

# 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	uY* ( ... ·âhV·) ...

Data in .img file

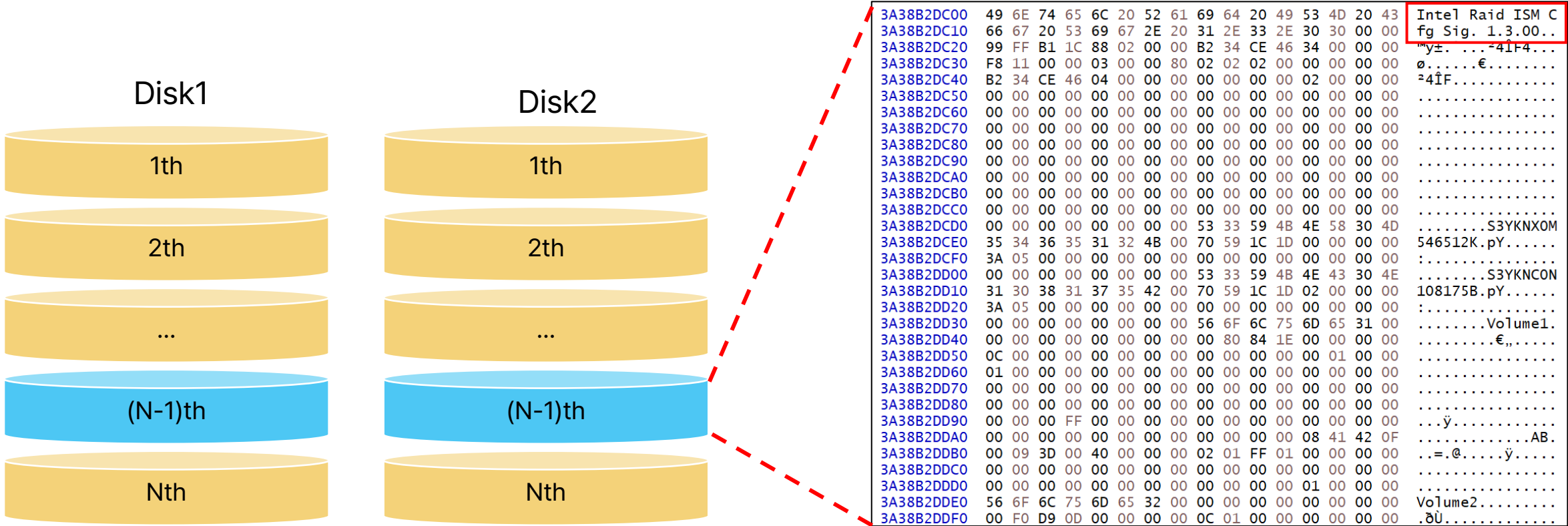
```
/* MPB == Metadata Parameter Block */
#define MPB_SIGNATURE "Intel Raid ISM Cfg Sig. "
#define MPB_SIG_LEN (strlen(MPB_SIGNATURE))
#define MPB_VERSION_RAID0 "1.0.00"
#define MPB_VERSION_RAID1 "1.1.00"
#define MPB_VERSION_MANY_VOLUMES_PER_ARRAY "1.2.00"
#define MPB_VERSION_30R4_DISK_ARRAY "1.2.01"
#define MPB_VERSION_RAIDS "1.2.02"
#define MPB_VERSION_50R6_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



# 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±.^...²4ÎF4...
3A38B2DC30	F8 11 00 00 03 00 00 80 02 02 02 00 00 00 00	ø.....€.....
3A38B2DC40	B2 34 CE 46 04 00 00 00 00 00 00 00 02 00 00	²4ÎF.....
3A38B2DC50	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	.....
3A38B2DC70	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	.....
3A38B2DC90	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	.....
3A38B2DCB0	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	.....S3YKNX0M
3A38B2DCD0	00 00 00 00 00 00 00 00 53 33 59 4B 4E 58 30 4D	546512K.pY.....
3A38B2DCE0	35 34 36 35 31 32 4B 00 70 59 1C 1D 00 00 00 00	.....
