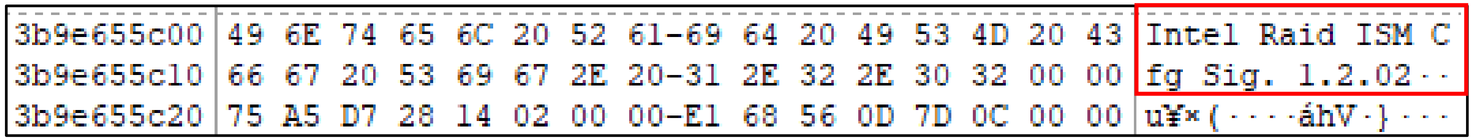
Intel RAID

Approach

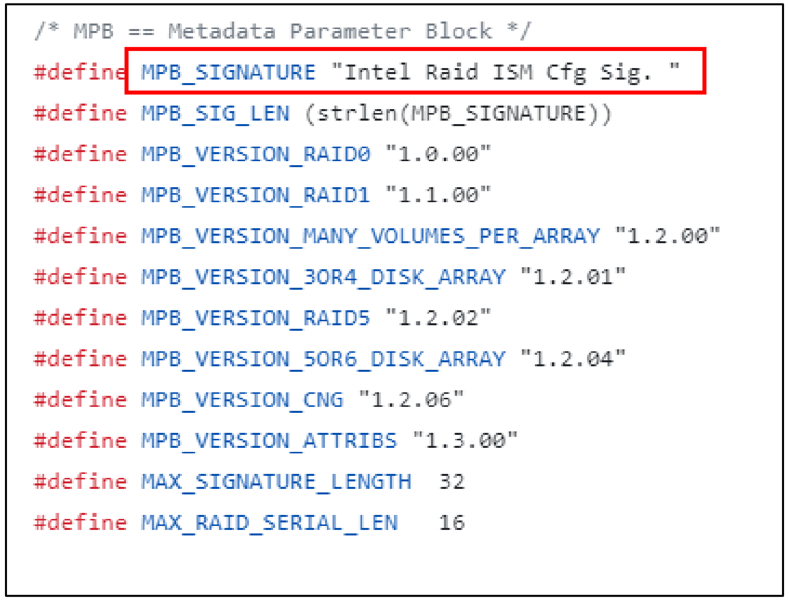
OSS (Open-source Software)

- **mdadm**
  - Linux utility for RAID control
  - Code for Intel RAID
    - https://github.com/neilbrown/mdadm/blob/master/super-intel.c
  - Can find same Signature in storage



```
3b9e655c00 | 49 6E 74 65 6C 20 52 61-69 64 20 49 53 4D 20 43 | Intel Raid ISM C
3b9e655c10 | 66 67 20 53 69 67 2E 20-31 2E 32 2E 30 32 00 00 | fg Sig. 1.2.02··
3b9e655c20 | 75 A5 D7 28 14 02 00 00-E1 68 56 0D 7D 0C 00 00 | u¥×(····áhV·}···
```

Data in .img file



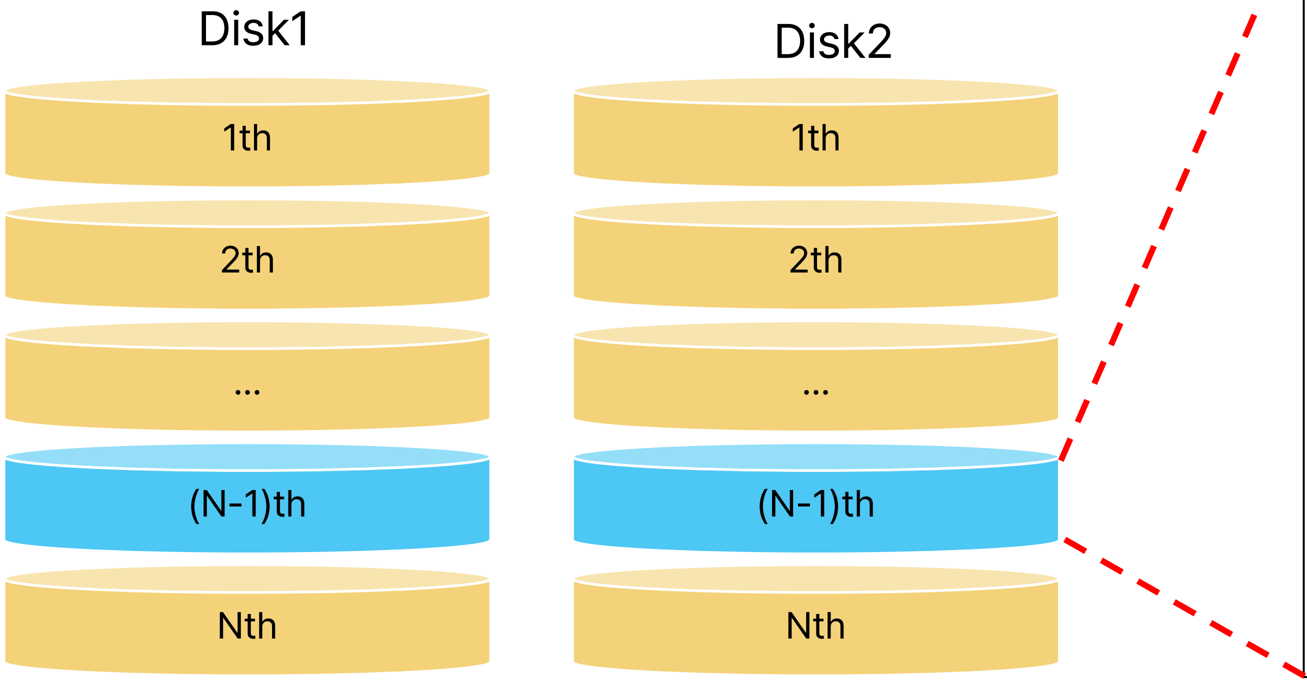
```c
/* MPB == Metadata Parameter Block */
#define MPB_SIGNATURE "Intel Raid ISM Cfg Sig. "
#define MPB_SIG_LEN (strlen(MPB_SIGNATURE))
#define MPB_VERSION_RAID0 "1.0.00"
#define MPB_VERSION_RAID1 "1.1.00"
#define MPB_VERSION_MANY_VOLUMES_PER_ARRAY "1.2.00"
#define MPB_VERSION_3OR4_DISK_ARRAY "1.2.01"
#define MPB_VERSION_RAID5 "1.2.02"
#define MPB_VERSION_5OR6_DISK_ARRAY "1.2.04"
#define MPB_VERSION_CNG "1.2.06"
#define MPB_VERSION_ATTRIBS "1.3.00"
#define MAX_SIGNATURE_LENGTH  32
#define MAX_RAID_SERIAL_LEN   16
```

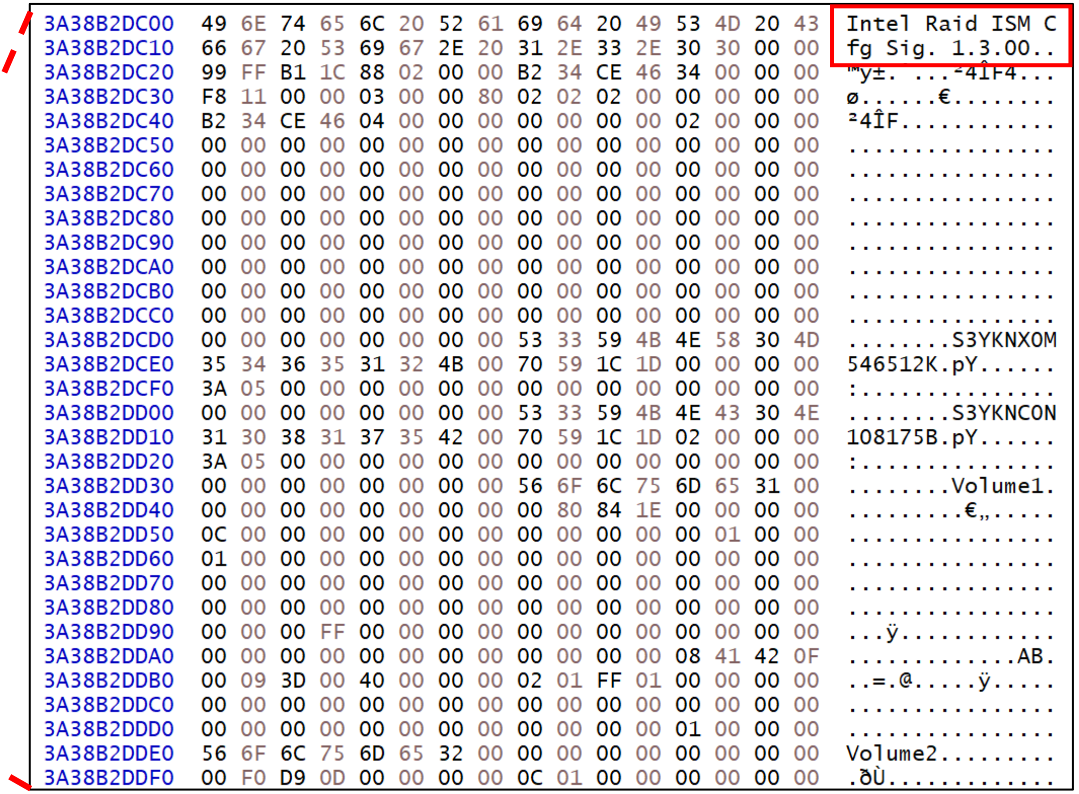
OSS code

Conclusion

Summary

- **Metadata exists in (N-1)th sector**



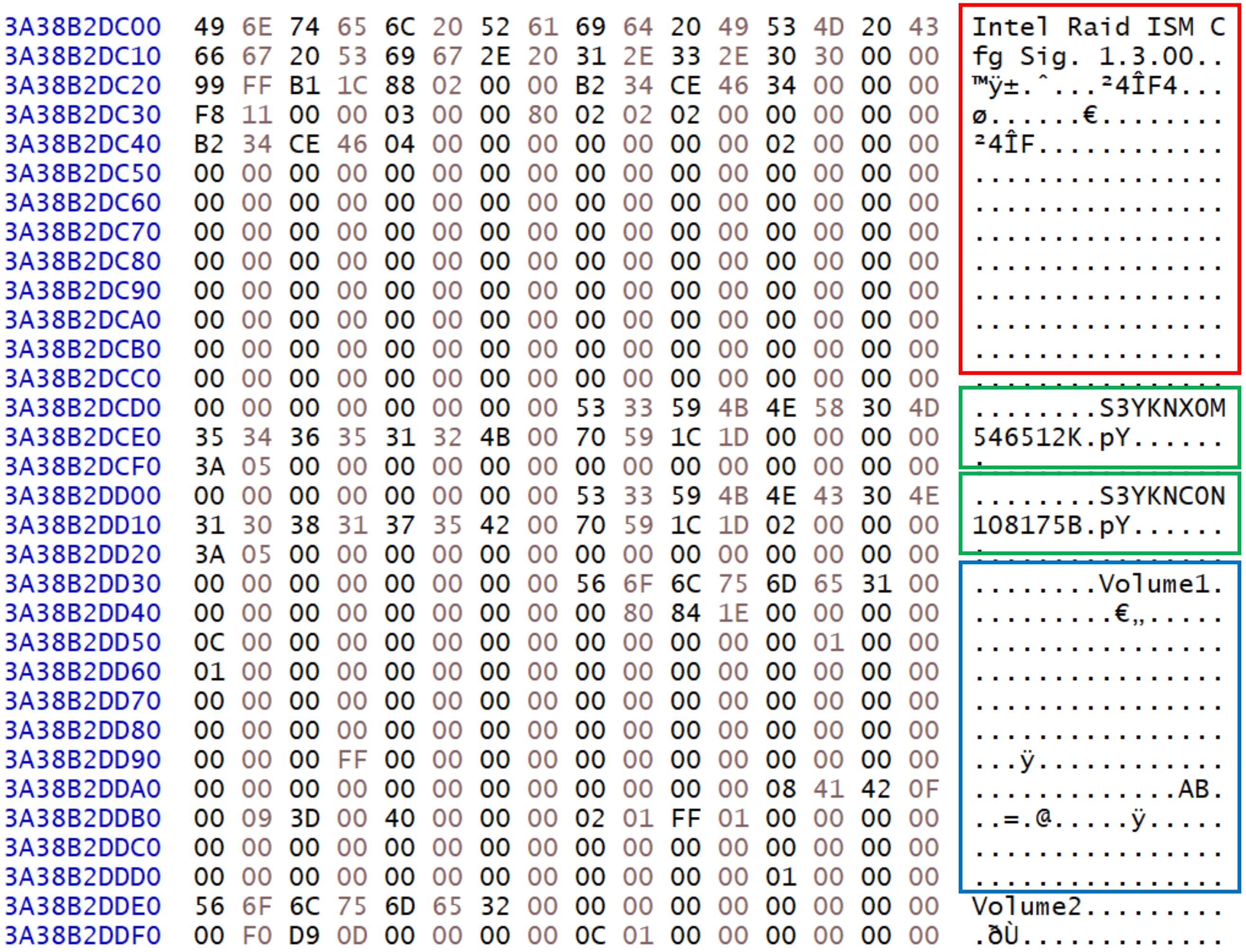


```
3A38B2DC00  49 6E 74 65 6C 20 52 61 69 64 20 49 53 4D 20 43  Intel Raid ISM C
3A38B2DC10  66 67 20 53 69 67 2E 20 31 2E 33 2E 30 30 00 00  fg Sig. 1.3.00..
3A38B2DC20  99 FF B1 1C 88 02 00 00 B2 34 CE 46 34 00 00 00  ™y±. ...²4ÎF4...
3A38B2DC30  F8 11 00 00 03 00 00 80 02 02 02 00 00 00 00 00  ø......€........
3A38B2DC40  B2 34 CE 46 04 00 00 00 00 00 00 00 02 00 00 00  ²4ÎF............
3A38B2DC50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCD0  00 00 00 00 00 00 00 00 53 33 59 4B 4E 58 30 4D  ........S3YKNX0M
3A38B2DCE0  35 34 36 35 31 32 4B 00 70 59 1C 1D 00 00 00 00  546512K.pY......
3A38B2DCF0  3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :...............
3A38B2DD00  00 00 00 00 00 00 00 00 53 33 59 4B 4E 43 30 4E  ........S3YKNC0N
3A38B2DD10  31 30 38 31 37 35 42 00 70 59 1C 1D 02 00 00 00  108175B.pY......
3A38B2DD20  3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :...............
3A38B2DD30  00 00 00 00 00 00 00 00 56 6F 6C 75 6D 65 31 00  ........Volume1.
3A38B2DD40  00 00 00 00 00 00 00 00 00 80 84 1E 00 00 00 00  .........€„.....
3A38B2DD50  0C 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
3A38B2DD60  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DD70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DD80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DD90  00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00  ...ÿ............
3A38B2DDA0  00 00 00 00 00 00 00 00 00 00 00 00 08 41 42 0F  .............AB.
3A38B2DDB0  00 09 3D 00 40 00 00 00 02 01 FF 01 00 00 00 00  ..=.@.....ÿ.....
3A38B2DDC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DDD0  00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00  ................
3A38B2DDE0  56 6F 6C 75 6D 65 32 00 00 00 00 00 00 00 00 00  Volume2.........
3A38B2DDF0  00 F0 D9 0D 00 00 00 00 0C 01 00 00 00 00 00 00  .ðÙ.............
```

Conclusion

Structure

- **Metadata**
  - Header(IMSM_SUPER)
    - Signature
  - Disk(IMSM_DISK)
    - Serial number
    - Size of disk
  - VDisk(IMVM_DEV)
    - Size of VDisk
    - Disk index consisting Vdisk

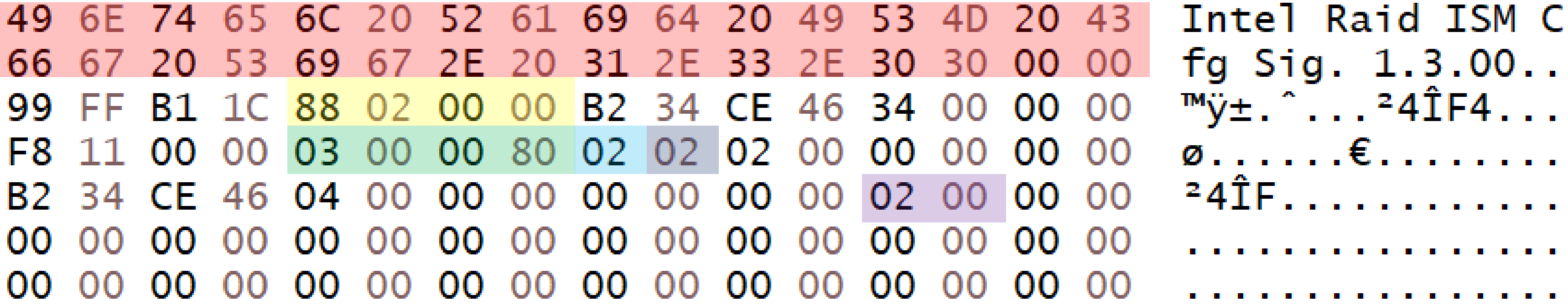


```
3A38B2DC00  49 6E 74 65 6C 20 52 61 69 64 20 49 53 4D 20 43  Intel Raid ISM C
3A38B2DC10  66 67 20 53 69 67 2E 20 31 2E 33 2E 30 30 00 00  fg Sig. 1.3.00..
3A38B2DC20  99 FF B1 1C 88 02 00 00 B2 34 CE 46 34 00 00 00  ™ÿ±.ˆ...²4ÎF4...
3A38B2DC30  F8 11 00 00 03 00 00 80 02 02 02 00 00 00 00 00  ø......€........
3A38B2DC40  B2 34 CE 46 04 00 00 00 00 00 00 00 02 00 00 00  ²4ÎF............
3A38B2DC50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DC90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DCD0  00 00 00 00 00 00 00 00 53 33 59 4B 4E 58 30 4D  ........S3YKNX0M
3A38B2DCE0  35 34 36 35 31 32 4B 00 70 59 1C 1D 00 00 00 00  546512K.pY......
3A38B2DCF0  3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :...............
3A38B2DD00  00 00 00 00 00 00 00 00 53 33 59 4B 4E 43 30 4E  ........S3YKNC0N
3A38B2DD10  31 30 38 31 37 35 42 00 70 59 1C 1D 02 00 00 00  108175B.pY......
3A38B2DD20  3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :...............
3A38B2DD30  00 00 00 00 00 00 00 00 56 6F 6C 75 6D 65 31 00  ........Volume1.
3A38B2DD40  00 00 00 00 00 00 00 00 00 80 84 1E 00 00 00 00  .........€„.....
3A38B2DD50  0C 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
3A38B2DD60  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DD70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DD80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DD90  00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00  ...ÿ............
3A38B2DDA0  00 00 00 00 00 00 00 00 00 00 00 00 08 41 42 0F  .............AB.
3A38B2DDB0  00 09 3D 00 40 00 00 00 02 01 FF 01 00 00 00 00  ..=.@.....ÿ.....
3A38B2DDC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
3A38B2DDD0  00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00  ................
3A38B2DDE0  56 6F 6C 75 6D 65 32 00 00 00 00 00 00 00 00 00  Volume2.........
3A38B2DDF0  00 F0 D9 0D 00 00 00 00 0C 01 00 00 00 00 00 00  .ðÙ.............
```

Conclusion

Structure

- **IMSM_SUPER**



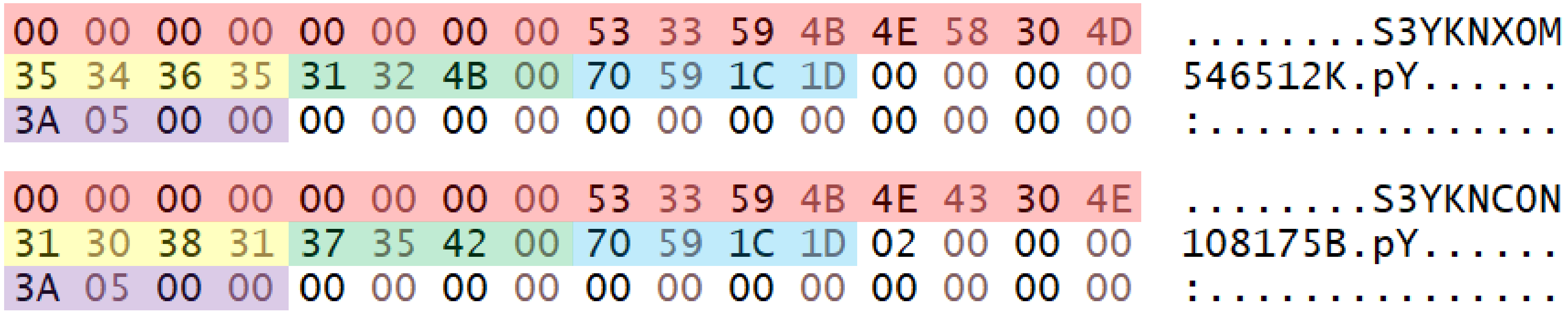
```
49 6E 74 65 6C 20 52 61 69 64 20 49 53 4D 20 43  Intel Raid ISM C
66 67 20 53 69 67 2E 20 31 2E 33 2E 30 30 00 00  fg Sig. 1.3.00..
99 FF B1 1C 88 02 00 00 B2 34 CE 46 34 00 00 00  ™ÿ±.ˆ...²4ÎF4...
F8 11 00 00 03 00 00 80 02 02 02 00 00 00 00 00  ø.......€.......
B2 34 CE 46 04 00 00 00 00 00 00 00 02 00 00 00  ²4ÎF............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Field Name	Description/Value
Signature	Signature: Magic + Version info
Metadata Block Size	Size of Meta Block
Attributes	Information about available RAID modes
Disk Number	Total number of disks used
Device Number	Number of volumes set
Created Device Number	Number of disks used for the RAID volume

Conclusion

Structure

- **IMSM_DISK**
  - A structure that contains information about the disk, such as the serial number, size



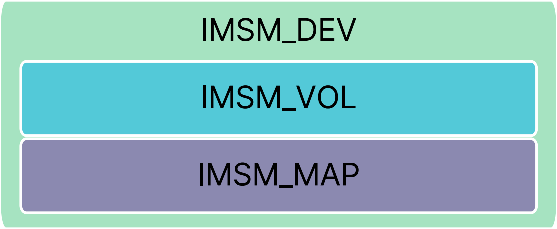
```
00 00 00 00 00 00 00 00 53 33 59 4B 4E 58 30 4D   ........S3YKNX0M
35 34 36 35 31 32 4B 00 70 59 1C 1D 00 00 00 00   546512K.pY......
3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00   :...............

