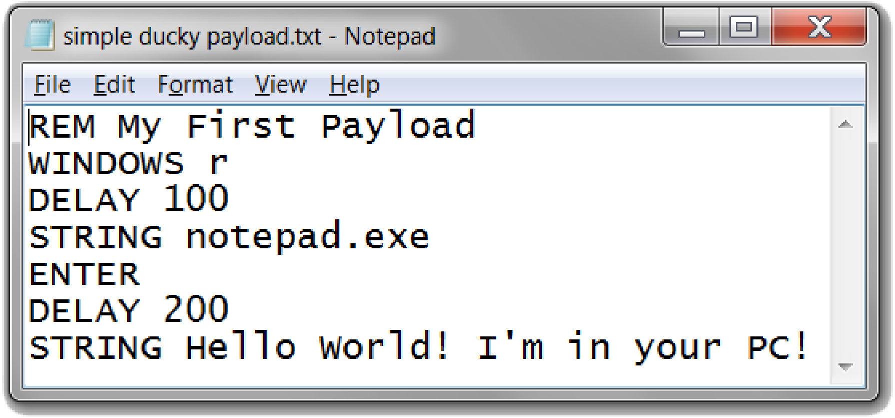
Remote Keystroke Injection and Key Logging

Christian Schwinne

What is USB Keystroke Injection?

- USB device with a MCU to emulate a keyboard
- Can send any desired keystrokes
- Simple standard macro language *Ducky script*


simple ducky payload.txt - Notepad
File Edit Format View Help

```
REM My First Payload
WINDOWS r
DELAY 100
STRING notepad.exe
ENTER
DELAY 200
STRING Hello World! I'm in your PC!
```

https://www.hackmod.de/WebRoot/Store12/Shops/78218349/MediaGallery/Hak5_USB_Rubber_Ducky_Pentest.png

Attack vectors

- The device needs to be connected to the target’s USB port. There are two possible methods:

  → The attacker has physical access to the target device

  → The user of the target device is socially engineered to plug in the KID

Examples of possible attacks

- Downloading and executing arbitrary code
- Sending WiFi network credentials to attacker via e-mail
- Deleting files
- Anything that can be done with keystrokes

Risk analysis

Attack potential

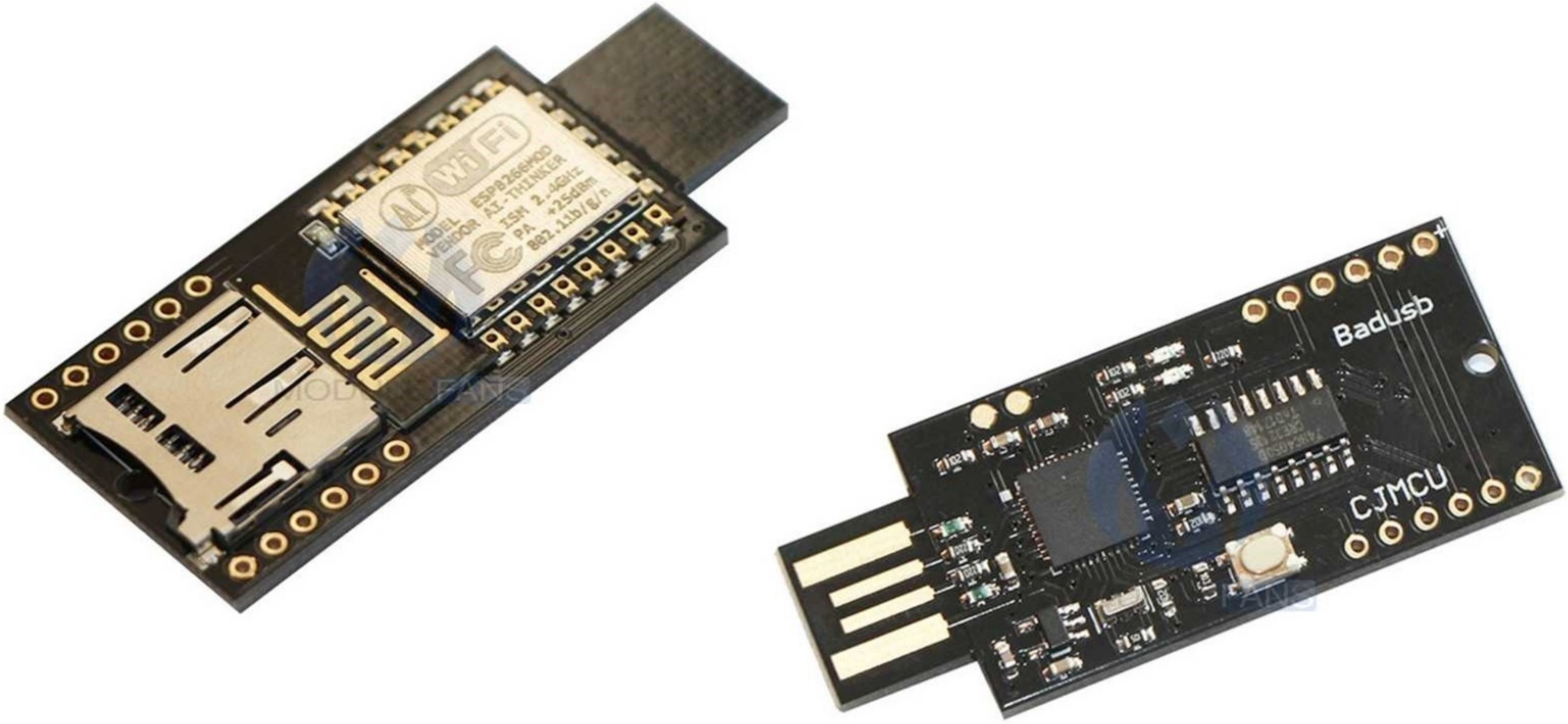
- Expertise: Flashing software to KID and simple keystroke scripting
- Time: Minimal
- Equipment: Specialized, but readily available device ($5)
- Damage potential: High, depends on criticality of data on the target system

Mitigations

- Locking the desktop when away
- Social engineering: Not plugging in unknown devices
- Disabling unused USB ports
- Software *(USB Keyboard Guard)*

→ Medium susceptability in business environments, high in private environments

The BadUSB board



https://github.com/SpacehuhnTech/WiFiDuck/issues/30#issuecomment-577283386

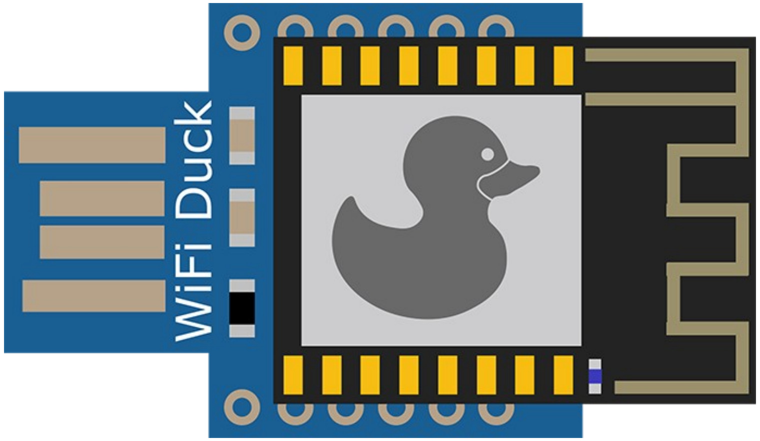
Specifications

- Small USB dongle
- ESP8266 microcontroller (WiFi AP+script storage)
- Atmega 32U4 (Keyboard emulation)
- Serial interface to connect both MCUs
- SD card slot (not required for this project)

