How does one checkm8?

Introduction

This is my analysis and writeup of the vulnerabilities exploited in the checkm8 BootROM exploit. I wrote this in order to help me gain a better understanding of the vulnerability so that I could design my own strategy for exploitation and write my own implementation of the exploit. The checkm8 exploit relies on a couple of vulnerabilities:

- The main use-after-free (not patched until A14)
- The memory leak (patched in A12)
  The memory leak is essential in order to exploit the use-after-free because it allows us to deterministically craft the heap in order to allow the exploit to work. The patching of said memory leak in the A12 and A13 BootROMs is what prevents checkm8 from being exploited on these later SoCs.

Before we start, there are some important resources that I used to help me understand the exploit:

- This technical analysis of checkm8 by a1exdandy
- This presentation about checkra1n's implementation by Luca Todesco
- This vulnerability writeup by littlelailo
- ipwndfu by axi0mX
- gaster by 0x7FF
- The leaked iBoot/BootROM source codes - not linked here for obvious reasons
- securerom.fun for their collection of BootROM dumps that I reverse engineered

Throughout this writeup, any code examples will be taken from the pseudocode generated from the BootROM dump reverse-engineering process for legal reasons, but the corresponding functions can also be easily found within the leaked iBoot/BootROM source codes.

DFU initialisation