www.inl.gov

Idaho National Laboratory

Enhancing Embedded Device Vulnerability Analysis through Hardware Interfaces

Even 'secure' software can be insecure with hardware-based vulnerabilities

Owen Parkins – New Mexico Institute of Mining and Technology
Robert Erbes – Idaho National Laboratory, Cyber Security R&D

Why use Hardware Interfaces?

- A device with a hardware vulnerability will be insecure even with secure software because the hardware cannot be trusted to perform requested actions and nothing else
- Hardware interfaces allow access to devices without malicious software deceiving researchers

What are Hardware Interfaces?

- Hardware Interfaces are exposed terminals on a device that allow interaction with running software from an external device
- These interfaces typically implement a communication protocol
- Examples of hardware interfaces include:
  - JTAG
    - Often used for debugging and programming
  - Universal Asynchronous Receiver and Transmitter (UART)
    - Provides a communication channel
  - Inter-Integrated Circuit (I^2C)
    - Simple two wire communication protocol
  - Serial Peripheral Interface (SPI)
    - Synchronous communication protocol

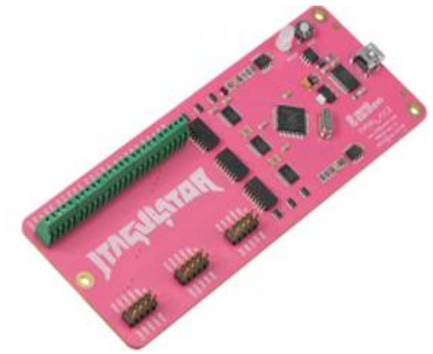

Figure 2. The JTAGulator used to help find JTAG pinouts
Photo Credit: parallax.com

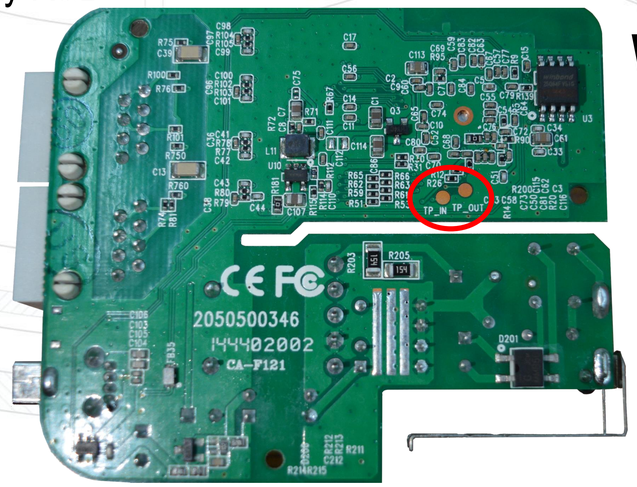

Figure 1. UART interface circled in **red** on a mobile router

What can you do with Hardware Interfaces?

- Hardware interfaces can facilitate:
  - Removing malicious firmware from a device
    - Install custom, trusted firmware
  - Firmware dumps
    - Allows direct access to passwords, binaries, and more
    - Possibly unbrick a device if something went wrong
  - Interrupting bootloaders
    - Interact with system before security measures are initialized
  - Direct communication with software on device
    - Useful for vulnerability analysis and exploit development
  - Manipulation of microcontrollers
    - Bypass normal authentication methods

Using Hardware Interfaces

- There are tools available to use hardware interfaces:
  - Shikra
    - The Shikra (see Figure 3) understands multiple protocols including JTAG, UART, and SPI
  - JTAGulator
    - The JTAGulator (see Figure 2) helps determine the JTAG pinout on an embedded device
    - This reduces research time because JTAG pinouts are not standardized among manufactures
  - GoodFET
    - An open-source adapter that specializes in communicating via the JTAG protocol
    - A newer version called the GreatFET is available

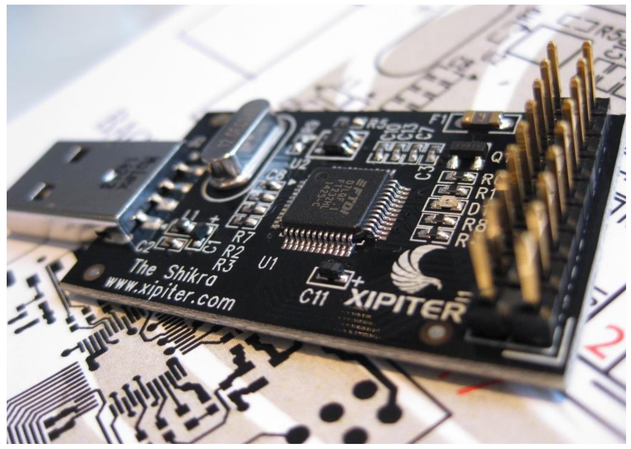

Figure 3. The Shikra tool used to communicate with a variety of protocols.
Photo Credit: int3.cc

References:
Grusin, Mike. "Serial Peripheral Interface (SPI)." *Serial Peripheral Interface (SPI)*, learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all.
Hord, Mike. "I2C." *I2C*, learn.sparkfun.com/tutorials/i2c/all.
Nanda, Umakanta, and Sushant Kumar Pattnaik. "Universal Asynchronous Receiver and Transmitter (UART)." *Universal Asynchronous Receiver and Transmitter (UART) - IEEE Conference Publication*, ieeexplore.ieee.org/document/7586376.
"What Is JTAG and How Can I Make Use of It? - XJTAG Tutorial." *XJTAG*, www.xjtag.com/about-jtag/what-is-jtag.

INL/EXP-19-55035