# Rubber Ducky Payload System

*Project Synopsis Submitted*

*to*

**MANIPAL ACADEMY OF HIGHER EDUCATION**

*For Partial Fulfillment of the Requirement for the*

*Award of the Degree*

*Of*

**Bachelor of Technology**

*in*

**Information Security Lab**

By


| **Ashwin Mittal** | **Yogesh Rane** | **Vansh Yadav** |
|---|---|---|
| **220953128** | **220953290** | **220953306** |

Department of I&CT Department of I&CT

Manipal Institute of Technology, Manipal Institute of Technology,

Manipal, Karnataka, India Manipal, Karnataka, India


MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

*A Constituent Unit of MAHE, Manipal*

**October 2024**

## Title:

Rubber Ducky Payload System

## Abstract:

This project demonstrates a keystroke injection attack simulation using a USB-based Rubber Ducky payload system. The system uses an Arduino Pro Micro to emulate a keyboard that injects platform-specific malicious commands into a target machine, depending on the mode addressed by number of external button presses. It simulates real-world attack scenarios, such as copying sensitive files, encrypting them with a secret passphrase, and leaving a ransom note. The purpose of this project is to illustrate the dangers of USB-based attacks and raise awareness about securing endpoints against such threats. This payload system can be used for penetration testing and cybersecurity education to help reinforce good security practices.

## Introduction:

USB-based keystroke injection allows attackers to compromise systems by simply plugging in a device that acts as a keyboard. Such attacks have been popularised through devices like the USB Rubber Ducky, which can execute pre-programmed scripts by simulating fast, automated keystrokes.

This project uses Arduino Pro Micro. When connected to a target system, the microcontroller emulates a keyboard and delivers platform-specific malicious commands based on the operating system of the target device. A single button press triggers the Linux payload, while a double button press triggers the Windows payload. Both payloads copy all .docx and .pdf files from a specified folder, encrypt them, and leave a ransom note demanding payment for the decryption key.

This project aims to demonstrate the ease with which such attacks can be executed and to stress the importance of proper endpoint protection measures, such as disabling USB ports or employing robust access control mechanisms.

## System Requirements:

**Hardware:**

- *Arduino Pro Micro:*
    - Serves as the main microcontroller for the project.
    - Has the capability to emulate a keyboard, making it ideal for simulating keystroke injection attacks.
    - The small form factor allows for discreet use.

- One Button:
    - A push button is used to trigger the payload. The number of presses determines whether the system is targeting a Linux or Windows machine.

- A single button press initiates the Linux-specific payload, and a double button press triggers the Windows-specific payload.

- Pull-up Resistor:
  - A pull-up resistor ensures that the input pin connected to the button remains in a stable state (HIGH) when not pressed.
  - This prevents accidental triggering of the payload when the button is not actively being pressed.

- Micro USB Cable:
  - The Micro USB cable connects the Arduino Pro Micro to the target system.
  - It powers the microcontroller and establishes the connection for simulating keystrokes on the target device.

**Software:**

- Arduino IDE:
  - Used for writing the script that controls the behaviour of the Arduino Pro Micro.
  - The IDE also allows for uploading the script to the microcontroller.

- Target Systems:
  - Linux OR Windows:
    Payload opens a terminal / Powershell , copies all .docx and .pdf files from the Documents directory, encrypts them, and leaves a ransom note.

## Algorithm:

1. Initialize System:
   - Setup Arduino Pro Micro to emulate a keyboard.
   - Configure button input and pull-up resistor for the button state.
   - Wait for user interaction (button press).

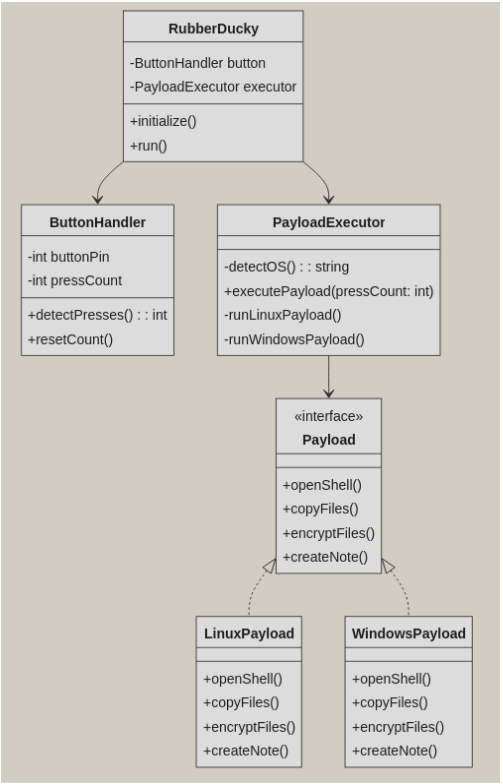
2. Detect Button Press:
   - If one button press is detected, execute the Linux payload.
   - If two button presses are detected, execute the Windows payload.

3. Linux Payload Execution (Single Button Press):
   - Open terminal (simulate pressing Ctrl + Alt + T).
   - Copy all .docx and .pdf files from ~/Documents/ to /tmp/ .
   - Encrypt the copied files using gpg with a symmetric passphrase.
   - Delete the original copied files from /tmp/.
   - Create a ransom note.

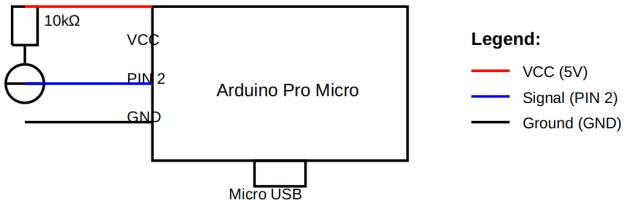
4. Windows Payload Execution (Double Button Press):
    - Open PowerShell (simulate pressing Win + R, type powershell, and press Enter).
    - Copy all .docx and .pdf files from Documents to a temporary directory.
    - Compress and encrypt the copied files using PowerShell.
    - Remove the original files.
    - Create a ransom note in the Documents folder.

5. End of Execution:
    - Disconnect the Arduino keyboard functionality.

## Class Model Diagram:




## Circuit Diagram:




## References:


- https://www.arduino.cc/reference/en/language/functions/usb/keyboard/
- https://docs.hak5.org/hak5-usb-rubber-ducky
- https://en.wikipedia.org/wiki/List_of_GNU_Core_Utilities_commands
- https://www.redhat.com/sysadmin/encryption-decryption-gpg