00 00 00 00 00 00 00 00 53 33 59 4B 4E 43 30 4E   ........S3YKNC0N
31 30 38 31 37 35 42 00 70 59 1C 1D 02 00 00 00   108175B.pY......
3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00   :...............
```

Field Name	Description/Value
Serial Number	Product serial number
Total Blocks Low	Total Sector Count(Low)
SCSI ID	Connected port number
status	Status
Total_blocks High	Total Sector Count(High)

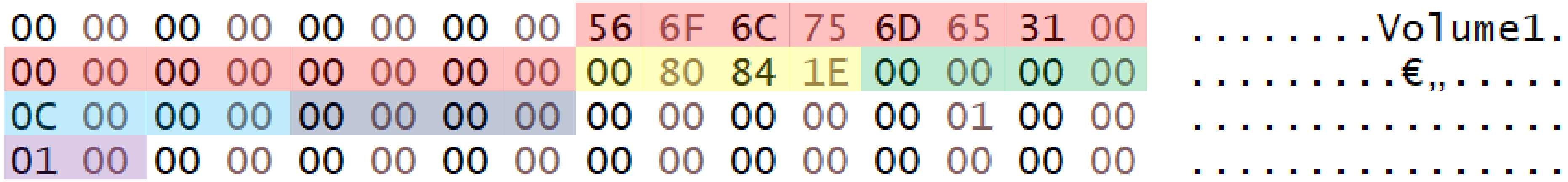
Conclusion

Structure

- **IMSM_DEV**
  - IMSM_VOL, information exists about Vdisk configured with IMSM_MAP



IMSM_DEV



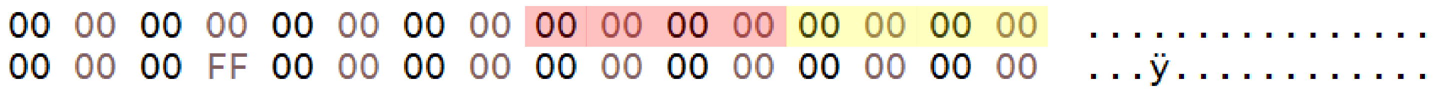
```
00 00 00 00 00 00 00 00 56 6F 6C 75 6D 65 31 00   ........Volume1.
00 00 00 00 00 00 00 00 00 80 84 1E 00 00 00 00   .........€„.....
0C 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00   ................
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Field Name	Description/Value
Volume	Volume Name
Size Low	Size of volume (Low)
Size High	Size of volume (High)
Status	Status
Reserved Blocks	Reserved block
Unique Volume Id	Volume ID

Conclusion

Structure

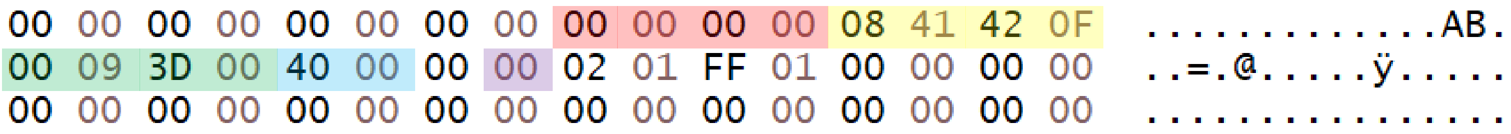
- **IMSM_VOL**



```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 FF 00 00 00 00 00 00 00 00 00 00 00 00  ...ÿ............
```

Field Name	Description/Value
Current migr unit	Migration
Checkpoint Id	Migration

- **IMSM_MAP**



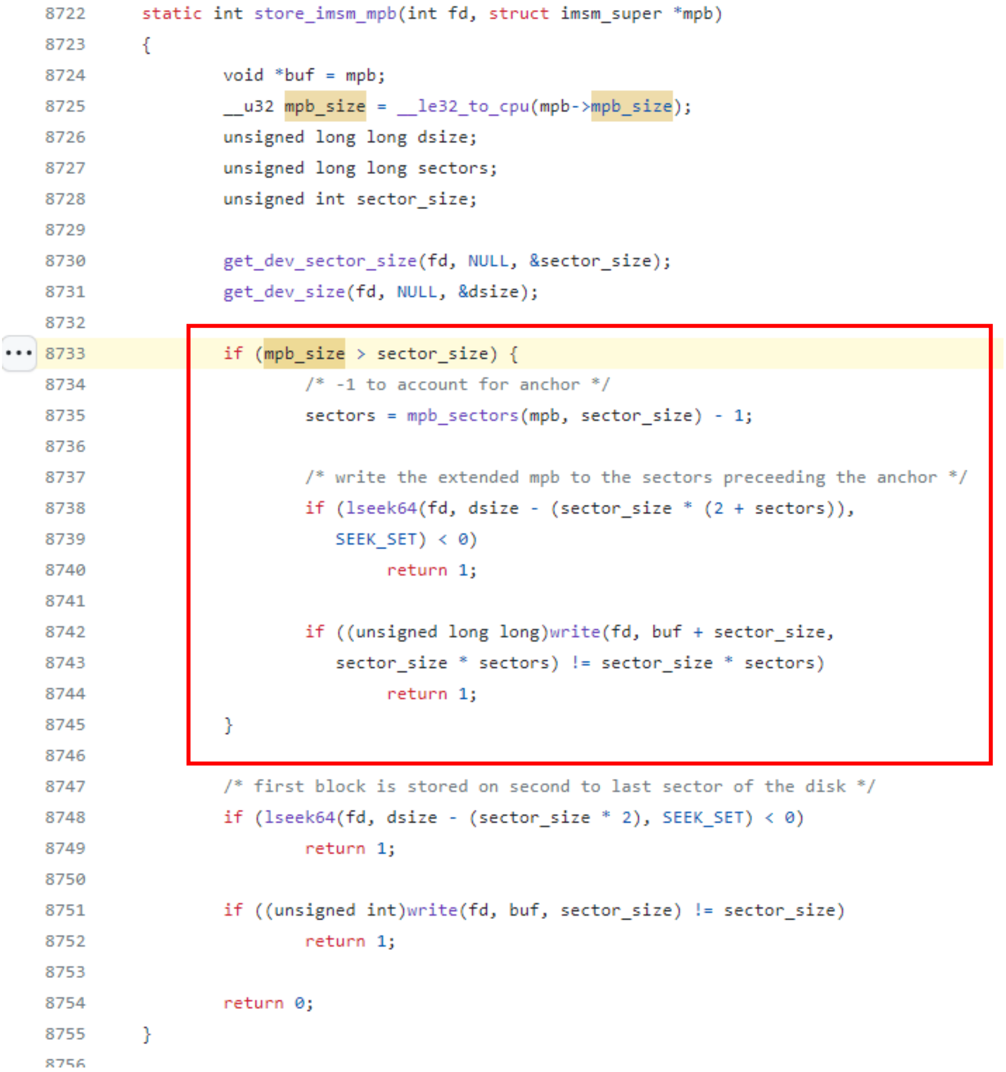
```
00 00 00 00 00 00 00 00 00 00 00 00 08 41 42 0F  .............AB.
00 09 3D 00 40 00 00 00 02 01 FF 01 00 00 00 00  ..=.@.....ÿ.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Field Name	Description/Value
Partiton LBA0 Low	The start of the volume LBA
Blocks Per Member Low	Number of blocks for each disk
Number of Data Stripes Low	Stripe Count
Blocks Per Strip	The number of blocks that make up a Strip
Raid_level	RAID Levels

Conclusion

How MBP(Metadata Block Parameter) is recorded

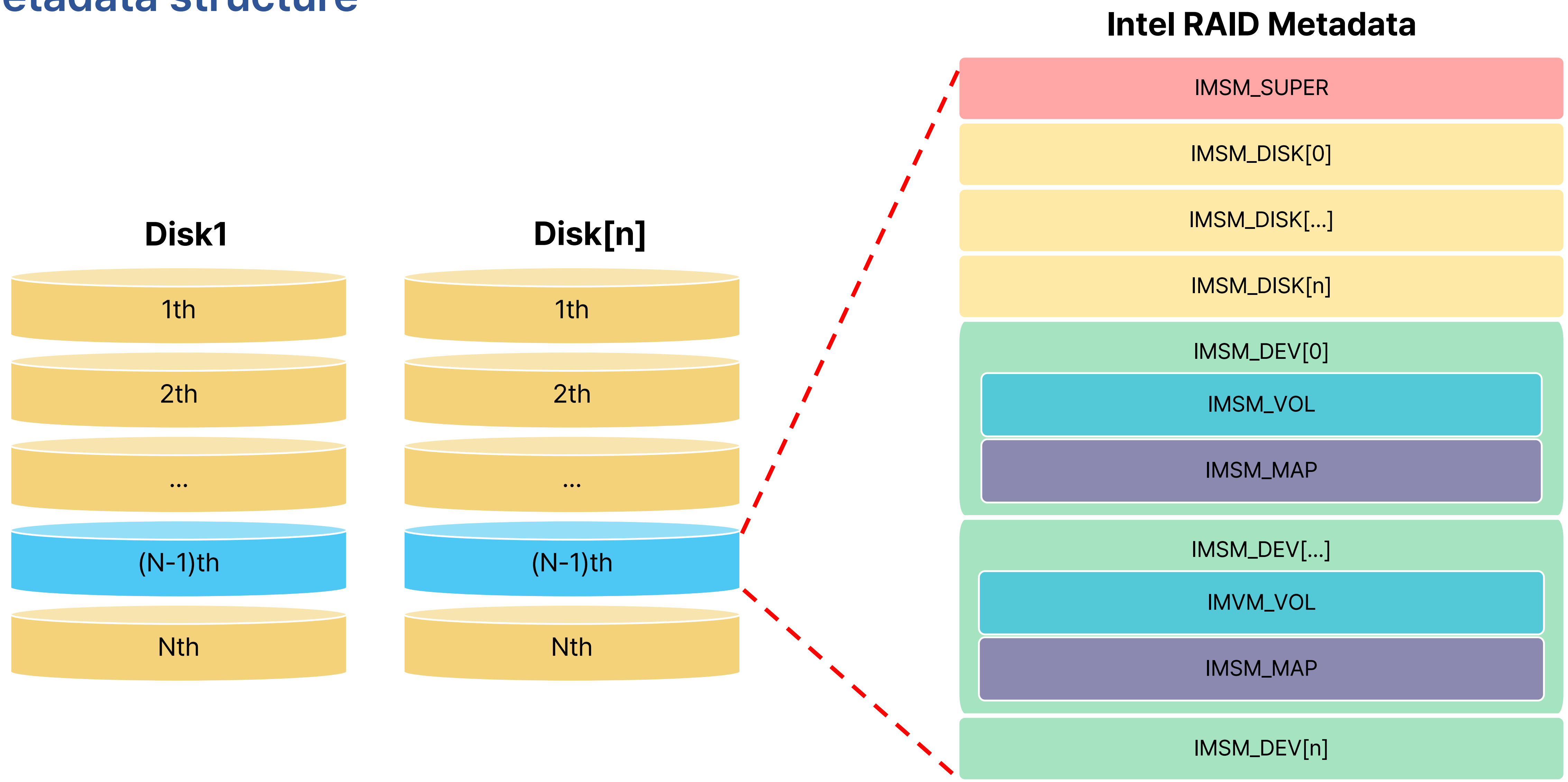
- Starts from (N-1)th sector
- If size exceed one sector
- Records on (N-2)th sector



```
static int store_imsm_mpb(int fd, struct imsm_super *mpb)
{
    void *buf = mpb;
    __u32 mpb_size = __le32_to_cpu(mpb->mpb_size);
    unsigned long long dsize;
    unsigned long long sectors;
    unsigned int sector_size;

    get_dev_sector_size(fd, NULL, &sector_size);
    get_dev_size(fd, NULL, &dsize);

    if (mpb_size > sector_size) {
        /* -1 to account for anchor */
        sectors = mpb_sectors(mpb, sector_size) - 1;

        /* write the extended mpb to the sectors preceeding the anchor */
        if (lseek64(fd, dsize - (sector_size * (2 + sectors)),
           SEEK_SET) < 0)
              return 1;

        if ((unsigned long long)write(fd, buf + sector_size,
           sector_size * sectors) != sector_size * sectors)
              return 1;
    }

    /* first block is stored on second to last sector of the disk */
    if (lseek64(fd, dsize - (sector_size * 2), SEEK_SET) < 0)
        return 1;

    if ((unsigned int)write(fd, buf, sector_size) != sector_size)
        return 1;

    return 0;
}
```

Conclusion

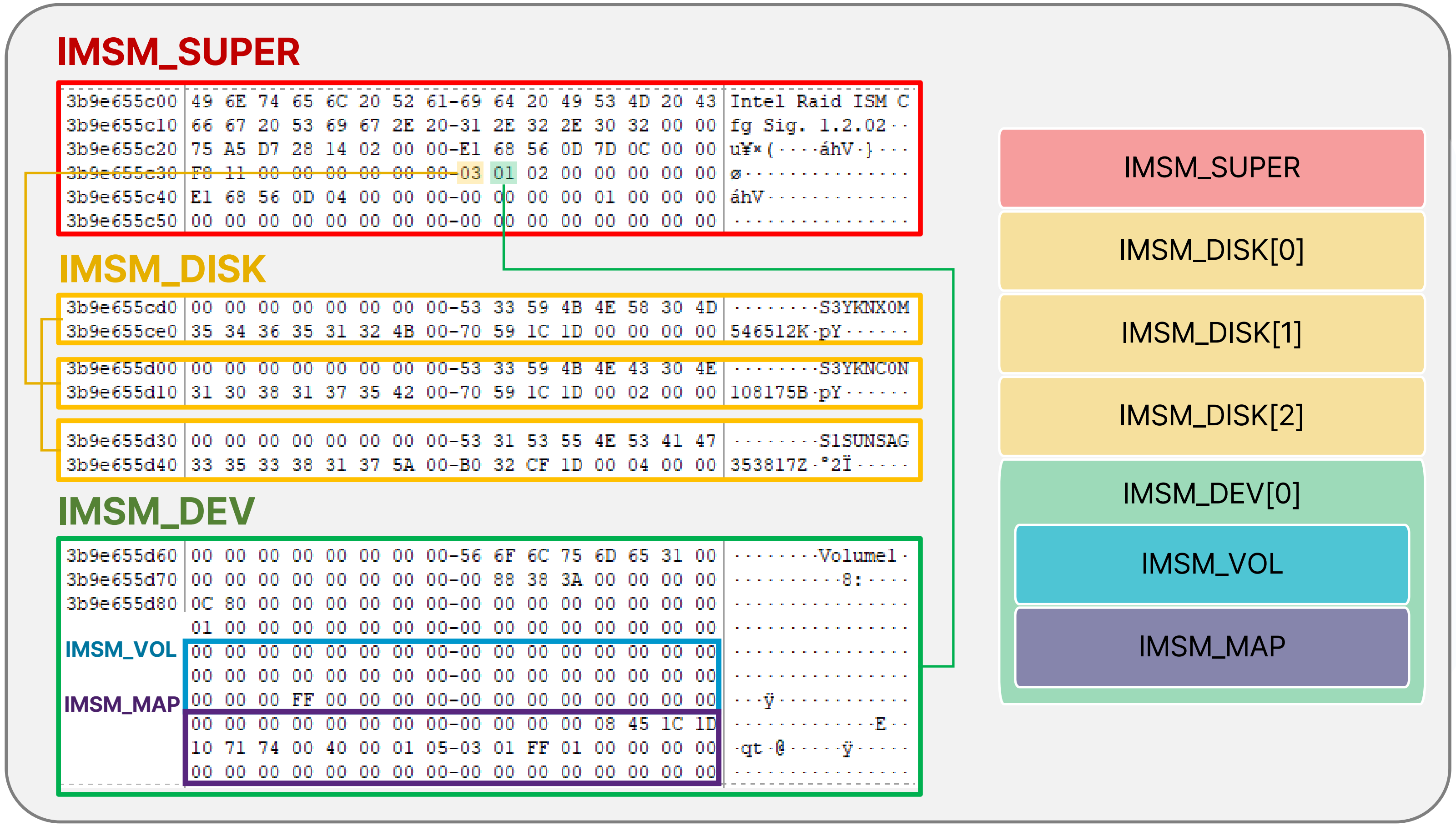
Metadata structure



Conclusion

Intel RAID - Volume (3 Disk - RAID 5)

- **Volume (RAID 5)**



IMSM_SUPER

```
3b9e655c00 49 6E 74 65 6C 20 52 61-69 64 20 49 53 4D 20 43 Intel Raid ISM C
3b9e655c10 66 67 20 53 69 67 2E 20-31 2E 32 2E 30 32 00 00 fg Sig. 1.2.02··
3b9e655c20 75 A5 D7 28 14 02 00 00-E1 68 56 0D 7D 0C 00 00 u¥×(····áhV·}···
3b9e655c30 F8 11 00 00 00 00 00 00-03 01 02 00 00 00 00 00 ø···············
3b9e655c40 E1 68 56 0D 04 00 00 00-00 00 00 01 00 00 00 00 áhV·············
3b9e655c50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
```

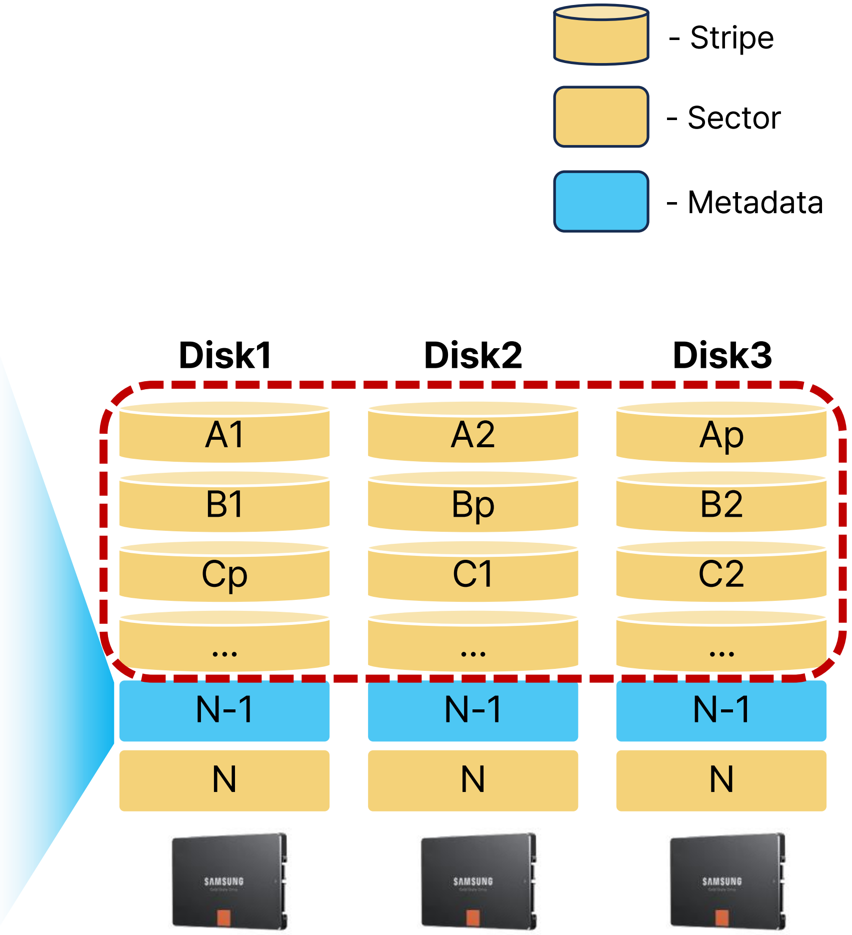
IMSM_DISK

```
3b9e655cd0 00 00 00 00 00 00 00 00-53 33 59 4B 4E 58 30 4D ········S3YKNX0M
3b9e655ce0 35 34 36 35 31 32 4B 00-70 59 1C 1D 00 00 00 00 546512K·pY······

3b9e655d00 00 00 00 00 00 00 00 00-53 33 59 4B 4E 43 30 4E ········S3YKNC0N
3b9e655d10 31 30 38 31 37 35 42 00-70 59 1C 1D 00 02 00 00 108175B·pY······

3b9e655d30 00 00 00 00 00 00 00 00-53 31 53 55 4E 53 41 47 ········S1SUNSAG
3b9e655d40 33 35 33 38 31 37 5A 00-B0 32 CF 1D 00 04 00 00 353817Z·°2Ï····
```

IMSM_DEV

