.o
[ 4%] Building CXX object CMakeFiles/Core.dir/audio-logger.cpp.o
[ 6%] Linking CXX static library libCore.a
[ 6%] Built target Core
[ 8%] Building CXX object CMakeFiles/Gui.dir/common-gui.cpp.o
[10%] Building CXX object CMakeFiles/Gui.dir/ingui/ingui.cpp.o
```

5. Ahora si utilizamos el comando `ls` podemos observar todo lo que se ha creado

```
Terminal
jose@jose-S15449 [~/.../kbd-audio/build] ± master ✓ [12:22:42]
> ls
CMakeCache.txt      compress-n-grams  keytap2-gui  keytap-gui  play      view-full-gui
CMakeFiles          key-detector     keytap3      libCore.a   play-full view-gui
cmake_install.cmake keytap           keytap3-app  libGui.a    record
compile_commands.json keytap2          keytap3-gui  Makefile    record-full
jose@jose-S15449 [~/.../kbd-audio/build] ± master ✓ [12:22:42]
>
```

## Ejecución KeyTap

Empezaremos ejecutando el binario `record` y pasaremos toda la información al archivo `output.kbd`, se realiza con el comando `./record output.kbd`. En este punto se debe estar totalmente en silencio porque empezará a recoger el sonido de las pulsaciones del teclado.

```
Terminal
jose@jose-S15449 [~/.../kbd-audio/build] ± master ✓ [14:04:13]
> ./record output.kbd
Usage: ./record output.kbd [-cN]
       -cN - select capture device N
       -CN - number N of capture channels N

Recording 11 frames per key press
Found 2 capture devices:
  - Capture device #0: 'Tiger Lake-LP Smart Sound Technology Audio Controller Headphones Stereo Microphone'
  - Capture device #1: 'Tiger Lake-LP Smart Sound Technology Audio Controller Digital Microphone'
Attempt to open capture device 0 : 'Tiger Lake-LP Smart Sound Technology Audio Controller Headphones Stereo Microphone' ...
Opened capture device succesfully!
  DeviceId: 2
  Frequency: 16000
  Format: 33056 (4 bytes)
  Channels: 2
  Samples: 512
  Audio Filter: 1
  Cutoff frequency: 100 Hz
Capturing audio ..
```



```
Terminal
-CN - number N of capture channels N

Recording 11 frames per key press
Found 2 capture devices:
- Capture device #0: 'Tiger Lake-LP Smart Sound Technology Audio Controller Headphones Stereo Microphone'
- Capture device #1: 'Tiger Lake-LP Smart Sound Technology Audio Controller Digital Microphone'
Attempt to open capture device 0 : 'Tiger Lake-LP Smart Sound Technology Audio Controller Headphones Stereo Microphone' ...
Opened capture device successfully!
DeviceId: 2
Frequency: 16000
Format: 33056 (4 bytes)
Channels: 2
Samples: 512
Audio Filter: 1
Cutoff frequency: 100 Hz
Capturing audio ..
