??????R-TT
http://www.r-tt.com/Articles/Finding...rs/index.shtml
??????:OSSLab aweij ,thx

????????Raid 5 ??

??:

1. 3 ??? ????
2. ????: NTFS (created by Windows XP/2003 ? Master Boot Record (MBR start block)
3. Type: Basic volume

?? Raid ??????
1. ????
2. ?? ?????(Strip size) ??
3. ????
4. ????(Offset)

????R Studio ??????????:

Disk1.arc
Disk2.arc
Disk3.arc

"?????????????,??R-Studio?Disk2.arc???Disk1 object,???????Raid???????"

?? MBR ?? RAID ???

1. ?????? ?? Text/hexadecimal edi
2. **??????,??????????**
3. Write down the Windows disk signature for each object to recognize later which Editor's window belongs to which object.

"?? ???? disk signature ????object?, ???????????????object?"

4. " ????HEX??????????? mbr 33 C0 8E D0 BC (??HEX???MBR???BLOCK, ??CASE??CODE?????) ; ???????????
?"From start position" ?? ??"0" ?" Search at offset(hex) "

5. Click OK to start searching.

Data in Search dialog box to begin search for the Master Boot Record (MBR)

Search results:

Disk1.arc ?????

Disk2.arc MBR pattern found.

Disk3.arc opened in the Text/hexadecimal editor. MBR pattern found.

The result is that the Text/hexadecimal editor finds this pattern at address 00 on Disk2.arc and Disk3.arc; Disk1.arc shows only zeros. That means that the offset is 0, and Disk1.arc cannot be the first disk in the RAID.

" ???Hexview ????????Disk 2 ??Disk 3???MBR Patterns , ?Disk1????????0 ,?????????? 0 , ?Disk 1 ????? ?? ???? ? "

"????? Disk2 ? Disk3 ? master bootstrap loader code. ?????? , ??DISK??????????MBR data "

"??? , ??????NTFS boot sector "

Take a look on the Sectors preceding partition field on the Template pane

" ??Template pane ??Sectors preceding partition ?? Windows disk signature "

Template pane for Disk2 and Disk3

For our case, the sector preceding the partition is 16,065.
" ???????, Sectors preceding partition? 16,065 "

If this value is larger than 63, we should divide it by N -1, where N is the number of disks (in our case, N = 3), which gives us 8,032. This is an approximate position to start searching for the NTFS boot sector. We will start the search from this position to avoid finding false NTFS boot sectors that may remain from previous NTFS partitions.

" ????????63 , ????????? N - 1 , (N ???RAID ? DISK?? ???CASE???3?) , ????????NTFS partitions????NTFS boot sectors,??????????(????????NTFS boot sector???),??? 8,032 ?. "

" ??HexView ?NTFS boot sector pattern??????????sector "

The Sectors search field in the Text/hexadecimal editor

On the Search dialog box, enter EB 52 90 4E 54 46 53 20 20 20 20 (the NTFS boot sector **always** starts from these bytes) into the HEX field, select From current position and enter 0 in Search at offset.

" ?????????, ???? "From current position"???" Search at offset(hex)" ??? 0 ,? ?? EB 52 90 4E 54 46 53 20 20 20 20 (??NTFS **boot sector ????bytes??) "**

Data in Search dialog box to start search for NTFS boot sector

"?????DISK 2 ?DISK 3 ??sector 8064

Now select the Boot sector NTFS pattern on the Template pane.

RAID Data Recovery

Click image to enlarge
Disk2.arc opened in the Text/hexadecimal editor. NTFS boot sector pattern found. The same pattern is found on Disk3.arc.

" ?DISK 2 ????Disk 3 ??? **NTFS boot sector pattern "**

"????**NTFS boot sector pattern** ????Bios parameters Block??????????????"

Required parameters that we have found
Bytes per sector: 512
Sectors per cluster: 8
Logical Cluster Number for the file $MFT: 786432

Previously found parameters:
RAID offset: 0

Next we need to find the MFT (master file table) on the disk:
" ?????DISK???MFT "

1. We will try to find an approximate MFT offset from the RAID start:

" 1. ???????RAID ????? ???MFT ???"

MFT offset from the partition start in sectors = Logical Cluster Number for the file $MFT * Sectors per cluster+RAID offset = 786,432*8+0 = 6,291,456 sector

" ??????sectors?????MFT ??? = Logical Cluster Number for the file $MFT * ??????+RAID ???=786,432*8+0 = 6,291,456 sector "

If the RAID offset is not 0, we need to add the offset to the result of the equation above.
MFT start on the first disk = MFT offset from the partition start in sectors/(N-1) = 6,291,456/2 = 3,145,728

"??Raid ????? 0 , ?????raid???????????. ?????

????????MFT = ??????????MFT ???/(N-1) = 6,291,456/2 = 3,145,728 "

2. We will begin to search for the exact MFT start at a position a couple thousand sectors less than this value. Say, sector 3,140,000.

" ??????3,140,000???? ??????????MFT??"

On the Search dialog box, enter "FILE" into the ANSI field, then select From current position and enter 0 in Search at offset.

" ?????????, ???? "From current position"???" Search at offset(hex)" ??? 0 ,? ?? "FILE"??ANSI? "

This pattern is found at sector 10,241,463 on Disk2 and at sector 3,153,792 on Disk3.

" ?????????????? , ?Disk2 ? **sector 10,241,463 , ?Disk3 ? sector 3,153,792 "**

First file record sector in Disk3. Start of a data block.
"?Disk3 ??????????Block(First MFT record) "

What is important: The signature FILE ends with 0, which means that the file record number is not overwritten with a fixup. If it had ended with * (FILE*), we would not have been able to proceed further in our search and would have needed to use another technique.

" ??? : ??????signature FILE ???? 0 ???, ??????file record number ????????????.

?????* (FILE*) ???, ?????????????,??????????????????case "

The pattern $.M.F.T. (HEX 24 00 4D 00 46 00 54) shows that this is a correct MFT beginning.

"??????$.M.F.T. (HEX 24 00 4D 00 46 00 54) ?????????MFT ???"

Because sector 3,153,792 is closer to our expected value of sector 3,145,728 than to sector 10,241,463, we may assume that Disk3 is the first disk in the RAID.

" ??Disk 3 ?????? (3,153,792) ????????3,145,728, ????????Disk3 ??Raid?????Disk "

To proceed further, we need to keep in mind that a file record in MFT occupies two sectors, and that data is written to a RAID 5 successively, one data block to one disk, then the next data block to the next disk, and a parity block to the third disk. We can represent an example of such a scheme in the following table ...

" ??????????, ??????? MFT ??????sectors , ????????raid? , ??????Disk ??data block ??? , ????????????? "

RAID DISK 1	Second RAID disk	Third RAID disk
PD	1	2
3	PD	4
5	6	PD

... where the numbers represent the order in which the data blocks are written to their respective disks, and PD stands for the "parity of data" block.
(This table represents only an example, and the block order may be arbitrary in a general case.)

" ??????????,??data blocks????????Disk?, PD??????Block"

" (??????????case?????????, ????????case) "

Here that means that the file record numbers in MFT will increase by one within one data block. Then the MFT will continue on another disk, where file record numbers will increase by one within its respective data block, the third disk containing the parity block. And so on.

" ???? MFT??????????????data block?, ??MFT?????????Disk? ,?????disk ???parity block ,??????"

So, to find the block size, we will look at the file record numbers on this disk to discover the place where they no longer increase by one. This place would mean the end of that data block. Then we will look at other disks to find the disk and the place on it where file record numbers in the MFT resume increasing by one. Then we will look at another disk to find where the MFT continues, and so on.

" ??,?????disk????????????????record???block size,??????data block???,???????disk?????????MFT file record number , ?????disk?MFT?????,??????"

Such a search can be done by scrolling down the text in the Editor by two sectors.

"????????????????2? sectors?????"

On Disk 3 the data block ends in sector 3,153,919 with file record number 3F 00.

"?disk 3?, data block???sector 3,153,919?, record number? 3F 00"

Last file record in Disk3. End of a data block is on the next sector (3,153,919).

Looking at other disks, we find that this MFT continues on Disk 1 in sector 3,153,792 with file record number 40 00 and ends in Sec: 3,153,919 with file record number 7F 00. And so on.

" ?????Disk? , ??Disk 1?MFT ??? 3,153,792 ,?? record number ? 40 00 ????? 3,153,919? ,record number ? 7F 00 , ???? "

File record continues in Disk1. Start of a data block.

Last file record in Disk1. End of the data block is on the next sector (3,153,919)

The final results are represented in the Table below:

"?????????????? : "

Disk1	**Disk2**	**Disk3**
Sec: 3,153,792 Rec: 40 00 Sec: 3,153,918 Rec: 7F 00 Sec: 3,153,919 End of stripe	Sec: 3,153,792 No records Sec: 3,153,918 No records Sec: 3,153,919: End of stripe	Sec: 3,153,792 Rec: 00 00 Sec: 3,153,918 Rec: 3F 00 Sec: 3,153,919 End of stripe
Sec: 3,153,920 Rec: No records Sec: 3,154,046 Rec: No records Sec: 3,154,047 End of stripe	Sec: 3,153,920 Rec: C0 00 Sec: Sec: 3,154,046 Rec: FF 00 Sec: 3,154,047 End of stripe	Sec: 3,153,920 Rec: 80 00 Sec: 3,154,046 Rec: BF 00 Sec: 3,154,047 End of stripe
Sec: 3,154,048 Rec: 00 01 Sec: 3,154,174 Rec: 3F 01 Sec: 3,154,175 End of stripe	Sec: 3,154,048 Rec: 40 01 Sec: Sec: 3,154,174 Rec: 7F 01 Sec: 3,154,175 End of stripe	Sec: 3,154,048 Rec: No records Sec: 3,154,174 Rec: No records Sec: 3,154,175 End of stripe

No records mean this is a parity block.

Example of a parity sector

?????

????
DISK ?? Disk3.arc
DISK ?? Disk1.arc
DISK ?? Disk2.arc

?? 0
Stripe size: 128 sectors, or 65,536KB (64KB)
Stripe order: (PD stands for Parity of Data)

First RAID disk	Second RAID disk	Third RAID disk

1	2	PD
3	PD	4
PD	5	6

?R-Studio??raid....