3A38B2DCF0	3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00	.....S3YKNC0N
3A38B2DD00	00 00 00 00 00 00 00 00 53 33 59 4B 4E 43 30 4E	108175B.pY.....
3A38B2DD10	31 30 38 31 37 35 42 00 70 59 1C 1D 02 00 00 00	.....
3A38B2DD20	3A 05 00 00 00 00 00 00 00 00 00 00 00 00 00	.....Volume1.
3A38B2DD30	00 00 00 00 00 00 00 00 56 6F 6C 75 6D 65 31 00	.....€.,.....
3A38B2DD40	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 01 00 00	.....
3A38B2DD60	01 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	.....
3A38B2DD80	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	...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	.....
3A38B2DDD0	00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	.....
3A38B2DDE0	56 6F 6C 75 6D 65 32 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	.ðÛ.....

# 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±.^...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	.....S3YKNCON
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



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

- **IMSM\_MAP**



# Conclusion

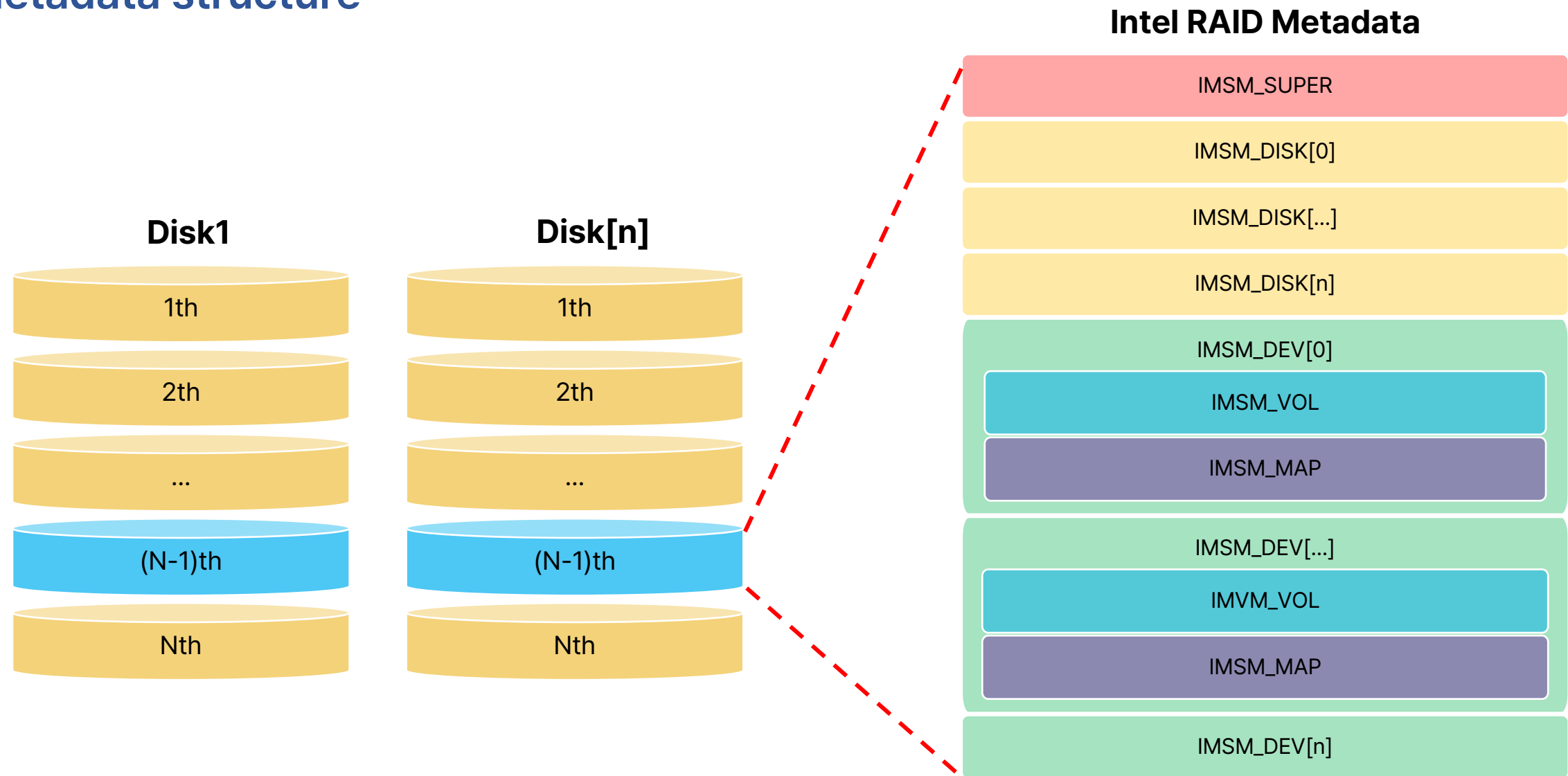
## How MBP(Metadata Block Parameter) is recorded

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