Last recorded key - 104 'h'. Total times recorded so far - 1. Total data saved: 0.0214844 MB
Last recorded key - 97 'a'. Total times recorded so far - 1. Total data saved: 0.0429688 MB
Last recorded key - 104 'h'. Total times recorded so far - 2. Total data saved: 0.0644531 MB
Last recorded key - 100 'd'. Total times recorded so far - 1. Total data saved: 0.0859375 MB
Last recorded key - 105 'i'. Total times recorded so far - 1. Total data saved: 0.107422 MB
Last recorded key - 101 'e'. Total times recorded so far - 1. Total data saved: 0.128906 MB
Last recorded key - 117 'u'. Total times recorded so far - 1. Total data saved: 0.150391 MB
Last recorded key - 114 'r'. Total times recorded so far - 1. Total data saved: 0.171875 MB
Last recorded key - 116 't'. Total times recorded so far - 1. Total data saved: 0.193359 MB
Last recorded key - 121 'y'. Total times recorded so far - 1. Total data saved: 0.214844 MB
Last recorded key - 112 'p'. Total times recorded so far - 1. Total data saved: 0.236328 MB
Last recorded key - 102 'f'. Total times recorded so far - 1. Total data saved: 0.257812 MB
Last recorded key - 109 'm'. Total times recorded so far - 1. Total data saved: 0.279297 MB
Last recorded key - 104 'h'. Total times recorded so far - 3. Total data saved: 0.300781 MB
Last recorded key - 97 'a'. Total times recorded so far - 2. Total data saved: 0.322266 MB
Last recorded key - 113 'q'. Total times recorded so far - 1. Total data saved: 0.34375 MB
Last recorded key - 108 'l'. Total times recorded so far - 1. Total data saved: 0.365234 MB
Last recorded key - 110 'n'. Total times recorded so far - 1. Total data saved: 0.386719 MB
Last recorded key - 118 'v'. Total times recorded so far - 1. Total data saved: 0.408203 MB
Last recorded key - 55 '7'. Total times recorded so far - 1. Total data saved: 0.429688 MB
Last recorded key - 100 'd'. Total times recorded so far - 2. Total data saved: 0.451172 MB
Last recorded key - 110 'n'. Total times recorded so far - 2. Total data saved: 0.472656 MB
Last recorded key - 106 'j'. Total times recorded so far - 1. Total data saved: 0.494141 MB
Last recorded key - 115 's'. Total times recorded so far - 1. Total data saved: 0.515625 MB
Last recorded key - 100 'd'. Total times recorded so far - 3. Total data saved: 0.537109 MB
Last recorded key - 100 'd'. Total times recorded so far - 4. Total data saved: 0.558594 MB
Last recorded key - 115 's'. Total times recorded so far - 2. Total data saved: 0.580078 MB
Last recorded key - 115 's'. Total times recorded so far - 3. Total data saved: 0.601562 MB
Last recorded key - 100 'd'. Total times recorded so far - 5. Total data saved: 0.623047 MB
Last recorded key - 106 'j'. Total times recorded so far - 2. Total data saved: 0.644531 MB
Last recorded key - 107 'k'. Total times recorded so far - 1. Total data saved: 0.666016 MB
Last recorded key - 112 'p'. Total times recorded so far - 2. Total data saved: 0.6875 MB
Last recorded key - 119 'w'. Total times recorded so far - 1. Total data saved: 0.708984 MB
Last recorded key - 119 'w'. Total times recorded so far - 2. Total data saved: 0.730469 MB
Last recorded key - 114 'r'. Total times recorded so far - 2. Total data saved: 0.751953 MB
Last recorded key - 107 'k'. Total times recorded so far - 2. Total data saved: 0.773438 MB
```

Una vez completada esta fase de entrenamiento, ejecutaremos el keytap gui pasándole el archivo output.kdb que es el que contiene los datos. El comando sería `./keytap-gui output.kdb`, nos abrirá una interfaz que será la que nos señalará las teclas que se pulsan.