Software

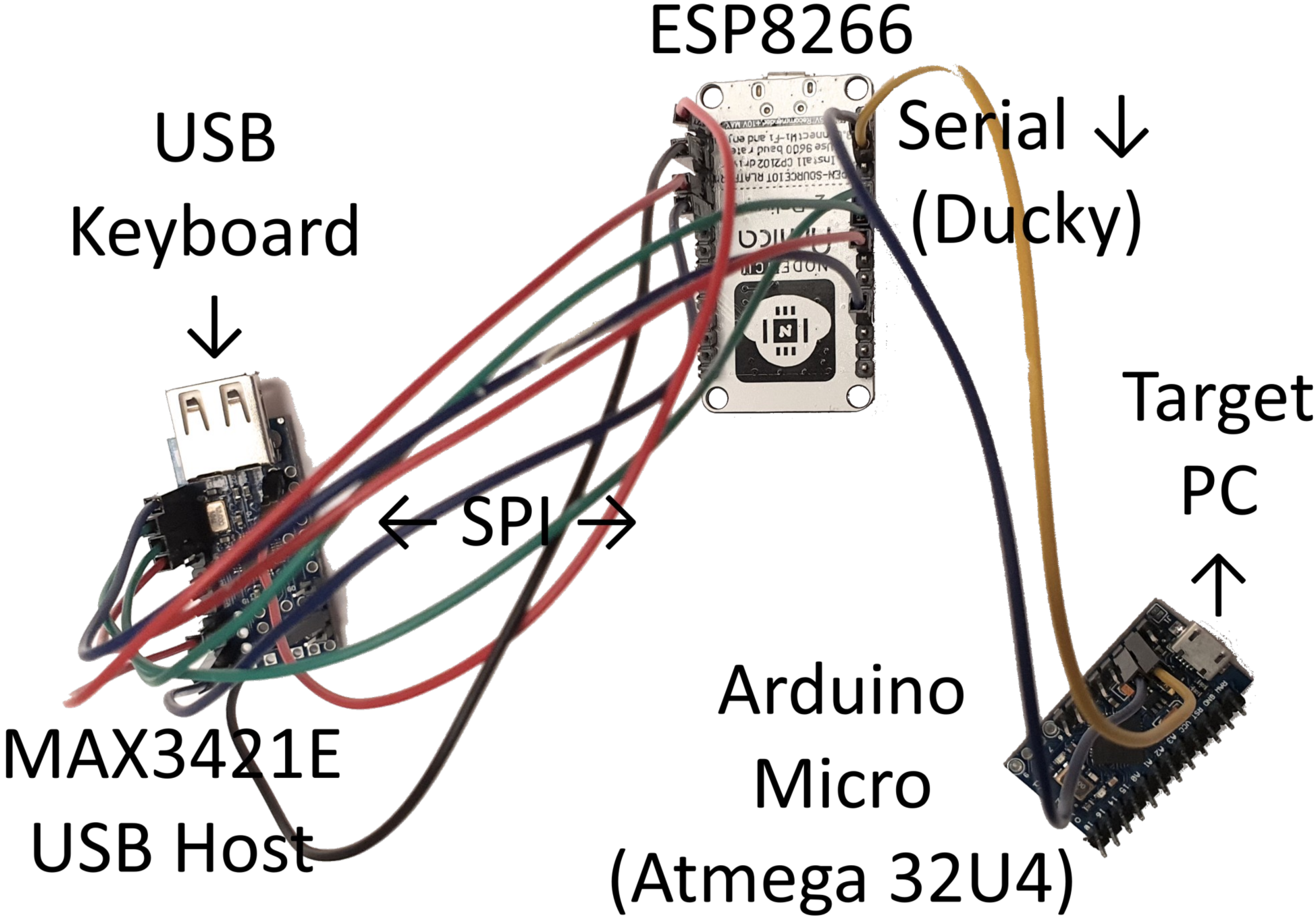
- Originally it was planned to develop an ESP8266 firmware to inject scripts from its internal filesystem
- However, an excellent open-source software for that exact purpose already exists:

  https://github.com/SpacehuhnTech/WiFiDuck



Project

- New goal: Based on WifiDuck, engineer a combined keylogger/keystroke injection device
- Therefore, the key injection setup was extended with the MAX3421E USB host shield to allow for connecting an external keyboard
- Proof-of-concept, but could be combined to a single USB-stick style PCB like BadUSB



ESP8266

USB
Keyboard
↓

Serial ↓
(Ducky)

Target
PC
↑

← SPI →

MAX3421E
USB Host

Arduino
Micro
(Atmega 32U4)

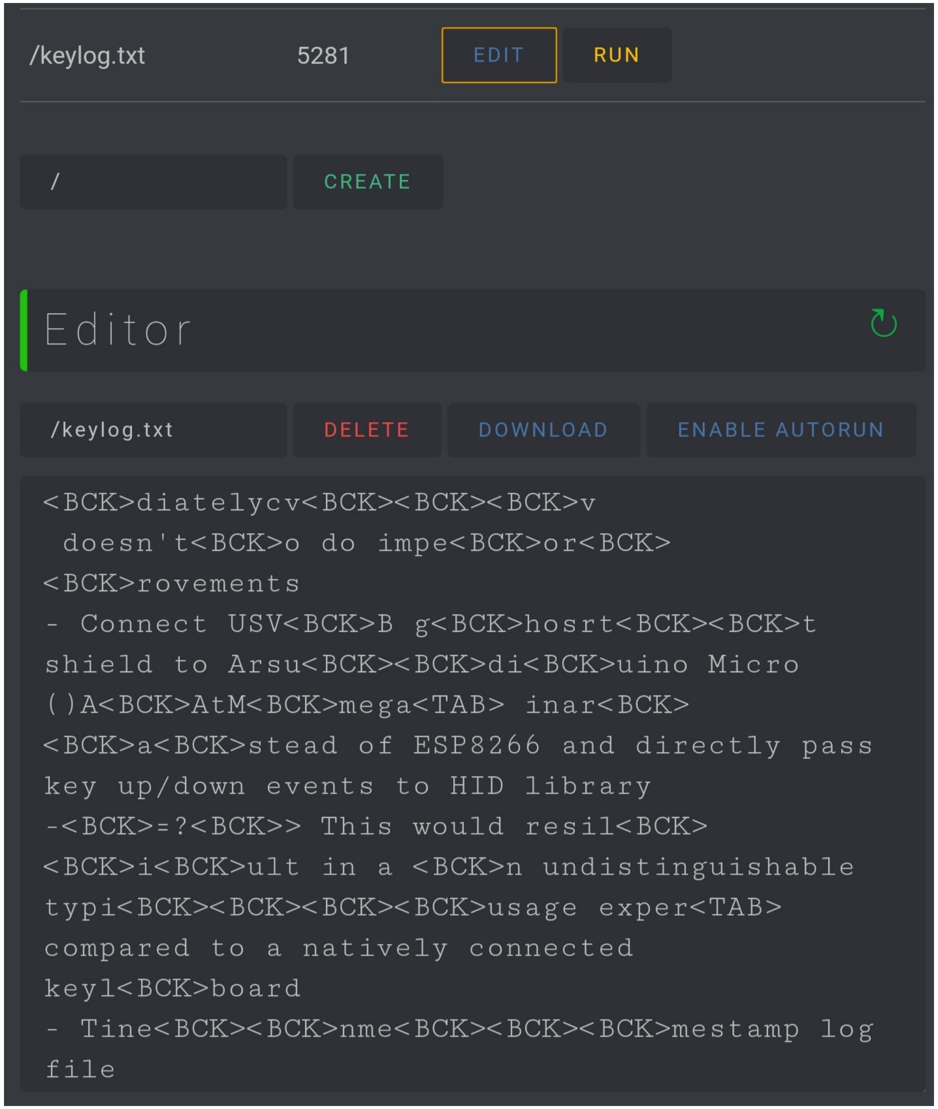
Project

- The keystrokes from the connected USB keyboard are received by the ESP8266 using the USB Host shield connected via SPI
- Keystrokes are then written to the ESP8266 filesystem in a `keylog.txt` file accessible via Wifi
- They are also passed to the 32U4 via a duckyscript `STRING` command to be printed to the host computer

Results

- Keylogging in principle works for all keys
- Passthrough to host PC works for ASCII characters
- Original feature set of WifiDuck (running Ducky scripts wirelessly) completely functional
- Wireless readout of key log
- No noticable latency compared to natively connected keyboard

Example key log



```
<BCK>diatelycv<BCK><BCK><BCK>v
 doesn't<BCK>o do impe<BCK>or<BCK>
<BCK>rovements
- Connect USV<BCK>B g<BCK>hosrt<BCK><BCK>t
shield to Arsu<BCK><BCK>di<BCK>uino Micro
()A<BCK>AtM<BCK>mega<TAB> inar<BCK>
<BCK>a<BCK>stead of ESP8266 and directly pass
key up/down events to HID library
-<BCK>=?<BCK>> This would resil<BCK>
<BCK>i<BCK>ult in a <BCK>n undistinguishable
typi<BCK><BCK><BCK><BCK>usage exper<TAB>
compared to a natively connected
keyl<BCK>board
- Tine<BCK><BCK>nme<BCK><BCK><BCK>mestamp log
file
```

Limitations and future improvements

- The implementation for this project was wired using 3 different off-the-shelf PCBs, which led to unreliability due to loose connections
- For use as a pentesting tool, all components should be on a compact and custom PCB
- This would ensure maximum reliability, usability and lower the risk of detection by the target user

Limitations and future improvements

- The project solution only passes through a limited set of keypresses to the host PC (ASCII+Backspace)
- This limitation drastically increases the risk of discovery by the target as important functions (Enter and arrow keys, key combinations, holding a key to print it repeatedly) do not work
- To overcome this limitation, raw key down and up events would need to be passed, making an extension of Duckyscript functionality necessary

Perspective and conclusion

- A single device plugged between a PC and an USB keyboard allowing for two high-impact attack vectors (keylogging + keystroke inj.), and do so wirelessly, is a very powerful pentesting tool!

Project code at https://github.com/Aircoookie/WiFiDuck

Thank you!

- Do you have any questions?