```
8722     static int store_ismm_mpb(int fd, struct ismm_super *mpb)
8723     {
8724         void *buf = mpb;
8725         __u32 mpb_size = __le32_to_cpu(mpb->mpb_size);
8726         unsigned long long dsize;
8727         unsigned long long sectors;
8728         unsigned int sector_size;
8729
8730         get_dev_sector_size(fd, NULL, &sector_size);
8731         get_dev_size(fd, NULL, &dsize);
8732
8733         if (mpb_size > sector_size) {
8734             /* -1 to account for anchor */
8735             sectors = mpb_sectors(mpb, sector_size) - 1;
8736
8737             /* write the extended mpb to the sectors preceeding the anchor */
8738             if (lseek64(fd, dsize - (sector_size * (2 + sectors)),
8739                 SEEK_SET) < 0)
8740                 return 1;
8741
8742             if ((unsigned long long)write(fd, buf + sector_size,
8743                 sector_size * sectors) != sector_size * sectors)
8744                 return 1;
8745         }
8746
8747         /* first block is stored on second to last sector of the disk */
8748         if (lseek64(fd, dsize - (sector_size * 2), SEEK_SET) < 0)
8749             return 1;
8750
8751         if ((unsigned int)write(fd, buf, sector_size) != sector_size)
8752             return 1;
8753
8754         return 0;
8755     }
8756
```

# Conclusion

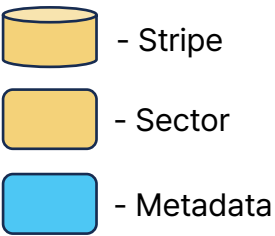
## Metadata structure



# 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	uŸ*{...áhV~}...
3b9e655c30	F0 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 00 01 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	.....S3YKNXOM
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	.....S3YKNCON
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~°2İ.....

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	.....
3b9e655d90	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	.....
IMSM_MAP	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.....
IMSM_MAP	00 00 00 00 00 00 00 00-00 00 00 08 45 1C 1D	.....E..
IMSM_MAP	10 71 74 00 40 00 01 05-03 01 FF 01 00 00 00 00	..qt~@.....Ÿ.....
IMSM_MAP	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	.....

IMSM\_SUPER

IMSM\_DISK[0]

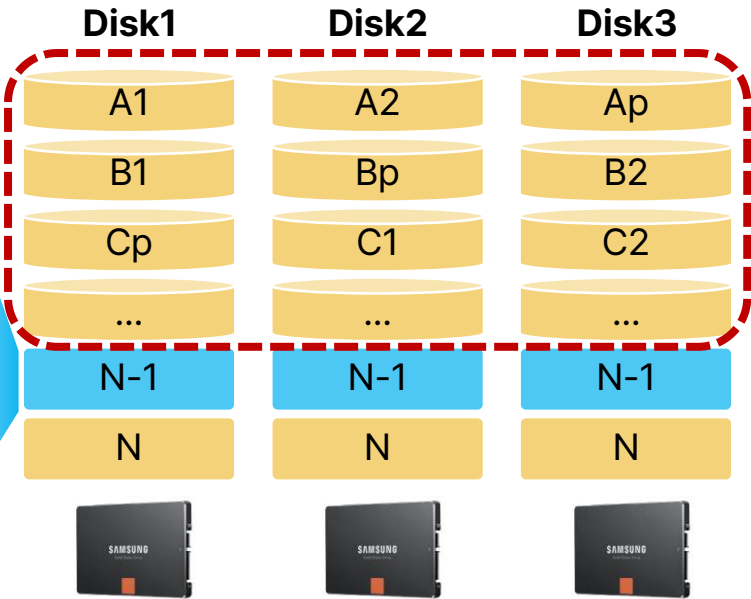
IMSM\_DISK[1]

IMSM\_DISK[2]

IMSM\_DEV[0]

IMSM\_VOL

IMSM\_MAP

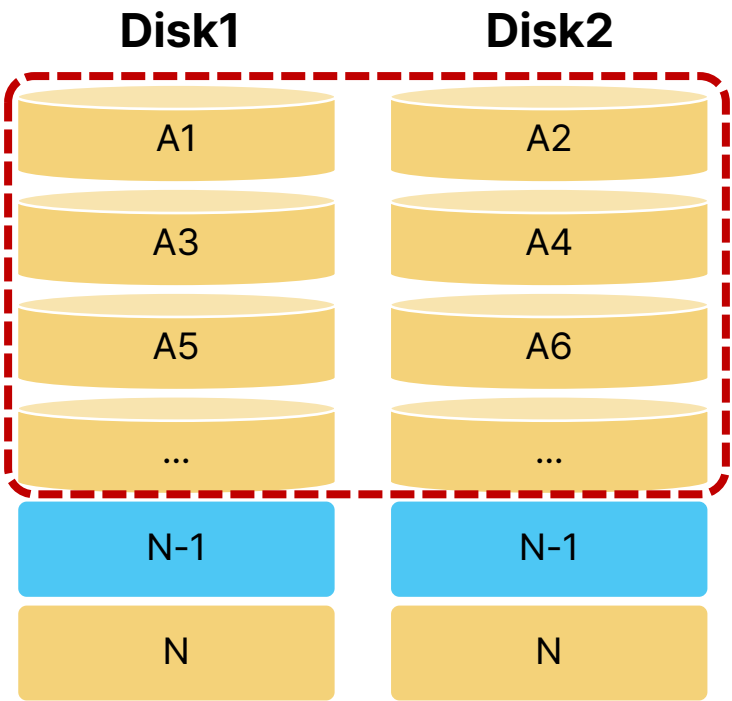


Intel RAID Metadata

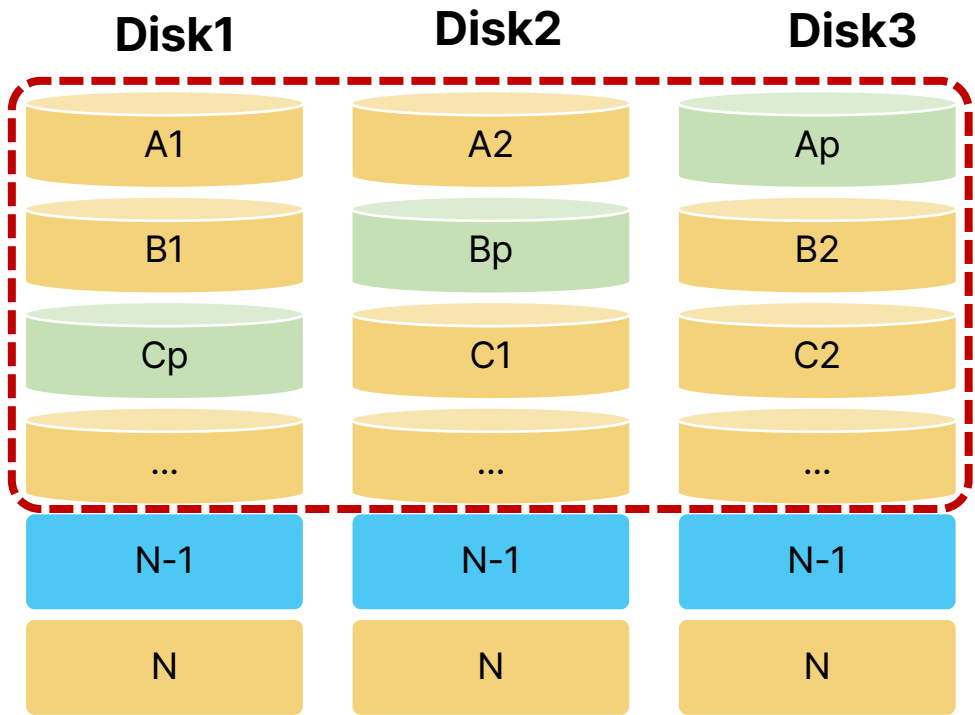
# Conclusion




## Intel RAID Disk Layout

- RAID 0



- RAID 5



 - Stripe  - Sector  - Metadata

# AMD RAID

---

# Conclusion

## Components of AMD RAID

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

