

Wooden Ships And Iron Men

© 1987 Avalon Hill

Crack Study By Six

Tools used (Modern):

VICE
DirMaster
Visualize1541
G64Conv
Rapidlok Ripper V2.0 <https://csdb.dk/release/?id=82051>

Tools I would have used in 1987:

C64
Super Snapshot
Disector <https://csdb.dk/release/?id=74829>
The GCR Editor <https://csdb.dk/release/?id=67858>
Kracker Jax Rapidlok Scanner <https://csdb.dk/release/?id=177127>

Starting with a G64 rip of my original, a direct conversion to D64 produces a non-working copy. A look at the BAM shows a disk full of errors. Closer analysis reveals that this is a Rapidlok protected disk. Rapidlok is one of those protections that has been heavily documented and thus isn't as interesting as some of the lesser known ones. See the extremely thorough doc at https://rittwege.com/RL6Handbook_v130/INDEX.HTM for more details.

A Rapidlok ripping tool exists, which I tested on this disk with no success. Specifically the "boot", "bb1.o", and "bb2.o" files didn't rip properly at first, and then upon inspection it became obvious that none of the other files ripped properly either.

Existing cracks of this game crash in the reloading phase. This is because of a two byte error on them at \$8d8. All of the known cracks have the bytes ea 08 at 8d8 (in file "the.o") but the correct bytes are bd b3. One of the things I set out to establish is if this would have happened during ripping of the rapidlok files – it wouldn't. This must be an artifact of a transfer or copy error that made it into all of the recracks. This issue exists on the OUG, PAPPILLIONS and FUSION releases. An interesting side note here – the version of this game available on 64 Preservation Project isn't the same as my version and the version the ESI rip was produced from. Looking at both versions, they don't differ in any obviously meaningful way (forensic analysis, BAM, etc), but some of the library code is different. Being unsure about the stability of the 64PP version, I opted for mine when ripping files.

The second side of the disk does not appear to be protected in any way, nor is it forensically interesting.

Ripping the files

The Rapidlok loader is already extensively documented, so while I did work through this myself, there's no reason to. If you want to know how rapidlok itself works, no summary I could give here would contain more information than the aforementioned docs on rittwage.com. I

In any case, you can set a breakpoint at \$309 and \$ffbd. When the system stops at \$ffbd it is preparing to load a file, you can get the filename from x:y. When the system stops at \$0309, it is starting to load the file, you can get the starting address from \$AE/\$AF. Once the file is completely loaded, you can get the ending address from \$AE/\$AF. This is the technique that was used to rip all of the files on the disk.

The first file loaded is bb1.o from \$cd00-cff4

At this point it launches into bb1.o, starting at \$cdf2 which displays a loading screen and then starts loading files (starting with the title picture) at around \$ce86. Now we can put a breakpoint at \$ffd5 since it is using a hijacked kernal load vector. This will allow us to inspect \$AE/\$AF after each load as well as see the filename for the next load.

Files loaded are: (in order, and start/end addr pulled out by inspecting \$AE/\$AF)

picture : \$6000-\$8711

arrows : \$5c00-\$5d80

sprites : \$4000-\$4fff

mapb: \$2b00-\$3bff (ship list, british 1)

small: \$5400-\$5c00

two.o: \$0400-\$28db

one.o: \$9243-\$ccd3

Where this goes from here depends on the options you pick in the menu.

Map1:\$8000-\$9243 (constitution vs guerriere)

map2:\$8000-\$9243 (bonhomme richard vs. serapis)

map3:\$8000-\$9243 (the battle of lissa)

map4: \$8000-\$9243 (all land/all sea/all shoal)

mape: \$2b00-\$3bff(ship list, british 2)

mapa:\$2b00-\$3bff (ship list, american)

maps:\$2b00-\$3bff (ship list, spanish)

mapf: \$2b00-\$3bff (ship list, french)

mapW: \$8000-\$9243 (Islands)

mapX: \$8000-\$9243 (forks)

mapY:\$8000-\$9243 (harbor)

mapZ:\$8000-\$9243 (reefland)

