一．HID

1.1 HID 基本概念与 USBHID 攻击

HID 表示人机接口设备，是一种典型的由人类用来向计算机输入数据或从计算机接收输出的设备。HID 标准被现代操作系统广泛认可，允许键盘和鼠标等设备无需专门的驱动程序即可使用。这个标准是 USB 规范的一部分，在简化连接外围设备到计算机的过程中起到了关键作用。

HID 设备和 USBHID 攻击之间的联系在于，攻击者可以利用 HID 设备的信任和通用性来执行恶意活动。通过模拟合法的 HID 设备，如 USB 键盘，攻击者能够注入恶意代码或命令到目标计算机，这种攻击通常被称为 USBHID 攻击。由于操作系统普遍信任并自动接受 HID 设备，这种攻击很难被及时发现和阻止，从而对系统安全构成了威胁。

1.2 常见的 USB 攻击

微型控制器改编类 USB 攻击：这类攻击使 USB 设备在外观上看起来像是一个普通设备，但实际上执行的是其他操作（如键盘注入键盘击键）。

恶意修改 USB 固件的攻击：通过修改 USB 设备的固件来执行恶意动作，例如下载恶意软件、数据窃取等。

不重编程的 USB 设备攻击：这类攻击不通过重新编程固件，而是利用操作系统与 USB 协议/标准交互时的漏洞。

电气攻击：某些 USB 设备被设计用来对计算机造成物理损害，例如通过发送电流冲击。

1.3 可软件防护的 USB 攻击类型

可软件防护的 USB 攻击一般具有一下特点:

（1） 基于恶意软件的攻击。

（2） 键盘模拟攻击。

（3） 自动运行脚本攻击。

（4） 网络仿冒攻击。

在本次项目中，我们采用检测输入频率的方法来检测 USB 设备自动运行的脚本，以此来防护 USB 攻击。

1.4 USB 攻击的一些防护手段

(1)从 USB 协议入手，利用状态机模型，分析 USB 协议枚举过程和 HID 协议运行过程中存在的设备权限管理和设备可靠性校验两个漏洞。

(2)设计时域聚合的数据预处理算法。根据人类用户自然击键习惯将数据流切分为特征稳定的短击键序列，原有掩藏在噪声中的数据通过数 据融合后汇聚成为有用的特征知识，为 SVM 分类器提供决策依据.基于这种算法的安全策略实现实时认证实时授权的防护机制。

(3)设计多元击键特征提取算法，从数据组表征的自然击键事件中得到丰富稳定的击键特征。这些特征不单是击键间隔还包括错误率、速度、节奏波动等多元特征。还可以应用 SVM 分类器等自动学习分类算法

二．Duckhunt 检测 hid 攻击

2.1 相关技术框架

2.1.1 Tkinter

Tkinter 是 Python 的标准 GUI（图形用户界面）工具包，它提供了一组用于创建和管理 GUI 应用程序的类和方法。Tkinter 是 Tcl/Tk 工具包的 Python 接口，Tcl/Tk 是一种流行的跨平台 GUI 工具包。Tkinter 提供了创建窗口、对话框、按钮、标签、文本框等各种 GUI 元素的功能。它还支持事件驱动编程，允许开发人员根据用户的交互响应来执行相应的操作。

Tkinter 的一些主要特点：

1. 简单易用：Tkinter 提供了直观的 API，使得创建 GUI 应用程序变得简单易用。它的语法清晰简洁，适合初学者和快速原型开发。

2. 跨平台性：Tkinter 是 Python 的标准库，因此它在几乎所有的主流操作系统上都可以使用，包括 Windows、macOS 和 Linux。

3. 强大的小部件库：Tkinter 提供了丰富的 GUI 小部件库，包括窗口、按钮、标签、文本框、滚动条、列表框等。这些小部件可以用于构建各种类型的用户界面。

2.1.2 pyhook

pyhook（全名为 PyHook - Python wrapper for global input hooks）是一个第三方库，它提供了在 Windows 操作系统上捕获和处理全局输入事件的功能。它基于 PyWin32（Python 对 Windows API 的封装）和 pyhk（Python 的

全局热键模块）。

pyhook 库的一些主要特性和功能：

钩子功能：pyhook 允许注册全局钩子来捕获并处理各种输入事件，例如按键、鼠标点击、鼠标移动等。通过注册钩子，可以监视和拦截系统中发生的输入事件，并根据需要采取相应的操作。