```
C:\Program Files (x86)\WRAIDXpert2>rcadm.exe --manage --query-all

<VERSIONS>
rcadm: 9.3.0-00296
rcraid: 9.3.0-00296
rcbottom: 9.3.0-00296

<CONTROLLER LIST>
```

Number	Type	Serial Number	License Key	Port Count	PCI Vendor Id	PCI Device Id	PCI SubVendor Id	PCI SubDevice Id	SAS Address (WWID)	BIOS 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>
```

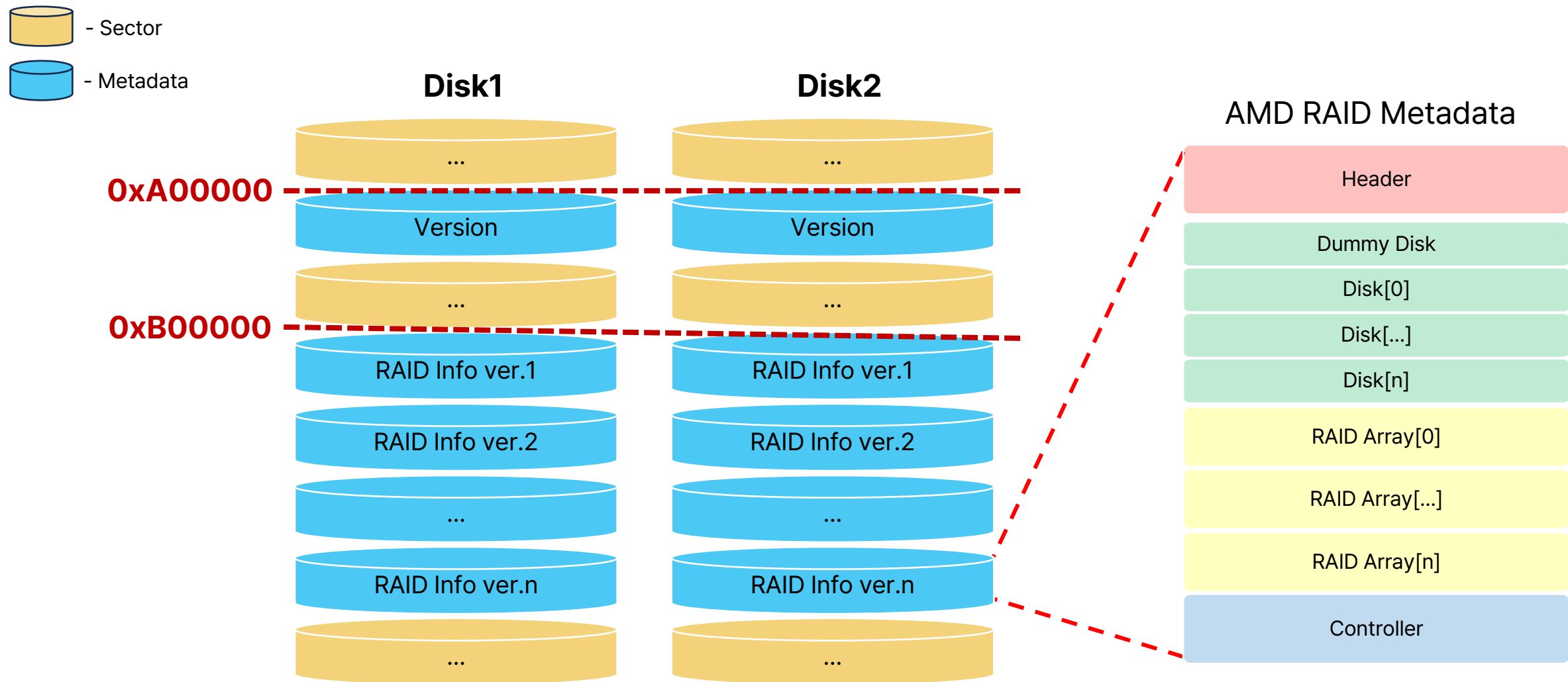
Disk	State	Disk Type	Port Type	Port Speed	Size	Largest Free Space	Free Space	G S Ca	SMART Poll	Ctrl Chan	Model Number	Firmware Version	Serial Number
0	Online	Disk	SATA/SSD	6Gb/sec	250.0GB	0.0MB	0.0MB	- NC	off	1:01	Samsung SSD 860 EVO 250GB	RYT04B6Q	S3YKNCON108175B
1	Online	Disk	SATA/SSD	6Gb/sec	250.0GB	0.0MB	0.0MB	- RW	on	1:05	Samsung SSD 860 EVO 250GB	RYT03B6Q	S3YKNXOM546512K

```
<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	0x167b0c2f20ad33f8	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	ïï 0è~!.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

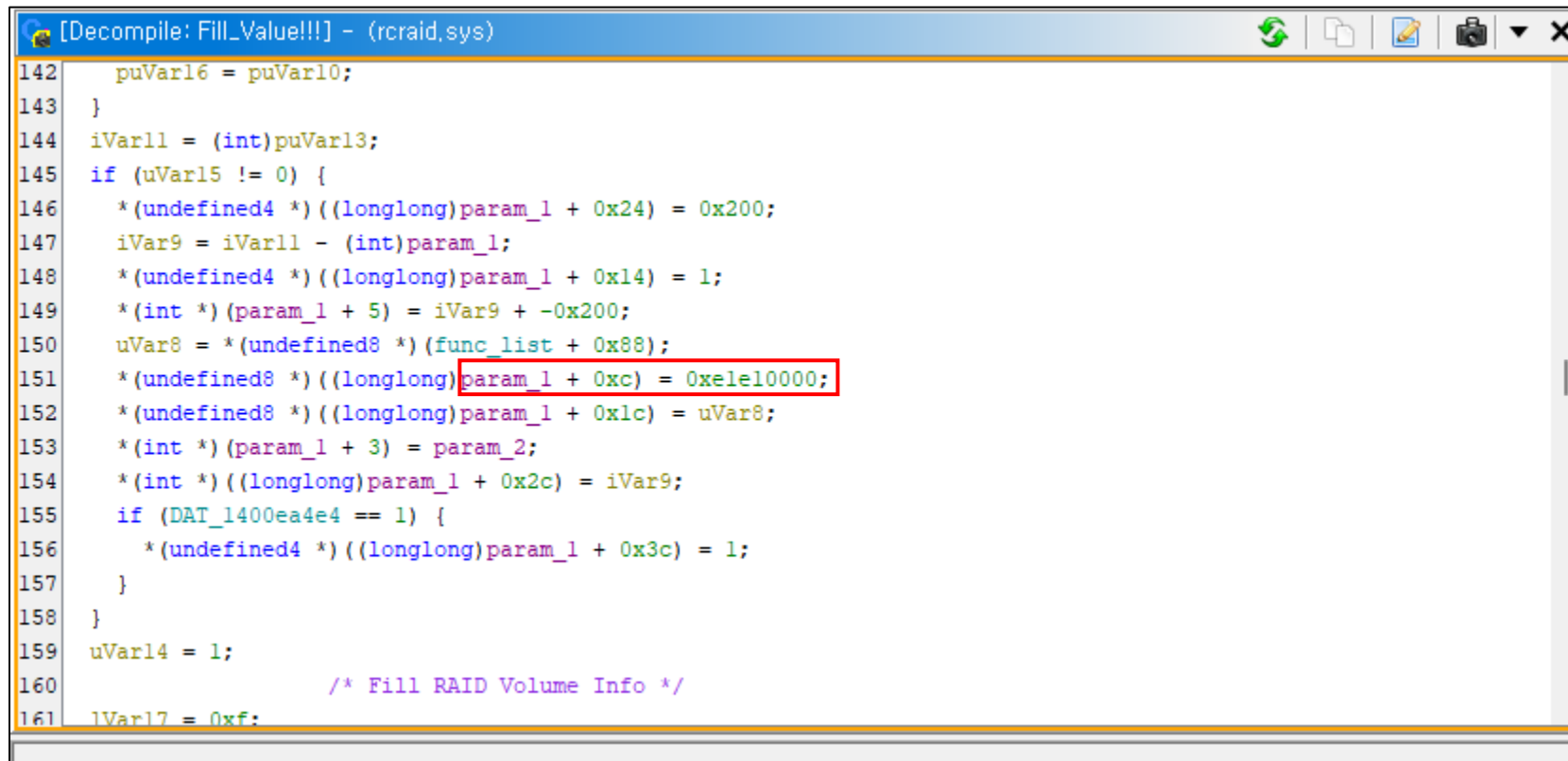
```
[Decompile: Fill_Value!!!] - (rcraid.sys)

1
2 void Fill_Value!!!(undefined8 *param_1,int param_2,undefined4 param_3,uint param_4)
3
4 {
5     undefined8 *puVar1;
6     longlong lVar2;
7     undefined4 uVar3;
8     undefined4 uVar4;
9     undefined4 uVar5;
10    uint uVar6;
11    ulonglong uVar7;
12    undefined8 uVar8;
13    int iVar9;
14    undefined8 *puVar10;
15    int iVar11;
16    undefined8 *puVar12;
17    undefined8 *puVar13;
18    uint uVar14;
19    uint uVar15;
20    undefined8 *puVar16;
21    longlong lVar17;
22    undefined auStack_e8 [32];
23    undefined4 local_c8;
24    int local_b8;
25    undefined local_a8 [80];
26    ulonglong local_58;
27
28        /* @param_1: metadata block pointer
29           @param_2: metadata block size
30           @param_3: Maybe Signature?
31           @param_4: Maybe OP code:
32               0x70: Get config
33               0x90: Update config */
34    puVar12 = glob_mem_metadata_list;
35    local_58 = glob_for_checksum_calc ^ (ulonglong)auStack_e8;
```

# Conclusion

## Header - Reversing

- **Fill\_Value function**
  - Signature

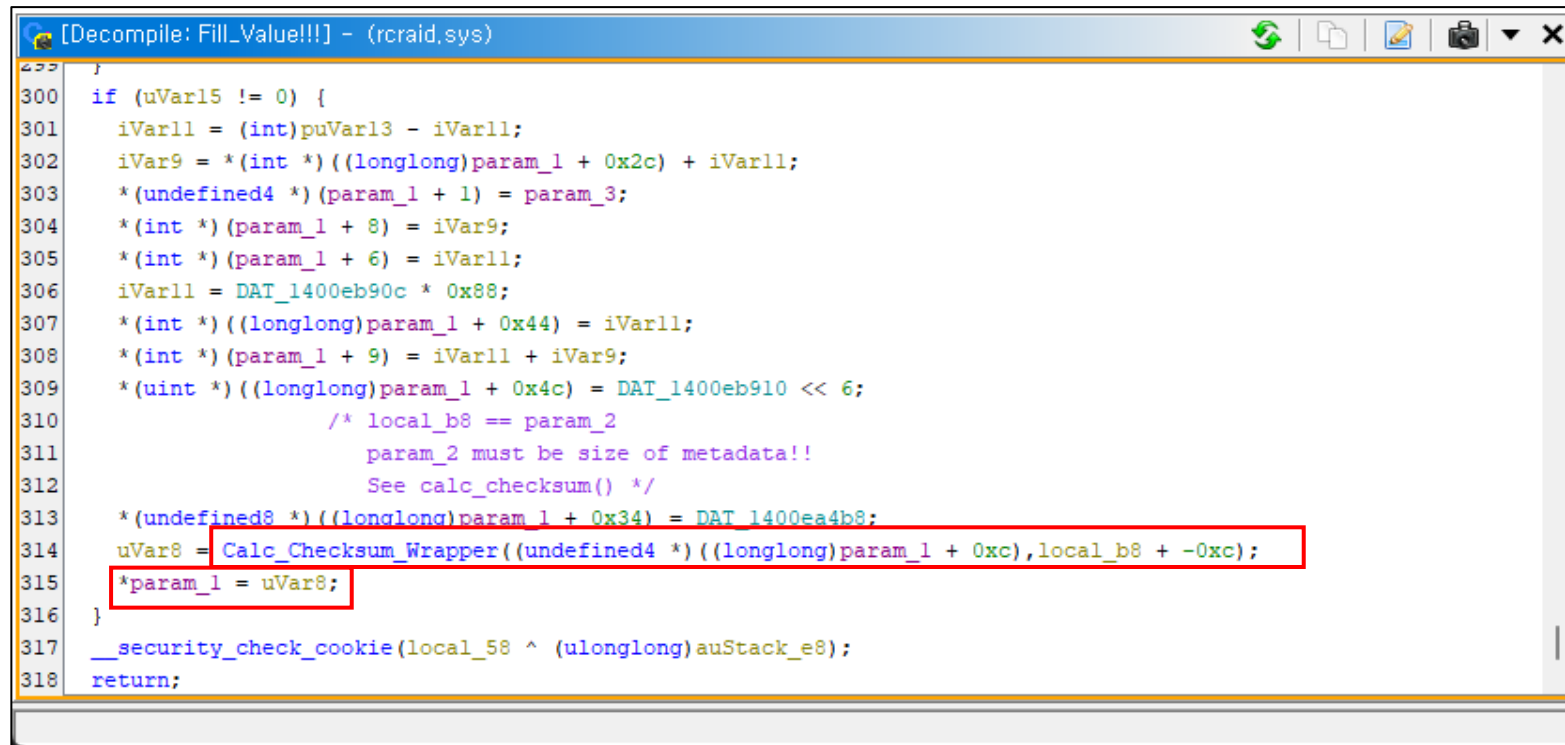


```
[Decompile: Fill_Value!!!] - (rcraid.sys)
142  puVar16 = puVar10;
143  }
144  iVar11 = (int)puVar13;
145  if (uVar15 != 0) {
146      *(undefined4 *) ((longlong)param_1 + 0x24) = 0x200;
147      iVar9 = iVar11 - (int)param_1;
148      *(undefined4 *) ((longlong)param_1 + 0x14) = 1;
149      *(int *) (param_1 + 5) = iVar9 + -0x200;
150      uVar8 = *(undefined8 *) (func_list + 0x88);
151      *(undefined8 *) ((longlong)param_1 + 0xc) = 0xe1e10000;
152      *(undefined8 *) ((longlong)param_1 + 0x1c) = uVar8;
153      *(int *) (param_1 + 3) = param_2;
154      *(int *) ((longlong)param_1 + 0x2c) = iVar9;
155      if (DAT_1400ea4e4 == 1) {
156          *(undefined4 *) ((longlong)param_1 + 0x3c) = 1;
157      }
158  }
159  uVar14 = 1;
160      /* Fill RAID Volume Info */
161  iVar17 = 0xf;
```

# Conclusion

## Header - Reversing

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

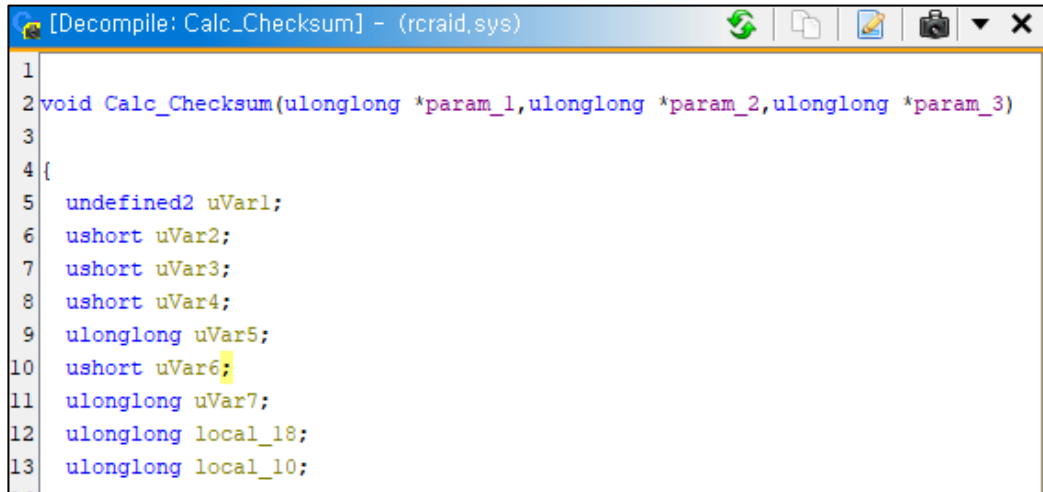


```
[Decompile: Fill_Value!!!] - (rcraid.sys)
300 if (uVar15 != 0) {
301     iVar11 = (int)puVar13 - iVar11;
302     iVar9 = *(int *)((longlong)param_1 + 0x2c) + iVar11;
303     *(undefined4 *) (param_1 + 1) = param_3;
304     *(int *) (param_1 + 8) = iVar9;
305     *(int *) (param_1 + 6) = iVar11;
306     iVar11 = DAT_1400eb90c * 0x88;
307     *(int *) ((longlong)param_1 + 0x44) = iVar11;
308     *(int *) (param_1 + 9) = iVar11 + iVar9;
309     *(uint *) ((longlong)param_1 + 0x4c) = DAT_1400eb910 << 6;
310     /* local_b8 == param_2
311        param_2 must be size of metadata!!
312        See calc_checksum() */
313     *(undefined8 *) ((longlong)param_1 + 0x34) = DAT_1400ea4b8;
314     uVar8 = Calc_Checksum_Wrapper((undefined4 *) ((longlong)param_1 + 0xc), local_b8 + -0xc);
315     *param_1 = uVar8;
316 }
317 __security_check_cookie(local_58 ^ (ulonglong)auStack_e8);
318 return;
```

# Conclusion

## Header - Reversing

- **Checksum Calculation Functions**
  - Calc\_Checksum(int start\_offset, int length)
  - Verify that the same value is acquired



```
[Decompile: Calc_Checksum] - (rcraid.sys)
1
2 void Calc_Checksum(ulonglong *param_1,ulonglong *param_2,ulonglong *param_3)
3
4 {
5     undefined2 uVar1;
6     ushort uVar2;
7     ushort uVar3;
8     ushort uVar4;
9     ulonglong uVar5;
10    ushort uVar6;
11    ulonglong uVar7;
12    ulonglong local_18;
13    ulonglong local_10;
```



```
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)
27
28      /* @param_1: metadata block pointer
29      @param_2: metadata block size
30      @param_3: Maybe Signature?
31      @param_4: Maybe OP code:
32      0x70: Get config
33      0x90: Update config */
34  puVar12 = glob_mem_metadata_list;
35  local_58 = glob_for_checksum_calc ^ (ulonglong)auStack_e8;
36  local_c8 = 0x20;
37  uVar15 = param_4 & 0x10;
38  puVar16 = param_1 + 0x40;
39      /* param_1 + 0x40 --> means First block metadata size is 0x200 */
```

```
66  puVar13 = puVar16 + 0x10;
67  puVar16 = puVar13;
68      /* Raid Volume datal */
69  }
70  for (; puVar12 != (undefined8 *)0x0; puVar12 = (undefined8 *)*puVar12) {
71  puVar10 = puVar16;
72  if ((* (uint *) (puVar12 + 10) & 0x8000) != 0) {
73      /* Storage SI */
74      *(undefined8 *) ((longlong)puVar13 + 0x1c) = *(undefined8 *) ((longlong)puVar12 + 0x54);
75      /* Storage ID */
76      *(undefined8 *) ((longlong)puVar13 + 4) = puVar12[1];
77      /* Unknown flag1 */
78      *(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

```
[Decompile: Fill_Value!!!] - (rcraid.sys)
133      *(undefined4 *)puVar13 = 0x25bc;
134      *(undefined4 *) (puVar13 + 3) = 0;
135      puVar13 = puVar16 + 0x10;
136      puVar10 = puVar16 + 0x10;
137      if (param_4 == 0) {
138          puVar13 = puVar1;
139          puVar10 = puVar16;
140      }
141  }
142  puVar16 = puVar10;
143  }
144  iVar11 = (int)puVar13;
145  if (uVar15 != 0) {
146      *(undefined4 *) ((longlong)param_1 + 0x24) = 0x200;
147      iVar9 = iVar11 - (int)param_1;
148      *(undefined4 *) ((longlong)param_1 + 0x14) = 1;
149      *(int *) (param_1 + 5) = iVar9 + -0x200;
150      uVar8 = *(undefined8 *) (func_list + 0x88);
151      *(undefined8 *) ((longlong)param_1 + 0xc) = 0xe1e10000;
152      *(undefined8 *) ((longlong)param_1 + 0x1c) = uVar8;
153      *(int *) (param_1 + 3) = param_2;
```

## Disk

- **Information about recognized disks**

- Size
- Disk Type (HDD/SSD)
- Port Type (SATA)
- Port Speed
- Firmware version

[illegible]



# 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

```
Decompile: Fill_Value!!! - (rcraid.sys)
69 }
70 for (; puVar12 != (undefined8 *)0x0; puVar12 = (undefined8 *)*puVar12) {
71     puVar10 = puVar16;
72     if ((* (uint *) (puVar12 + 10) & 0x8000) != 0) {
73         /* Storage SI */
74         *(undefined8 *) ((longlong)puVar13 + 0x1c) = *(undefined8 *) ((longlong)puVar12 + 0x54);
75         /* Storage ID */
76         *(undefined8 *) ((longlong)puVar13 + 4) = puVar12[1];
77         /* Unknown flag1 */
78         *(undefined2 *) ((longlong)puVar13 + 0xc) = *(undefined2 *) (puVar12 + 2);
79         /* Unknown flag2 */
80         *(undefined2 *) ((longlong)puVar13 + 0xe) = *(undefined2 *) ((longlong)puVar12 + 0x12);
81         *(undefined4 *) (puVar13 + 2) = 0;
82         *(undefined4 *) ((longlong)puVar13 + 0x24) = *(undefined4 *) (puVar12 + 6);
83         *(undefined4 *) ((longlong)puVar13 + 0x2c) = *(undefined4 *) (puVar12 + 7);
84         *(undefined4 *) (puVar13 + 6) = *(undefined4 *) ((longlong)puVar12 + 0x3c);
85         /* Storage FE */
86         *(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	%%.....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	.°i".....
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	...Ä.....
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	.£@3A".....
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 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 00 00 00 00	Ž.\8.....
0000B0E1C0	00 00 00 00	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	.°i".....
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 00 00 00 00	...À.....
0000B0E210	90 02 00 00	00 00 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 00 00 00 00	.....
0000B0E230	44 46 52 43	41 4D 44 52 41 49 44 30	00 00 00 00 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	.°i".....
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

```
[Decompile: get_raid_level] - (rcadm.exe)

7
8 iVar1 = *(int *) (param_1 + 0xc);
9 if (iVar1 == 0x1bfa) {
10     if ((*uint *) (param_1 + 0x74) == 1) && (2 < *(uint *) (param_1 + 0x78))) {
11         return PTR_s_RAID6_14014dd20;
12     }
13     if ((1 < *(uint *) (param_1 + 0x74)) && (2 < *(uint *) (param_1 + 0x78))) {
14         return PTR_s_RAID60_14014dd28;
15     }
16 }
```

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)
  - Rcadm.exe: Can check flag in Fill\_config\_or\_log\_string\_buffer
  - 0x200 means Normal
- RAID Signature(\*param\_2)

```
[Decompile: Fill_RAID_Volume_Value!!!] - (rcraid.sys)
17  longlong lVar13;
18
19  *param_1 = 0x25bd;
20  param_1[1] = 0x200;
21  param_1[2] = *param_2;
22                                     /* Raid Level */
23  param_1[3] = param_2[1];
24  param_1[4] = param_2[2];
25  param_1[5] = param_2[3];
```

```
[Decompile: Fill_config_or_log_string_buffer -
37  FUN_1400369f0(local_168,0,0x50);
38  if (param_8 == 0) {
39      iVar5 = FUN_1400175f0(param_4
40      iVar3 = *(int *) (param_1 + 8);
41      if (iVar3 == 0x200) {
42          pcVar13 = "NORMAL";
43      }
44      else if (iVar3 == 0x201) {
45          pcVar13 = "CRITICAL";
46      }
47      else if (iVar3 == 0x202) {
48          pcVar13 = "OFFLINE";
49      }
```



# 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)
265 if ((*local_338 - 0x1bf70 & 0xffffffffb) != 0) {
266     local_320 = local_3ac * local_320;
267 }
268 if (local_320 < *(ulonglong *) (piVar18 + 0x1c)) {
269     (**(code **)) (func_list + 0xac)
270     ("RC_CreateTransformRaidArray: Error - Array size too small: %I64x, old size %I64x\n",
271      local_320);
272     goto RETURN;
273 }
```

```
[Decompile: Fill_RAID_Volume_Value!!!] - (rcraid.sys)
38 param_1[0x12] = 0;
39 *(undefined2 *) (param_1 + 0x13) = *(undefined2 *) (param_2 + 0x1a);
40 *(undefined2 *) ((longlong)param_1 + 0x4e) = *(undefined2 *) ((longlong)param_2 + 0x6a);
41     /* Maybe Array Size
42      See RC_CreateTransformRaidArray() Error RC_CreateTransformRaidArray: Error -
43      Array size too small: %I64x, old size %I64x\n */
44 *(undefined8 *) (param_1 + 0x14) = *(undefined8 *) (param_2 + 0x1c);
45 *(undefined8 *) (param_1 + 0x16) = *(undefined8 *) (param_2 + 0x1e);
```



# 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

```

74 do {
75     *(undefined *)puVar5 =
76         *(undefined *) ((longlong)param_2 + (0x140 - (longlong)param_1) + (longlong)puVar5);
77     puVar5 = (undefined4 *) ((longlong)puVar5 + 1);
78     lVar6 = lVar6 + -1;
79 } while (lVar6 != 0);
80 memset(param_1 + 0x45, 0, 0xec);
81     /* Space between RAID Volume is 0x200 */

```

[illegible]

# Conclusion

## Array Metadata - Reversing

### ■ Fill\_RAID\_Volume\_Value

- First Count
- Second Count
- Dummy Disk Count

```
Decompile: RC_CreateRaidArray - (rcraid.sys)
...
486     uVar19 = 0;
487     uVar18 = (uint)uVar24;
488     if (local_15c == 0) {
489         *(ulonglong *) (logical_device_start + 0x1c) = uVar22;
490         /* 1. Set Raid Level and Disk Count */
491         logical_device_start[0x2c] = raid_hdd_count * uVar18;
492         logical_device_start[1] = *(undefined4 *) (configstruct + 4);
493         logical_device_start[0x2d] = uVar18;
```

```
Decompile: RC_CreateRaidArray - (rcraid.sys)
...
522     logical_device_start[0x45] = local_174;
523     /* 2. Set disk count */
524     logical_device_start[0x2e] = raid_hdd_count;
525     *logical_device_start = 0x200;
526     *(undefined8 *) (logical_device_start + 0x50) = 0;
```

```
Decompile: RC_CreateRaidArray - (rcraid.sys)
...
516     puVar13[4] = DAT_1400ed2e0;
517     puVar13 = logical_device_start + 0x7c;
518     /* 3. Set dummy disk count */
519     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 00 A8 76 11 00 00 00	....."v.....
0000B0E3A0	00 50 10 00 00 00 00 00 00 00 58 76 11 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 00 A8 76 11 00 00 00	....."v.....
0000B0E3E0	00 50 10 00 00 00 00 00 00 00 58 76 11 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

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

91 do {
92     lVar13 = uVar10 * 0x44;
93     /* Maybe double pointer(**) */
94     lVar6 = *(longlong *) (*(longlong *) (param 2 + 0x76) + uVar12 * 8);
95     /* Set Disk ID */
96     *puVar9 = *(undefined8 *) (lVar13 + lVar6);
97     /* Disk info Values */
98     *(undefined2 *) (puVar8 + -2) = *(undefined2 *) (lVar13 + 8 + lVar6);
99     uVar2 = *(undefined2 *) (lVar13 + 10 + lVar6);
100    *(undefined4 *) ((longlong)puVar8 + -0xc) = 0;
101    *(undefined2 *) ((longlong)puVar8 + -0xe) = uVar2;
102    /* Array of RAID volume
103       begin & end */
104    auVar1 = *(undefined (*) [16]) (lVar13 + 0x14 + lVar6);
105    puVar8[-1] = auVar1._0_8_;
106    *puVar8 = auVar1._8_8_;
107    auVar1 = *(undefined (*) [16]) (lVar13 + 0x24 + lVar6);
108    puVar8[1] = auVar1._0_8_;
109    puVar8[2] = auVar1._8_8_;
110    *(uint *) (puVar8 + 3) = *(uint *) (lVar13 + 0x34 + lVar6) & 0xffff;
111    /* Zero padding */
112    *(undefined8 *) ((longlong)puVar8 + 0x1c) = 0;
113    *(undefined4 *) ((longlong)puVar8 + 0x24) = 0;
```

