CHALLANGE WRITE-UP FOR

Debugging Sensei

Author: Andrei VLĂDESCU

1 ABOUT THE CHALLANGE

1.1 Challange name

Debugging Sensei

1.2 Description

I got bored at my workplace and decided rules are for the uncreative. So, I started hacking into the company's prototype microcontroller just to see what I could find. Anyways, they use some kind of trusted platform module from ***NXP***, that's managing the encryption keys. Luckily I have a logic analyzer on hand and I can debug it through the conveniently left-out pins. I don't really know the protocol, since it's an unmarked PCB, but it has ***two wires*** only for ***debug*** interface. The first signal seems like a clock, the other one is certainly data. Right now they are extracting data from it, so the chip must be **writing** to their interface.

1.3 Hint

- AN11553 Successful write operation
- WDATA is in LSB order

1.4 Flag format

CTF{message}

2 WRITE-UP

2.1 Proof of flag

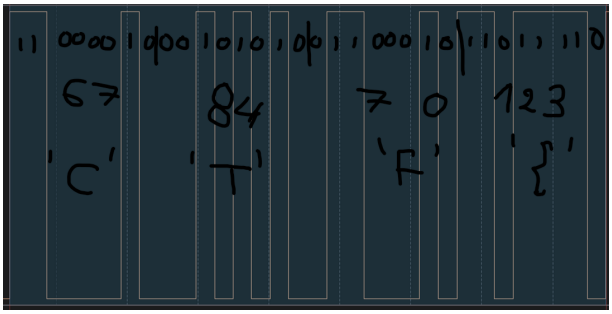


Рис. 1: WDATA0

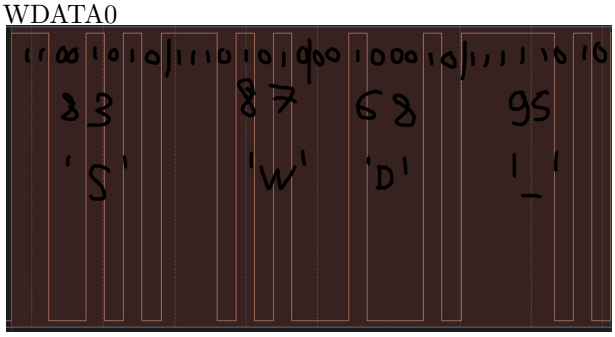


Рис. 2: WDATA1

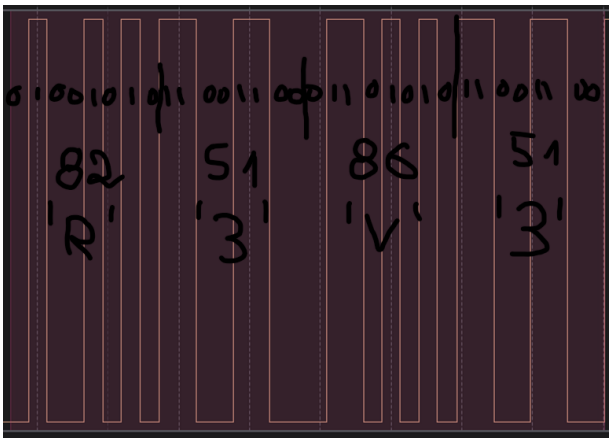


Рис. 3: WDATA2

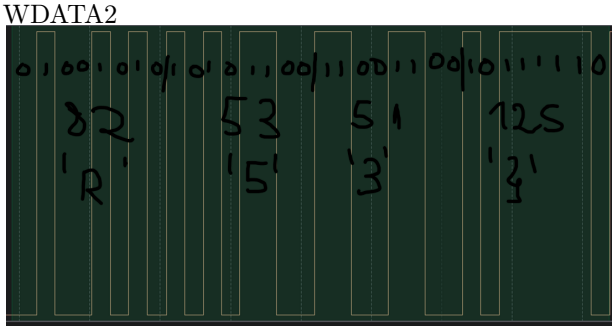


Рис. 4: WDATA3

2.2 Flag

CTF{SWD_R3V3R53}

2.3 Summary of the approach

- Find out the protocol, and it's standard datasheet.
- Open the .sal file in Logic Analyzer.
- Decode each packet, from the -33 bit to the -1 bit, those are the relevant bytes.

2.4 Proof of solving

The first hints come to mind when reading the description. It says in italic, bold, "NXP ... two wires ... debug". If you search that on the internet with google search, the first 3 links are:

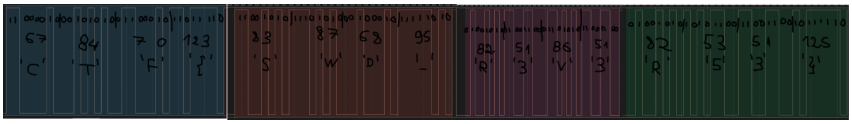
- Webpage: Serial Wire debug - NXP Community
- Webpage: MCU-Link Debug Probe
- PDF File: AN11553 - NXP Community

From the first one, the player will gather that the protocol in use is "SWD serial wire debug. It requires two pins, SCK (Serial ClocK) and SDIO (Serial Data Input-Output).
The second page is irrelevant, it doesn't contain any useful info.
The third page is the most interesting, since it contains the protocol specification for SWD, and it gives detailed info about the protocol packets, together with graphical examples.

Knowing that it's a SWD from NXP, and we have the standard protocol for it, we go and search for "write as the description says the MCU is writing to the SWD bus. The first example of writing is at page 15/61, called "7.1.2 Successful write operation (OK response)". It stipulates all the bit info, and the interesting data is in "WDATA which is a 32 bit section, written in LSB form. The last bit is parity, so we get the last 33 bits, minus the very last one, and we decode them in 4 bytes.

To open the capture file, we need "Logic Analyzer"from Saleae.

There are multiple approaches in getting the data, like with an automated script, but the simplest is to decode it visually.



We got the flag: **CTF{SWD_R3V3R53}**.