ARMIN
EBRAHIMPOUR
AZIZI

استفاده از برد attiny85 به عنوان کلید اجرای برنامه های ویندوزی :

در ابتدا با استفاده از یک اسکریپت پایتون در سیستم یک هاست ساختیم تا منتظر اتصال برد مذکور باشد بصورتی که با استفاده از unique instance ID ای که سیستم عامل ویندوز به hardware های متصل به سیستم اختصاص میدهد استفاده کرده :

```
import win32com.client


def is_device_attached():
    # connenct to wmi
    wmi = win32com.client.GetObject("winmgmts:")

    # query for PnP devices
    devices = wmi.ExecQuery("SELECT * FROM Win32_PnPEntity")

    # checking if the specific device is attached
    for device in devices :
        if (device.DeviceID == r"USB\VID_16D0&PID_0753\5&54725E2&0&4"):
            return True

    return False
```

در این قطعه کد با استفاده از کتابخانه win32com.client در لیست device های متصل شده به سیستم به دنبال unique id برد Attiny می باشیم .

اما مشکل این سیستم به تنهایی این است که unique instance ID هایی که ویندوز برای Attity85 های مختلفی که استفاده میکند با یکدیگر یکسانند این به این معنی است که کلید برنامه در دسترس عموم بوده و صرفا با تهیه یک برد Attiny85 و متصل کردن آن به سیستم برنامه به صورت خودکار unlock میشود.

با توجه به مشکل پیش آمده از اسکریپت دیگری استفاده کردیم که علاوه بر سیستم بالا منتظر اطلاعات دیگری از سمت برد باشد تا به دنبال کلید منحصر به فردی باشیم.

```
IMPORT KEYBOARD
FROM TIME IMPORT TIME, SLEEP

# CONFIGURATION
TRIGGER_COMBO = {'CTRL', 'ALT', 'SHIFT', 'A'}
DEBOUNCE_TIME = 0.1
LAST_TRIGGER_TIME = 0

DEF CHECK_KEY():
    GLOBAL LAST_TRIGGER_TIME

    PRINT("PRESS ESC TO EXIT")

    WHILE TRUE:
        PRESSED = {NAME FOR NAME IN TRIGGER_COMBO IF
KEYBOARD.IS_PRESSED(NAME)}

        IF PRESSED == TRIGGER_COMBO:
            CURRENT_TIME = TIME()

            IF CURRENT_TIME - LAST_TRIGGER_TIME > DEBOUNCE_TIME:
                PRINT("KEY DETECTED!")
                LAST_TRIGGER_TIME = CURRENT_TIME




                WHILE ALL(KEYBOARD.IS_PRESSED(K) FOR K IN
TRIGGER_COMBO):
                    SLEEP(0.001)  # SHORTER SLEEP FOR RAPID RELEASE CHECK
                RETURN
        IF KEYBOARD.IS_PRESSED('ESC'):
            BREAK

        SLEEP(0.01)
```

در این اسکریپت ما از کیبورد به دنبال یک ترکیبی خاص از دکمه های کیبورد میباشیم که از طریق برد Attiny به سیستم وارد میشود. این ترکیب میتواند با توجه به پروگرم برنامه تغییر کند .

در این مرحله باید برد Attiny پروگرم شود که با توجه به نیاز هاست تنها کافیست کلید های خاصی از کیبورد به سیستم وارد شود که هاست در انتظار آنهاست .

```
#INCLUDE "DIGIKEYBOARD.H"
VOID SETUP() {
  DIGIKEYBOARD.SENDKEYSTROKE(0);
}

VOID LOOP() {
  IF (DIGITALREAD(0) == LOW) {
    // SEND CTRL+ALT+SHIFT+[KEY] COMBINATION
    DIGIKEYBOARD.SENDKEYSTROKE(KEY_A, MOD_CONTROL_LEFT |
MOD_ALT_LEFT | MOD_SHIFT_LEFT);
    DELAY(5);
  }
  DIGIKEYBOARD.DELAY(100);
}
```

در این مرحله باید اسکریپت های پایتونی را که نوشته ایم کد گذاری کنیم تا پروسه ی reverse engineering را دشوار تر کنیم با استفاده از کتابخانه های cython,distutils فایل های اسکریپتی نوشته شده را به کد گذاری کرده و به فایل هایی با فرمت pyd. تبدیل میکنیم .

از این فایل های کد گذاری شده ی جدید میتوانیم در برنامه مورد نیاز import کنیم.

در نهایت با import از فایل های مذکور در برنامه خود میشود از functionality های نوشته شده بهره برد.

```
    TRY:
        IF (KEYFINDER.IS_DEVICE_ATTACHED()):
            TRIGER.CHECK_KEY()
    EXCEPT KEYBOARDINTERRUPT:
        PRINT("\NEXITING...")
```