Disabling the PrtSc key by blocking input

devblogs.microsoft.com/oldnewthing/20131216-00

December 16, 2013

Raymond Chen

A customer asked how to disable the `PrtSc` key in the On-Screen Keyboard.

There is no way to disable the `PrtSc` key in the On-Screen Keyboard. The On-Screen Keyboard shows a keyboard, and you can click any virtual key you like. There is no policy to remove specific keys from the On-Screen Keyboard.

But this was a case of a customer breaking down a problem and asking a question about a specific part of the problem instead of presenting the entire problem so that a solution to the overall problem could be developed.

The customer's real goal was to disable the `PrtSc` key in general. They had figured out how to disable it on their physical keyboards by using the PS/2 scan code mapper, but they found that users could circumvent the feature by using the On-Screen Keyboard, so they asked if there was something they could do about the On-Screen Keyboard.

They didn't mention this when they asked the original question, so I replied by saying, "The mechanism for blocking the screen capture functionality in the window manager should work regardless of whether the request came from the physical keyboard or the virtual keyboard."

Naturally, the customer liaison decided to direct follow-up questions to me directly, even though I was replying from my phone while doing a quick email check while on vacation, and I had to remind him that my response to your message should not be interpreted as meaning that I had taken responsibility for driving your issue to resolution. I had to steer the thread back to the distribution list so that somebody else could pick up the ball while I was out of the office. (Either that, or the customer ends up waiting until the next time I feel like checking email while on vacation, which could be quite a while.)

The solution to the original problem is not to try to identify every possible source of a `PrtSc` keypress and block it there, because even a simple script can use the `SendKeys` method to inject the `PrtSc` key.

You block the message in the window manager by installing a low-level keyboard hook that rejects the `VK_SNAPSHOT` key.

Today’s Little Program is a printscreen blocker. Remember, Little Programs have little to no error checking, because that’s the way they roll.

Take our scratch program and add the following lines of code:

```
HHOOK g_hhk;
LRESULT CALLBACK KHook(int nCode, WPARAM wParam, LPARAM lParam)
{
 if (nCode == HC_ACTION) {
 if (wParam == WM_KEYDOWN || wParam == WM_SYSKEYDOWN) {
  PKBDLLHOOKSTRUCT phs = (PKBDLLHOOKSTRUCT)lParam;
  if (phs->vkCode == VK_SNAPSHOT) {
   MessageBeep(0); // annoying beep
   return 1; // block the key
  }
 }
 }
 return CallNextHookEx(g_hhk, nCode, wParam, lParam);
}
...
    ShowWindow(hwnd, nShowCmd);
    g_hhk = SetWindowsHookEx(WH_KEYBOARD_LL, KHook,
                             NULL, 0);
    while (GetMessage(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    UnhookWindowsHookEx(g_hhk);
```

This program installs a low-level keyboard hook which listens for presses of the `VK_SNAPSHOT` key, and if it sees one, it beeps and then returns `1` to block further processing.

Note that we are using a global solution to a local problem. If you want to block `PrtSc` because you have sensitive information in your application that you don’t want captured, then tag your window to be excluded from screen capturing. That way, the user can still capture other windows. It also tells other screen capturing programs to exclude your window from the capture.

Raymond Chen

Follow