View-full

▼ Main

record.kbd
Audio file Load ☒

Last predicted key: ? (-1.000000)

0.350

Threshold CC

Last detected key stroke: 4.332 seconds ago  
 Average background level: 0.0001967476888

10.000

Threshold background

Tasks in queue: 0

Display confidence

`	1	2	3	4	5	6	7	8	9	0	-	=	[<-]
q	w	e	r	t	y	u	i	o	p	[	]	\	
a	s	d	f	g	h	j	k	l	;	'	[enter]		
z	x	c	v	b	n	m	,	.	/				

Last 24 predicted keys: Clear

► Last prediction

► Average key sound

▼ Training statistics

Key:	a	Average CC: 0.413422	Waveforms: 9 / 9
Key:	b	Average CC: 0.581119	Waveforms: 3 / 3
Key:	c	Average CC: 0.414798	Waveforms: 6 / 6
Key:	d	Average CC: 0.384549	Waveforms: 7 / 7
Key:	g	Average CC: 0.580982	Waveforms: 3 / 3
Key:	j	Average CC: 0.504577	Waveforms: 4 / 4
Key:	k	Average CC: 0.582125	Waveforms: 3 / 3
Key:	l	Average CC: 0.580234	Waveforms: 3 / 3
Key:	m	Average CC: 0.451489	Waveforms: 5 / 5
Key:	n	Average CC: 0.451847	Waveforms: 5 / 5
Key:	o	Average CC: 0.504304	Waveforms: 4 / 4
Key:	p	Average CC: 0.451921	Waveforms: 5 / 5
Key:	s	Average CC: 0.384648	Waveforms: 7 / 7
Key:	x	Average CC: 0.505050	Waveforms: 4 / 4
Key:	z	Average CC: 0.504841	Waveforms: 4 / 4
Key:	?	Average CC: 0.579267	Waveforms: 3 / 3
Key:	?	Average CC: 0.579267	Waveforms: 3 / 3

View-full

▼ Main

record.kbd

Audio file  ☒

Last predicted key: ? (-1.000000)

0.350

Threshold CC

Last detected key stroke: 26.839 seconds ago

Average background level: 0.0002074520503

10.000

Threshold background

Tasks in queue: 0

Display confidence

`	1	2	3	4	5	6	7	8	9	0	-	=	[<-]
	q	w	e	r	t	y	u	i	o	p	[	]	\
	a	s	d	f	g	h	j	k	l	;	'	[enter]	
	z	x	c	v	b	n	m	,	.	/			
					[space]								

Last 24 predicted keys:

► Last prediction

▼ Average key sound