```
3b9e655d60 00 00 00 00 00 00 00 00-56 6F 6C 75 6D 65 31 00 ········Volume1·
3b9e655d70 00 00 00 00 00 00 00 00-00 88 38 3A 00 00 00 00 ··········8:····
3b9e655d80 0C 80 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
           01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
IMSM_VOL   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
           00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
IMSM_MAP   00 00 00 FF 00 00 00 00-00 00 00 00 00 00 00 00 ···ÿ············
           00 00 00 00 00 00 00 00-00 00 00 08 45 1C 1D ···············E··
           10 71 74 00 40 00 01 05-03 01 FF 01 00 00 00 00 ·qt·@·····ÿ·····
           00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
```

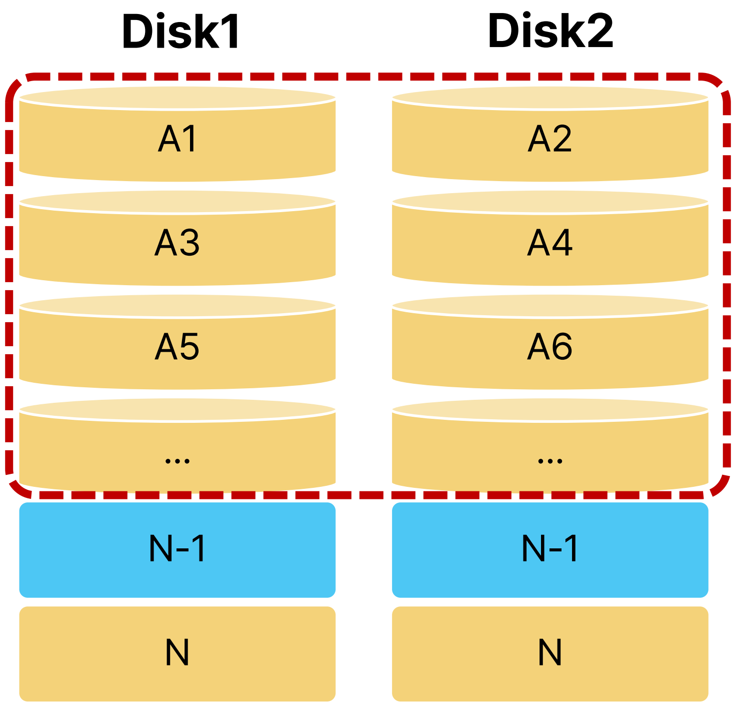
Intel RAID Metadata



Conclusion

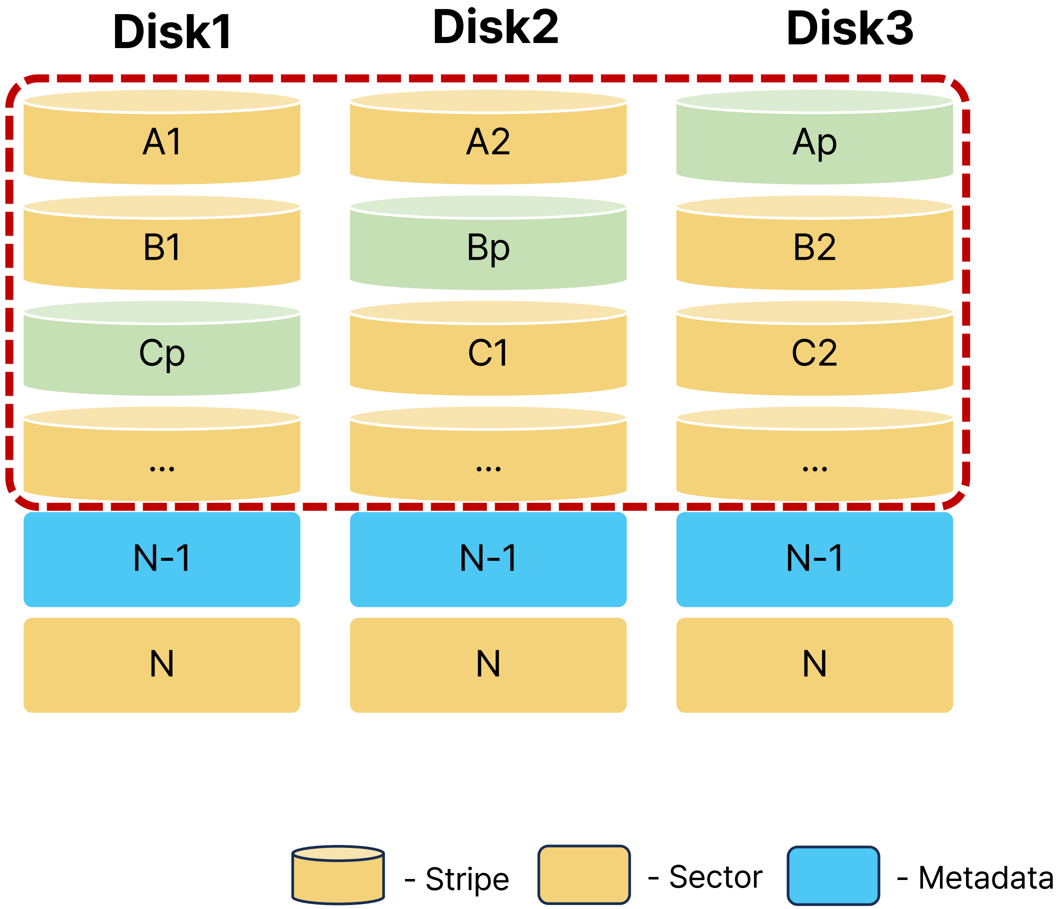
Intel RAID Disk Layout

- **RAID 0**



Disk1	Disk2
A1	A2
A3	A4
A5	A6
…	…
N-1	N-1
N	N

- **RAID 5**



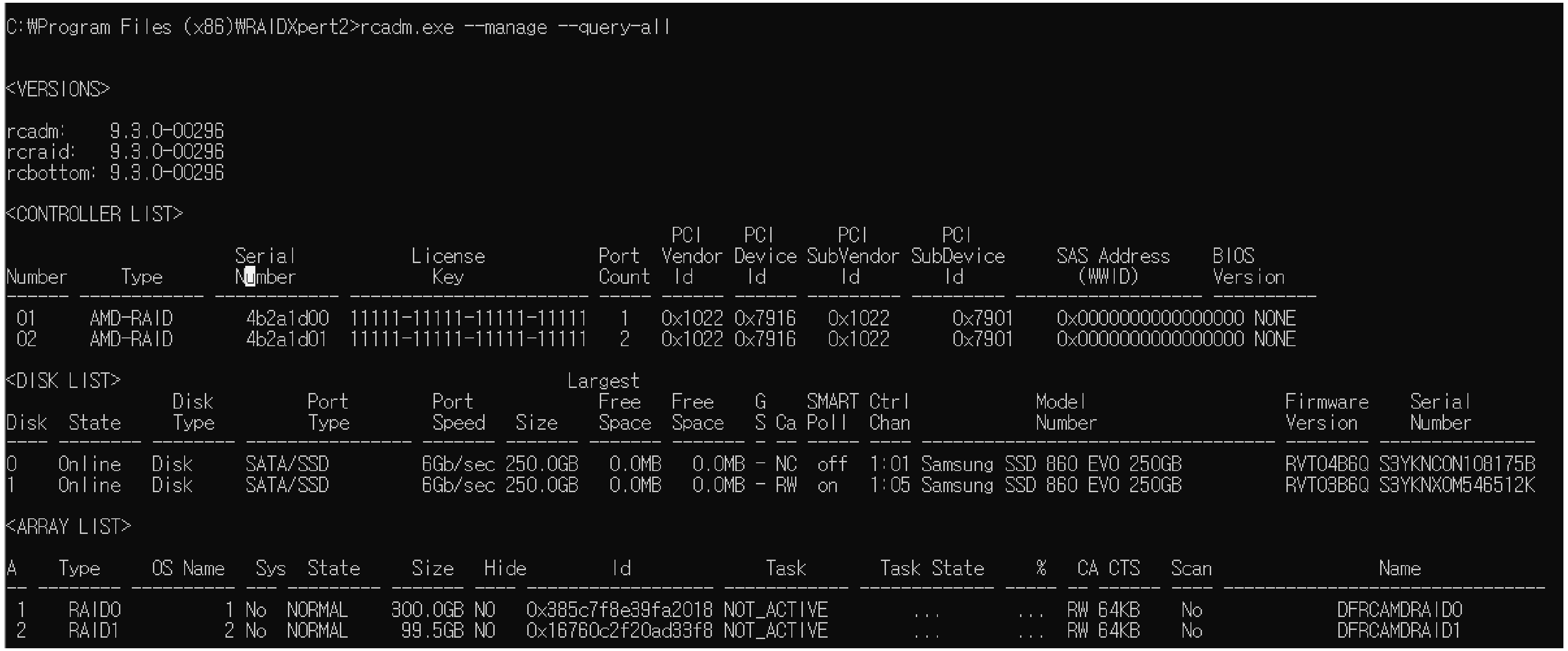
Disk1	Disk2	Disk3
A1	A2	Ap
B1	Bp	B2
Cp	C1	C2
…	…	…
N-1	N-1	N-1
N	N	N

AMD RAID

Conclusion

Components of AMD RAID

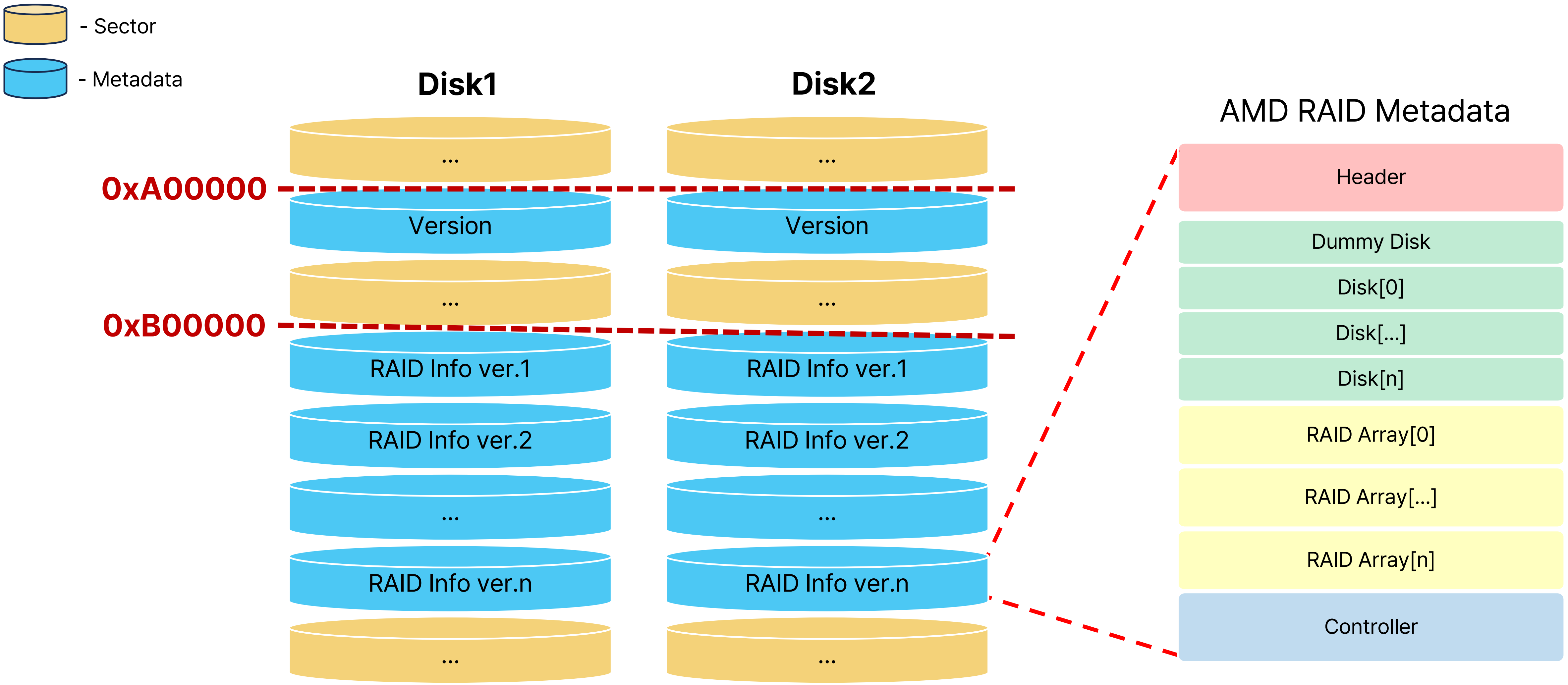
- **Version**
  - Firmware, driver version
- **Controller**
  - RAID Controllers
- **Disk**
  - Disk Info
- **Array**
  - Vdisk Info



```
C:₩Program Files (x86)₩RAIDXpert2>rcadm.exe --manage --query-all

<VERSIONS>

rcadm:    9.3.0-00296
rcraid:   9.3.0-00296
rcbottom: 9.3.0-00296

<CONTROLLER LIST>
                                                                PCI    PCI      PCI        PCI
                      Serial           License           Port  Vendor Device SubVendor SubDevice     SAS Address    BIOS
Number     Type       Number             Key             Count  Id     Id       Id         Id          (WWID)       Version
------ ------------ ------------ ----------------------- ----- ------ ------ --------- --------- ------------------ ----------
 01     AMD-RAID      4b2a1d00  11111-11111-11111-11111   1   0x1022 0x7916   0x1022      0x7901    0x0000000000000000 NONE
 02     AMD-RAID      4b2a1d01  11111-11111-11111-11111   2   0x1022 0x7916   0x1022      0x7901    0x0000000000000000 NONE

<DISK LIST>                                           Largest
                Disk         Port        Port            Free   Free    G    SMART Ctrl               Model                   Firmware    Serial
Disk  State     Type         Type        Speed   Size    Space  Space   S Ca Poll  Chan               Number                  Version     Number
---- -------- -------- ---------------- ------- ------- ------- ------- - -- ----- ---- ------------------------------------- -------- ----------------
0    Online   Disk     SATA/SSD         6Gb/sec 250.0GB  0.0MB   0.0MB - NC  off  1:01 Samsung SSD 860 EVO 250GB          RVT04B6Q S3YKNC0N108175B
1    Online   Disk     SATA/SSD         6Gb/sec 250.0GB  0.0MB   0.0MB - RW  on   1:05 Samsung SSD 860 EVO 250GB          RVT03B6Q S3YKNX0M546512K

<ARRAY LIST>

A    Type     OS Name   Sys  State     Size  Hide       Id           Task       Task State    %   CA CTS    Scan             Name
-- -------- ---------- --- --------- ------- ---- ------------------ ---------- ------------- ----- -- ------- ---- -----------------------------
 1    RAID0          1 No  NORMAL    300.0GB NO   0x385c7f8e39fa2018 NOT_ACTIVE       ...        ...  RW 64KB    No          DFRCAMDRAID0
 2    RAID1          2 No  NORMAL     99.5GB NO   0x16760c2f20ad33f8 NOT_ACTIVE       ...        ...  RW 64KB    No          DFRCAMDRAID1
```

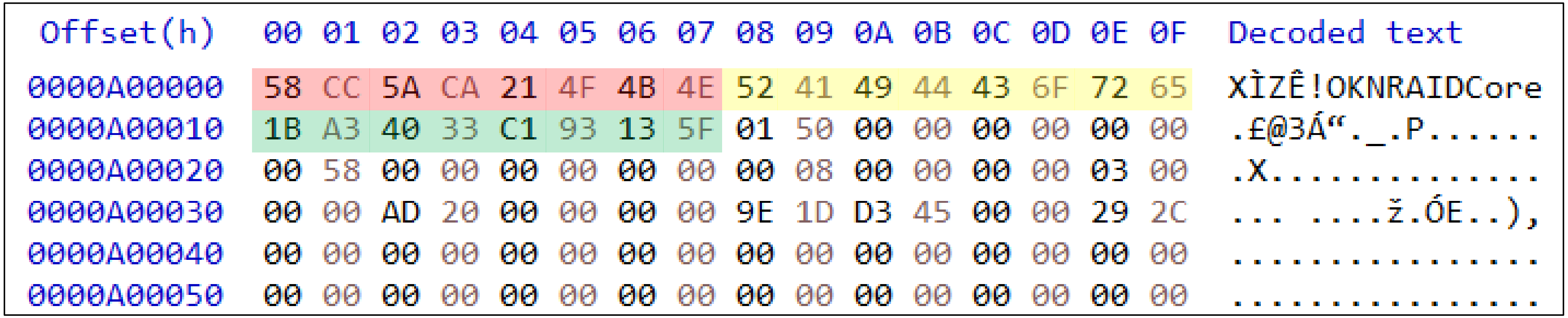
Conclusion

Metadata structure



Conclusion

Version

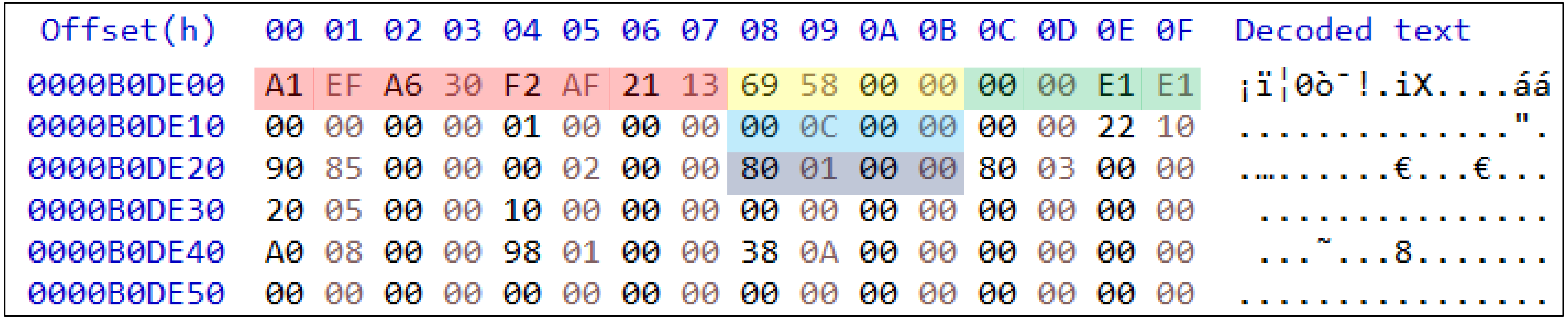


```
Offset(h)   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

0000A00000  58 CC 5A CA 21 4F 4B 4E 52 41 49 44 43 6F 72 65  XÌZÊ!OKNRAIDCore
0000A00010  1B A3 40 33 C1 93 13 5F 01 50 00 00 00 00 00 00  .£@3Á“._.P......
0000A00020  00 58 00 00 00 00 00 00 00 08 00 00 00 00 03 00  .X..............
0000A00030  00 00 AD 20 00 00 00 00 9E 1D D3 45 00 00 29 2C  ... ....ž.ÓE..),
0000A00040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000A00050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Field Name	Description/Value
Checksum	Version Block Checksum
Signature	Signature(RAIDCore)
H/W ID	The ID of the current hardware