gd: \$fff0-\$ffff

bb2.o: \$8f00-\$8fdf

the.o: \$0400-\$3972

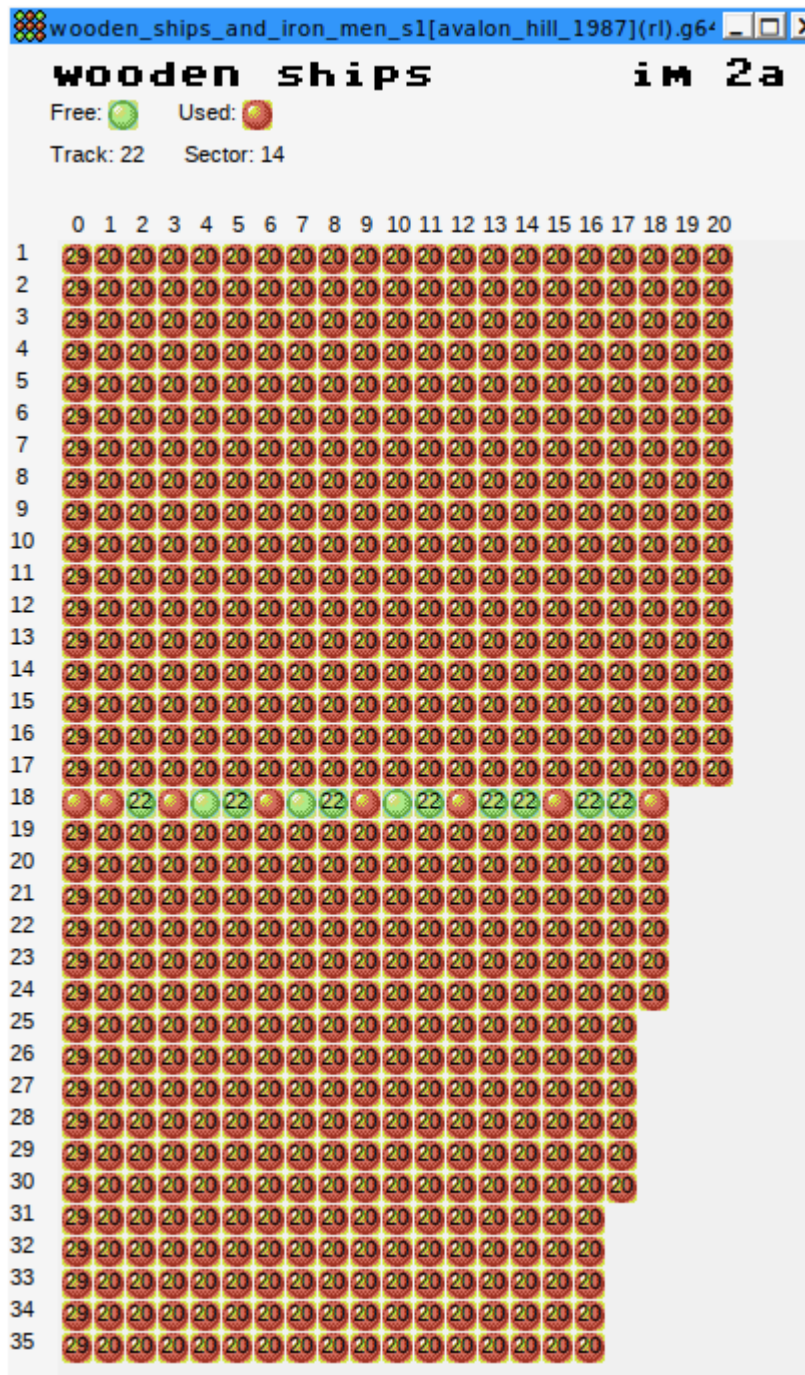
When you quit a game:

v.o: \$1000-\$1b3e

old: \$4000-\$4400

All of these files were ripped and put on a freshly formatted disk (image). The file "bb1.o" was linked with exomizer. Crack complete.

Forensic Data, BAM:



Forensic Data, GCR Scan, side 1:

