VIRUS ANALYSIS 1

RMNS - The Perfect Couple

Eugene Kaspersky
Kami Associates

The evolution of the programs we call ‘computer viruses’ continues relentlessly. Today, there are at least two which have gone beyond a single-celled basis and started to replicate by dividing their code into two different components - they are known as ‘multicellular’ viruses (not to be confused with multipartite viruses).

The first of the ‘multicellular’ viruses is Dichotomy [*see Virus Bulletin, December 1994, p.8*], which has two components: ‘odd’ and ‘even’. When a file infected with the ‘odd’ component is executed, the virus looks for a file infected with ‘even’ code, installing itself into memory only if that part is found.

Now, viruses may be abandoning their purely monosexual existence: the RMNS virus may be a further step towards more complex electronic creations. It appears that the word ‘virus’ may no longer be the best term to describe such programs - RMNS does not look like a biological virus, but more like an ‘electronic creature’. This begins another branch of electronic evolution: the era of viruses of a specific sex.

The Virus and its (His?) Sex

RMNS gets its name from the internal text string which is placed at its end. Like Dichotomy, the code of the RMNS virus is divided into two parts (‘male’ and ‘female’). Here, however, the similarity ends. The two parts of RMNS install themselves into memory independently of each other.

The names ‘male’ and ‘female’ derive from text descriptions held within the two parts of the virus:

male:	`R.M.N.S Test virus R.M.N.S MW Man`
female:	`R.M.N.S Test virus R.M.N.S MW Woman`

Infection can only take place if both sections of the code are resident in memory at the same time and on the same computer.

The male and female parts of the virus are very similar: they are both placed at the end of COM files, they both receive control when the infected file is executed, they both issue ‘Are you there?’ calls, and they both hook Int 21h and stay resident. They also have similar lengths - the male code is 297 bytes long, and the female is 353).

The differences between the two parts are few but important: the male does not infect files, but only intercepts their execution; the female does not intercept execution of the files, but infects them on request from the male.

Installation and Int 21h Hooking

When an infected file is executed, that part of the virus with which it is infected receives control with a JMP instruction. The virus then restores the three bytes at the beginning of the host program which were overwritten on infection.

Next, the virus decides whether or not to go resident: it is made up of two parts, and each part will only go resident in memory if it is not there already. The virus code in the infected program issues an ‘Are you there?’ call using Int 21h - for the male code, the AX register is set to 4BBCh, and for the female it is 4BBDh.

Both the male and female sections of the virus return the ID value BBB4h in the AX register to show that they are present in memory.

The segments of the virus install themselves at the top of system memory, using the standard methods of direct manipulation of Memory Control Blocks, and hooking Int 21h. After this, control passes to the beginning of the host program.

The male and female parts each intercept only one Int 21h function: AH=4Bh (Load and Execute). Both parts check the subfunctions of the Load and Execute call and execute the following corresponding routines:

- Male code:
  a) AL = BCh. ‘Are you there?’ call, returns BBB4h in the AX register.
  b) AL = 0, 1, 2, or 3. The file being loaded is checked, and the female part called, using Int 21h with 4BBEh in the AX register, to infect the file.
- Female code:
  a) AL = BDh. ‘Are you there?’ call, returns BBB4h in the AX register.
  b) AL = BEh. Performs the infection routine.

Note that only the male part intercepts the system generated Load and Execute subfunctions (i.e. 0, 1, 2 or 3).

Infection

On a Load and Execute call, the male part opens the corresponding file, reads three bytes from the beginning, and compares the first byte with the character ‘M’ in order to prevent infection of EXE files. Then the virus checks the date and time stamp of the file for the value 00FF00FFh (31.07.80; 12:07am). This is the virus’ ID stamp, and if it is found, the file will not be infected.

If the file concerned is not an EXE file, and it is not yet infected, the virus calls the female part of its code with an ‘Infect it’ call (Int 21h, AX=4BB4h). The male part passes

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

its length in the CX register (CX=0129h), the segment address of its code in the DS register, and the file’s handle in the BX register.

After receiving the ‘Infect it’ request, the female section of the virus checks the length of the file. If the file is longer than 65024 (FE00h) bytes, it will not be infected.

The infection routine then selects the part of the virus with which to infect the file, by using the system timer. It will, 50% of the time, write the male code (using the length and segment address received in the CX and DS registers), and 50% of the time it will write the female code (by overwriting the values in CX and DS with the appropriate values for the female section of the code).

Then, the infection routine overwrites the head of the file with a JMP VIRUS instruction, sets the file date and time stamp to 31.07.80, 12:07am, and returns control to the male part of the code. Thus, the file is infected either with the male or the female code, but not with both at the same time.

The virus does not perform standard virus routines, such as hooking Int 24h during infection to prevent the DOS error message whenever an attempt is made to write to a write-protected disk. It neither saves, clears, nor restores the file’s attributes, and overwrites the file time and mask stamp with its ID value. However, the many minor defects in this virus cannot belittle its importance in the history of these electronic creatures.

RMNS

Aliases: RMNS MW.

Type: Memory-resident, parasitic file infector.

Infection: COM files.

Self-recognition in Files:

Compares file’s date and time stamp with 00FF 00FFh (31.07.80, 12:07am).

Self-recognition in Memory:

‘Are you there?’ calls with Int 21h, AX=4BBCh, AX=4BBDh. The memory-resident code returns BBB4h in the AX register.

Hex Pattern in Files and Memory:

```
BF84 0101 F78A 05A2 0001 478B
05A3 0101 B8B? 4BCD 213D B4BB
```

(The wildcard is replaced in the ‘woman’ by D, and in the ‘man’ by C.)

Intercepts: Int 21h for infection.

Trigger: None.

Removal: Under clean system conditions, identify and replace infected files.

VIRUS BULLETIN **©1995 Virus Bulletin Ltd, 21 The Quadrant, Abingdon, Oxfordshire, OX14 3YS, England. Tel. +44 (0)1235 555139. /95/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.