Conclusion

Header



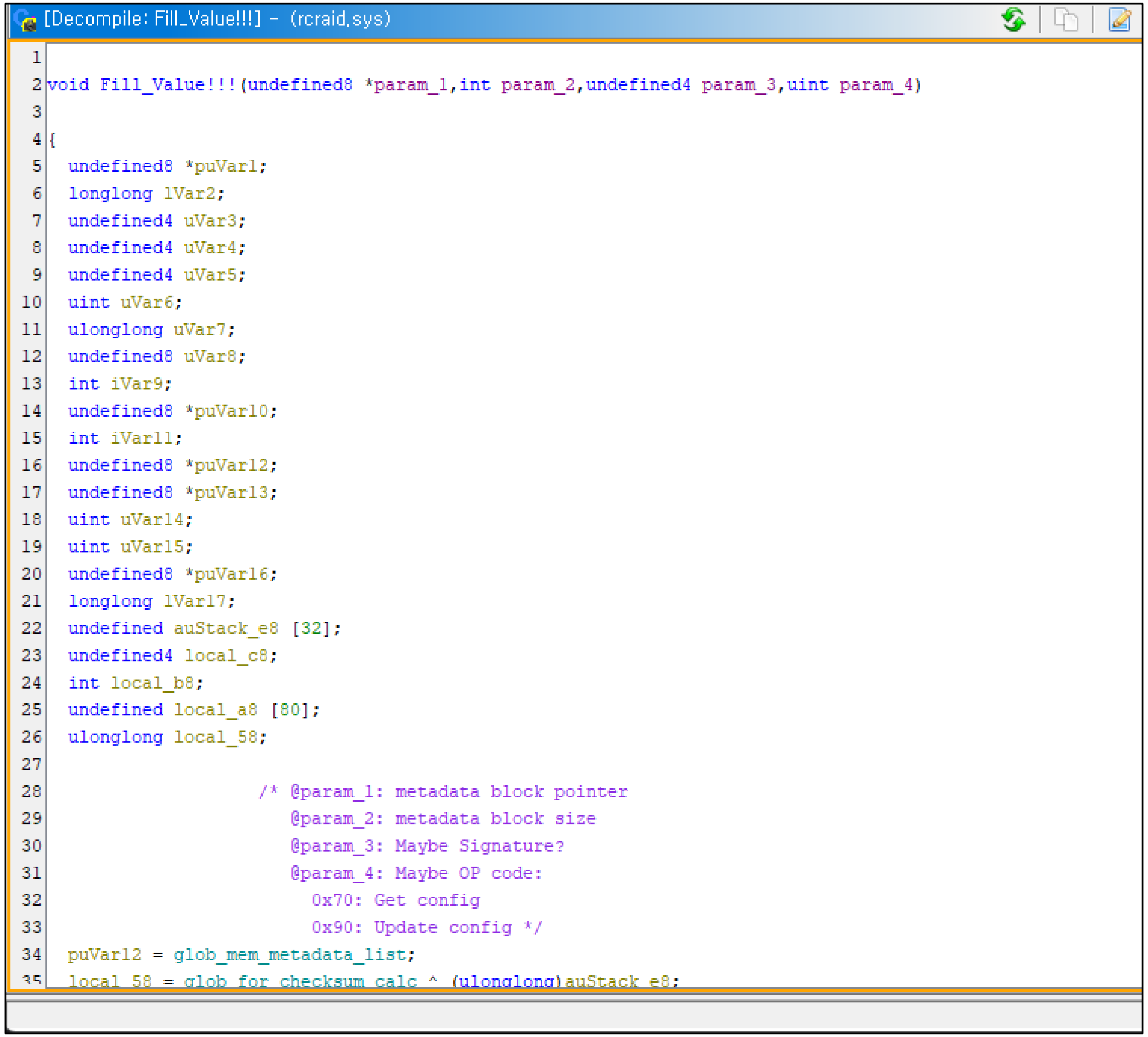
```
 Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
0000B0DE00  A1 EF A6 30 F2 AF 21 13 69 58 00 00 00 00 E1 E1  ¡ï¦0ò¯!.iX....áá
0000B0DE10  00 00 00 00 01 00 00 00 00 0C 00 00 00 00 22 10  ..............".
0000B0DE20  90 85 00 00 00 02 00 00 80 01 00 00 80 03 00 00  .…......€...€...
0000B0DE30  20 05 00 00 10 00 00 00 00 00 00 00 00 00 00 00   ...............
0000B0DE40  A0 08 00 00 98 01 00 00 38 0A 00 00 00 00 00 00   ...˜...8.......
0000B0DE50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Field Name	Description/Value
Checksum	Metadata checksum
Checksum Parameter	Parameters for calculating checksums
Signature	Signature
Metadata Size	Metadata size
Disk Block Size	Disk block size

Conclusion

Header - Reversing

- **Fill_Value function**
  - Param_1: metadata block pointer
  - Param_2: metadata block size
  - Param_3: Maybe Signature
  - Param_4: Maybe OP code



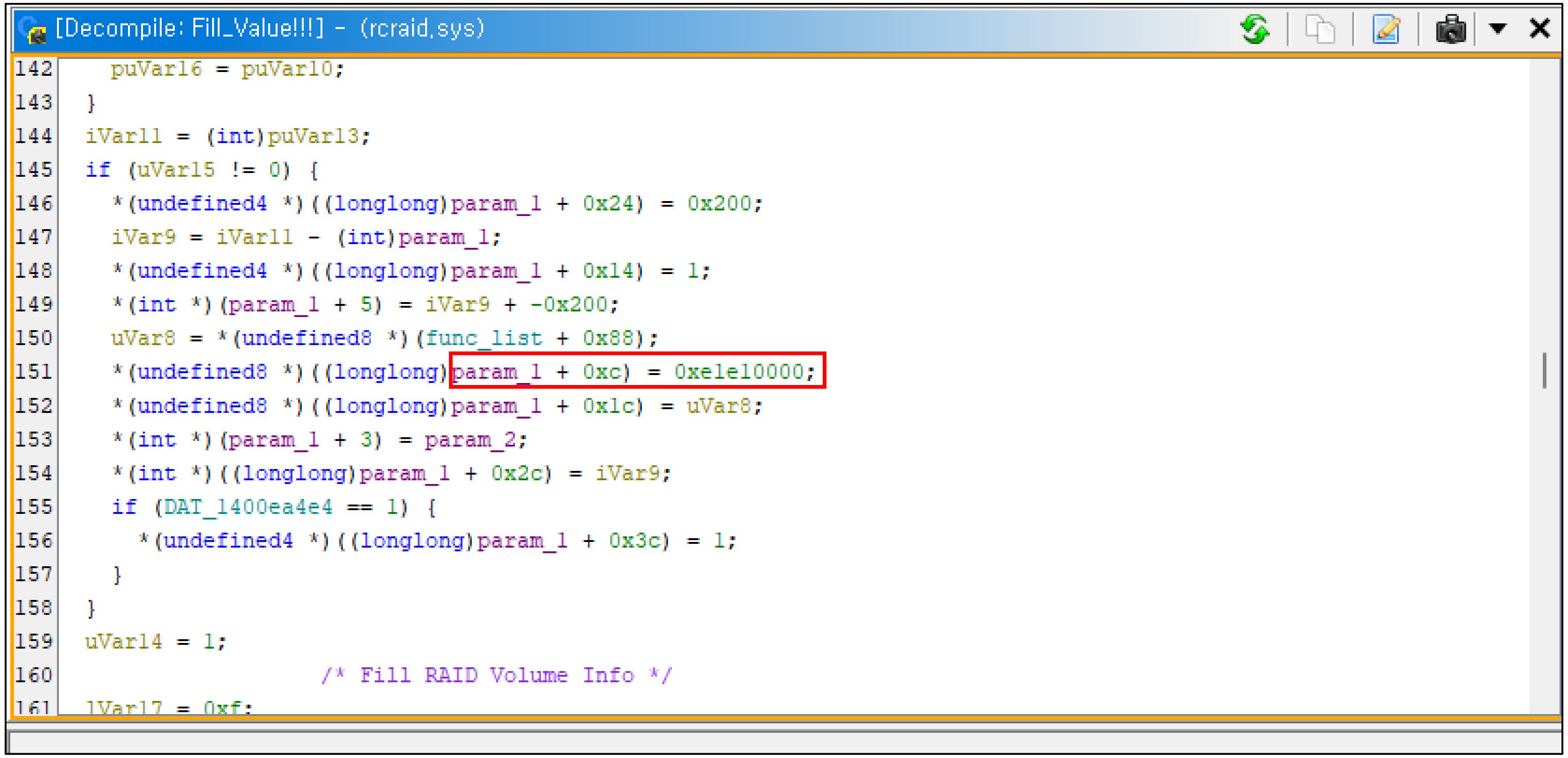
[Decompile: Fill_Value!!!] - (rcraid.sys)

```

void Fill_Value!!!(undefined8 *param_1,int param_2,undefined4 param_3,uint param_4)

{
  undefined8 *puVar1;
  longlong lVar2;
  undefined4 uVar3;
  undefined4 uVar4;
  undefined4 uVar5;
  uint uVar6;
  ulonglong uVar7;
  undefined8 uVar8;
  int iVar9;
  undefined8 *puVar10;
  int iVar11;
  undefined8 *puVar12;
  undefined8 *puVar13;
  uint uVar14;
  uint uVar15;
  undefined8 *puVar16;
  longlong lVar17;
  undefined auStack_e8 [32];
  undefined4 local_c8;
  int local_b8;
  undefined local_a8 [80];
  ulonglong local_58;

                    /* @param_1: metadata block pointer
                       @param_2: metadata block size
                       @param_3: Maybe Signature?
                       @param_4: Maybe OP code:
                         0x70: Get config
                         0x90: Update config */
  puVar12 = glob_mem_metadata_list;
  local_58 = glob_for_checksum_calc ^ (ulonglong)auStack_e8;
```

Conclusion

Header - Reversing

- **Fill_Value function**
  - Signature

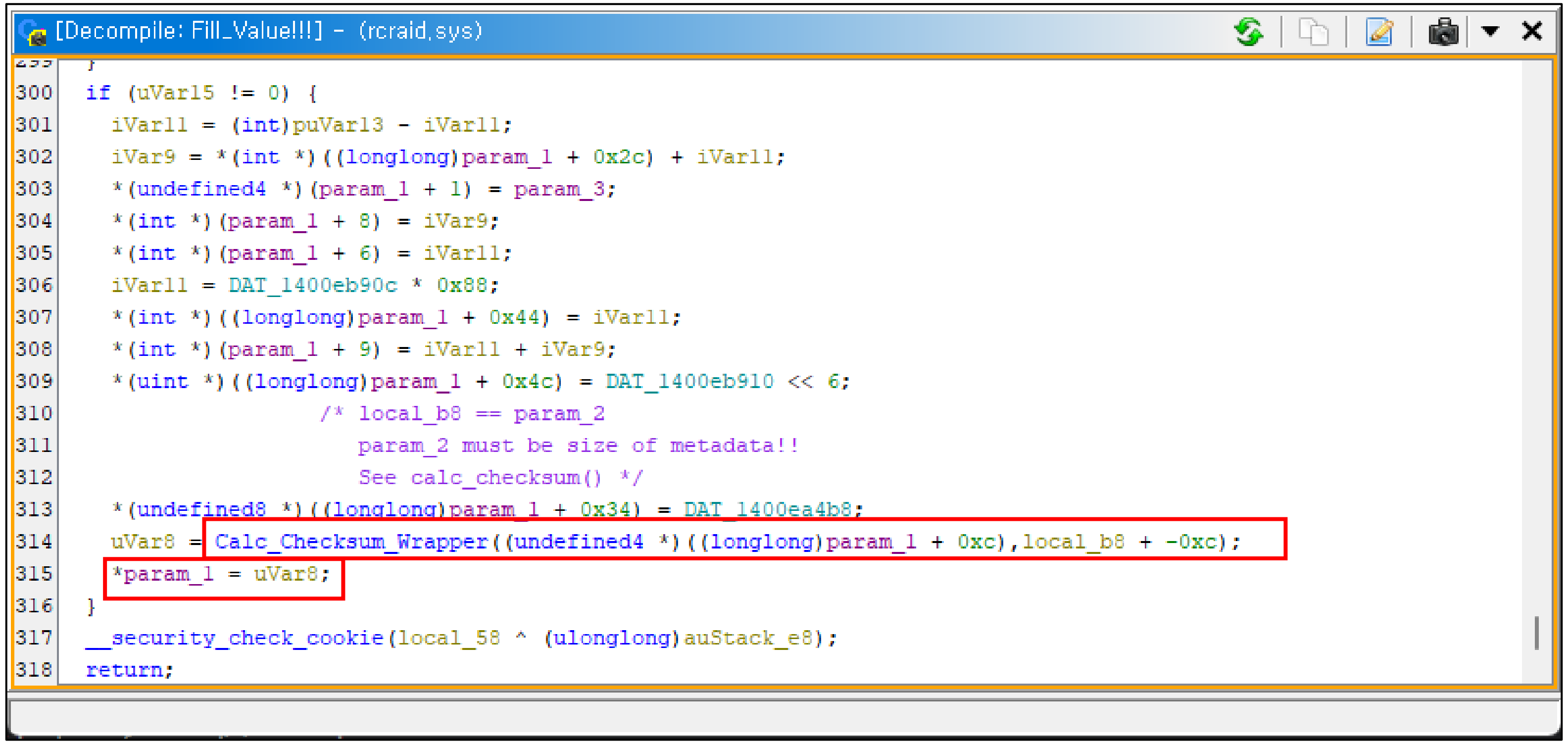


```
    puVar16 = puVar10;
  }
  iVar11 = (int)puVar13;
  if (uVar15 != 0) {
    *(undefined4 *)((longlong)param_1 + 0x24) = 0x200;
    iVar9 = iVar11 - (int)param_1;
    *(undefined4 *)((longlong)param_1 + 0x14) = 1;
    *(int *)(param_1 + 5) = iVar9 + -0x200;
    uVar8 = *(undefined8 *)(func_list + 0x88);
    *(undefined8 *)((longlong)param_1 + 0xc) = 0xe1e10000;
    *(undefined8 *)((longlong)param_1 + 0x1c) = uVar8;
    *(int *)(param_1 + 3) = param_2;
    *(int *)((longlong)param_1 + 0x2c) = iVar9;
    if (DAT_1400ea4e4 == 1) {
      *(undefined4 *)((longlong)param_1 + 0x3c) = 1;
    }
  }
  uVar14 = 1;
                    /* Fill RAID Volume Info */
  lVar17 = 0xf;
```

Conclusion

Header - Reversing

- **Checksum**
  - Checksum calculation from the Offset 0xC
  - Insert the checksum into the Offset 0x0

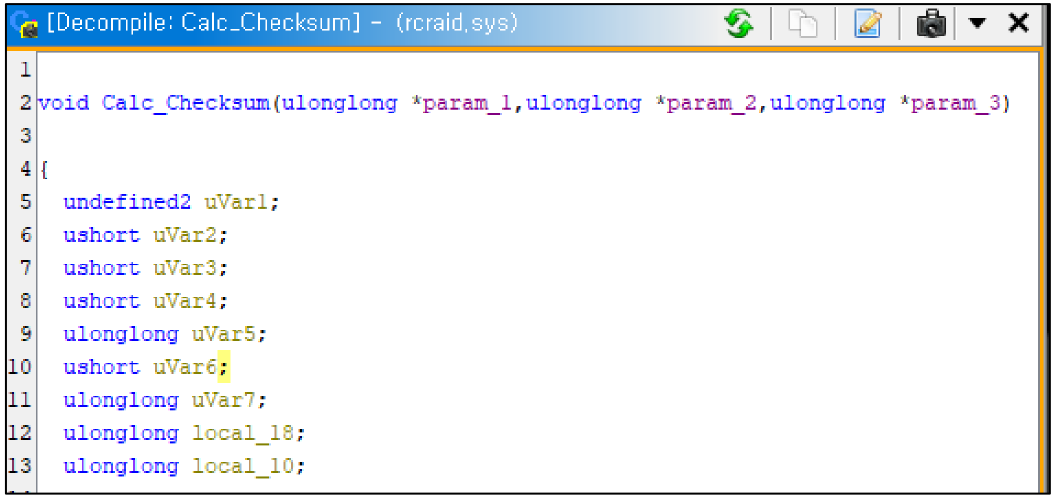


```
}
if (uVar15 != 0) {
  iVar11 = (int)puVar13 - iVar11;
  iVar9 = *(int *)((longlong)param_1 + 0x2c) + iVar11;
  *(undefined4 *)(param_1 + 1) = param_3;
  *(int *)(param_1 + 8) = iVar9;
  *(int *)(param_1 + 6) = iVar11;
  iVar11 = DAT_1400eb90c * 0x88;
  *(int *)((longlong)param_1 + 0x44) = iVar11;
  *(int *)(param_1 + 9) = iVar11 + iVar9;
  *(uint *)((longlong)param_1 + 0x4c) = DAT_1400eb910 << 6;
                /* local_b8 == param_2
                   param_2 must be size of metadata!!
                   See calc_checksum() */
  *(undefined8 *)((longlong)param_1 + 0x34) = DAT_1400ea4b8;
  uVar8 = Calc_Checksum_Wrapper((undefined4 *)((longlong)param_1 + 0xc),local_b8 + -0xc);
  *param_1 = uVar8;
}
__security_check_cookie(local_58 ^ (ulonglong)auStack_e8);
return;
```

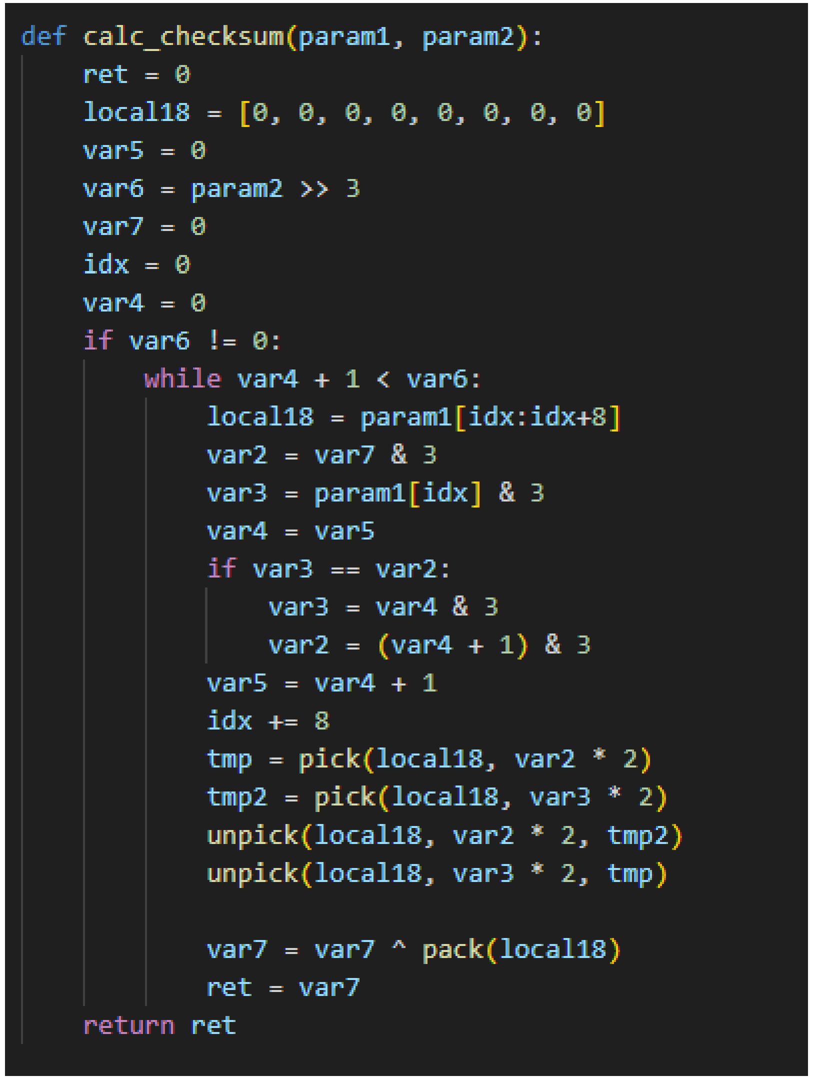
Conclusion

Header - Reversing

- **Checksum Calculation Functions**
  - Calc_Checksum(int start_offset, int length)
  - Verify that the same value is acquired



```

void Calc_Checksum(ulonglong *param_1,ulonglong *param_2,ulonglong *param_3)

{
  undefined2 uVar1;
  ushort uVar2;
  ushort uVar3;
  ushort uVar4;
  ulonglong uVar5;
  ushort uVar6;
  ulonglong uVar7;
  ulonglong local_18;
  ulonglong local_10;
```

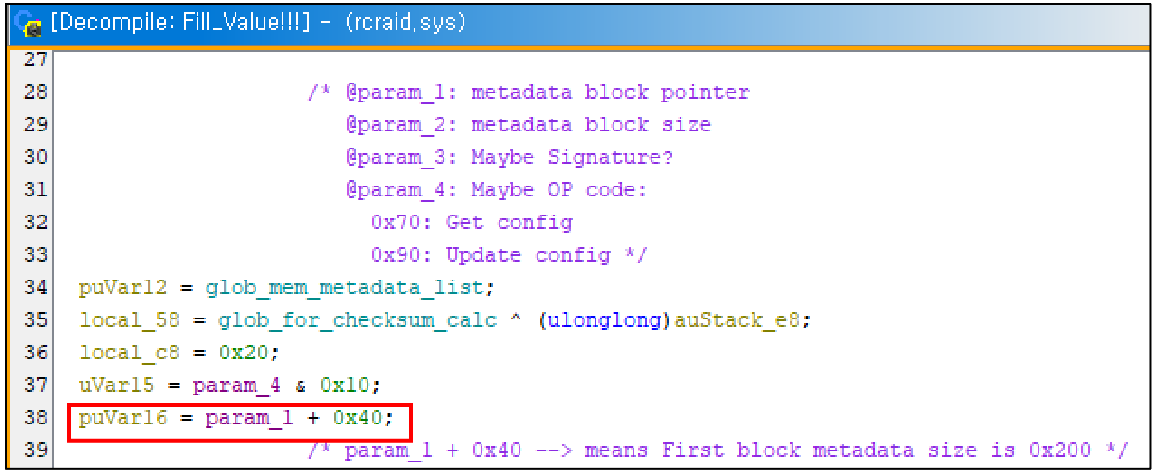


```python
def calc_checksum(param1, param2):
    ret = 0
    local18 = [0, 0, 0, 0, 0, 0, 0, 0]
    var5 = 0
    var6 = param2 >> 3
    var7 = 0
    idx = 0
    var4 = 0
    if var6 != 0:
        while var4 + 1 < var6:
            local18 = param1[idx:idx+8]
            var2 = var7 & 3
            var3 = param1[idx] & 3
            var4 = var5
            if var3 == var2:
                var3 = var4 & 3
                var2 = (var4 + 1) & 3
            var5 = var4 + 1
            idx += 8
            tmp = pick(local18, var2 * 2)
            tmp2 = pick(local18, var3 * 2)
            unpick(local18, var2 * 2, tmp2)
            unpick(local18, var3 * 2, tmp)

            var7 = var7 ^ pack(local18)
            ret = var7
    return ret
```

Conclusion

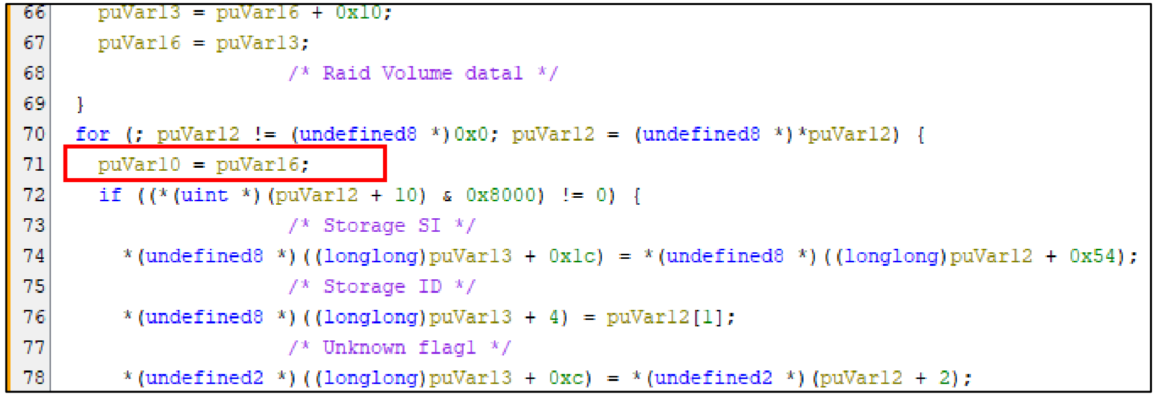
Header - Reversing

- **Disk Block Size**
  - Record Disk info in Fill_Value
  - Pointer after 0x40 of disk information pointer
    - Imply that the header size is 0x200
  - After that, repeat the number of disks and proceed with the recording process.



[Decompile: Fill_Value!!!] - (rcraid.sys)