a




b

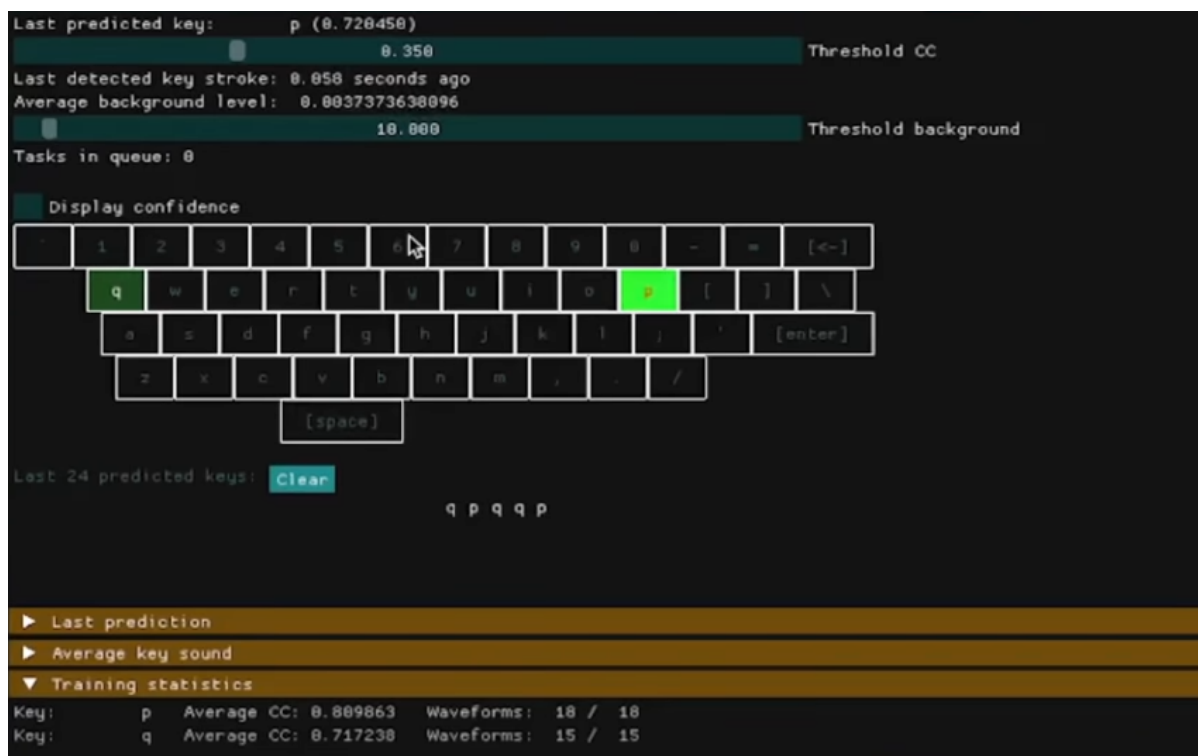


c



d

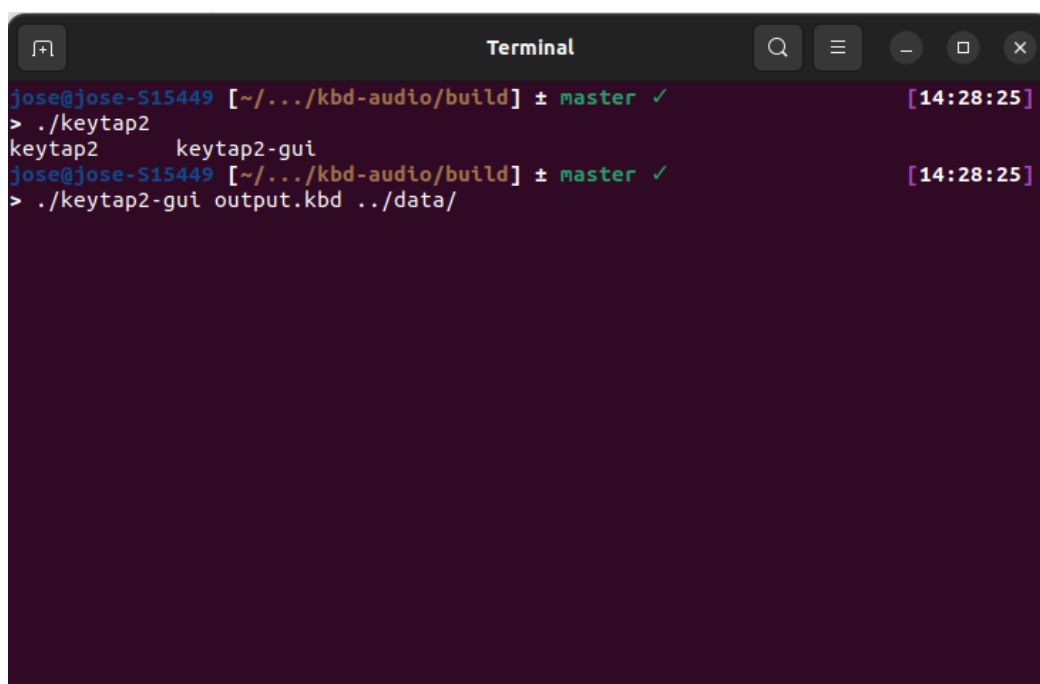


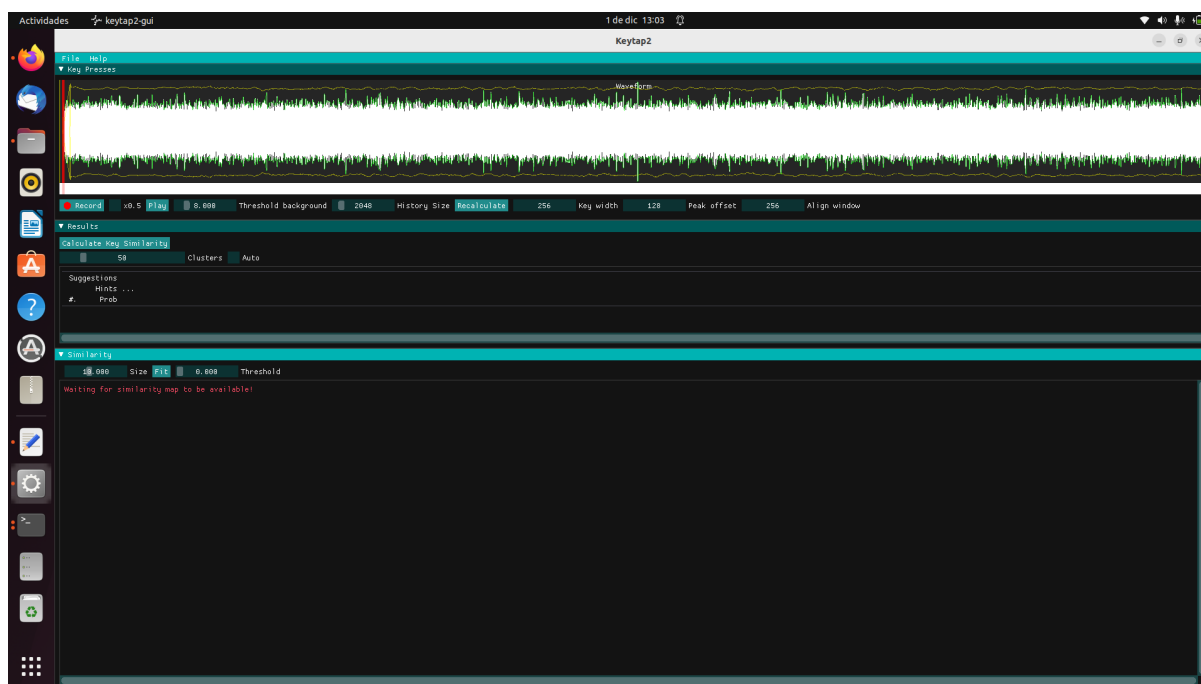


## Ejecución KeyTap2

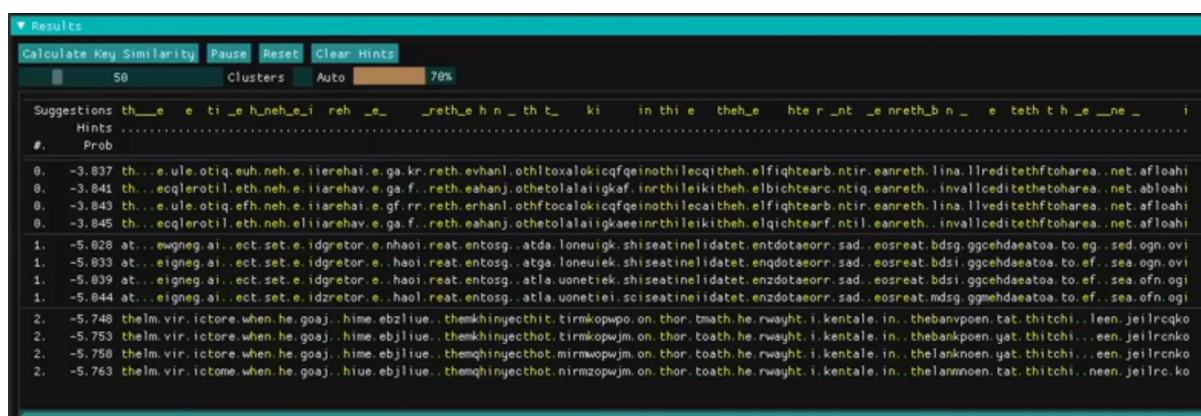
Esta herramienta incorporada en KeyTap no necesita recopilar información de entrenamiento, sino que mediante la información estadística de las frecuencias de las letras y los enagramas del inglés, te permite reconocer las pulsaciones de letras realizadas.

Para ejecutarlo debemos utilizar el siguiente comando `./keytap2-gui output.kbd ../data/`.





Al darle al play comenzará a escuchar lo que estamos escribiendo, si introducimos un texto en inglés comenzará a realizar operaciones para lograr descifrar lo que ha sido presionado.



## Vídeos

<https://www.youtube.com/watch?v=2Ojzl9m7W10>

<https://www.youtube.com/watch?v=5aphvxpSt3o>