键盘事件：pyhook 可以捕获键盘事件，包括按键按下和释放事件。可以注册按键钩子来监视特定的按键操作，例如捕获特定的快捷键、记录用户的键盘输入等。

鼠标事件：pyhook 可以捕获鼠标事件，包括鼠标按钮点击、鼠标移动和滚动等。可以注册鼠标钩子来监视和响应特定的鼠标操作，例如实现自定义的鼠标手势、跟踪鼠标位置等。

2.2 检测模式

Paranoid: 当检测到攻击时，锁定后续按键输入，直到正确密码输入，(在 .conf 文件中设置密码)，攻击会被记录。

Normal: 当检测到攻击时，键盘输入会被临时禁止，(在攻击看似结束之后，键盘输入将会重新允许)，攻击会被记录。

Sneaky: 当检测到攻击时，会掉几个字符（足够让攻击者出错），攻击会被记录。

LogOnly: 当检测到攻击时，仅记录攻击，不在任何方式阻止它。

2.3 HID 攻击

攻击脚本

```
#include "DigiKeyboard.h"


void setup() {
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
//按下 Win+R 建
  DigiKeyboard.delay(200); //等待 200 毫秒
  DigiKeyboard.println("https://www.baidu.com"); //
输入网址
  DigiKeyboard.sendKeyStroke(KEY_ENTER); //回车
  DigiKeyboard.sendKeyStroke(KEY_ENTER); //两次回车以
防是中文输入法
}


void loop() {
}
```

利用 Arduino 开源软件将其写入 Digistump 开发板，重新将其插入电脑效果如下:

成功打开百度网页

2.4 duckhunt 脚本检测攻击

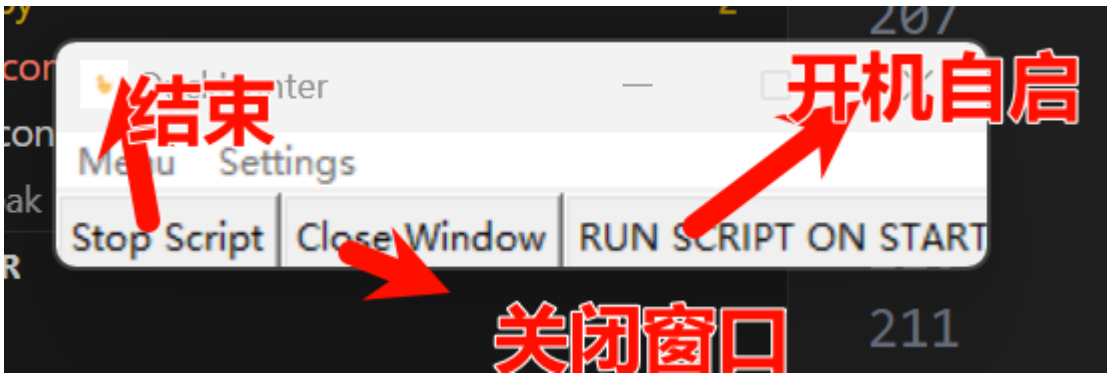
检测部分关键代码逻辑:

```
# 类型 Speed = NewKeyTime - OldKeyTime
    history[i] = event.Time - prevTime
    print(event.Time, "-", prevTime, "=",
history[i])
    prevTime = event.Time
    speed = sum(history) / float(len(history))
    i = i + 1
```

```
    print("\rAverage Speed:", speed)


    # 如果速度低于阈值则认为是入侵
    if (speed < threshold):
        return caught(event)
    else:
        intrusion = False
    # 继续执行
    return True
```

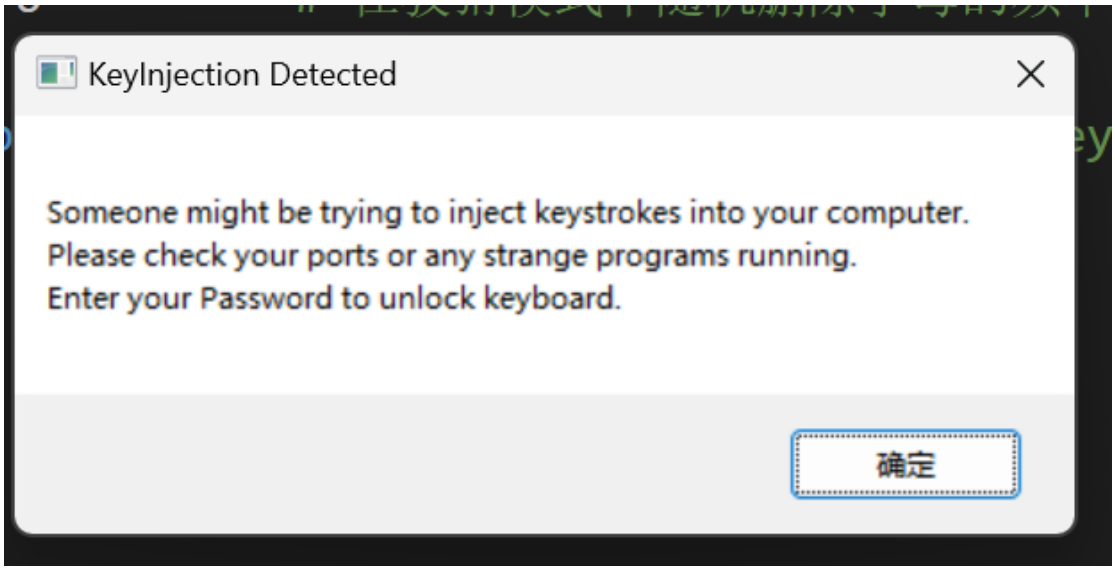
点击 start 后:

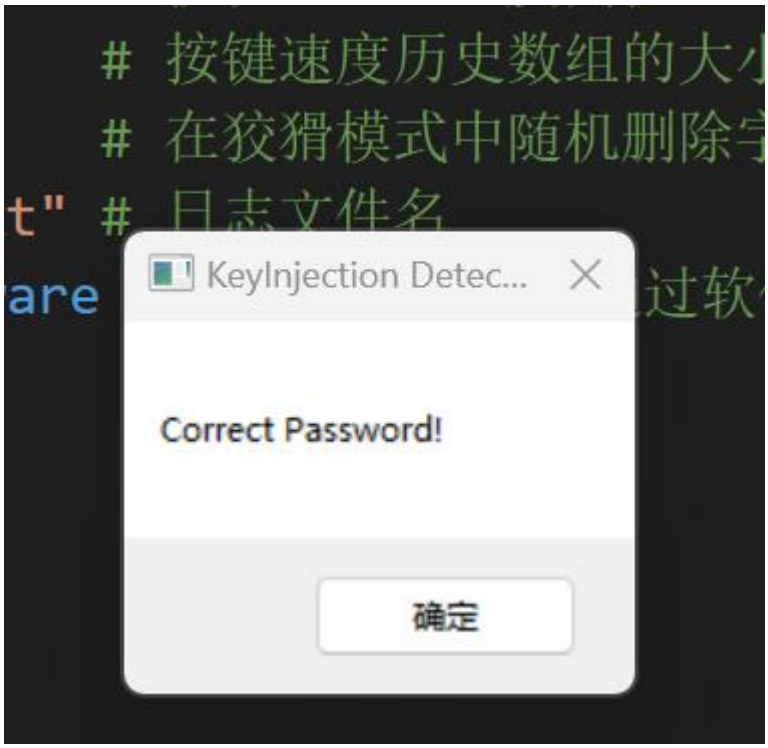


点击关闭窗口则命令行可见键入的事件

控制效果：Paranoid 模式：

插入设备后检测出 hid 攻击，进行提示（需要输入密码后才可以重新键入）：





三 模拟生成攻击框架

3.1 Flashsploit

Flashsploit 是一种利用 ATtiny85 HID 设备（如 Digispark USB 开发板）进行攻击的漏洞利用框架。Flashsploit 根据用户输入生成 Arduino IDE 兼容的（.ino）脚本，如果脚本需要，它会在 Metasploit-Framework 中启动监听器。

3.2 P4wnp1

P4wnP1 是一个高度可定制的 USB 攻击平台，基于低成本的 Raspberry Pi Zero 或 Raspberry Pi Zero W（用于 HID 后门功能）。同时支持 windows 和 linux 平台

在我们的项目中将使用上述攻击框架进行来生成攻击代码