```

                     /* @param_1: metadata block pointer
                        @param_2: metadata block size
                        @param_3: Maybe Signature?
                        @param_4: Maybe OP code:
                          0x70: Get config
                          0x90: Update config */
  puVar12 = glob_mem_metadata_list;
  local_58 = glob_for_checksum_calc ^ (ulonglong)auStack_e8;
  local_c8 = 0x20;
  uVar15 = param_4 & 0x10;
  puVar16 = param_1 + 0x40;
                     /* param_1 + 0x40 --> means First block metadata size is 0x200 */
```

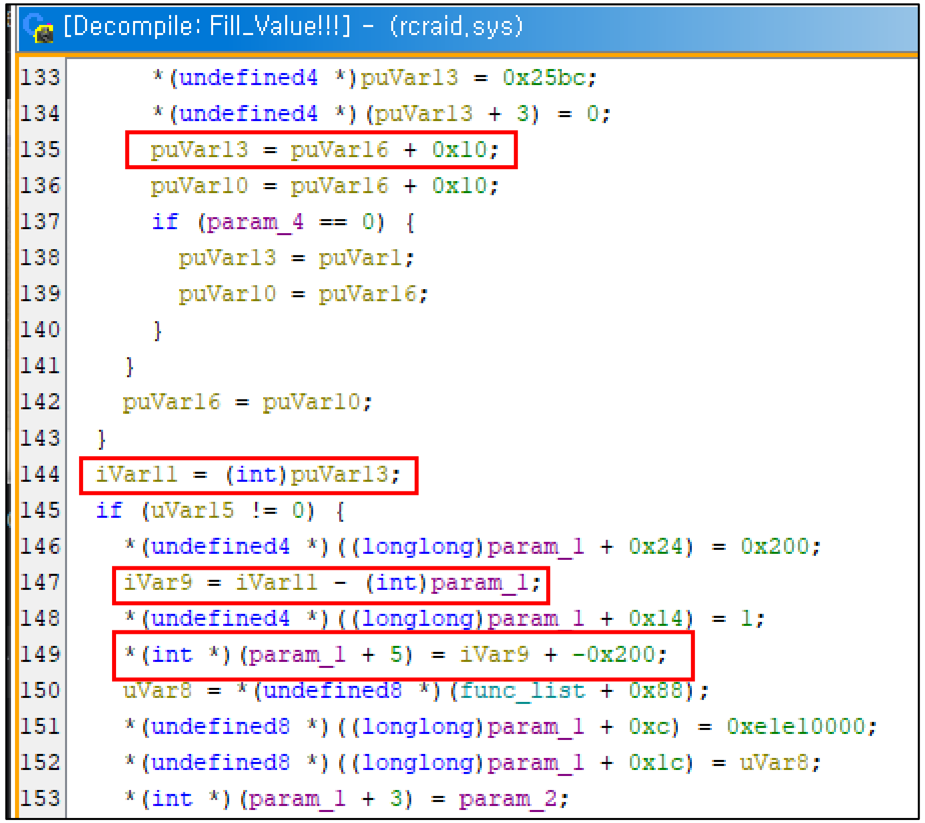


```
    puVar13 = puVar16 + 0x10;
    puVar16 = puVar13;
                     /* Raid Volume data1 */
  }
  for (; puVar12 != (undefined8 *)0x0; puVar12 = (undefined8 *)*puVar12) {
    puVar10 = puVar16;
    if ((*(uint *)(puVar12 + 10) & 0x8000) != 0) {
                     /* Storage SI */
      *(undefined8 *)((longlong)puVar13 + 0x1c) = *(undefined8 *)((longlong)puVar12 + 0x54);
                     /* Storage ID */
      *(undefined8 *)((longlong)puVar13 + 4) = puVar12[1];
                     /* Unknown flag1 */
      *(undefined2 *)((longlong)puVar13 + 0xc) = *(undefined2 *)(puVar12 + 2);
```

Conclusion

Header - Reversing

- **Disk Block Size**
  - Insert iVar9 – 0x200 at Param_1 + 0x28
  - If you go up the variable, you can see puVar13
  - puVar13 is the end of the Disk Block pointer
    - Imply size of Disk Block is 0x80

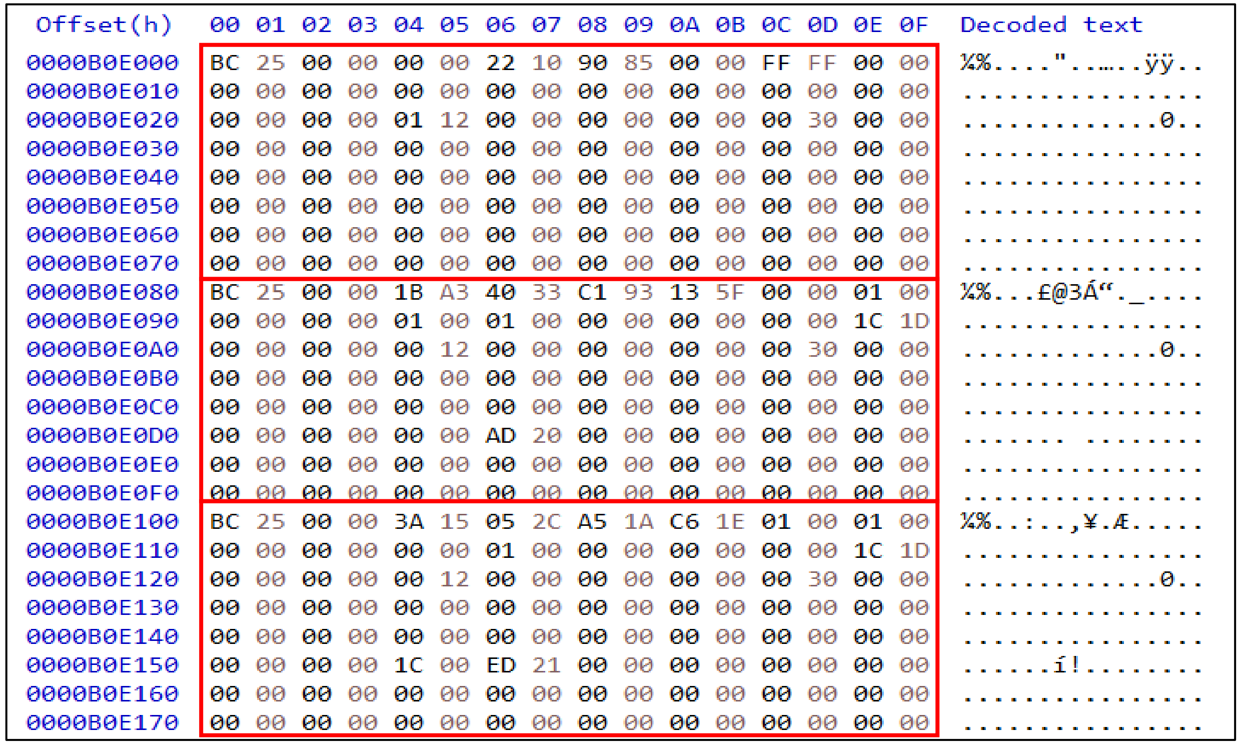


```
      *(undefined4 *)puVar13 = 0x25bc;
      *(undefined4 *)(puVar13 + 3) = 0;
      puVar13 = puVar16 + 0x10;
      puVar10 = puVar16 + 0x10;
      if (param_4 == 0) {
        puVar13 = puVar1;
        puVar10 = puVar16;
      }
    }
    puVar16 = puVar10;
  }
  iVar11 = (int)puVar13;
  if (uVar15 != 0) {
    *(undefined4 *)((longlong)param_1 + 0x24) = 0x200;
    iVar9 = iVar11 - (int)param_1;
    *(undefined4 *)((longlong)param_1 + 0x14) = 1;
    *(int *)(param_1 + 5) = iVar9 + -0x200;
    uVar8 = *(undefined8 *)(func_list + 0x88);
    *(undefined8 *)((longlong)param_1 + 0xc) = 0xe1e10000;
    *(undefined8 *)((longlong)param_1 + 0x1c) = uVar8;
    *(int *)(param_1 + 3) = param_2;
```

Conclusion

Disk

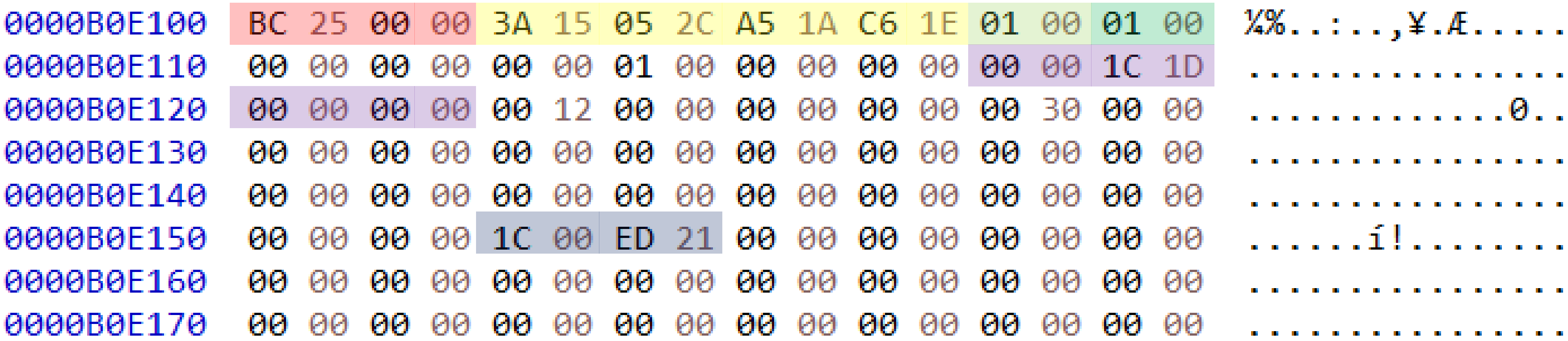
- **Information about recognized disks**
  - Size
  - Disk Type (HDD/SSD)
  - Port Type (SATA)
  - Port Speed
  - Firmware version



```
 Offset(h)   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   Decoded text
0000B0E000  BC 25 00 00 00 00 22 10 90 85 00 00 FF FF 00 00   ¼%...."..….ÿÿ..
0000B0E010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E020  00 00 00 00 01 12 00 00 00 00 00 00 00 30 00 00   .............0..
0000B0E030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E080  BC 25 00 00 1B A3 40 33 C1 93 13 5F 00 00 01 00   ¼%...£@3Á“._....
0000B0E090  00 00 00 00 01 00 01 00 00 00 00 00 00 00 1C 1D   ................
0000B0E0A0  00 00 00 00 00 12 00 00 00 00 00 00 00 30 00 00   .............0..
0000B0E0B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E0C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E0D0  00 00 00 00 00 00 AD 20 00 00 00 00 00 00 00 00   ....... ........
0000B0E0E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E0F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E100  BC 25 00 00 3A 15 05 2C A5 1A C6 1E 01 00 01 00   ¼%..:..,¥.Æ.....
0000B0E110  00 00 00 00 00 00 01 00 00 00 00 00 00 00 1C 1D   ................
0000B0E120  00 00 00 00 00 12 00 00 00 00 00 00 00 30 00 00   .............0..
0000B0E130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E150  00 00 00 00 1C 00 ED 21 00 00 00 00 00 00 00 00   ......í!........
0000B0E160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000B0E170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Conclusion

Disk



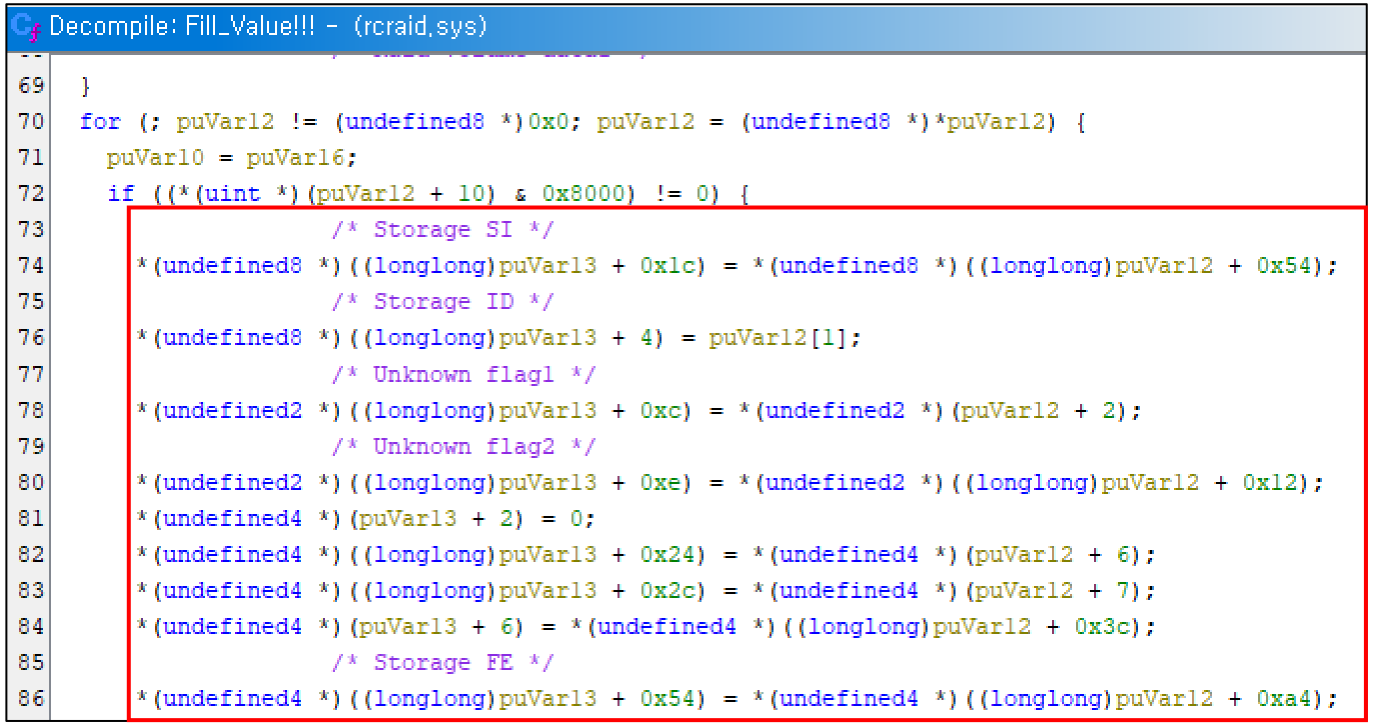
```
0000B0E100  BC 25 00 00 3A 15 05 2C A5 1A C6 1E 01 00 01 00  ¼%..:..,¥.Æ.....
0000B0E110  00 00 00 00 00 00 01 00 00 00 00 00 00 00 1C 1D  ................
0000B0E120  00 00 00 00 00 12 00 00 00 00 00 00 00 30 00 00  .............0..
0000B0E130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E150  00 00 00 00 1C 00 ED 21 00 00 00 00 00 00 00 00  ......í!........
0000B0E160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Field Name	Description/Value
Signature	Metadata checksum
Disk ID	Disk ID
?	?
?	?
FE(Feature)	?
SI(Capacity)	Disk capacity

Conclusion

Disk

- **Storage**
  - capacity
  - ID
  - Feature(FE)
  - Other flags

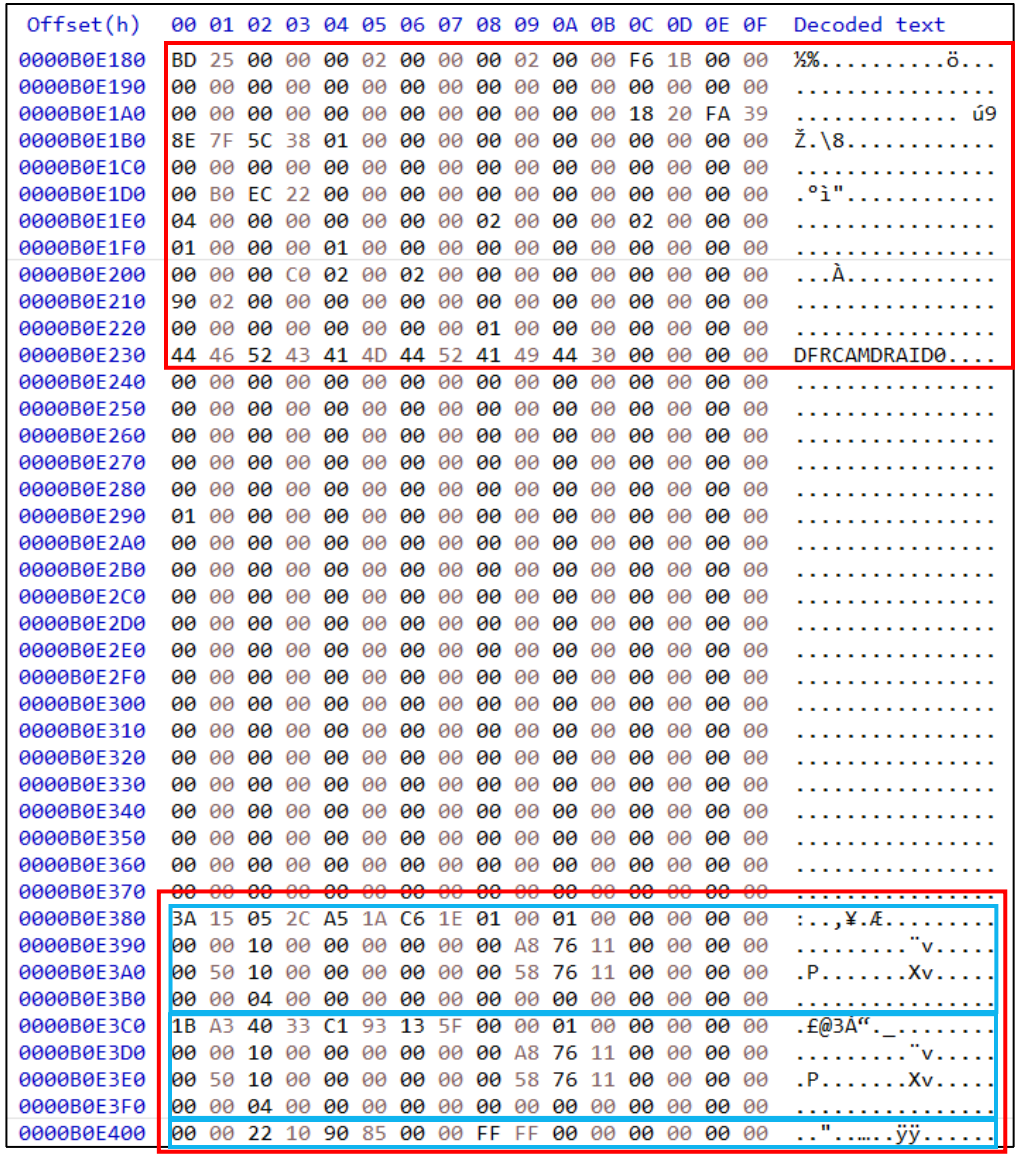


```
}
for (; puVar12 != (undefined8 *)0x0; puVar12 = (undefined8 *)*puVar12) {
  puVar10 = puVar16;
  if ((*(uint *)(puVar12 + 10) & 0x8000) != 0) {
                    /* Storage SI */
    *(undefined8 *)((longlong)puVar13 + 0x1c) = *(undefined8 *)((longlong)puVar12 + 0x54);
                    /* Storage ID */
    *(undefined8 *)((longlong)puVar13 + 4) = puVar12[1];
                    /* Unknown flag1 */
    *(undefined2 *)((longlong)puVar13 + 0xc) = *(undefined2 *)(puVar12 + 2);
                    /* Unknown flag2 */
    *(undefined2 *)((longlong)puVar13 + 0xe) = *(undefined2 *)((longlong)puVar12 + 0x12);
    *(undefined4 *)(puVar13 + 2) = 0;
    *(undefined4 *)((longlong)puVar13 + 0x24) = *(undefined4 *)(puVar12 + 6);
    *(undefined4 *)((longlong)puVar13 + 0x2c) = *(undefined4 *)(puVar12 + 7);
    *(undefined4 *)(puVar13 + 6) = *(undefined4 *)((longlong)puVar12 + 0x3c);
                    /* Storage FE */
    *(undefined4 *)((longlong)puVar13 + 0x54) = *(undefined4 *)((longlong)puVar12 + 0xa4);
```

Conclusion

Array

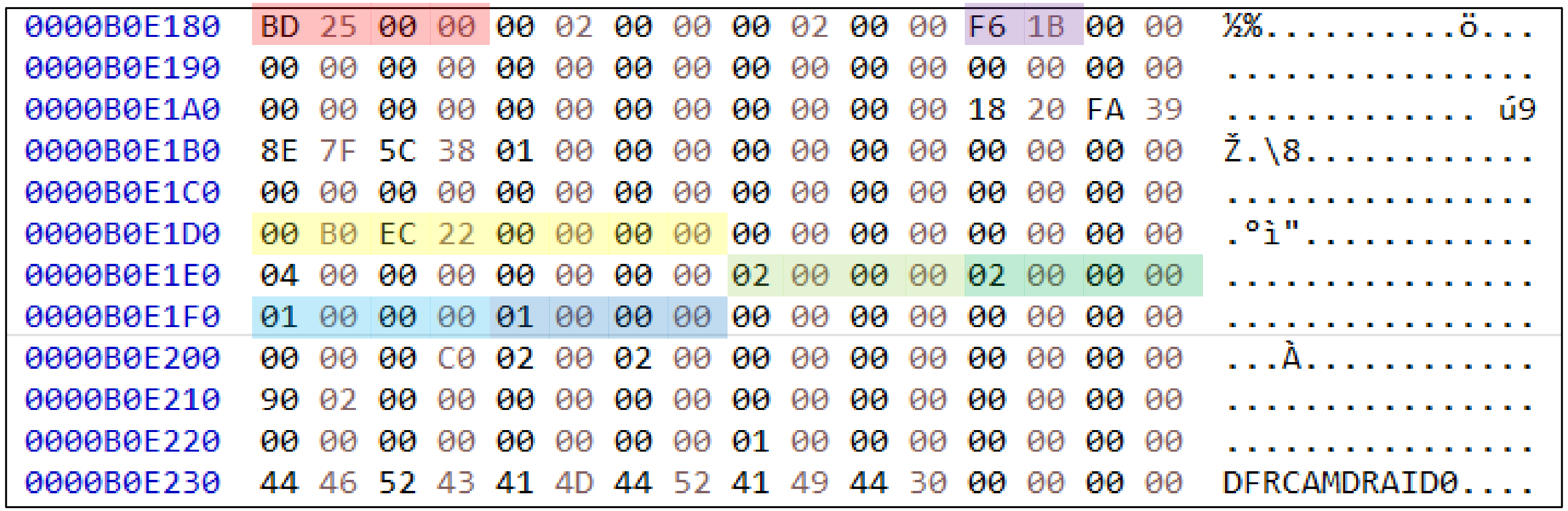
- **Array Metadata**
- **Disk Info**
  - Disk1
  - Disk2
  - Dummy



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
0000B0E180 BD 25 00 00 00 02 00 00 00 02 00 00 F6 1B 00 00  ½%..........ö...
0000B0E190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1A0 00 00 00 00 00 00 00 00 00 00 00 00 18 20 FA 39  ............. ú9
0000B0E1B0 8E 7F 5C 38 01 00 00 00 00 00 00 00 00 00 00 00  Ž.\8............
0000B0E1C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1D0 00 B0 EC 22 00 00 00 00 00 00 00 00 00 00 00 00  .°ì"............
0000B0E1E0 04 00 00 00 00 00 00 00 02 00 00 00 02 00 00 00  ................
0000B0E1F0 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E200 00 00 00 C0 02 00 02 00 00 00 00 00 00 00 00 00  ...À............
0000B0E210 90 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E220 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
0000B0E230 44 46 52 43 41 4D 44 52 41 49 44 30 00 00 00 00  DFRCAMDRAID0....
0000B0E240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E290 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E380 3A 15 05 2C A5 1A C6 1E 01 00 01 00 00 00 00 00  :..,¥.Æ.........
0000B0E390 00 00 10 00 00 00 00 00 00 A8 76 11 00 00 00 00  .........¨v.....
0000B0E3A0 00 50 10 00 00 00 00 00 00 58 76 11 00 00 00 00  .P.......Xv.....
0000B0E3B0 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E3C0 1B A3 40 33 C1 93 13 5F 00 00 01 00 00 00 00 00  .£@3Á“._........
0000B0E3D0 00 00 10 00 00 00 00 00 00 A8 76 11 00 00 00 00  .........¨v.....
0000B0E3E0 00 50 10 00 00 00 00 00 00 58 76 11 00 00 00 00  .P.......Xv.....
0000B0E3F0 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E400 00 00 22 10 90 85 00 00 FF FF 00 00 00 00 00 00  .."...….ÿÿ......
```

Conclusion

Array – Array Metadata

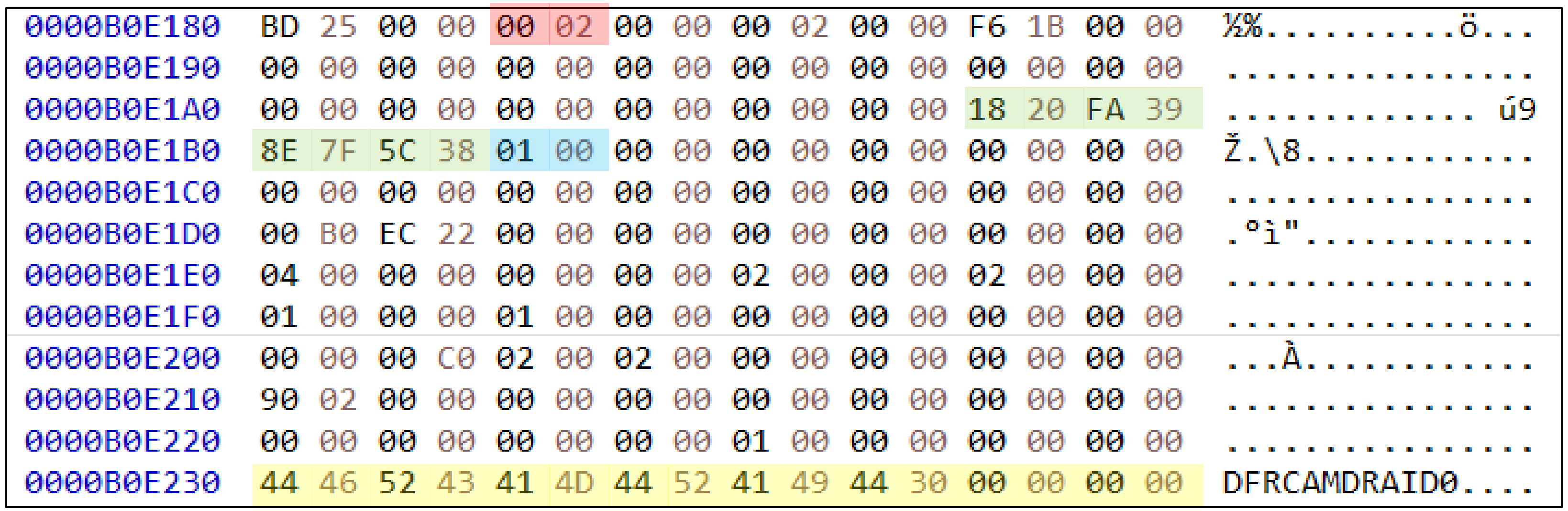


```
0000B0E180  BD 25 00 00 00 02 00 00 00 02 00 00 F6 1B 00 00  ½%..........ö...
0000B0E190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1A0  00 00 00 00 00 00 00 00 00 00 00 00 18 20 FA 39  ............. ú9
0000B0E1B0  8E 7F 5C 38 01 00 00 00 00 00 00 00 00 00 00 00  Ž.\8............
0000B0E1C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1D0  00 B0 EC 22 00 00 00 00 00 00 00 00 00 00 00 00  .°ì"............
0000B0E1E0  04 00 00 00 00 00 00 00 02 00 00 00 02 00 00 00  ................
0000B0E1F0  01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E200  00 00 00 C0 02 00 02 00 00 00 00 00 00 00 00 00  ...À............
0000B0E210  90 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E220  00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
0000B0E230  44 46 52 43 41 4D 44 52 41 49 44 30 00 00 00 00  DFRCAMDRAID0....
```

Field Name	Description/Value
Signature	Signature(0x25BD)
Array Size	Vdisk size
First Count x Second Count	First Count x Second Count
First Count	Support Disk(Number of storage devices)
Second Count	RAID Level
Dummy Count	Dummy Data Count
RAID Signature	RAID Signature

Conclusion

Array – Array Metadata



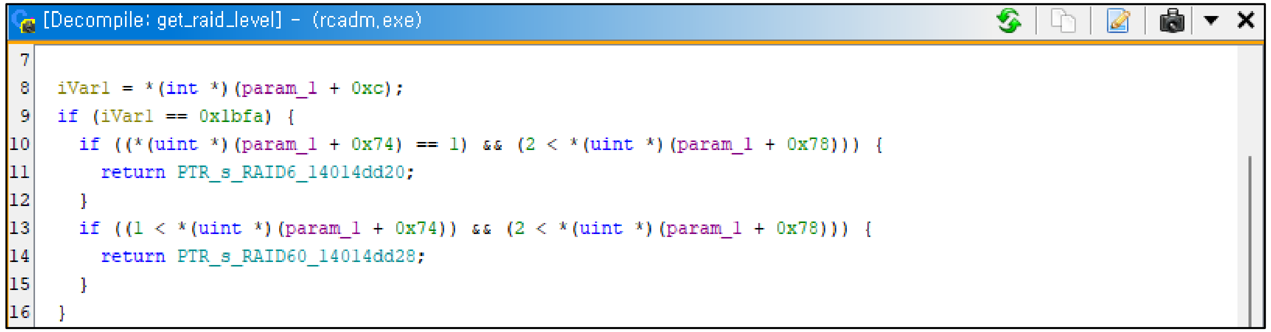
```
0000B0E180  BD 25 00 00 00 02 00 00 00 02 00 00 F6 1B 00 00  ½%..........ö...
0000B0E190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1A0  00 00 00 00 00 00 00 00 00 00 00 00 18 20 FA 39  ............. ú9
0000B0E1B0  8E 7F 5C 38 01 00 00 00 00 00 00 00 00 00 00 00  Ž.\8............
0000B0E1C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1D0  00 B0 EC 22 00 00 00 00 00 00 00 00 00 00 00 00  .°ì"............
0000B0E1E0  04 00 00 00 00 00 00 00 02 00 00 00 02 00 00 00  ................
0000B0E1F0  01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E200  00 00 00 C0 02 00 02 00 00 00 00 00 00 00 00 00  ...À............
0000B0E210  90 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E220  00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
0000B0E230  44 46 52 43 41 4D 44 52 41 49 44 30 00 00 00 00  DFRCAMDRAID0....
```

Field Name	Description/Value
Status	Status
Array Name	Array name
Array ID	Array ID
CTS(Cache tag size)	Strip size 1 = 32KB 2 = 64KB 3 = 128KB

Conclusion

Array – Disk info

- **RAID Level**
  - rcadm.exe: get_raid_level function



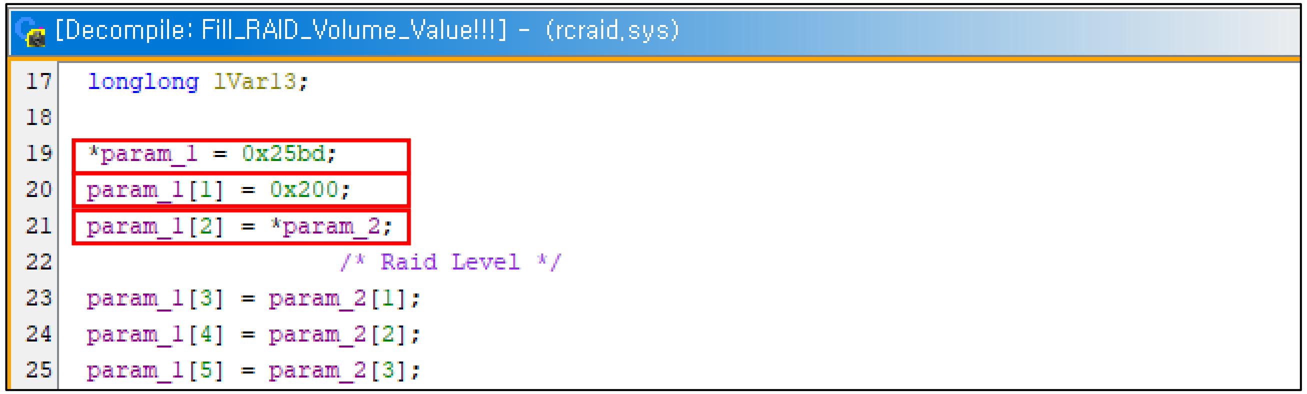
```

iVar1 = *(int *)(param_1 + 0xc);
if (iVar1 == 0x1bfa) {
  if ((*(uint *)(param_1 + 0x74) == 1) && (2 < *(uint *)(param_1 + 0x78))) {
    return PTR_s_RAID6_14014dd20;
  }
  if ((1 < *(uint *)(param_1 + 0x74)) && (2 < *(uint *)(param_1 + 0x78))) {
    return PTR_s_RAID60_14014dd28;
  }
}
```

RAID Signature	First Count	Second Count	RAID Level
0x1bfa	== 1	> 2	RAID 6
	> 1	> 2	RAID 60
0x1bf5	== 1	> 2	RAID 5
	> 1	> 2	RAID 50
0x1bf6	== 1	== 2	RAID 1
	== 1	> 2	RAID 1N
	> 1	== 1	RAID 0
	> 1	== 2	RAID 10
	> 1	> 2	RAID 10N
0x1bf7	"	"	Volume
0x1bf9	"	"	Legacy
0x1bfb	"	"	Raidable
0x1bfd	"	"	Raidtier
0x1bfc			Promise

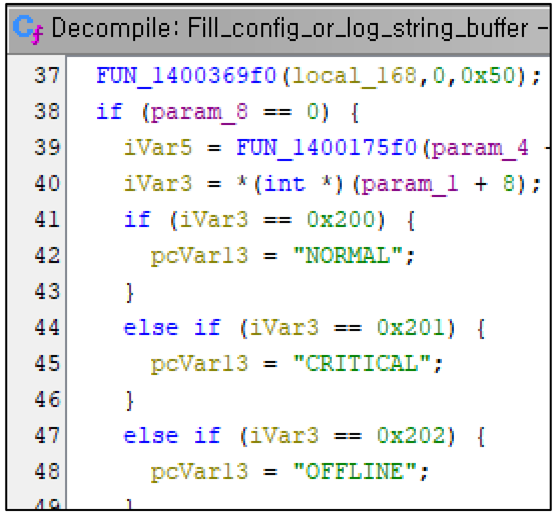
Conclusion

Array Metadata - Reversing

- **Fill_RAID_Volume_Value function**
  - Signature(0x25bd)
  - Status(0x200)
    - Rcadm.exe: Can check flag in Fill_config_or_log_string_buffer
    - 0x200 means Normal
  - RAID Signature(*param_2)



```
longlong lVar13;

*param_1 = 0x25bd;
param_1[1] = 0x200;
param_1[2] = *param_2;
                  /* Raid Level */
param_1[3] = param_2[1];
param_1[4] = param_2[2];
param_1[5] = param_2[3];
```

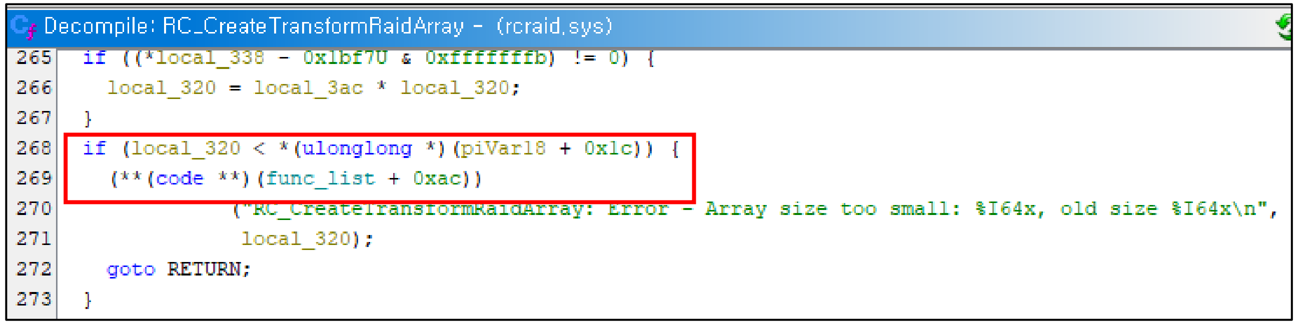


```
FUN_1400369f0(local_168,0,0x50);
if (param_8 == 0) {
  iVar5 = FUN_1400175f0(param_4
  iVar3 = *(int *)(param_1 + 8);
  if (iVar3 == 0x200) {
    pcVar13 = "NORMAL";
  }
  else if (iVar3 == 0x201) {
    pcVar13 = "CRITICAL";
  }
  else if (iVar3 == 0x202) {
    pcVar13 = "OFFLINE";
  }
```

Conclusion

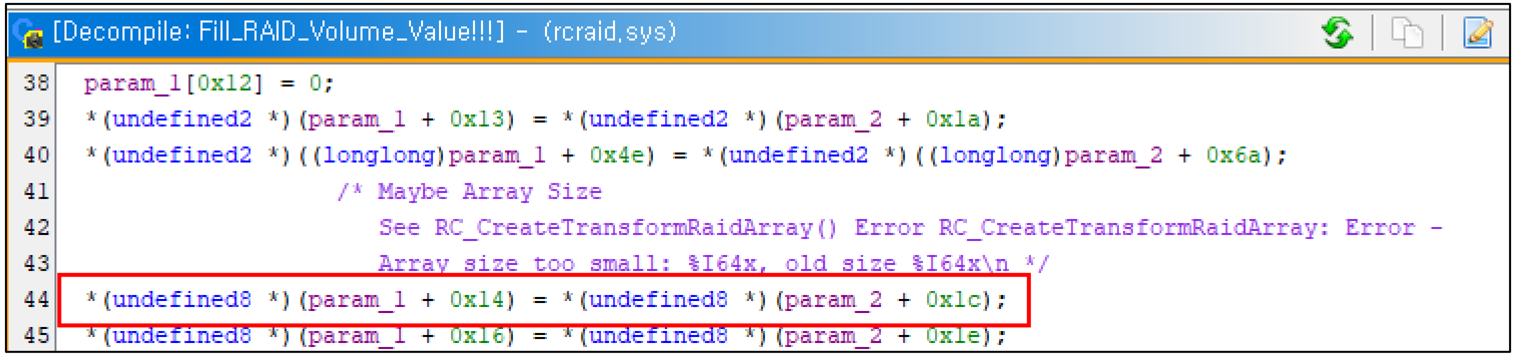
Array Metadata - Reversing

- **Fill_RAID_Volume_Value function**
  - Array Size
    - RC_CreateTransformRaidArray function
      - In the case of piVar18, it behaves as a memory struct
      - Offset 0x1c: Array size
      - Offset 0x14: value at 0x1c (type: uint64_t)



Decompile: RC_CreateTransformRaidArray - (rcraid.sys)

```
if ((*local_338 - 0x1bf70 & 0xfffffffb) != 0) {
  local_320 = local_3ac * local_320;
}
if (local_320 < *(ulonglong *)(piVar18 + 0x1c)) {
  (**(code **)(func_list + 0xac))
            ("RC_CreateTransformRaidArray: Error - Array size too small: %I64x, old size %I64x\n",
             local_320);
  goto RETURN;
}
```



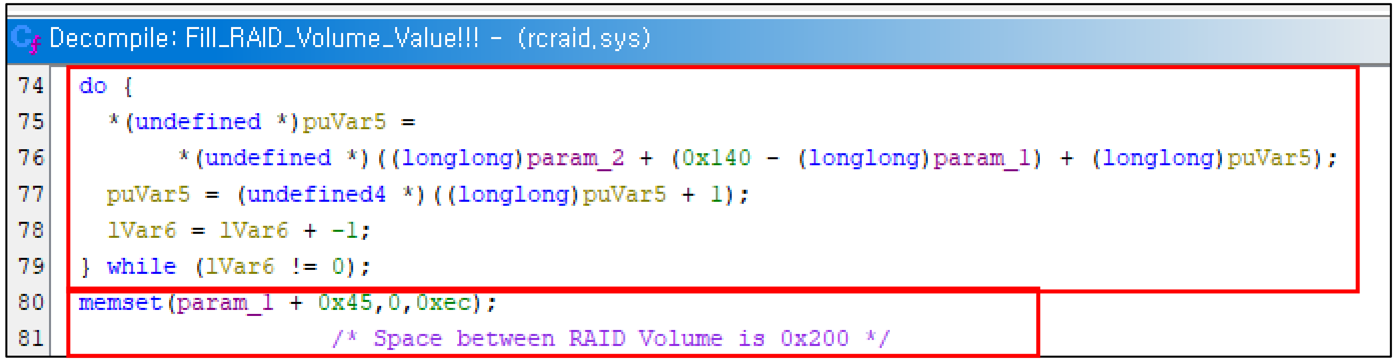
[Decompile: Fill_RAID_Volume_Value!!!] - (rcraid.sys)

```
param_1[0x12] = 0;
*(undefined2 *)(param_1 + 0x13) = *(undefined2 *)(param_2 + 0x1a);
*(undefined2 *)((longlong)param_1 + 0x4e) = *(undefined2 *)((longlong)param_2 + 0x6a);
                    /* Maybe Array Size
                       See RC_CreateTransformRaidArray() Error RC_CreateTransformRaidArray: Error -
                       Array size too small: %I64x, old size %I64x\n */
*(undefined8 *)(param_1 + 0x14) = *(undefined8 *)(param_2 + 0x1c);
*(undefined8 *)(param_1 + 0x16) = *(undefined8 *)(param_2 + 0x1e);
```

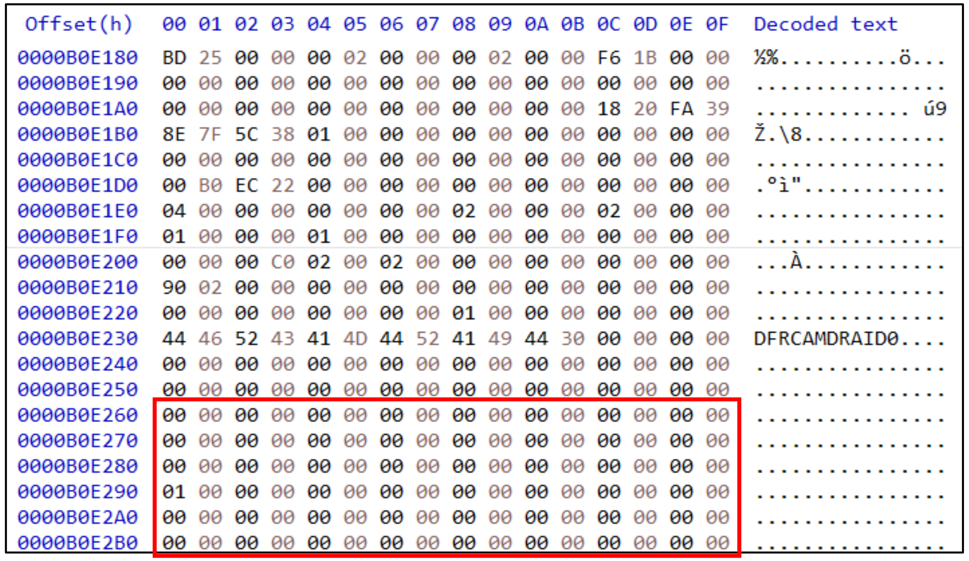
Conclusion

Array Metadata - Reversing

- **Fill_RAID_Volume_Value**
  - Array Name
    - Logic for copying strings can be checked
    - Maximum Length: 0x20
  - Array Padding
    - Implies that the array size is 0x200



```
do {
  *(undefined *)puVar5 =
       *(undefined *)((longlong)param_2 + (0x140 - (longlong)param_1) + (longlong)puVar5);
  puVar5 = (undefined4 *)((longlong)puVar5 + 1);
  lVar6 = lVar6 + -1;
} while (lVar6 != 0);
memset(param_1 + 0x45,0,0xec);
                    /* Space between RAID Volume is 0x200 */
```

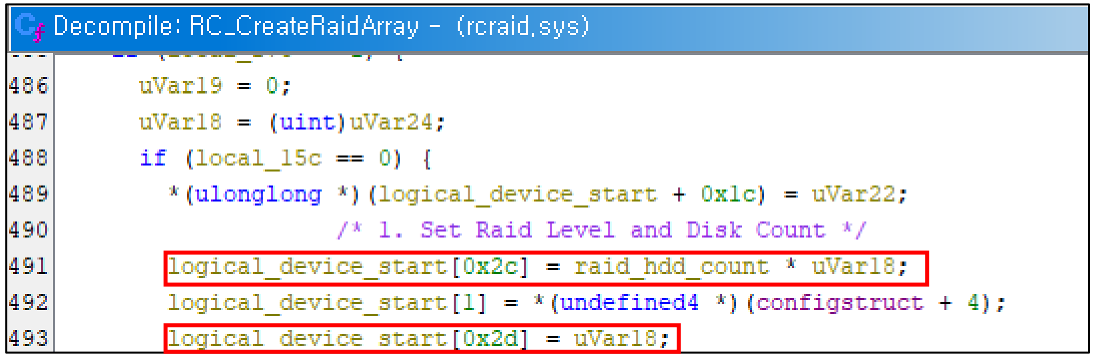


```
 Offset(h)   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
0000B0E180  BD 25 00 00 00 02 00 00 00 02 00 00 F6 1B 00 00  ½%..........ö...
0000B0E190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1A0  00 00 00 00 00 00 00 00 00 00 00 00 18 20 FA 39  ............. ú9
0000B0E1B0  8E 7F 5C 38 01 00 00 00 00 00 00 00 00 00 00 00  Ž.\8............
0000B0E1C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1D0  00 B0 EC 22 00 00 00 00 00 00 00 00 00 00 00 00  .°ì"............
0000B0E1E0  04 00 00 00 00 00 00 00 02 00 00 00 02 00 00 00  ................
0000B0E1F0  01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E200  00 00 00 C0 02 00 02 00 00 00 00 00 00 00 00 00  ...À............
0000B0E210  90 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E220  00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
0000B0E230  44 46 52 43 41 4D 44 52 41 49 44 30 00 00 00 00  DFRCAMDRAID0....
0000B0E240  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E250  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E260  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E270  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E280  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E290  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E2B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

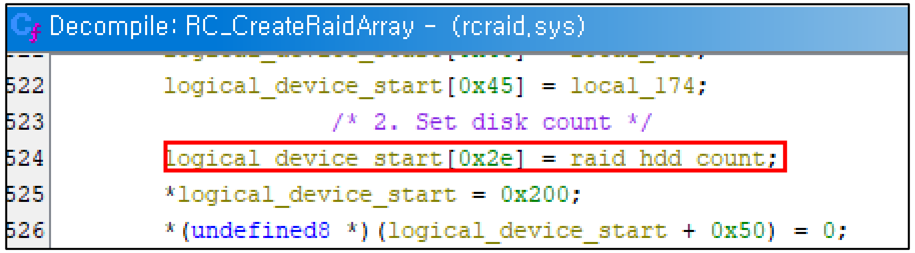
Conclusion

Array Metadata - Reversing

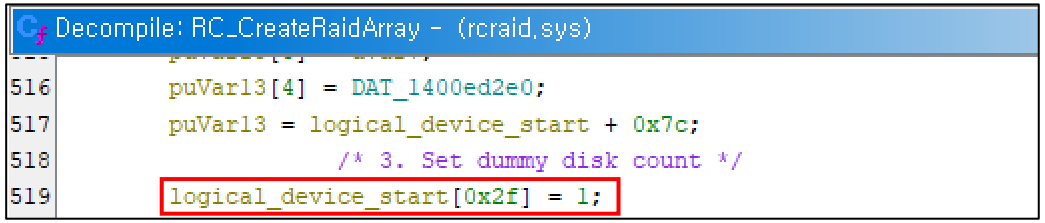
- **Fill_RAID_Volume_Value**
  - First Count
  - Second Count
  - Dummy Disk Count



```
    uVar19 = 0;
    uVar18 = (uint)uVar24;
    if (local_15c == 0) {
      *(ulonglong *)(logical_device_start + 0x1c) = uVar22;
                    /* 1. Set Raid Level and Disk Count */
      logical_device_start[0x2c] = raid_hdd_count * uVar18;
      logical_device_start[1] = *(undefined4 *)(configstruct + 4);
      logical_device_start[0x2d] = uVar18;
```



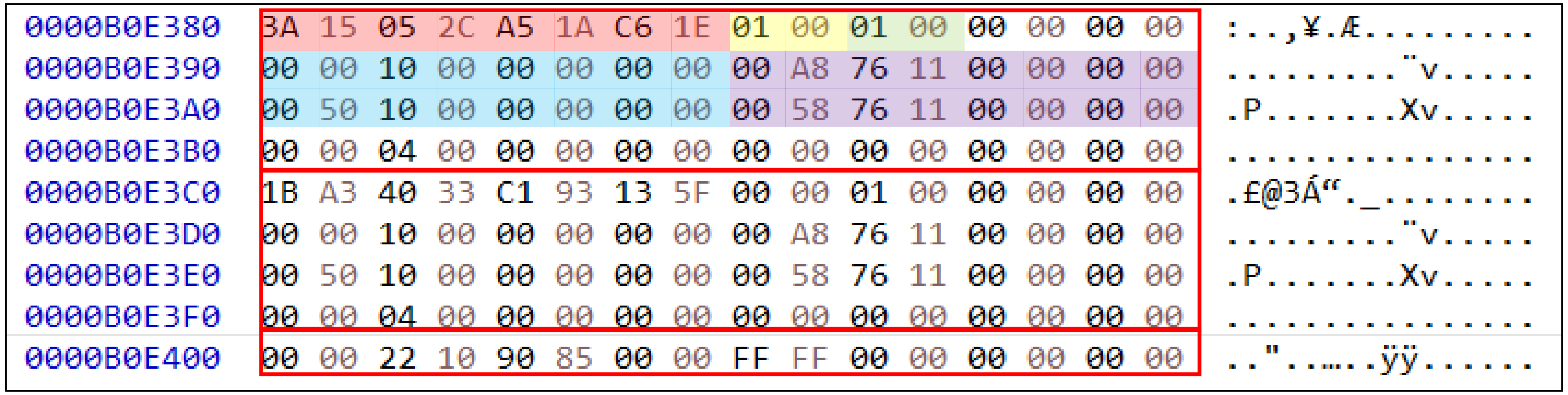
```
      logical_device_start[0x45] = local_174;
                    /* 2. Set disk count */
      logical_device_start[0x2e] = raid_hdd_count;
      *logical_device_start = 0x200;
      *(undefined8 *)(logical_device_start + 0x50) = 0;
```



```
      puVar13[4] = DAT_1400ed2e0;
      puVar13 = logical_device_start + 0x7c;
                    /* 3. Set dummy disk count */
      logical_device_start[0x2f] = 1;
```

Conclusion

Array – Disk info



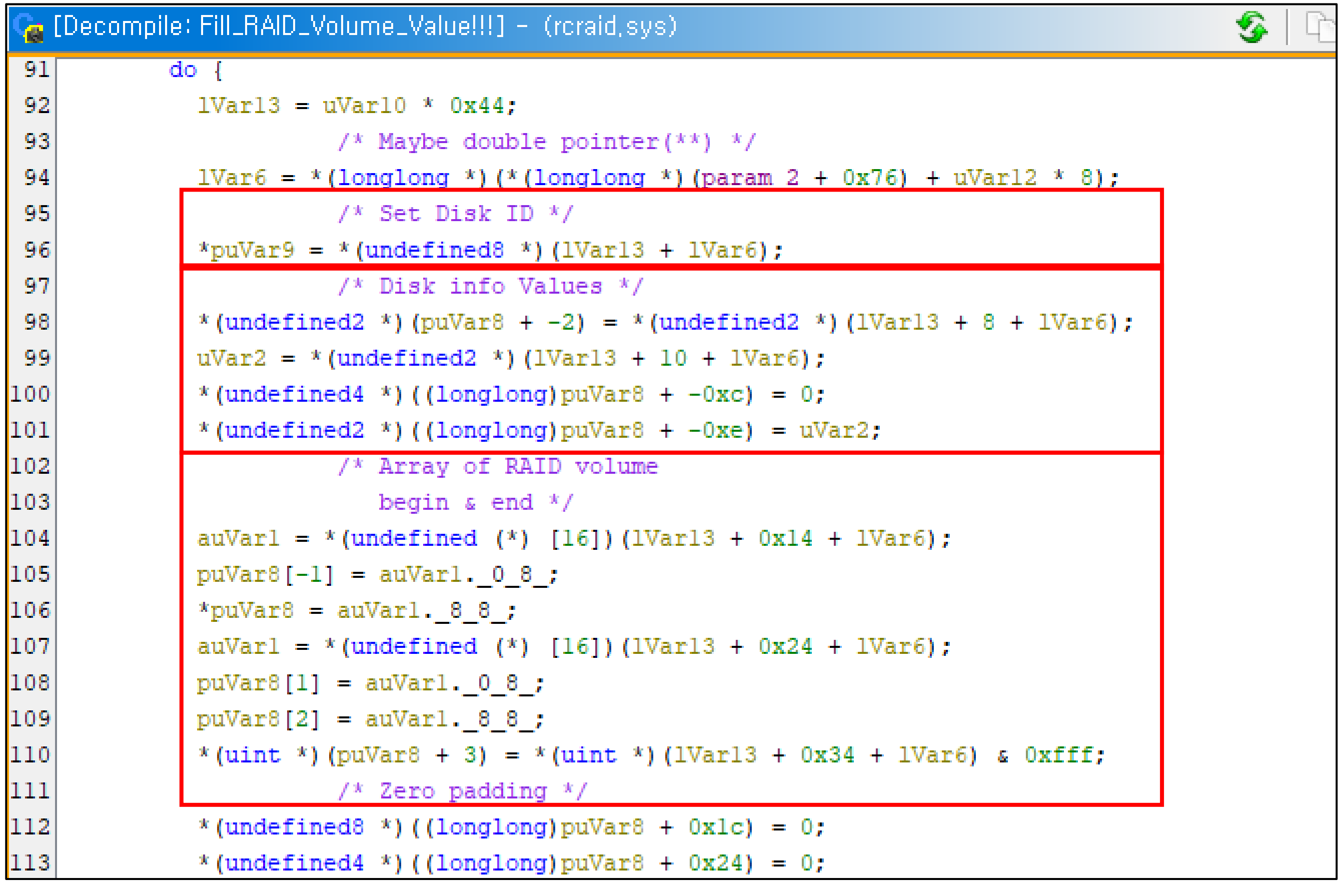
```
0000B0E380  3A 15 05 2C A5 1A C6 1E 01 00 01 00 00 00 00 00  :..,¥.Æ.........
0000B0E390  00 00 10 00 00 00 00 00 00 A8 76 11 00 00 00 00  .........¨v.....
0000B0E3A0  00 50 10 00 00 00 00 00 00 58 76 11 00 00 00 00  .P.......Xv.....
0000B0E3B0  00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E3C0  1B A3 40 33 C1 93 13 5F 00 00 01 00 00 00 00 00  .£@3Á“._........
0000B0E3D0  00 00 10 00 00 00 00 00 00 A8 76 11 00 00 00 00  .........¨v.....
0000B0E3E0  00 50 10 00 00 00 00 00 00 58 76 11 00 00 00 00  .P.......Xv.....
0000B0E3F0  00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E400  00 00 22 10 90 85 00 00 FF FF 00 00 00 00 00 00  .."..…..ÿÿ......
```

Field Name	Description/Value
ID	ID of the disk used for the RAID configuration
HD(DeviceRoute)	Order of RAID Configurations
RT(CoreRoute)	?
Begin	Array Start Address
End	Array End Address

Conclusion

Array Metadata - Reversing

- **Fill_RAID_Volume_Value**
    - ID
    - Flags
    - Array begin & end

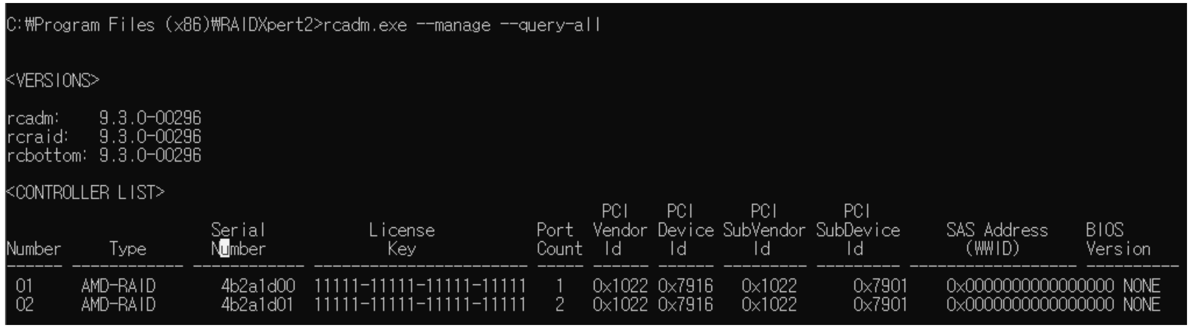


```
        do {
          lVar13 = uVar10 * 0x44;
                    /* Maybe double pointer(**) */
          lVar6 = *(longlong *)(*(longlong *)(param_2 + 0x76) + uVar12 * 8);
                    /* Set Disk ID */
          *puVar9 = *(undefined8 *)(lVar13 + lVar6);
                    /* Disk info Values */
          *(undefined2 *)(puVar8 + -2) = *(undefined2 *)(lVar13 + 8 + lVar6);
          uVar2 = *(undefined2 *)(lVar13 + 10 + lVar6);
          *(undefined4 *)((longlong)puVar8 + -0xc) = 0;
          *(undefined2 *)((longlong)puVar8 + -0xe) = uVar2;
                    /* Array of RAID volume
                       begin & end */
          auVar1 = *(undefined (*) [16])(lVar13 + 0x14 + lVar6);
          puVar8[-1] = auVar1._0_8_;
          *puVar8 = auVar1._8_8_;
          auVar1 = *(undefined (*) [16])(lVar13 + 0x24 + lVar6);
          puVar8[1] = auVar1._0_8_;
          puVar8[2] = auVar1._8_8_;
          *(uint *)(puVar8 + 3) = *(uint *)(lVar13 + 0x34 + lVar6) & 0xfff;
                    /* Zero padding */
          *(undefined8 *)((longlong)puVar8 + 0x1c) = 0;
          *(undefined4 *)((longlong)puVar8 + 0x24) = 0;
```

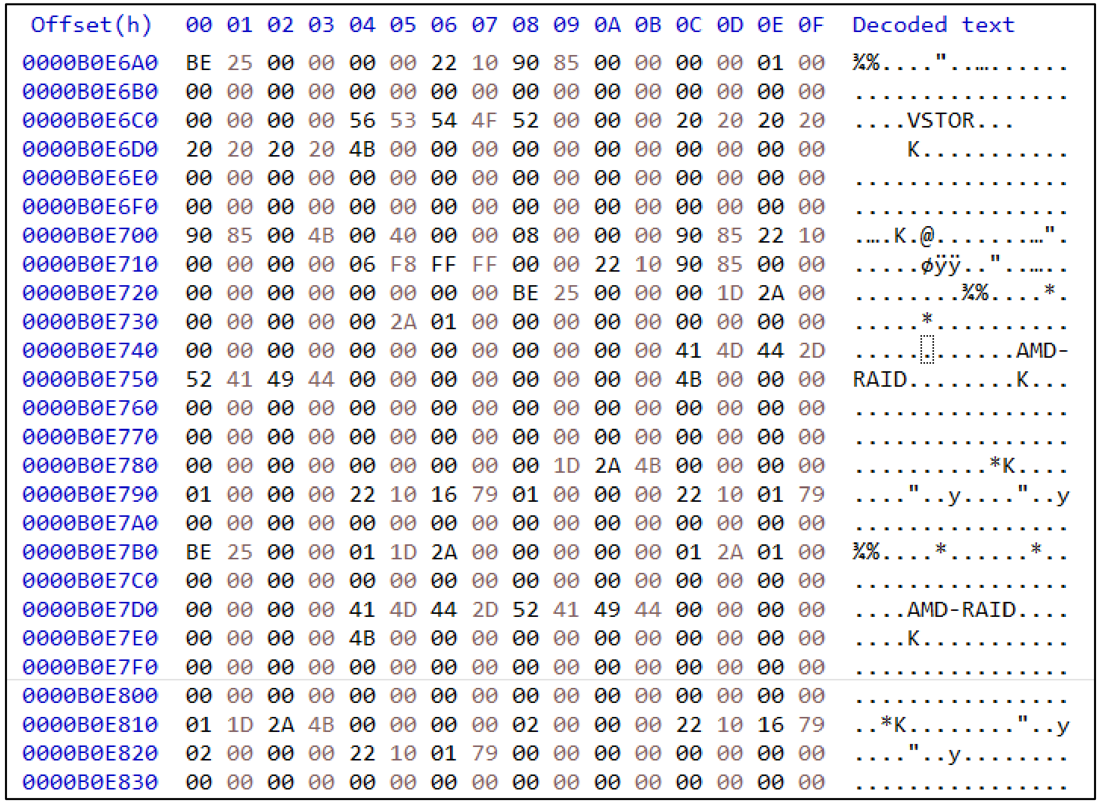
Conclusion

Controller

- **About RAID Controllers**
- **No special data found**
- **Same in query results**
  - Type, Serial Number, License
  - Port Count, PCI Vender Id, etc



```
C:₩Program Files (x86)₩RAIDXpert2>rcadm.exe --manage --query-all


<VERSIONS>

rcadm:    9.3.0-00296
rcraid:   9.3.0-00296
rcbottom: 9.3.0-00296

<CONTROLLER LIST>
                                                             PCI    PCI      PCI      PCI
                        Serial           License        Port Vendor Device SubVendor SubDevice      SAS Address      BIOS
Number      Type        Number             Key          Count  Id     Id       Id       Id            (WWID)          Version
------ ------------ ------------ ----------------------- ----- ------ ------ --------- --------- ------------------ ----------
 01     AMD-RAID       4b2a1d00 11111-11111-11111-11111   1    0x1022 0x7916   0x1022      0x7901   0x0000000000000000 NONE
 02     AMD-RAID       4b2a1d01 11111-11111-11111-11111   2    0x1022 0x7916   0x1022      0x7901   0x0000000000000000 NONE
```

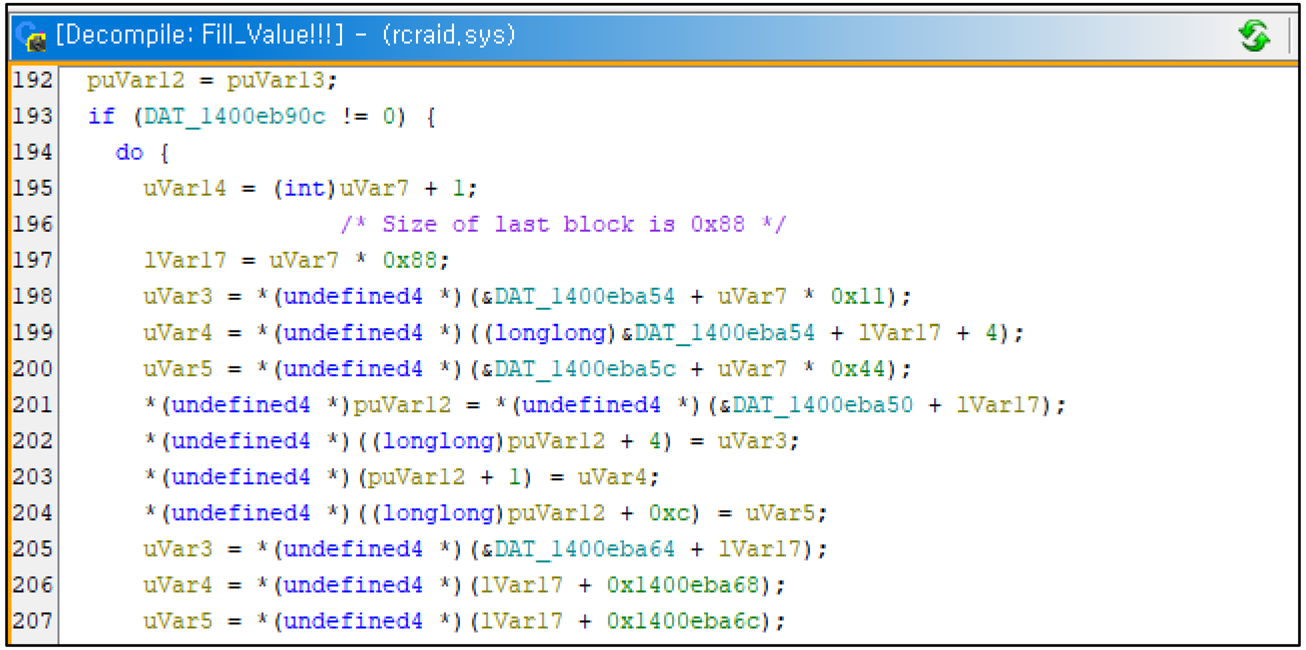


```
 Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
0000B0E6A0  BE 25 00 00 00 00 22 10 90 85 00 00 00 00 01 00  ¾%...."..…......
0000B0E6B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E6C0  00 00 00 00 56 53 54 4F 52 00 00 00 20 20 20 20  ....VSTOR...    
0000B0E6D0  20 20 20 20 4B 00 00 00 00 00 00 00 00 00 00 00      K...........
0000B0E6E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E6F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E700  90 85 00 4B 00 40 00 00 08 00 00 00 90 85 22 10  .….K.@.......…".
0000B0E710  00 00 00 00 06 F8 FF FF 00 00 22 10 90 85 00 00  .....øÿÿ.."..…..
0000B0E720  00 00 00 00 00 00 00 00 BE 25 00 00 00 1D 2A 00  ........¾%....*.
0000B0E730  00 00 00 00 00 2A 01 00 00 00 00 00 00 00 00 00  .....*..........
0000B0E740  00 00 00 00 00 00 00 00 00 00 00 00 41 4D 44 2D  ............AMD-
0000B0E750  52 41 49 44 00 00 00 00 00 00 00 00 4B 00 00 00  RAID........K...
0000B0E760  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E770  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E780  00 00 00 00 00 00 00 00 00 1D 2A 4B 00 00 00 00  ..........*K....
0000B0E790  01 00 00 00 22 10 16 79 01 00 00 00 22 10 01 79  ...."..y...."..y
0000B0E7A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E7B0  BE 25 00 00 01 1D 2A 00 00 00 00 00 01 2A 01 00  ¾%....*......*..
0000B0E7C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E7D0  00 00 00 00 41 4D 44 2D 52 41 49 44 00 00 00 00  ....AMD-RAID....
0000B0E7E0  00 00 00 00 4B 00 00 00 00 00 00 00 00 00 00 00  ....K...........
0000B0E7F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E800  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E810  01 1D 2A 4B 00 00 00 00 02 00 00 00 22 10 16 79  ..*K........"..y
0000B0E820  02 00 00 00 22 10 01 79 00 00 00 00 00 00 00 00  ...."..y........
0000B0E830  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Conclusion

Controller Metadata - Reversing

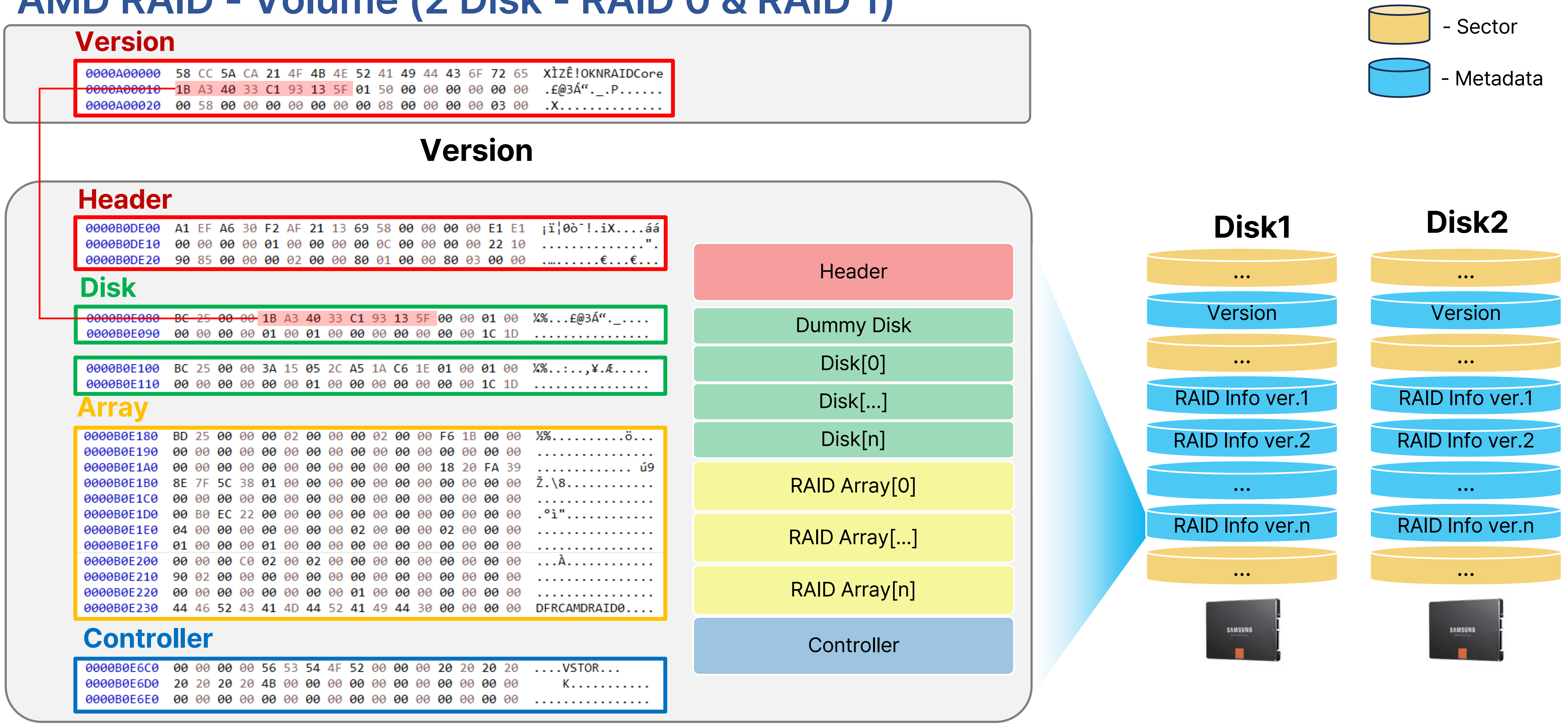
- **Fill_Value**



```
  puVar12 = puVar13;
  if (DAT_1400eb90c != 0) {
    do {
      uVar14 = (int)uVar7 + 1;
                    /* Size of last block is 0x88 */
      lVar17 = uVar7 * 0x88;
      uVar3 = *(undefined4 *)(&DAT_1400eba54 + uVar7 * 0x11);
      uVar4 = *(undefined4 *)((longlong)&DAT_1400eba54 + lVar17 + 4);
      uVar5 = *(undefined4 *)(&DAT_1400eba5c + uVar7 * 0x44);
      *(undefined4 *)puVar12 = *(undefined4 *)(&DAT_1400eba50 + lVar17);
      *(undefined4 *)((longlong)puVar12 + 4) = uVar3;
      *(undefined4 *)(puVar12 + 1) = uVar4;
      *(undefined4 *)((longlong)puVar12 + 0xc) = uVar5;
      uVar3 = *(undefined4 *)(&DAT_1400eba64 + lVar17);
      uVar4 = *(undefined4 *)(lVar17 + 0x1400eba68);
      uVar5 = *(undefined4 *)(lVar17 + 0x1400eba6c);
```

Conclusion

AMD RAID - Volume (2 Disk - RAID 0 & RAID 1)



Version

```
0000A00000  58 CC 5A CA 21 4F 4B 4E 52 41 49 44 43 6F 72 65  XÌZÊ!OKNRAIDCore
0000A00010  1B A3 40 33 C1 93 13 5F 01 50 00 00 00 00 00 00  .£@3Á“._.P......
0000A00020  00 58 00 00 00 00 00 00 00 08 00 00 00 00 03 00  .X..............
```

Version

Header

```
0000B0DE00  A1 EF A6 30 F2 AF 21 13 69 58 00 00 00 00 E1 E1  ¡ï¦0ò¯!.iX....áá
0000B0DE10  00 00 00 00 01 00 00 00 00 0C 00 00 00 00 22 10  ..............".
0000B0DE20  90 85 00 00 00 02 00 00 80 01 00 00 80 03 00 00  .…......€...€...
```

Disk

```
0000B0E080  BC 25 00 00 1B A3 40 33 C1 93 13 5F 00 00 01 00  ¼%...£@3Á“._....
0000B0E090  00 00 00 00 01 00 01 00 00 00 00 00 00 00 1C 1D  ................
```

```
0000B0E100  BC 25 00 00 3A 15 05 2C A5 1A C6 1E 01 00 01 00  ¼%..:..,¥.Æ.....
0000B0E110  00 00 00 00 00 00 01 00 00 00 00 00 00 00 1C 1D  ................
```

Array

```
0000B0E180  BD 25 00 00 00 02 00 00 00 02 00 00 F6 1B 00 00  ½%..........ö...
0000B0E190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1A0  00 00 00 00 00 00 00 00 00 00 00 00 18 20 FA 39  ............. ú9
0000B0E1B0  8E 7F 5C 38 01 00 00 00 00 00 00 00 00 00 00 00  Ž.\8............
0000B0E1C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E1D0  00 B0 EC 22 00 00 00 00 00 00 00 00 00 00 00 00  .°ì"............
0000B0E1E0  04 00 00 00 00 00 00 00 02 00 00 00 02 00 00 00  ................
0000B0E1F0  01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E200  00 00 00 C0 02 00 02 00 00 00 00 00 00 00 00 00  ...À............
0000B0E210  90 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000B0E220  00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
0000B0E230  44 46 52 43 41 4D 44 52 41 49 44 30 00 00 00 00  DFRCAMDRAID0....
```

Controller

```
0000B0E6C0  00 00 00 00 56 53 54 4F 52 00 00 00 20 20 20 20  ....VSTOR...
0000B0E6D0  20 20 20 20 4B 00 00 00 00 00 00 00 00 00 00 00      K...........
0000B0E6E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

RAID Info Metadata

Tips for deleted volume recovery

How to guess stripe size

Guessing

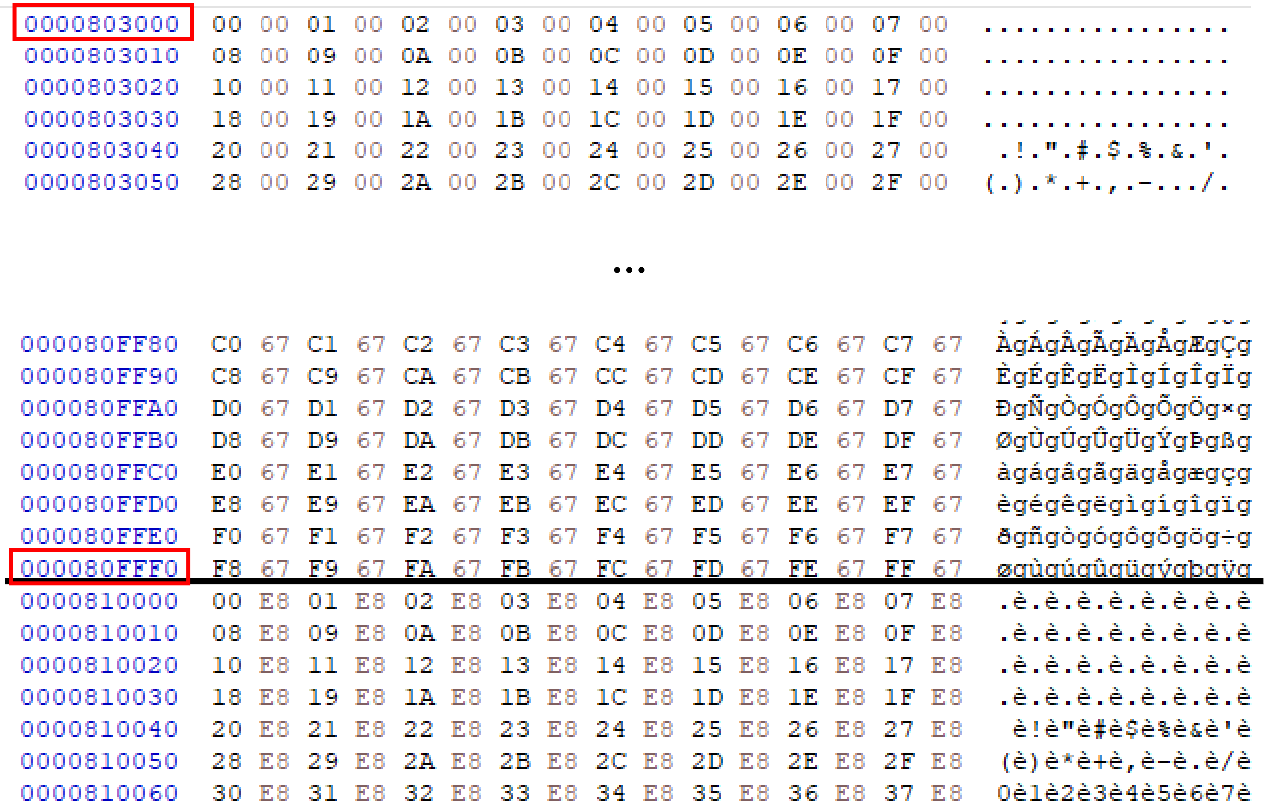
Method

- **Typical value of stripe size**
  - Fixed size: 8KB, 16KB, 32KB, 64KB … (2^nK)
  - Can be guessed by referring file system
    - ex) $Upcase file in NTFS file system

Guessing

Case study: NTFS

- $Upcase
  - **128KB** file full of capital letters
  - Guess stripe size by
    - Start offset
    - End offset

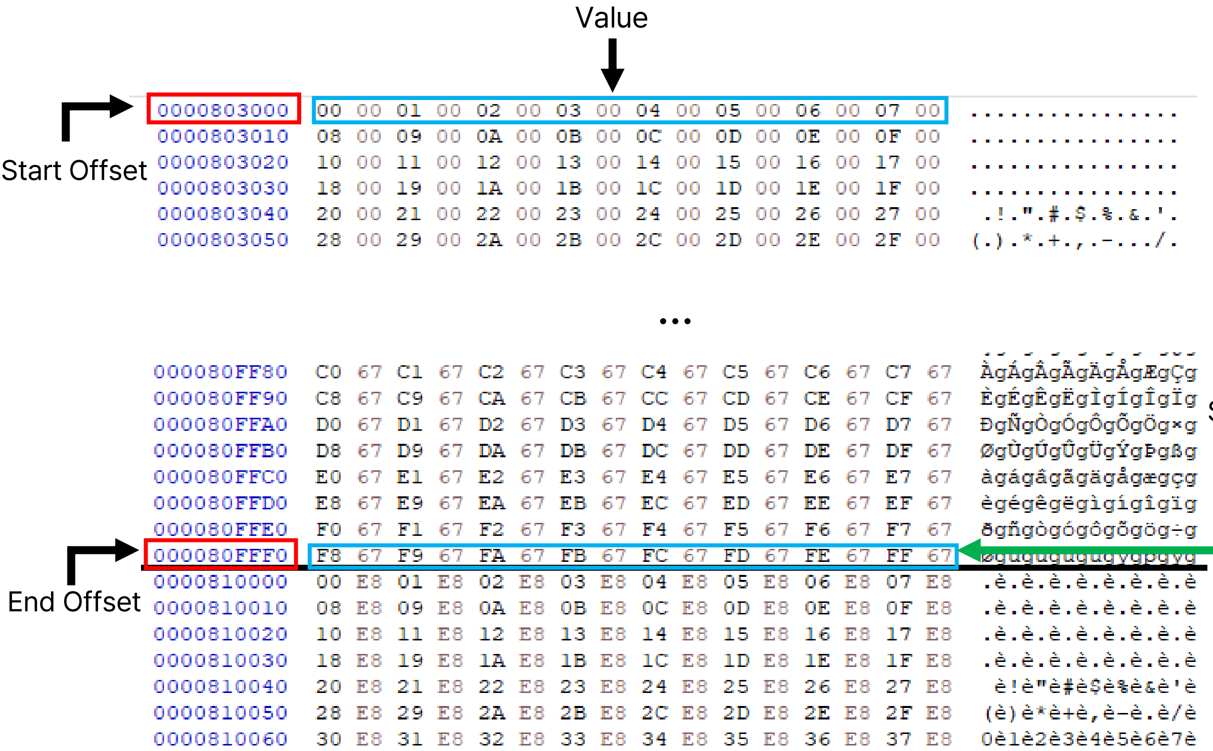


```
0000803000  00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00  ................
0000803010  08 00 09 00 0A 00 0B 00 0C 00 0D 00 0E 00 0F 00  ................
0000803020  10 00 11 00 12 00 13 00 14 00 15 00 16 00 17 00  ................
0000803030  18 00 19 00 1A 00 1B 00 1C 00 1D 00 1E 00 1F 00  ................
0000803040  20 00 21 00 22 00 23 00 24 00 25 00 26 00 27 00   .!.".#.$.%.&.'.
0000803050  28 00 29 00 2A 00 2B 00 2C 00 2D 00 2E 00 2F 00  (.).*.+.,.-.../.

...

000080FF80  C0 67 C1 67 C2 67 C3 67 C4 67 C5 67 C6 67 C7 67  ÀgÁgÂgÃgÄgÅgÆgÇg
000080FF90  C8 67 C9 67 CA 67 CB 67 CC 67 CD 67 CE 67 CF 67  ÈgÉgÊgËgÌgÍgÎgÏg
000080FFA0  D0 67 D1 67 D2 67 D3 67 D4 67 D5 67 D6 67 D7 67  ÐgÑgÒgÓgÔgÕgÖg×g
000080FFB0  D8 67 D9 67 DA 67 DB 67 DC 67 DD 67 DE 67 DF 67  ØgÙgÚgÛgÜgÝgÞgßg
000080FFC0  E0 67 E1 67 E2 67 E3 67 E4 67 E5 67 E6 67 E7 67  àgágâgãgägågægçg
000080FFD0  E8 67 E9 67 EA 67 EB 67 EC 67 ED 67 EE 67 EF 67  ègégêgëgìgígîgïg
000080FFE0  F0 67 F1 67 F2 67 F3 67 F4 67 F5 67 F6 67 F7 67  ðgñgògógôgõgögg÷g
000080FFF0  F8 67 F9 67 FA 67 FB 67 FC 67 FD 67 FE 67 FF 67  øgùgúgûgügýgþgÿg
0000810000  00 E8 01 E8 02 E8 03 E8 04 E8 05 E8 06 E8 07 E8  .è.è.è.è.è.è.è.è
0000810010  08 E8 09 E8 0A E8 0B E8 0C E8 0D E8 0E E8 0F E8  .è.è.è.è.è.è.è.è
0000810020  10 E8 11 E8 12 E8 13 E8 14 E8 15 E8 16 E8 17 E8  .è.è.è.è.è.è.è.è
0000810030  18 E8 19 E8 1A E8 1B E8 1C E8 1D E8 1E E8 1F E8  .è.è.è.è.è.è.è.è
0000810040  20 E8 21 E8 22 E8 23 E8 24 E8 25 E8 26 E8 27 E8   è!è"è#è$è%è&è'è
0000810050  28 E8 29 E8 2A E8 2B E8 2C E8 2D E8 2E E8 2F E8  (è)è*è+è,è-è.è/è
0000810060  30 E8 31 E8 32 E8 33 E8 34 E8 35 E8 36 E8 37 E8  0è1è2è3è4è5è6è7è
```

Guessing

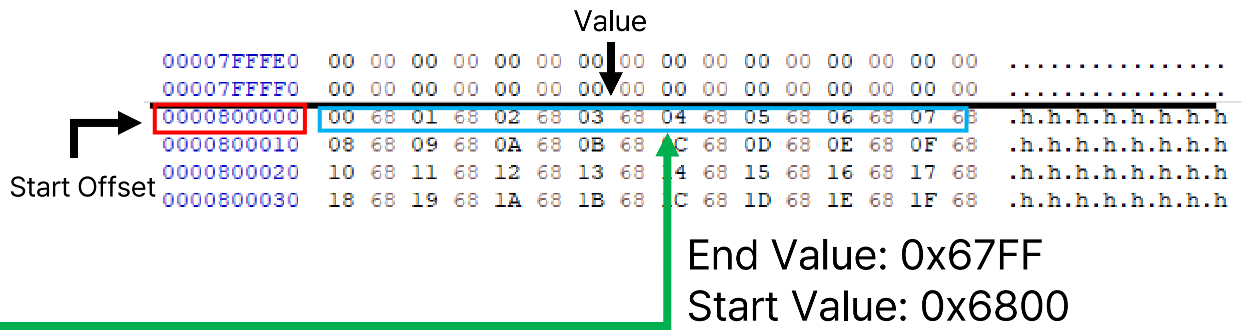
Case Study: NTFS

- **Disk1**



```
0000803000  00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00  ................
0000803010  08 00 09 00 0A 00 0B 00 0C 00 0D 00 0E 00 0F 00  ................
0000803020  10 00 11 00 12 00 13 00 14 00 15 00 16 00 17 00  ................
0000803030  18 00 19 00 1A 00 1B 00 1C 00 1D 00 1E 00 1F 00  ................
0000803040  20 00 21 00 22 00 23 00 24 00 25 00 26 00 27 00   .!.".#.$.%.&.'.
0000803050  28 00 29 00 2A 00 2B 00 2C 00 2D 00 2E 00 2F 00  (.).*.+.,.-.../.

...

000080FF80  C0 67 C1 67 C2 67 C3 67 C4 67 C5 67 C6 67 C7 67  ÀgÁgÂgÃgÄgÅgÆgÇg
000080FF90  C8 67 C9 67 CA 67 CB 67 CC 67 CD 67 CE 67 CF 67  ÈgÉgÊgËgÌgÍgÎgÏg
000080FFA0  D0 67 D1 67 D2 67 D3 67 D4 67 D5 67 D6 67 D7 67  ÐgÑgÒgÓgÔgÕgÖg×g
000080FFB0  D8 67 D9 67 DA 67 DB 67 DC 67 DD 67 DE 67 DF 67  ØgÙgÚgÛgÜgÝgÞgßg
000080FFC0  E0 67 E1 67 E2 67 E3 67 E4 67 E5 67 E6 67 E7 67  àgágâgãgägågægçg
000080FFD0  E8 67 E9 67 EA 67 EB 67 EC 67 ED 67 EE 67 EF 67  ègégêgëgìgígîgïg
000080FFE0  F0 67 F1 67 F2 67 F3 67 F4 67 F5 67 F6 67 F7 67  ðgñgògógôgõgögö÷g
000080FFF0  F8 67 F9 67 FA 67 FB 67 FC 67 FD 67 FE 67 FF 67  øgùgúgûgügýgþgÿg
0000810000  00 E8 01 E8 02 E8 03 E8 04 E8 05 E8 06 E8 07 E8  .è.è.è.è.è.è.è.è
0000810010  08 E8 09 E8 0A E8 0B E8 0C E8 0D E8 0E E8 0F E8  .è.è.è.è.è.è.è.è
0000810020  10 E8 11 E8 12 E8 13 E8 14 E8 15 E8 16 E8 17 E8  .è.è.è.è.è.è.è.è
0000810030  18 E8 19 E8 1A E8 1B E8 1C E8 1D E8 1E E8 1F E8  .è.è.è.è.è.è.è.è
0000810040  20 E8 21 E8 22 E8 23 E8 24 E8 25 E8 26 E8 27 E8   è!è"è#è$è%è&è'è
0000810050  28 E8 29 E8 2A E8 2B E8 2C E8 2D E8 2E E8 2F E8  (è)è*è+è,è-è.è/è
0000810060  30 E8 31 E8 32 E8 33 E8 34 E8 35 E8 36 E8 37 E8  0è1è2è3è4è5è6è7è
```

- **Disk2**



```
00007FFFE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00007FFFF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000800000  00 68 01 68 02 68 03 68 04 68 05 68 06 68 07 68  .h.h.h.h.h.h.h.h
0000800010  08 68 09 68 0A 68 0B 68 0C 68 0D 68 0E 68 0F 68  .h.h.h.h.h.h.h.h
0000800020  10 68 11 68 12 68 13 68 14 68 15 68 16 68 17 68  .h.h.h.h.h.h.h.h
0000800030  18 68 19 68 1A 68 1B 68 1C 68 1D 68 1E 68 1F 68  .h.h.h.h.h.h.h.h
```

End Value: 0x67FF
Start Value: 0x6800

Can guess the Stripe Size with

1. Start offset
2. Value

→ 0x10000(64KB)