

# Forum:[Hack Forums](#)

Thread: [How to disable Ctrl+Alt+Delete using Python](#)

**f4keg0d** - November 4, 2019, 12:58 am

Hello, Strangers so im trying to make Program tp disable Ctrl+Alt+Delete on widows.

Note: Pyhook is not working for some reason

**TheWarOwl** - November 9, 2019, 1:42 pm

I gave it a shot, and while my tests did block the keyboard input it doesn't prevent the ctrl+alt+delete popup

With a little bit of research I discovered that this is handled at kernel level due to security reasons. Basically, You need a low level keyboard hook.

Are you trying to prevent the user from opening task manager? If so you may just simply end the task manager process if it ever starts, ctrl+alt+delete isn't the only way to open task manager anyways.

**f4keg0d** - November 9, 2019, 2:07 pm

(Nov 9, 2019 02:42 PM)TheWarOwl Wrote: [\[CITE\]](#) I gave it a shot, and while my tests did block the keyboard input it doesn't prevent the ctrl+alt+delete popup

With a little bit of research, I discovered that this is handled at kernel level due to security reasons. Basically, You need a low level keyboard hook.

Are you trying to prevent the user from opening the task manager? If so you may just simply end the task manager process if it ever starts, ctrl+alt+delete isn't the only way to open task manager anyways.well I found something but its need rest (reboot)

how you can disable it :

Code reg add os.system('"HKLM\SYSTEM\CurrentControlSet\Control\Keyboard Layout" /v "Scancode Map" /t REG\_BINARY /d 0000000000000000030000004de01de04be01d0000000000 /f')  
P.S: problems with this method are its gonna disable all of the hotkeys, and you need to logout & login

(if anyone knows how can I make this req stuff work without logout/in please share <3

P.S 2: can you share your method with me (the one that looks at everything.

**magicparty** - November 9, 2019, 6:57 pm

Maybe using a System Call.

[Executing user-mode instructions] -> [System call] -> [Execute kernel-mode instructions] -> [Return back to user-mode]

The user mode / kernel mode is actually signified by a physical bit that is set in main memory. System calls in the C programming language on the other hand runs instructions specifically built for the operating system which is obviously in kernel mode.

Tl;dr - Use a System Call in the C programming language to see if you can get what you want in Kernel Mode

If you want to know more I can send you reading material or help you understand on a deeper level.  
Just PM me.

**f4keg0d** - November 11, 2019, 3:32 am

(Nov 9, 2019 07:57 PM)magicparty Wrote: [CITE](#) Maybe using a System Call.

[Executing user-mode instructions] -> [System call] -> [Execute kernel-mode instructions] -> [Return back to user-mode]

The user mode / kernel mode is actually signified by a physical bit that is set in main memory. System calls in the C programming language on the other hand runs instructions specifically built for the operating system which is obviously in kernel mode.

Tl;dr - Use a System Call in the C programming language to see if you can get what you want in Kernel Mode

If you want to know more I can send you reading material or help you understand on a deeper level.  
Just PM me.

sorry but can't massage you im still under lvl for that XD