# 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)\RAID\pert2>rcadm.exe --manage --query-all

<VERSIONS>

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

<CONTROLLER LIST>
```

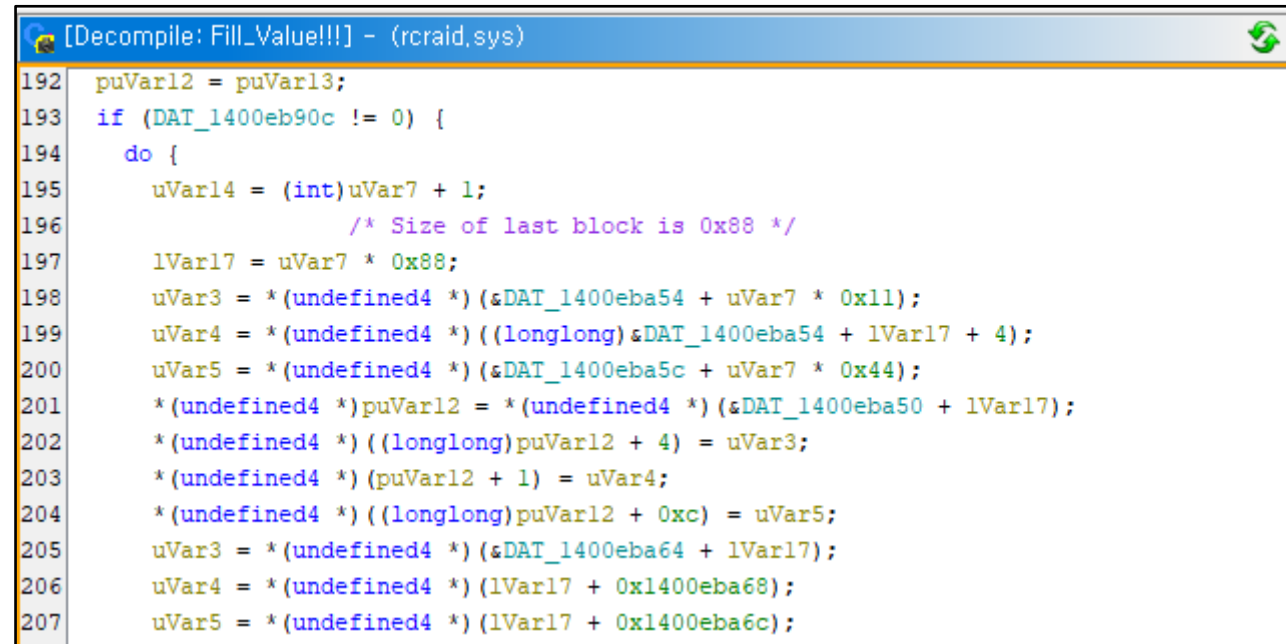
Number	Type	Serial Number	License Key	Port Count	PCI Vendor Id	PCI Device Id	PCI SubVendor Id	PCI SubDevice Id	SAS Address (WWID)	BIOS Version
01	AMD-RAID	4b2a1d00	11111-11111-11111-11111	1	0x1022	0x7916	0x1022	0x7901	0x0000000000000000	NONE
02	AMD-RAID	4b2a1d01	11111-11111-11111-11111	2	0x1022	0x7916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

[illegible]

# Conclusion

## Controller Metadata - Reversing

- **Fill\_Value**



The screenshot shows a decompiled C code snippet titled "[Decompile: Fill\_Value!!!] - (rcraid.sys)". The code is displayed in a window with a blue title bar and a green refresh icon in the top right corner. The code is color-coded: keywords like 'if', 'do', and 'int' are in blue; variables like 'puVar12', 'uVar7', 'uVar3', 'uVar4', 'uVar5', 'lVar17', 'DAT\_1400eba54', 'DAT\_1400eba5c', 'DAT\_1400eba50', 'DAT\_1400eba64', 'DAT\_1400eba68', and 'DAT\_1400eba6c' are in green; and literals and comments are in purple. The code is as follows:

```
192 puVar12 = puVar13;
193 if (DAT_1400eb90c != 0) {
194     do {
195         uVar4 = (int)uVar7 + 1;
196         /* Size of last block is 0x88 */
197         lVar17 = uVar7 * 0x88;
198         uVar3 = *(undefined4 *)(&DAT_1400eba54 + uVar7 * 0x11);
199         uVar4 = *(undefined4 *)((longlong)&DAT_1400eba54 + lVar17 + 4);
200         uVar5 = *(undefined4 *)(&DAT_1400eba5c + uVar7 * 0x44);
201         *(undefined4 *)puVar12 = *(undefined4 *)(&DAT_1400eba50 + lVar17);
202         *(undefined4 *)((longlong)puVar12 + 4) = uVar3;
203         *(undefined4 *) (puVar12 + 1) = uVar4;
204         *(undefined4 *)((longlong)puVar12 + 0xc) = uVar5;
205         uVar3 = *(undefined4 *)(&DAT_1400eba64 + lVar17);
206         uVar4 = *(undefined4 *) (lVar17 + 0x1400eba68);
207         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	i 0ð~!.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

0000B0E000	BC 25 00 00 1B A3 40 33 C1 93 13 5F 00 00 01 00	%%...£@3Á"_.P.....
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	%%.....¥.£.....
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	%%.....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	.°i".....
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	...Ä.....
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 40 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 48 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

 - Sector

 - Metadata

